AVEVA™ Operations Management Interface (OMI)

2023 R2 P01

© 2015-2024 by AVEVA Group Limited or its subsidiaries. All rights reserved.

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

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

U.S. GOVERNMENT RIGHTS

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

Publication date: Thursday, November 7, 2024

Publication ID: 1389410

Contact information

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

https://sw.aveva.com/

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

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

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

Contents

Chapter 1 Welcome to AVEVA Operations Management Interface. 15

Chapter 2 Design standards. 16

Configure components: Galaxies. 16
Create a Galaxy. 17
About Galaxies. 19
Connect to a Galaxy. 19
View the configuration file. 21
Set single sign-on for OMI ViewApps. 21
Turn off multithreaded processing for OMI ViewAPP. 21

About the integrated development environment (IDE). 21
Install and Activate Your License. 22
Manage Security Certificate Verification. 22
View the Galaxy. 22
Template Folder. 23
Visualization Folder. 24
AVEVA CONNECT. 24
IDE Application Views. 24
Customize Your Workspace. 33
Dock Views. 33
Float Views. 33
Hide Views. 33
Reset the Workspace. 34
Synchronize Views. 34
Configure User Preferences. 34
Use the simplified layout. 36
Log on with security enabled. 37
Change Users. 37
Search for content. 38
About the Industrial Graphic Editor. 39

Platform common services. 41
Configure and deploy AVEVA services. 41
Create an instance of a service. 42
Assign the service instance to a node. 42
Deploy the instance. 42
Configure service TCP ports. 42
Configure the ASBGRBrowsing service. 43

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure the ASBMxDataProvider service. . . . 45
Configure the ASBAuthentication service. . . . 45
IOMBLSService. . . . 45
Configure and deploy the OPC UA service. . . . 46
Client access rules and galaxy security. . . . 48
Automation objects. . . . 49
View Object Properties. . . . 50
About Templates and Instances. . . . 51
Instances. . . . 52
Templates. . . . 52
Propagation. . . . 53
About base templates. . . . 53
About Derived Templates. . . . 56
Working with Objects. . . . 57
Manage Folders. . . . 57
Using Industrial Graphics in AVEVA Connect. . . . 59
Create Derived Templates. . . . 65
Create Contained Templates. . . . 67
Edit Objects. . . . 76
Browse for Attributes. . . . 88
Browse for Graphics. . . . 91
Browse for Element Properties. . . . 92
Create a Filter for the Galaxy Browser. . . . 93
Change How Information is Shown in the Galaxy Browser. . . . 94
Manage objects. . . . 95
Check out objects. . . . 95
Check in objects. . . . 95
Validate objects. . . . 96
Create an instance. . . . 98
Object names. . . . 98
Delete an object. . . . 99
Export objects. . . . 99
Import objects. . . . 102
Configuring Objects. . . . 114
Add Attributes to an Object. . . . 114
Adding Features to Attributes. . . . 118
Using I/O Auto Assignment. . . . 139
Writing and Editing Scripts. . . . 150
Creating and Working with Content. . . . 155
Rearrange objects. . . . 162
Assign alias name to an object. . . . 163
Object Wizards. . . . 166
About Object Wizards. . . . 167
Object wizard features. . . . 169
Create a basic object wizard. . . . 169
Object wizard configuration best practices. . . . 171
Object wizard GUI panes. . . . 172
Object wizard user interface details. . . . 174

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Object wizard components. . . . 177
Add content for use with an object wizard. . . . 178
Add a new object-owned symbol. . . . 179
Link to shared content in the visualization folder. . . . 179
Associate content with a choice or option. . . . 180
Disassociate content from a choice or option. . . . 181
Multiple content item associations. . . . 182
Map attribute and graphic configurations to a choice or option. . . . 182
Attribute and graphic overrides. . . . 182
Delete content from an object. . . . 183
Delete an attribute or graphic from a template. . . . 183
Example 1: Map a Simple Symbol Wizard to an Object Wizard. . . . 184
Example: Configure a Linked Graphic Associated with Multiple Choices. . . . 188
Conditional visibility expressions for choice groups, choices, and options. . . . 190
Add a basic visibility expression. . . . 191
Add a visibility expression with logical operators. . . . 191
Operators, rules, and behavior for visibility expressions. . . . 192
Object wizard visibility settings in a derived template. . . . 193
Trimming. . . . 194
Enable Trimming. . . . 195
Attribute and graphic overrides. . . . 196
Enter overrides and set visibility for attribute and graphic values. . . . 198
Attribute and graphic overrides. . . . 199
Configurable settings for attribute and graphic values. . . . 200
Object wizard derivation. . . . 201
Set visibility of choices and options in a derived template. . . . 201
Reasons to derive a template from an object wizard 202
Changes to object wizards in derived templates. . . . 202
Behavior of object wizard components in a derived template. . . . 203
Object wizard test mode. . . . 203
Verify behavior of visibility expressions. . . . 203
About object wizard behavior in test mode. . . . 204
Propagation of object wizard changes to derived objects. . . . 205
Object wizard behavior when a default choice is deleted. . . . 205
Propagation of attribute and graphic settings. . . . 205
Configure instances. . . . 206
Configure an instance with the IDE. . . . 206
Guidelines for configuring instances with an object wizard. . . . 207
Override default settings. . . . 207
Create a new instance graphically. . . . 208
VisualBuild Object options. . . . 211
VisualBuild graphic options. . . . 212
Add a configured asset to an overview graphic. . . . 212
Symbol Wizard Editor. . . . 213
Creating Symbol Wizards. . . . 215
Embedding Symbol Wizards into an Application. . . . 215
Integrate Symbol Wizards with Object Wizards 216
Galaxy Style Library. . . . 217

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Open Galaxy Style Libraries. . . . 218
Manage Galaxy Style Libraries. . . . 218
Duplicate a Galaxy Style Library. . . . 219
Set the Default Galaxy Style Library. . . . 219
Rename a Galaxy Style Library. . . . 219
Export a Galaxy Style Library. . . . 220
Export All Galaxy Style LIbraries. . . . 220
Import a Galaxy Style Library. . . . 221
Quality and Status Styles. . . . 222
Understand Quality and Data Status. . . . 223
Set Quality and Status Styles. . . . 224
Element Styles. . . . 228
Details of Element Styles. . . . 228
Numeric Formats. . . . 229
Configure Global Number Styles. . . . 230
Working with User-defined Global Number Styles. . . . 232
Switch to Another Galaxy Style Library During Runtime. . . . 233

Chapter 3 Build applications. . . . 235

About screen profiles. . . . 236
About the Screen Profile Editor. . . . 237
Create a screen profile. . . . 238
Create a screen profile from the visualization folder. . . . 239
Configure a screen profile. . . . 240
Add a screen to a screen profile. . . . 240
Delete a screen from a screen profile. . . . 241
Rearrange screens in a screen profile. . . . 241
Configure a screen. . . . 242
Resize a screen. . . . 245
Set a primary screen. . . . 246
Set additional screens. . . . 247
Manage a screen profile. . . . 249
Delete a screen profile. . . . 250
Rename a screen profile. . . . 251
Duplicate a screen profile. . . . 251
Export a screen profile. . . . 252
Import a screen profile. . . . 253

AppFramework layouts. . . . 254
About the Layout Editor. . . . 255
About fixed vs. responsive layouts. . . . 259
Create a layout 1. . . . 264
Create a layout 2. . . . 264
Manage a layout. . . . 265
Delete a layout. . . . 266
Duplicate a layout. . . . 267
Rename a layout. . . . 268
Export a layout. . . . 268
Import a layout. . . . 269

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure a layout . . . 270
Add panes to layout . . . 271
Configure a layout: slide-in panes . . . 272
Rename a pane . . . 276
Remove a pane from a layout . . . 277
Resize a pane . . . 277
Extend panes in a layout . . . 280
Set the default pane of a layout . . . 283
Set the pane presentation style . . . 283
Set layout properties . . . 284
Set layout properties values . . . 291
Set pane properties 1 . . . 294
Set pane properties 2 . . . 298
Set layout divider properties . . . 301
Add content to panes . . . 302
Browse for content to add to layout . . . 303
Search for content to add to layout . . . 304
About adding content to an empty pane . . . 307
Add content to multi-content panes . . . 307
Move content to another pane . . . 308
About the responsive layout mode . . . 309
Build a responsive layout . . . 309
About responsive layout methods . . . 313
About responsive layout propagation . . . 314
Responsive Layout: Mostly Fluid . . . 315
Configure a column drop responsive layout . . . 317
Configure a layout shifter responsive layout . . . 321
Responsive layout limitations . . . 324
Build layout scripts . . . 325
Types of layout scripts . . . 326
Namespaces for use with layout scripts . . . 328
About Built-in scripts . . . 329
Add a named script . . . 331
About the ShowContent() function . . . 332
Layout event handler scripts . . . 337
Layout execution order . . . 342
Data binding in a layout . . . 343
About the MyContent namespace . . . 343
Add property binding . . . 344
Set custom property binding type . . . 345
Use a layout as content . . . 346
Effects of autofill on a layout linked to an asset . . . 346
About ViewApp namespaces . . . 346
About the ViewApp namespace editor . . . 348
Create a ViewApp namespace . . . 349
Add and configure custom ViewApp namespace attributes . . . 350
Export a ViewApp namespace . . . 352
Import a ViewApp namespace . . . 352

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Create references to ViewApp namespace attributes. . . . 352
About retentive attributes. . . . 353
About alarms. . . . 354
Understanding Events. . . . 354
Types of events. . . . 354
Understanding Alarms. . . . 356
Types of alarms. . . . 357
Setting Alarm State with Object Attributes. . . . 361
Setting Alarm State for Individual Alarms. . . . 363
Enabling, Silencing, and Disabling Alarms. . . . 364
Shelving Alarms. . . . 365
Throttling Alarms. . . . 369
Propagating Timestamps with Alarms and Events. . . . 369
Configuring Alarms. . . . 371
Configuring Alarming for System Objects. . . . 371
Configuring Alarms and Events for Application Objects. . . . 376
Setting Alarms on the Attributes Page. . . . 378
Distributing Alarms and Events. . . . 380
Subscribing to Alarms and Events from a Client. . . . 380
Using InTouch HMI as the Alarm and Event Client. . . . 381
Alarms and Event Implemenation in Application Server and Run-Time Applications. . . . 383
Configuring Plant State-Based Alarms. . . . 385
Mapping Alarm Modes to Plant States. . . . 385
Configuring State-Based Alarming on an Area Object. . . . 386
Viewing Plant State-Based Alarms at Run Time. . . . 387
Configuring Alarm Historization, Mapping, and Shelving Priority Ranges. . . . 388
Configuring Historization for Alarms and Events. . . . 388
Monitoring Alarm Severities at Run Time. . . . 390
Understanding How Alarms are Ranked at Run Time. . . . 390
Using Aggregated Alarm Information. . . . 390
Configuring Alarm State Aggregation. . . . 391
Aggregating Alarm State Information. . . . 391
Monitoring Alarm State Information at Run Time. . . . 399
About graphic elements. . . . 400
About the Graphic Editor. . . . 401
About the graphic element property commonalities. . . . 402
About graphic-specific and element-specific properties. . . . 403
About zoom percentage. . . . 403
Monitoring and Showing Quality and Status. . . . 412
Change images at runtime. . . . 416
Use animation. . . . 416
Create and manage graphics. . . . 417
Create a graphic. . . . 417
Create graphics in the visualization folder. . . . 418
Create graphics in AutomationObject templates. . . . 418
Create graphics in AutomationObject instances. . . . 419
Organizing Graphics. . . . 419
Create folders in the visualization folder. . . . 419

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Use graphic thumbnails. . . . 420
Move graphics between folders. . . . 420
Rename graphics. . . . 421
Copy graphics. . . . 421
Delete a graphic. . . . 421
Viewing a Graphic in Read-Only Mode. . . . 422
Rename graphic folders. . . . 422
Delete graphic folders. . . . 422
Move graphic folders. . . . 423
Customize graphic folders. . . . 423
Creating Multiple Configurations of a Graphic. . . . 423
Understanding Visual and Functional Graphic Configurations. . . . 423
About graphic scaling. . . . 424
About the Industrial Graphic Editor. . . . 425
Create a fixed-size graphic for a ViewApp pane with a known size. . . . 426
Create a fixed-size graphic for a ViewApp pane size. . . . 427
About fixed-size graphics when pane is resized. . . . 428
About fixed-size graphics during runtime. . . . 429
Create faceplate graphics. . . . 430
About faceplate graphics creation. . . . 430
About graphics visibility during runtime. . . . 432
About settings attributes. . . . 434
About custom properties for graphics. . . . 436
Use custom properties to show historical summary data. . . . 437
Analog statistical summary data. . . . 437
State statistical summary data. . . . 438
Historical summary period. . . . 439
Show statistical summary data. . . . 439
Use binding in custom properties. . . . 441
Changing the Expression or Reference of a Custom Property at Run Time. . . . 443
About using an External Content item 444
External content example. . . . 444
Create an External Content item. . . . 447
About the External Content editor. . . . 448
Configure an External Content item. . . . 450
Link an External Content item to an object. . . . 452
Associate External Content to a ViewApp. . . . 453
Manage external content. . . . 455
Create an app for external content. . . . 455
Script editor. . . . 456
Add a script to a graphic. . . . 456
Working with the SignedWrite() function for secured and verified writes. . . . 457
How to write to an attribute configured for Secured Write or Verified Write security classification. 461
Example of Changing Element Properties using Scripts. . . . 462
ViewApp editor. . . . 462
About the ViewApp Editor. . . . 463
Configure the ViewApp Editor. . . . 466
About ViewApp and derived templates. . . . 467

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Create a ViewApp. . . . 468
Manage a ViewApp. . . . 469
Delete a ViewApp. . . . 469
Rename a ViewApp. . . . 469
Export a ViewApp. . . . 470
Import a ViewApp. . . . 470
Configure a ViewApp. . . . 471
Manage screen profiles in a ViewApp. . . . 471
Manage layouts in a ViewApp. . . . 472
Set additional properties for a ViewApp. . . . 473
Add content to a ViewApp. . . . 474
Browse content to add to a ViewApp. . . . 475
Search for content. . . . 476
About adding content to an empty pane. . . . 478
Add content to multi-content panes. . . . 479
Show or hide ViewApp navigation items. . . . 480
About navigation item security. . . . 480
Configure navigation item security. . . . 481
User permissions and navigation item security. . . . 483
About previewing a ViewApp. . . . 483
Preview a ViewApp. . . . 484
About ViewApp previews at different stages. . . . 486
Switch ViewApp language at runtime. . . . 490
Define and configure galaxy languages. . . . 490
Graphics language switching. . . . 490
About localization workflow. . . . 490
Alarm app language switching. . . . 491
Configure languages for a Galaxy. . . . 492
Add a language to a Galaxy. . . . 493
Remove a language from a Galaxy. . . . 494
Modify the font for a language. . . . 494
Change the default language for a Galaxy. . . . 495
Export files for offline translation. . . . 495
Types of language dictionary files. . . . 496
Export language data for all Industrial Graphics in a Galaxy. . . . 497
Export language data for specific objects. . . . 497
Export graphic language data for a managed InTouch application. . . . 499
Export graphic language data for a published InTouch application. . . . 499
Export text to an existing dictionary file. . . . 500
Translate exported graphic language files. . . . 500
Translate exported graphic text dictionary files. . . . 500
Import translated language files. . . . 501
Import translated graphic dictionary files. . . . 501
Examples of graphic or object mismatch handling during language imports. . . . 504
Language data handling for galaxy operations. . . . 505
Export alarm comments for offline translation. . . . 506
Guidelines and Recommendations. . . . 506
About the alarm comments language file. . . . 507

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Export alarm comments from very large galaxies. . . . 507
Export all galaxy alarm comments. . . . 507
Exporting Alarm Comments by Area. . . . 508
Re-export alarm comments. . . . 509
Translate exported alarm comment language files. . . . 510
Import translated alarm comment language files. . . . 512
Testing the Language Switching Functionality at Run Time. . . . 513
About predefined language attributes. . . . 513
About attribute setting. . . . 513
About ViewApp navigation. . . . 514
About the Layout Editor and navigation. . . . 516
About the Layout Editor and navigation components. . . . 517
About ViewApp hierarchical navigation. . . . 518
About ViewApp hierarchical navigation. . . . 520
Trigger hierarchical navigation actions. . . . 520
About auto-fill navigation. . . . 521
About content types and navigation. . . . 523
Assign a content type. . . . 523
Action sequencing when content or other types are applied in ViewApp. . . . 524
About ViewApp custom navigation. . . . 525
Add custom navigation to a ViewApp. . . . 525
Configure properties for custom navigation items. . . . 526
ViewApp custom navigation rules. . . . 528
Hide assets in ViewApp navigation during runtime. . . . 529
About ViewApp navigation hierarchical display. . . . 530
About display of controls in the ViewApp navigation hierarchy. . . . 531
About predefined navigation attributes. . . . 533
Predefined navigation attribute rules. . . . 533
Use navigation attributes in action scripts. . . . 537
About client controls embedded into a graphic. . . . 540
Import client controls: options. . . . 542
Import client controls. . . . 542
Import previously exported client controls. . . . 542
Organizing Client Controls. . . . 543
Viewing and Changing the Properties of Client Controls. . . . 543
Example of changing a property of the TagPicker control. . . . 544
Binding Client Control Properties to Attributes or Element References. . . . 544
Example of data binding in the TagPicker control. . . . 545
Configuring Client Control Event Scripts. . . . 546
Example of configuring an event script for the TagPicker control. . . . 546
Animating Client Controls. . . . 547
Export client controls. . . . 548
Secure client controls. . . . 548
Including Dynamically Loaded Assemblies with the Client Control. . . . 548
Requirements for Both Inclusion Methods. . . . 549
Sample XML for a Dynamically Loaded Assembly List. . . . 549
XML Schema for the Dynamically Loaded Assembly List. . . . 549
Embedding the XML Manifest Resource in the Primary Assembly. . . . 550

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Including the XML Manifest Resource in an External Configuration File. . . . 550
Preventing Dynamically Loaded Assembly Import Issues. . . . 550
View additional client control information. . . . 551
View client control assemblies. . . . 551
View class name, vendor, and version of a client control. . . . 552
View objects and graphics referencing client controls. . . . 552
Ensure dynamically loaded assemblies for client controls are imported. . . . 553
Embedding Client Controls. . . . 555
Embed the TagPicker client control. . . . 555
OMI apps and widgets. . . . 555
Considerations when creating a ViewApp for the OMI web client. . . . 556
Prerequisites for using the OMI web client. . . . 556
Install a security certificate on Windows. . . . 557
Security considerations for the OMI web client. . . . 559
General OMI Web Client Limitations. . . . 560
Place the web client in an iFrame. . . . 560
Known Issues in the OMI web client. . . . 561
Industrial Graphics limitations in the OMI web client. . . . 561
Graphic element limitations. . . . 562
Animation limitations. . . . 562
Embedded widget limitations. . . . 563
Script limitations. . . . 563
Font limitations. . . . 565
OMI App limitations in the OMI web client. . . . 565
AlarmApp limitations. . . . 566
ContentPresenterApp limitations. . . . 566
GraphicRepeaterApp limitations. . . . 566
HistoricalTrendApp limitations. . . . 567
MapApp limitations. . . . 567
TitleBarApp limitations. . . . 570
WebBrowserApp limitations. . . . 570
Widget limitations in the OMI web client. . . . 570
Layout script limitations in the OMI web client. . . . 570
ViewApp namespace attribute limitations. . . . 570
OMI web client responsive layout limitations. . . . 571
About use of touch screens in ViewApp. . . . 571
Build a ViewApp that supports touch gestures during runtime 571
About ViewApp touch interactions. . . . 572
About touch interactions supported during runtime. . . . 574
Runtime touch interaction limitations. . . . 575
About InTouch Access Anywhere. . . . 576
About InTouch Access Anywhere architecture. . . . 576
Verify an InTouch Access Anywhere installation. . . . 577
Configure InTouch Access Anywhere. . . . 578
About InTouch Access Anywhere on MobileDevices. . . . 586
Configure InTouch Access Anywhere. . . . 587
InTouch Access Anywhere server limitations. . . . 595
Build an InTouch Access Anywhere Secure Gateway 597

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

02Build_ITAA_PostInstallation. . . . 598
About the InTouch Access Anywhere Secure Gateway Configuration Portal. . . . 603
About InTouch Access Anywhere Secure Gateway ports and SSL certificates. . . . 613
Configure Failover Gateways. . . . 619
Configure the built-In InTouch Access Anywhere web server. . . . 620
InTouch Access Anywhere Secure Gateway known limitations. . . . 623
About using InTouch Access Anywhere. . . . 624
About InTouch Access Anywhere Supported Browsers. . . . 624
About InTouch Access Anywhere and ViewApps. . . . 625
View an application with InTouch Access Anywhere. . . . 626
About device support for InTouch Access Anywhere. . . . 647

Chapter 4 About deployment. . . . 657

Prepare to deploy a ViewApp. . . . 657
Prepare to deploy a ViewApp: security. . . . 657
Prepare to deploy a ViewApp: galaxy status. . . . 658
Assign an engine to a ViewApp. . . . 658
Prepare to deploy a ViewApp: assign a ViewApp to an engine. . . . 659

About deploying a ViewApp. . . . 659
Deploy a ViewApp 660
Determine deployment status. . . . 663

Deploy an updated ViewApp. . . . 663

About undeploying a ViewApp. . . . 664
Undeploy a ViewApp. . . . 664

Chapter 5 Run a ViewApp. . . . 666

View deployed ViewApps. . . . 666
Launch a ViewApp. . . . 667
About the ViewApps Application Manager. . . . 668
Display a web ViewApp on a mobile device. . . . 669

About the InTouch web client. . . . 671

Launch a ViewApp in a web browser. . . . 671
Using ViewApp in a web client vs. on the desktop. . . . 672
Troubleshoot OMI web client connection issues. . . . 673

Launch ViewApp in the OMI Web Client using connected experience. . . . 675

Log in with AVEVA Identity Manager. . . . 676
Use Azure Active Directory credentials to sign on. . . . 676
Use Azure Active Directory credentials for secured write authentication. . . . 677
Use Azure Active Directory credentials for verified write authentication. . . . 679

Open a ViewApp. . . . 681

Search a ViewApp NavTree. . . . 681
Enable runtime navigation model searches. . . . 682
About search results. . . . 683

Process late data at runtime. . . . 685

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Enable historical playback . . . 685
About playback attributes . . . 687
Manually deploy the playback service . . . 688

Manage OMI applications . . . 689

Manage product licensing and user authorization . . . 689
Connected experience and operations control . . . 690
Configure operations control and connected experience . . . 690
Authentication and entitlement . . . 696
Authenticate with the AVEVA Identity Manager . . . 697
Single Sign-On (SSO) . . . 702
License mode . . . 702
Start and run the IDE . . . 703
Connect to a galaxy . . . 703
Connect to a galaxy with no security configured . . . 704
Connect to a galaxy not using Operations Control connected experience (configuration mismatch) . . . 705
Sign out of the IDE . . . 705
Start Application Manager . . . 705
Launch a ViewApp using the connected experience . . . 705
Launch ViewApp in the OMI Web Client using connected experience . . . 708
Sign out from an OMI ViewApp . . . 709
Switch OMI ViewApp users . . . 710
Secured and verified writes . . . 710
Deploy scenarios . . . 710
Change Operations Control connected experience mode . . . 711
Using the OCMC with connected experience . . . 712

Galaxy migration to support connected experience . . . 712

Licensing and the OMI web client . . . 712

About ViewApp security . . . 713
About Galaxy security . . . 714
Configure Galaxy security . . . 715
About Galaxy security groups . . . 716
Manage security roles . . . 716
About the default security role . . . 718
About the Credentials tab . . . 718
Apply log in security . . . 722
About security attributes . . . 724

Manage industrial graphics . . . 727
Import and export graphics . . . 727
Import graphics from .aaPKG files . . . 728
Export graphics . . . 728
Programmatically import and export industrial graphics . . . 729
Implement and use the GraphicAccess API . . . 730
Exporting and Importing Overridden Text Strings . . . 730
Configure security for graphics . . . 733

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

AVEVA

Chapter 1

Welcome to AVEVA Operations Management Interface

AVEVA Operations Management Interface (OMI) extends the core foundation of Human Machine Interface (HMI) and Supervisory Control and Data Acquisition (SCADA) products. You can use AVEVA OMI to create visual applications (ViewApps) that incorporate the functionality needed by both operational and information technology teams in a production environment.

This document is organized by workflow:

- Create standards: Design standards
- Build applications: Build applications
- Deploy applications: About deployment
- Run an application: Run a ViewApp
- Manage applications: Manage OMI applications

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Chapter 2

Design standards

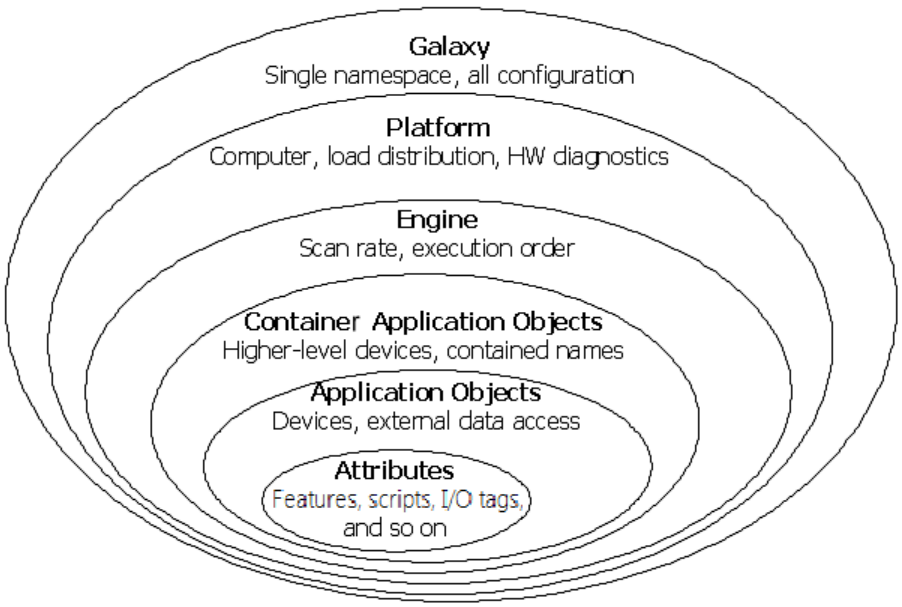
During the design phase, you can create a set of reusable graphics and objects that serve as templates for the content that appears in a ViewApp. Instead of creating unique graphics and objects for each ViewApp, you can configure these reusable graphic and object templates to suit the requirements of each ViewApp.

Galaxy Infrastructure	Configure components: Galaxies	About the integrated development environment (IDE)	Platform common services	
Standards	Automation objects	Object Wizards	Symbol Wizard Editor	
Visual Styles	Galaxy Style Library	Element Styles	Numeric Formats	Quality and Status Styles

Configure components: Galaxies

A Galaxy represents your entire production environment, including all computers and components that run your view application. A Galaxy is a collection of graphics, objects, engines, templates, and attributes you define as part of a specific view application. Persistent information about your project's collection of graphics and objects is stored in a database called the Galaxy Repository.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



When you start the IDE, you must select an existing Galaxy or create a new Galaxy. You cannot open the IDE without opening a Galaxy.

Create a Galaxy

Creating a new Galaxy requires you to specify a Galaxy Repository (GR) node name and the name of the Galaxy. The Galaxy database is created and is ready for you to connect to and use.

Guidelines for Creating a New Galaxy

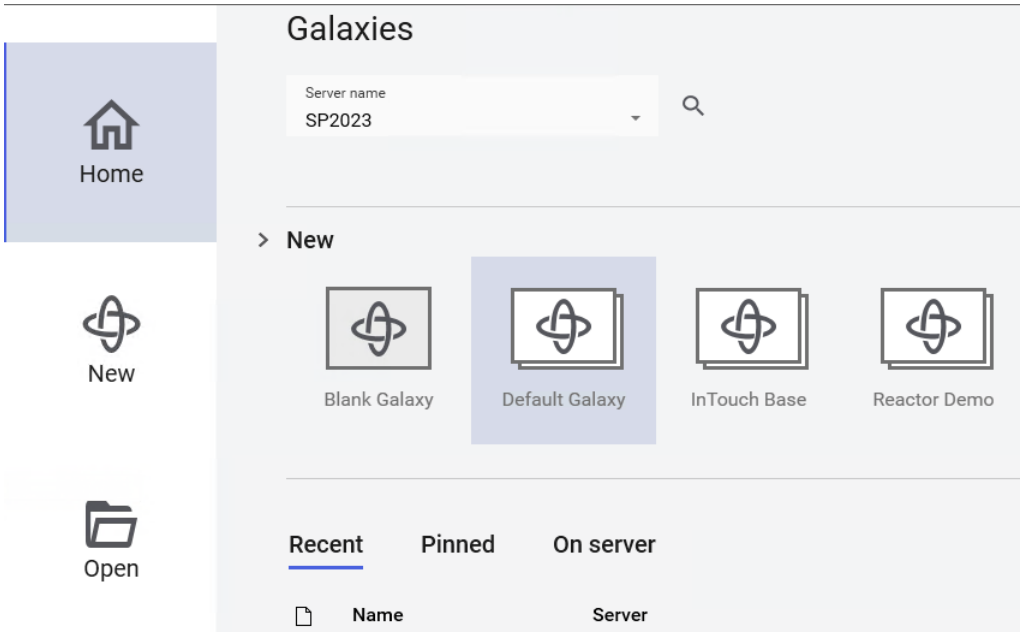
- Activate a valid development license before opening the IDE.
- You can only create a new Galaxy on a computer with the Bootstrap and the Galaxy Repository software installed.
- To be able to create a new Galaxy, TCP/IP must be enabled on the computer hosting the SQL Server database.

  The TCP/IP protocol setting can be verified with **SQL Server Configuration Manager**. See the *System Platform Installation* Guide for instructions on enabling the TCP/IP protocol for SQL Server.
- New Galaxies are created without security. To learn more about setting security for your Galaxy, see Working with Security.
- A computer name, as well as the name of the domain on which the computer exists, should include only English language characters, alphabetic and numeric. Use of characters from other language character sets that are not also included in the English language character set can result in a failure to create the new Galaxy.

To create a new Galaxy

1. From the **Start** menu, go to the **AVEVA System Platform** folder, and then select **System Platform IDE**. The **Galaxies** dialog box appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. If the **Server name** shown is not the Galaxy Repository server where you want to create the galaxy, either type the correct server name or click the down arrow and select it from the list.
3. In the **New** area, select the template to use for the new galaxy:
   - **Blank Galaxy**: Creates a baseline Galaxy that contains only base template objects. This is intended primarily as a recovery mechanism for recreating a damaged galaxy. After creating the baseline galaxy, you can import existing objects into it from aaPKG files.
   - **Default Galaxy**: This is the *recommended* starting point for a new Galaxy that will use AVEVA OMI run-time applications (ViewApps). It creates default screen profiles, templates, a basic equipment model, and an example AVEVA OMI ViewApp that provides an introduction to key features.
   - **InTouch Base**: Creates an InTouch HMI Galaxy. This is mainly for use if you are already using InTouch HMI applications, and can be used to convert legacy InTouch applications to managed applications.
   - **Reactor Demo**: This is the same as the **InTouch Base** option, but includes a demonstration run-time application, "$ReactDemo," and all the configuration objects and graphic files used to create the demo. $ReactDemo is derived from the InTouchViewApp object.

   If you have other galaxy templates available, click **More Templates** and select the one you want.

   The **New Galaxy** dialog box opens.
4. Type the name for the new Galaxy. The name can be up to 32 alphanumeric characters, including _ (underscore), $, and #. The first character must be a letter. It cannot contain a blank space

   Note: You cannot use the following reserved names as Galaxy names: Me, MyContainer, MyArea, MyHost, MyPlatform, MyEngine and System. You cannot use a name that conflicts with an existing object in the template.
5. To open the galaxy after it is created, make sure **Connect to this Galaxy** is selected.
6. Click **Create**. The **Create Galaxy** dialog box opens and shows the progress of the Galaxy database being created.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: If you use a backup file that was created in a previous version of Application Server as the galaxy template, and if Enhanced Security mode is in effect, you will be prompted to enter SQL SysAdmin credentials before proceeding.

7. After the Galaxy is created, click **Close**. If **Connect to this Galaxy** was checked, the Galaxy opens in the IDE. If it was not, you can open the Galaxy and begin creating your application. You must be a member of the aaConfigTools OS group to connect to a Galaxy, or you must be running the IDE as administrator. For more information about opening a Galaxy, see Connect to a Galaxy.

About Galaxies

A Galaxy database resides on a single network computer. A Galaxy database can reside on any computer on your network with the SQL Server, Bootstrap, and Galaxy Repository software installed. But, you cannot store parts of a Galaxy database on several computers.

A Galaxy Repository (GR) is the name of the single computer where a Galaxy database is located.

You can deploy Galaxy components, such as platforms and engines, on multiple computers to share the work load while applications are running.

A Galaxy's namespace is the set of unique object and attribute identifiers. The namespace and the values of each of its identifiers define a Viewapp, and can be accessed by clients of the configuration system as well as the Application Server Message Exchange in a deployed system.

A key benefit of a namespace is that it enables objects and process data to be referenced by scripts and animation links from any computer in a Galaxy without the reference needing to specify the object's location.

Galaxies also include security, which is inactive by default. Using security enables you to limit what users can do. You can add more users, security roles, and security groups later if you want.

Creating a new Galaxy requires you to specify a Galaxy Repository (GR) node name and the name of the Galaxy. The Galaxy database is created and is ready for you to connect to and use.

Connect to a Galaxy

You must connect to a Galaxy before you can perform work in it. To connect to a Galaxy, you must be a member of the **aaConfigTools** OS group *on the GR node*. To add a user to the **aaConfigTools** group, use the Windows Control Panel. If you are not a member of **aaConfigTools**, you can instead start the IDE as administrator and then connect to the Galaxy.

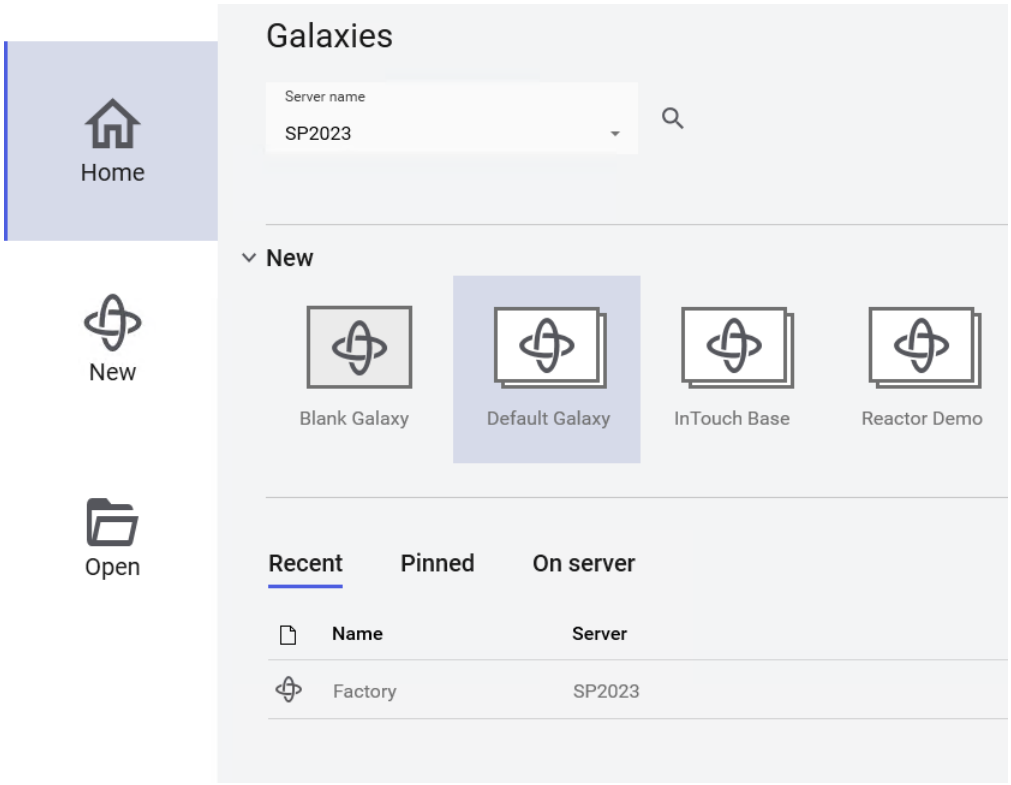
Important: You will not be able to connect to the Galaxy if you are not a member of the **aaConfigTools** group on the GR node. In a domain-based network, the **aaConfigTools** OS group must exist on the GR node.

If security is enabled for an existing Galaxy, you cannot open it without logging in, even if you are a system administrator or member of the **aaConfigTools** group. If you do not have log on rights to a Galaxy, you cannot log in to that Galaxy. For more information about Galaxy security, see 01Standardize_Security_Configure Security.

To connect to a Galaxy

1. From the **Start** menu, go to the **AVEVA System Platform** folder, then select **System Platform IDE**. The **Galaxies** page appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. If the galaxy you want to connect to is shown, select it, then select the folder icon that appears to its right.
3. If the galaxy is not shown, but is hosted on the computer you are on, or on a network server that you have defined to System Platform (see below):
   a. Select **On server**.
   b. If necessary, in **Server name**, click the down arrow and select the server that hosts the galaxy.
   c. Select the galaxy, then select the folder icon that appears to its right.
4. If the galaxy is on a server you have not yet defined:
   a. Click **Open**. On the **Open** page:
   b. Select **Add a server**.
   c. In **Add a network server**, type the server name. Select **Add**.
   d. Under **Network Servers**, select the new server name.
   e. Select the galaxy from the list that appears on the right side of the page, then select the folder icon that appears to its right.

   If security is not defined for the galaxy, the IDE opens and you can begin working.

   If the galaxy does have security defined, you must log in.

To log in to a galaxy

1. On the **Sign in to <galaxy>** dialog box, type your user name and select **Next**. If you need to include your domain,enter your name in the format *<Domain>\<User>*.
2. Type your password and select **Sign in**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Once the Galaxy opens in the IDE you can start working with your Galaxy. The IDE provides different ways to view the structure of the Galaxy and how it will be deployed to run time. See IDE Application Views for more information.

View the configuration file

A configuration file, view.exe.config, is used define various settings for use by AVEVA OMI ViewApp applications at runtime. Most of these settings are used to to load different runtime dependent assemblies, and do not require any changes. The location of view.exe.config will vary, depending on the selected installation directory. The default location for the file is:
C:\Program Files (x86)\ArchestrA\Framework\Bin\ViewAppFramework

There are two settings in the view.exe.config that you may need to alter. One is used to enable single sign-in (SSO) for AVEVA OMI apps that are registered with the AVEVA Identity Manager (AIM), and the other is used to enable/disable layout multithreading. Configuration of the file is required to enable single sign-on via AIM, or to disable multi-threading if needed for compatibility with older controls that were not written to take advantage of multithreading.

Set single sign-on for OMI ViewApps

Single sign-on for OMI ViewApps is configured by setting the values for three keys in the view.exe.config file (file location: C:\Program Files (x86)\ArchestrA\Framework\Bin\ViewAppFramework):

```
<add key="TokenServiceBase" value="https://your_token_server/IdentityManager" />
<add key="ClientId" value="your_client_id" />
<add key="Scope" value="openid profile system offline_access" />
```

These settings are commented out in the view.exe.config file, so you will need to uncomment them after setting the values.

Turn off multithreaded processing for OMI ViewAPP

By default, the OMI ViewApp is a multithreaded application, where each layout within the ViewApp runs on its own UI thread. Thus, each layout in a multi-screen ViewApp runs on a unique thread. For example, a ViewApp configured for three screens will run on three separate threads when this setting is enabled. This is the default, and should not normally be changed.

However, if your ViewApp contains controls that are not optimized for multithreading, such as controls that were used for legacy AVEVA InTouch applications, then you can disable the setting to avoid unpredictable results at runtime. With the setting disabled, all layouts in a ViewApp will run on the same thread. To disable multithreading, change the value for the key from true to false.

```
<add key="LayoutMultiThreadingModeEnabled" value="true" />
```

About the integrated development environment (IDE)

The integrated development environment (IDE) is the single tool from which all objects are configured and deployed to target computers. You work from the IDE to create, configure, and maintain the objects that comprise your application and the underlying infrastructure that supports your application.

Using the System Platform IDE, you can import new types of objects in to the Galaxy Repository, configure new ones, and deploy them to computers in your network. Multiple users can work concurrently on different sets of objects from different System Platform IDEs.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When you start the IDE, you must select an existing Galaxy or create a new Galaxy. You cannot open the IDE without opening a Galaxy.

Install and Activate Your License

Before you can open the IDE, you must have a valid license. For information about licensing, see "License Installation and Activation" in the *AVEVA System Platform Installation Guide*. If you need additional information, see the *AVEVA Enterprise Licensing Guide*.

If the Galaxy Repository process (aaGR) cannot obtain a license for any reason, it will send requests periodically to the license server. These attempts to acquire a license will continue until either a license is acquired successfully, or the aaGR process is stopped. This retry process also applies if access to a license is lost after it was acquired.

System Platform supports both perpetual and Flex licensing. See Flex Licenses View and Allocating AppEngines to Platforms for information about using Flex licenses.

System Platform also supports AVEVA Operations Control and the connected experience. For more information, see Manage product licensing and user authorization..

Manage Security Certificate Verification

The System Platform IDE may take 30 seconds to open on a system that is not connected to the Internet. When the IDE starts, the operating system attempts to verify the digital certificates for internal components against a Certificate Revocation List (CRL) located on a public website. If your system cannot access the public site within 30 seconds, the IDE startup process resumes and completes.

To maintain a better security profile by checking components against a CRL, keep your internet connection enabled and allow the verification to proceed.

To avoid the potential delay, in **Internet Explorer** or through **Control Panel**, open **Internet Options**, **Advanced Options**, and uncheck the **Security** option to **Check for publisher's certificate revocation**.

View the Galaxy

Once you connect to a Galaxy, the IDE opens. When the IDE opens, it shows two windows that contain tools that you use to build and modify a galaxy, and three application views that let you see the structure of your Galaxy. The tool windows are:

- **Template folder:** this contains different object templates that you use to build a digital twin of your physical plant. The digital twin is a set of objects that interfaces with PLCs, RTUs, or other controllers. Expand the top level folders to see the different templates in the folders. For more information, see Template Folder. For a complete discussion of template and instance objects, see About Templates and Instances.
- **Visualization folder:** this contains different graphic models that you use to construct the run-time graphical representation of your physical plant, including gauges, equipment, and valves. The graphic models in the folder include Industrial Graphics and Situational Awareness Library graphics. For more information, see Visualization Folder.

When the Galaxy opens, the application views are shown by default in the lower left pane. See IDE Application Views for more information.

- **Model view:** this shows the object relationship to the automation scheme layout. The objects are organized into Areas that typically represent the physical plant layout.

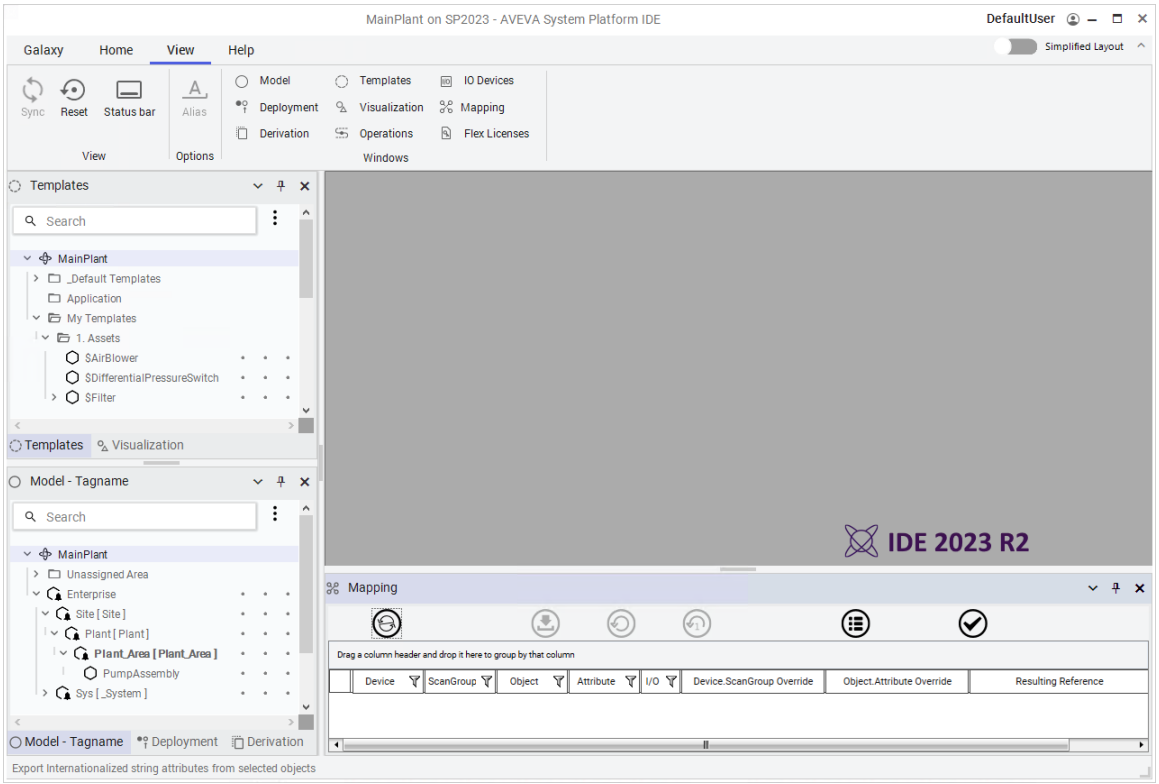
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **Deployment view:** this shows the object relationship to the computers that comprise the deployed system that the objects run on.
- **Derivation view:** this shows the derivation path from base template to the instances. This view allows a user to see all object instances that were based on a given template. All templates and instances appear in this view.

You can open additional application views:

- **IO Devices view:** this shows the object relationship to scan groups and DI objects. This view lets you assign objects that were configured for I/O automatic assignment in the Object Editor to scan groups, and assign scan groups to Device Integration (DI) objects. The IDE then builds the I/O references for each object.
- **Mapping view:** this shows the relationships of objects and attributes to scan groups and DI objects. This view lets you see, validate, and edit the I/O references of object attributes that have been automatically configured.
- **Operations view:** this shows the results of validating the configuration of objects.

A status bar at the bottom of the IDE window shows messages, user name, Galaxy name and node, and license information. You can turn off the Status bar by clicking **Status bar** on the **View** ribbon.



Template Folder

The Template folder lists template folders, which contain object templates. The Template folder shows a tree view of template categories in the Galaxy. Expand the folders to see their contents.

A new Galaxy is automatically populated with the content defined by the cab file (template) that was used to create it. At a minimum, the newly-created Galaxy will include base templates.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- The Default Galaxy includes first level derived templates in addition to base templates, to simplify the process of building your Galaxy, and a sample AVEVA OMI ViewApp.
- The Blank Galaxy includes only base templates. See About base templates for additional information.
- The InTouch Base and Reactor Demo cab files include only the base templates needed to support an InTouch HMI application.

Visualization Folder

The Visualization Folder lists graphic-related content. The Visualization Folder shows a tree view of the graphic folders. As with the Template Folder, a new Galaxy is populated with the content defined by the cab file used to create it. Expand the folders to see their contents.

- The Default Galaxy includes several graphic libraries, including the Industrial Graphics Library, the Situational Awareness Library, and an icon library. The Default Galaxy also includes various Screen Profiles, Layouts, Apps and Controls.
- The Blank galaxy includes only a basic set of Apps and Controls.

The InTouch Base and Reactor Demo include the Industrial Graphics and Situational Awareness Libraries. The Reactor Demo also includes additional graphics used for the InTouch HMI reactor demo app.

AVEVA CONNECT

AVEVA Connect can be configured within the Visualization folder to provide centralized storage and access for reusable Industrial Graphics. You can use these graphics as templates to simplify building applications. For more information, see Using Industrial Graphics in AVEVA Connect.

IDE Application Views

Base templates and non-base templates are included with Application Server. The templates are automatically imported into the IDE from the Cab file you use to create a Galaxy.

Base templates appear in the Template folder and in the Derivation view with a $ as the first character of their name. You cannot directly modify base templates but you can use them to create your own objects or derived templates.

When you move from one view to another, the selected object in the first view is selected in the second view, if the object exists in that view. For example, templates are not shown in the Deployment view, Model view and IO Devices view.

You can open views from the **View** menu or by using keyboard shortcuts.

View	Shortcut
Model View	Ctrl+Shift+M
Deployment View	Ctrl+Shift+D
Derivation View	Ctrl+Shift+R
Template Folder	Ctrl+Shift+T

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

View	Shortcut
Visualization Folder	Ctrl+Shift+V
Operations View	Ctrl+Shift+O
IO Devices View	Ctrl+Shift+I
Mapping view	Ctrl+Shift+G

Application Object Icons

All IDE views (Templates, Visualization, Model, Deployment, Derivation, and IO Devices) allow you to choose whether to show objects in tree (hierarchical) view or in a list (flat) view. By default, objects are shown in a tree. To change to the view selection, click the three vertical dots at the top right of the view window. Default objects are shown in bold.

For objects shown in any view, icons to the left of the object name help to identify the object type.

System object icons

Galaxy (project)

GRPlatform

WinPlatform

Cloud Platform

AppEngine

Primary (1) Redundant AppEngine

Backup (2) Redundant AppEngine

Area

Application Object (asset)

View engine icons

View Engine

Webview Engine

ViewApp icons

OMI ViewApp

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

InTouch ViewApp

Device integration icons

DDE SuiteLink Client
InTouch Proxy
OPC Client

Redundant DI

Scan group

Visualization icons

Screen profile

Layout

Symbol

External content

Namespace

Client control

OMI app

Widget

Status Icons

Status icons are shown to the right of the object names. These icons are arranged in three columns are used to show validation state (first column), editing state (middle column), and deployment state (right column) of the objects.

Each status icon can also show a tooltip, identifying the icon. Hover your cursor over the icon to view the tooltip.

Validation Status	Editing Status	Deployment Status
No error	Checked in	Deployed
Good	Checked out by me	Not deployed
Warning	Checked out by other	Pending update
Error	Protected	Pending software update

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Redundant AppEngine Validation Status

See Working with AppEngine Redundancy for more information about working with redundant AppEngines.

Both redundant AppEngines are not deployed

This AppEngine is not deployed, but its redundant partner is deployed.

This AppEngine is deployed, but its redundant partner is not deployed.

Deployed, but the redundant AppEngines have a pending configuration update.

This AppEngine has a pending configuration update, but its redundant partner is up-to-date.

Both redundant AppEngines have pending software updates.

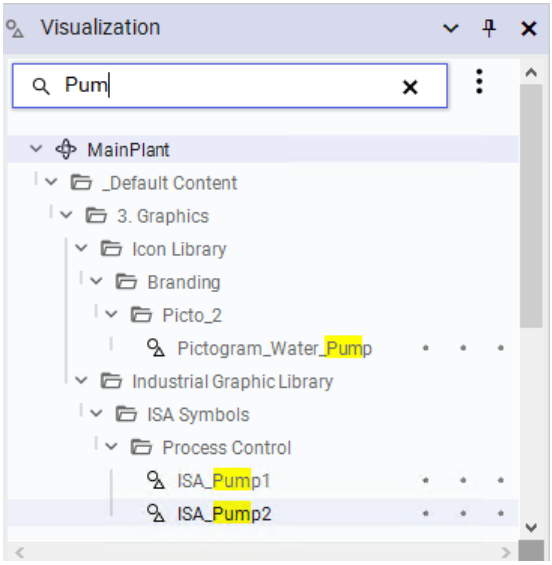
This AppEngine has a pending software update, but its redundant partner is up-to-date.

This AppEngine is up-todate, but its redundant partner has a pending software update.

Searching in a View

You can search in all IDE application views except for Mapping view and Operation view.

To search, just type the string to search for. As you type each character, the view expands the tree as needed to show items that match your entry, and highlights the matches.



Note: In Model view, the search includes either object tagnames or aliases, whichever is currently shown.

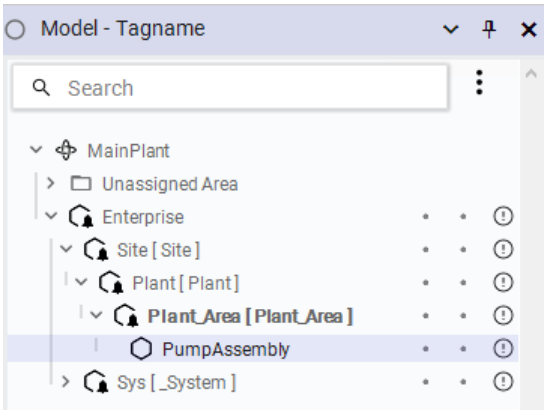
Model View

The Model view organizes your Galaxy by Area object assignment. It shows objects in terms of their physical or containment relationships, which can be organized through a folder structure. This Model view most accurately

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

represents an application perspective of the processes that users are emulating: for instance, specific process areas, tanks, valves, pumps and their relationships based on containment. For more information about containment, see Create Contained Templates.

Note: You must undeploy an object before reassigning it to another object.



The tree structure acts like a standard Microsoft Windows Explorer tree. Initially, it shows only the Galaxy name and the Unassigned Area folder.

In the Model view, all objects are grouped by areas and by containment relationship. The Model view shows these relationships in the following ways:

- The top of the tree is the Galaxy.
- Top-level Areas are shown under the Galaxy.
- Within each Area, contained Areas are listed. Areas support hierarchical composition; that is, they support sub-Area construction. Areas can be nested only 10 levels (after the sub-area is 10-levels deep, you cannot add another sub-level).
- Objects that belong to an Area are listed under the Area.
- Objects contained by other objects are listed under their respective containers. Multiple levels are allowed. For more information about containment, see Create Contained Templates.

  Note: Contained objects belong to the same Area as the object that contains them.

  Some objects' hierarchical, or contained, names are truncated if you have multiple levels shown. To view the entire hierarchical name, select the object and select **Properties** on the **Home** ribbon. The entire hierarchical name is shown in the **Properties** dialog box. For more information about hierarchical names, see Using Contained Names.
- Objects that currently do not belong to an Area are listed under Unassigned Area. Containment relationships between parent and child objects are shown there.
- In each branch of the tree, objects are listed in alphabetical order. Default objects are shown in bold.

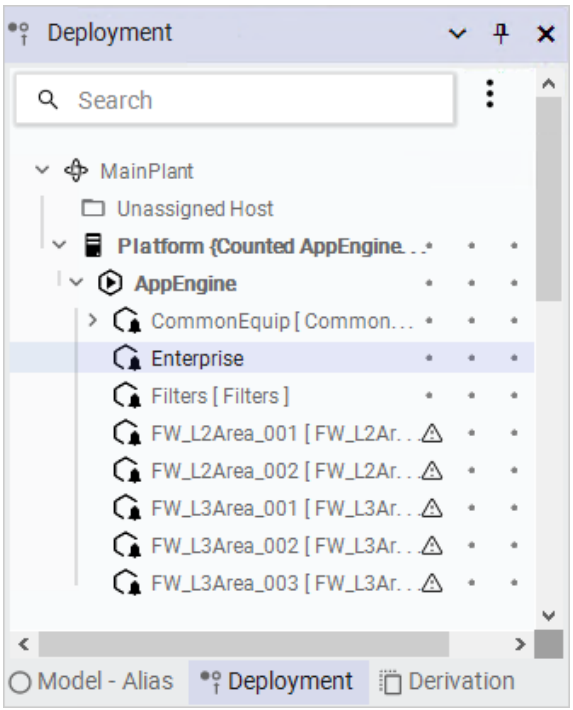
  To assign an object to another, drag it onto the host object. If the object is an inappropriate assignment match, the international Not symbol appears. To unassign an object, drag it to the **Unassigned Host** folder.

Model view shows either the tagname or the alias name for each object. To change which is shown, right-click any object in the view and select **Alias** from the context menu.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Deployment View

The Deployment view shows which object instances reside on which computer (platforms), and AppEngine assignments to platforms. In the System Platform environment, the physical location of object instances is not required to approximate the real-world environment it models. The Deployment view does not need to reflect your physical plant environment. If you are using Flex licensing, your available licenses will help you determine how to assign AppEngines to platforms. Deployment view shows instances only in terms of their assignment relationships. This view enables you to organize those objects through a folder structure.



The tree structure acts like a standard Windows Explorer tree. It is initially divided into two levels: <Galaxy Name> and the Unassigned Host folder.

In the Deployment view, objects appear in a tree according to their distribution relationships in a multi-node system in the following ways:

- The top of the tree is the Galaxy.
- WinPlatforms are shown under the Galaxy.
- Under each WinPlatform, assigned AppEngines are listed.
- Under each AppEngine, assigned Areas and objects configured to communicate with PLCs, RTUs, etc. such as OPCClient objects, are listed.

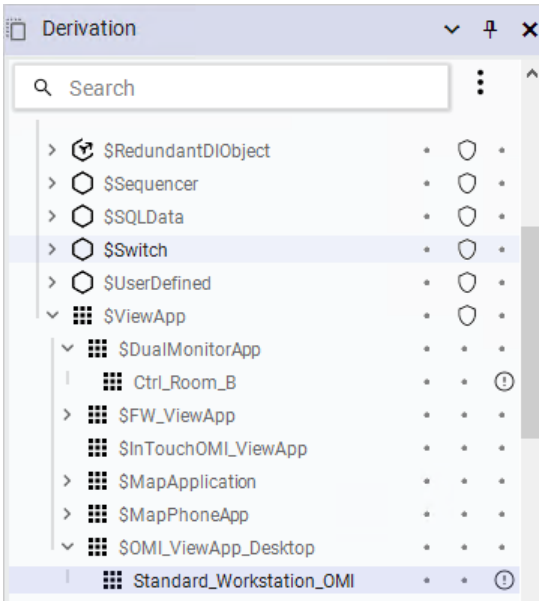
  Note: DINetwork Objects, such as ABCIP and SIDIRECT, have been deprecated. Use an OPCClient object instead.

- Under each Area, assigned ApplicationObjects are listed.
- Under each ApplicationObject, contained ApplicationObjects are listed. Multiple levels are allowed.
- Unassigned objects are grouped together in the **Unassigned Host** folder. Area and containment relationships are maintained in this view.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Derivation View

Derivation view shows objects and templates in terms of their parent/child relationship. An object derived from another object appears in a hierarchy level under it.



The tree structure acts like a standard Windows Explorer tree, and initially is divided into three levels: <Galaxy Name>, <Used Base Templates>, and the **Unused Base Templates** folder.

In the Derivation view, objects appear according to their parent-child relationship in the following ways:

- The top of the tree is the Galaxy.
- Base templates with associated child objects, either derived templates or instances, are shown under the Galaxy.
- Under each base template, derived templates and instances created from the base template are listed. Multiple levels are allowed. Instances created from derived templates are listed under their parents.
- Templates with no associated derived templates or instances are grouped together in the **Unused Base Templates** folder.

Objects with names that start with a "$" are templates, either base or derived. Under each branch of the tree, child objects are listed in alphabetical order.

As in other views, dragging one object onto another in the **Derivation view** associates the two objects based on the predefined rules of the object types. For example, you can drag ApplicationObjects onto other ApplicationObjects but you cannot drag ApplicationObjects to an Engine.

IO Devices View

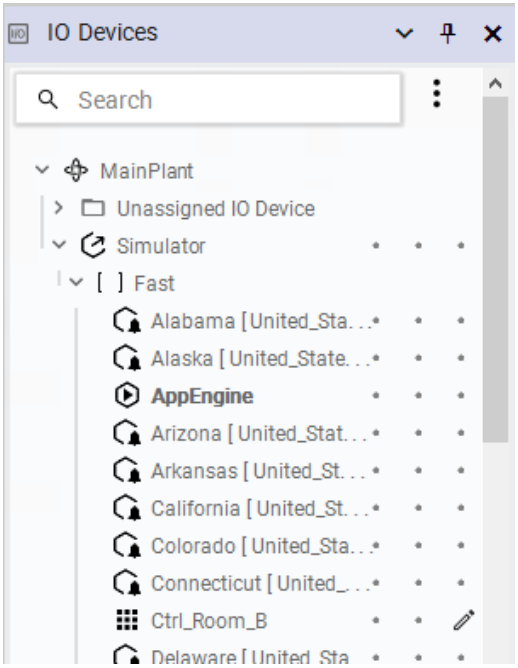
The **IO Devices** view is used for I/O auto assignment and shows the relationship of system objects and application objects to scan groups and Device Integration objects. Before using this view, system and application objects must be configured to use I/O auto assignment by adding one or more attributes to the object and activating the I/O feature. See Edit Objects and Using I/O Auto Assignment for additional information.

You will also use the Object Editor to add scan groups to DI objects (OPCClient, DDESuiteLinkClient, or RedundantDIObject) before using the **IO Devices** view.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Once you have configured system and application objects for I/O auto assignment, and added scan groups to at least one DI object, you will assign the system and application objects to scan groups in the **IO Devices** view. This links auto-binds the system and application objects to PLCs, through the DI objects.

To open **IO Devices** view, select **IO Devices** on the **View** ribbon.



System and application objects appear as child objects to scan groups, and scan groups appear as child objects to DI objects.

Note: Only instances, and not templates, appear in the **IO Devices** view.

Device linkage, which is established by assigning a system or application object to a scan group, ensures that all of the object's attributes that have been prepared for automatic I/O assignment will match up with a corresponding input source or output destination in the scan group. The I/O references are then referred to as auto-bound.

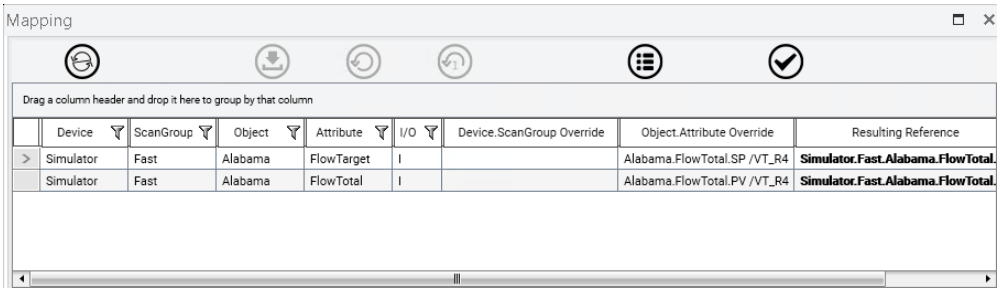
Important: Objects must be undeployed before they can be assigned to a scan group.

System and application objects that are not yet assigned to a scan group are placed under the folder "Unassigned IO Devices." To assign objects to a scan group, and thus allow I/O automatic assignment to occur, simply select objects in the Unassigned area, and drag and drop them to the applicable scan group. Once an object has been assigned to a scan group, the I/O reference for each I/O attribute is automatically configured.

Mapping View

The **Mapping** view shows I/O references for attributes that have been configured for I/O auto assignment, and that have been assigned to a scan group through the **IO Devices** view. To open this view, select **Mapping** on the **View** ribbon. The **Mapping** view will also open if you select a DI object, scan group, or object assigned to a scan group in the **IO Devices** view.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



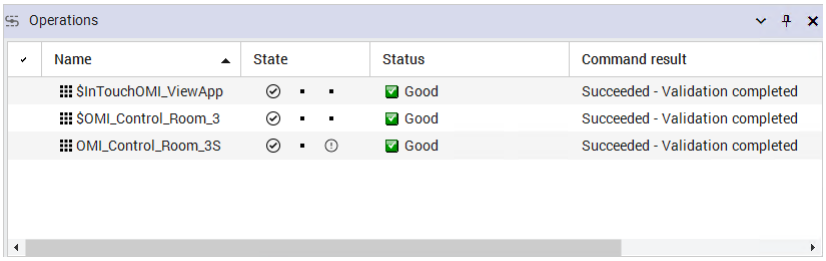
The **Mapping** view shows I/O auto assigned attributes with resolved I/O references for objects that have been assigned to scan groups. The **Mapping** view lets you enter override values to change the I/O reference. You can also validate the I/O references.

You select the references that are displayed in the **Mapping** view by selecting objects in the **IO Devices** view. Manually configured references are not displayed in the **Mapping** view, nor are attributes associated with objects that are not assigned to a scan group. Only I/O references for the object or group of objects currently selected in the **IO Devices** view are displayed. You can select DI objects, scan groups, and application and system objects in the **IO Devices** view. If only the top level item in a device hierarchy is selected, all subordinate items are automatically selected. If, however, you select a subordinate item along with the top level item, only I/O references for the selected items are displayed. You can select multiple items at different hierarchical levels.

Note: Selecting a subordinate object will exclude non-selected objects within the device hierarchy, even if the parent object is selected.

Operations View

The **Operations** view shows the results of validating the configuration of objects. You may need to open it before you see it. On the **View** ribbon, select **Operations**.



During the validation of an object, its icon and name appear with the state and status of the object and the status of the operation.

Important: You can validate both templates and instances if they are checked in.

- The icons in the **State** column show the current state of the object. See Application Object Icons for the meaning of each icon.
- The status of the object (**Status** column) is shown with an icon and a descriptive word or phrase.

When validation is complete, the **Command Result** column shows a "Succeeded" or "Failed" message and additional information about the validation results. For more information about validating objects, see Validate objects.

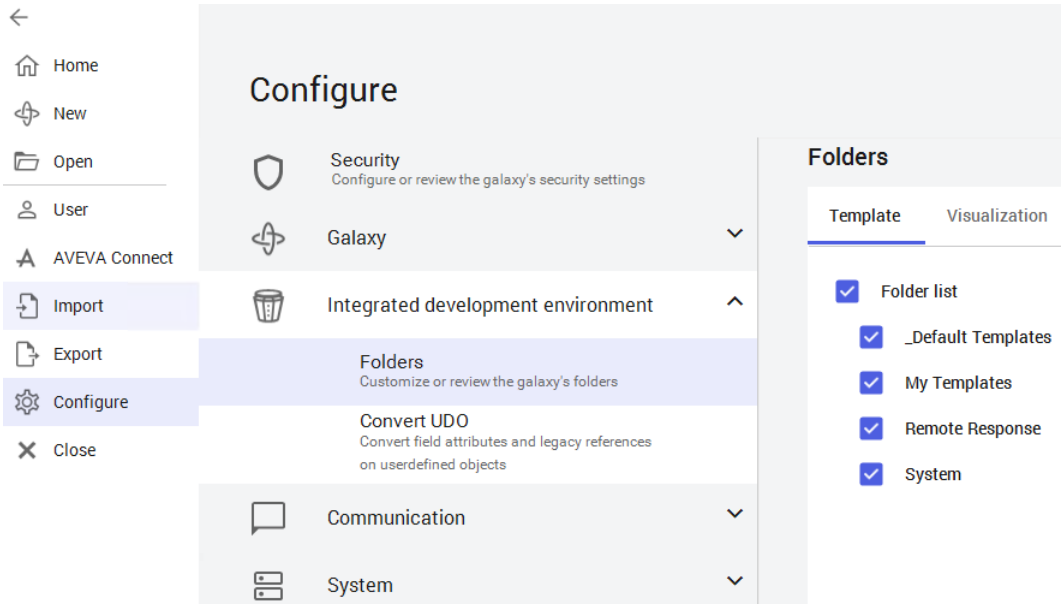
Note: You can validate all objects in the Galaxy by running the Validate operation on the Galaxy object. In that case, Command Result messages are shown after all objects in the Galaxy are validated.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The **Operations view**, like the **Template folder** and **Applications views**, is also updated as the status and conditions of objects in the Galaxy change.

Customize Your Workspace

You can dock, float or hide the different IDE views and folders to customize your workspace. You can also select which folders will be shown by default for the Galaxy by selecting **Galaxy** from the menu, then select **Configure**, and then navigate to **Folders** under **Integrated development environment**.



Dock Views

To dock a view to a new location:

1. Place the cursor on the title bar of the view (or folder) and hold down the left mouse button.
2. As you drag the view from its position, blue directional arrows appear on the screen. While continuing to hold down the mouse button, position the cursor on one of the arrows. The position on the screen that correlates with the arrow will be highlighted.
3. Release the button when the area where you want to dock the view is highlighted. The view will snap to highlighted area.

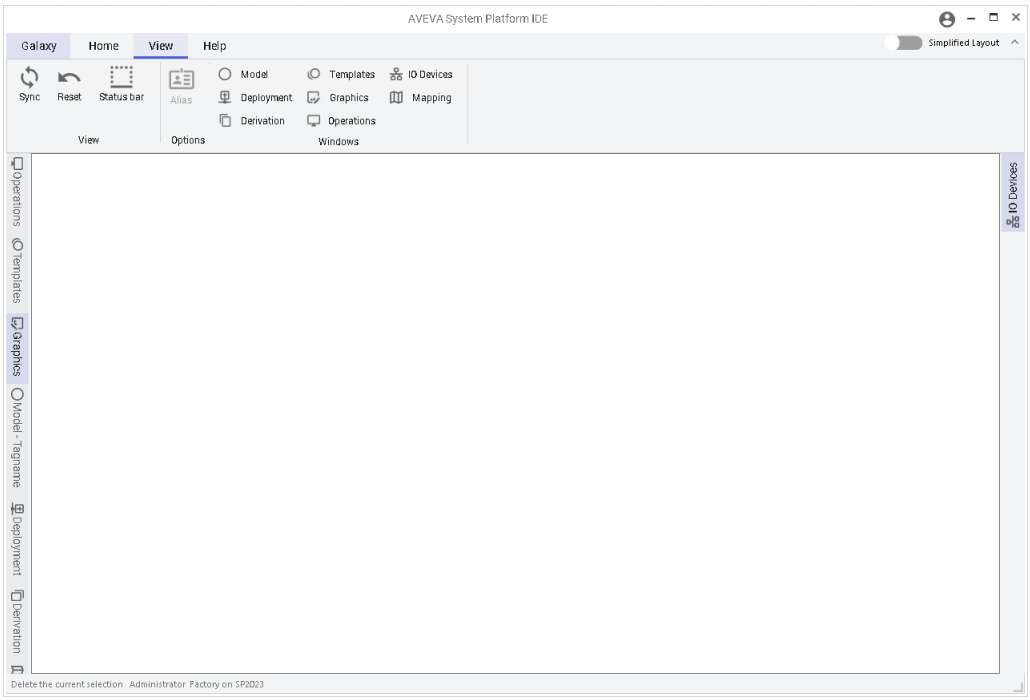
Float Views

You can also float a view. With your cursor over the view you want to float, click the arrow. On the menu that appears, click **Floating**.

Hide Views

To hide a view, click the **Pin** icon. The views "hide" as tabs on the side of the window. The Operations and Mapping views hide at the bottom of the window, and the IO Devices view hides at the right side. The Visualization and Template folders, and Model, Derivation, and Deployment views are at the left side of the window.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



When you move the mouse over the tab, the view expands into the workspace.

Reset the Workspace

You can easily reset the IDE workspace and restore views back to their default locations.

To return the views to the default docking

- On the **View** ribbon, select **Reset**.

Synchronize Views

You can specify that a selected object stay selected as you move through the views. In any of the views, select the object you want to synchronize. On the **View** ribbon, select **Sync**. Now as you move from one view to another, that object stays selected.

You can also configure an option in your user information to automatically keep the same object selected as you switch between the **Model**, **Deployment**, and **Derivation** views. See Configure User Preferences for details.

Configure User Preferences

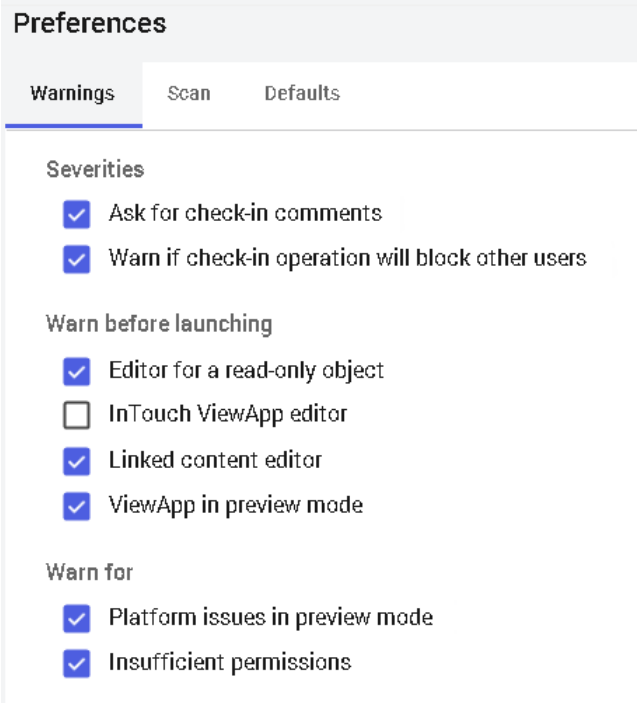
You can configure options for a Galaxy, including prompts for check-in comments and user defaults. These user options apply to the currently open Galaxy and do not change options for other Galaxies.

If you specify a security group, that security group must already exist. For more information about security and security groups, see Working with Security.

To configure user information

1. From the ribbon, select **Galaxy**, then **User**, then **Preferences**. The **Preferences** pane appears. The pane has three tab pages.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. On the **Warnings** page, select the check box for each option you want:
   - **Ask for check-in comments** to be prompted to type comments when checking in templates and objects.
   - **Warn if check-in operation will block other users** if you want to be warned if this occurs.
   - **Editor for a read-only object** to see a prompt that informs you if you are opening an instance or template as read-only. The prompt warns you if you open an instance or template while someone else is working on it. If someone else is working in the instance or template, you cannot make changes.
   - **InTouchViewApp editor** to see a prompt each time you attempt to edit an InTouchView application instance. You can edit the associated template or cancel the operation. If you do not select this check box, the request to edit the InTouchViewApp instance is automatically redirected to the associated template.
   - **Linked content editor** to see a prompt each time you attempt to edit a Visualization folder item, such as a graphic or layout, that is linked to an object. You can edit the linked content or cancel the operation. If you do not select this check box, the linked content opens in its editor (graphics in the Graphic Editor, layouts in the Layout Editor, etc.) without a warning, and you will be able to make changes to the content. If you then save the content, the item in the Visualization folder is updated with your edits, and the changes propagate to all objects that link to the content.
   - **ViewApp in preview mode** to restore the first warning message that appears after requesting a preview of a ViewApp. You need to select this prompt to display the message again after you select the check box on the warning message to hide the message.
   - **Platform issues in preview mode** to restore the warning message that appears after requesting a preview without the WinPlatform deployed. You need to select this prompt to display the message again after you select the check box on the warning message to hide the message.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Select the **Warn for insufficient permissions** to see a prompt that tells you if you have permission to create or modify InTouchView applications. These permissions authorize or prevent you from creating and modifying InTouchView Applications.

3. On the **Scan** page:
   - Select the **Initial scan state for deployed objects**. You can select **On Scan** or **Off Scan**. You can change this setting on an individual basis in the **Deploy** dialog box when you deploy.
   - Select the **Scan state defaults** when undeploying or redeploying instances. **Force Off Scan** will attempt to take the target object offscan when an already deployed object is redeployed. **Don't Force Off Scan** does not force the target to go off scan when you deploy.

     Note: Redeployment of objects that are currently deployed on-scan will be canceled unless this option is selected.

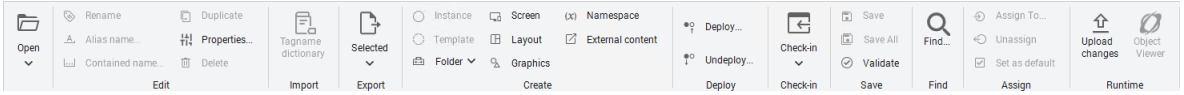
   - Enable or disable **Auto context selection**. This setting is enabled by default:
     - When **Auto context select** is enabled (checked), an object selected in the **Model**, **Derivation**, or **Deployment** view remains selected when you switch between views.
     - When **Auto context select** is disabled (unchecked), the **Model**, **Derivation**, and **Deployment** views do not keep the same object selected as you switch between views, so a different object can be retained in each view.
   - In the **Field Attributes** section, select the **Show Field Attributes** check box to display the **Field Attributes** tab in the User Defined Object editor. By default, the **Field Attributes** tab is displayed only if field attributes are already defined for the object. You can override the default by selecting this option. For more information about working with legacy field attributes, see About the Field Attributes, UDAs, and Extensions Pages in the Application Server help file
4. On the **Defaults** page, type the object names of the Framework objects you want to be the defaults with respect to assignment relationships.
   - **Platform** name.
   - **Application engine** name.
   - **Area** name.
   - **View engine** name.
   - Your **Security group** name, if any.

   Leave a field blank if you do not want a default for that type of object.

   Your changes are saved automatically.
5. When you are done, click **OK**.

Use the simplified layout

By default, the ribbons at the top of the Application Server IDE window show all the features available to you, identified with both an icon and text, and grouped into sections. For example, this is the Home ribbon:



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To decrease the size of the ribbon and make more room for the other parts of the IDE display, you can turn on Simplified Layout. This collapses the ribbon so it shows only icons for each section. Here is the Home ribbon in Simplified Layout:

To use a feature in Simplified Layout, select the down arrow for the section, then select the feature.

To turn Simplified Layout on or off, use the slider at the top right of the IDE display:

Note: You can hide the ribbon completely by selecting the caret (^) next to the **Simplified Layout** slider.

Log on with security enabled

Some Galaxies have security associated with them. If you try to open a Galaxy with security, you need to log on to the Galaxy to open it.

Note: If you do not have logon rights, you cannot open a Galaxy.

For information about setting up security in the System Platform environment, see Work with security.

To log on to a Galaxy

1. When you open a Galaxy that has security enabled, the **Sign in to <Galaxy>** dialog box appears.
2. Type your **User name** and select **Next**. If you need to include your domain,enter your name in the format *<Domain>**<User>*.
3. Type your **Password** and select **Sign in.**

   Your logon data is validated by the Galaxy Repository and the IDE opens. If the GR does not recognize your user name or password, you are prompted to enter them again.

Change Users

You can change users in a Galaxy at any time. Any security restrictions associated with a user change when the user logs on or logs off from the Galaxy. For more information about users and security, see Working with Security.

If the Galaxy has not been configured to enable security, you see a message. All users in an open-security environment are treated as the DefaultUser by the Galaxy. In an open-security environment, all users have full access to everything.

To change users

1. From the ribbon, select **Galaxy,** then select **User**.
2. Select **Change User**.
3. The **Sign in to <Galaxy>** dialog box appears.
4. Type your **User name** and select **Next**. If you need to include your domain,enter your name in the format *<Domain>**<User>*.
5. Type your **Password** and select **Sign in.**

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

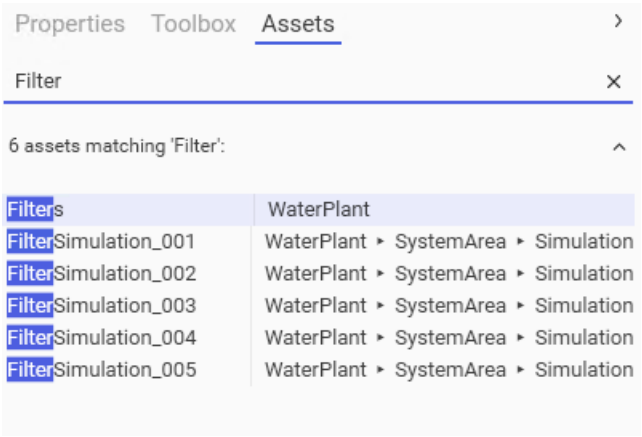
Search for content

You can search within the currently displayed folders or assets by entering a search term in the **Search** field. Typing in the **Search** field hides the hint text in preparation to enter a search string.

While you are entering a search string, an X appears in the **Search** field to indicate the string can be deleted. If you press the X or the Esc key, the string is cleared from the **Search** field and no search is conducted.

Asset Searches

An Assets search is applied across the entire Galaxy. The search results show the first batch of Areas and Assets containing the search term in a multi-column list displayed in ascending alphabetical order. A message beneath the **Search** field indicates the number of items found in the search that match the search string.

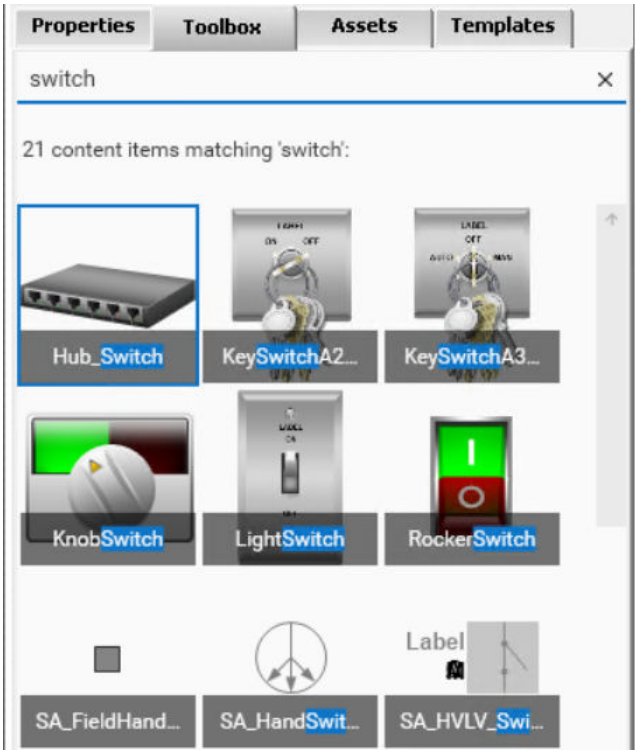


The search results show the search term highlighted within their names. You can select an item from the search results list. Selecting an item updates the content view with all content that is owned or linked to the asset you selected.

Toolbox Searches

A Toolbox search is applied across the entire Visualization folder. The search results show thumbnails of content that match the search term in ascending, alphabetical order. A message beneath the **Search** field indicates the number of matching items found by the search.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Search result items in the content area can be selected and dropped in a layout pane.

About the Industrial Graphic Editor

The Industrial Graphic Editor provides a tabbed browse function to filter graphics, object instances and templates into lists whose items can be selected.

- **Properties grid**

  The **Properties** grid shows the properties of the currently selected element on the Industrial Graphic Editor's canvas. If no element is selected, the properties of an entire graphic appear in the view.

- **Toolbox Tab**

  The **Toolbox** tab shows the folder structure of the Visualization folder. By default, the **Toolbox** view shows two main folders: "_Default Content" and "MyContent."

  - _Default Content contains numbered subfolders for default Screen Profiles, Layouts, Graphics, Apps, and ViewApp Namespaces (the content of ViewApp Namespaces is not applicable to the Industrial Graphics Editor and thus is not displayed).
  - MyContent contains a similar set of (empty) numbered subfolders. These are provided as a ready-to-use repository for any new content that you add or derive from the default content.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Any folders you create also appear in the **Toolbox** view. A breadcrumb trail above the folder structure shows the current selected folder.

Selecting a folder from the browse tree expands any subordinate folders beneath it. Selecting a folder containing graphics shows their thumbnails in the content area beneath the folder tree. You can drag graphics from the content area onto the canvas to become part of the edited graphic.

- **Assets Tab**

  The **Assets** tab shows the tree structure of the Galaxy's asset containment model. The **Assets** view is equivalent to the **Model** view of the Template folder consisting of a hierarchical arrangement of object instances.

  Selecting an object from the browse tree expands any subordinate object instances beneath it. Selecting an object instance containing any owned or linked graphics shows their thumbnails in the content area beneath the object tree. Graphics can be selected from the content area and dragged onto the canvas to become part of the edited graphic. The owning object is set to the object instance containing the graphic.

- **Templates Tab**

  The **Templates** tab shows the folder tree structure of the Galaxy's Template folder consisting of folders and object templates. As in the **Toolbox** tab, the **Templates** view organizes the templates under two main folders, "_Default Content" and "My Content."

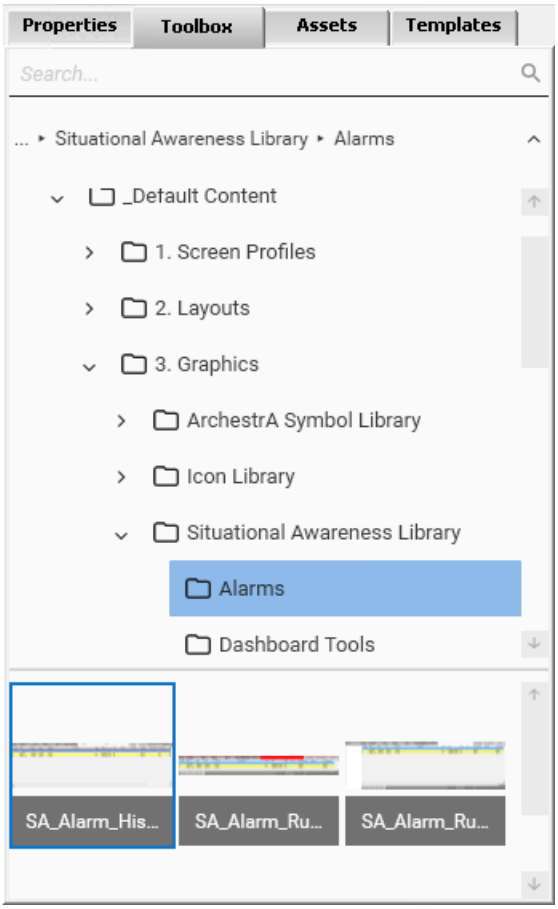
  - _Default Content contains numbered subfolders for Asset, System, Device Integration, and ViewApp templates.
  - MyContent contains a similar set of (empty) numbered subfolders. These are provided as a ready-to-use repository for any new templates that you derive from the defaults.
  - Any template folders you create also appear in the **Templates** view. A breadcrumb trail above the folder structure shows the current selected toolset.

  Selecting an object from the browse tree expands any subordinate object instances beneath it. Selecting an object instance containing any owned or linked graphics shows their thumbnails in the content area beneath the object tree. Graphics can be selected from the content area and dragged onto the canvas to become part of the edited graphic. The owning object is set to the object instance containing the graphic.

If a listed item contains sub-items, an expand arrow appears to the left of the item. Clicking or touching the expand arrow retrieves and displays the sub-items. The current selection does not change until an item is selected. If the number of items exceeds the amount of vertical space available, a scrollbar appears to scroll the list .

If there is insufficient horizontal space to display an item name, a horizontal scrollbar appears at the bottom of the tree to enable you to scroll the content into view.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Platform common services

Platform Common Services (PCS)—formerly the AVEVA Service Bus (ASB)—is a framework that hosts services based on general principles of a service-oriented architecture (SOA).

- SOA provides a mesh of interoperable services that can be used within a single system or extended to external systems based on an established trust relationship between the systems.
- SOA also generally provides a way for services to discover other available SOA-based services.

PCS components are installed with Application Server as part of the System Platform installation. PCS functionality appears as menu items in the IDE.

PCS components include a set of core runtime services, a catalog of user-configurable services, and a scheme that enables connectivity, interoperability and exchange of data among the services as well as with internal and external applications

Configure and deploy AVEVA services

The configuration tool and workflow are the same for all user-configurable AVEVA Services. To configure an AVEVA Service, you will need to perform the following tasks:

1. Create an instance of a PCS service
2. Assign the service instance to a node

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3. Deploy the instance

You can make multiple instances of a single service, and deploy them to multiple nodes. You can also deploy a single instance to multiple nodes. This approach helps reduce service management. On the same machine, you can have more than one service simply by creating new instances.

Create an instance of a service

You must create an instance of an AVEVA PCS Service (for example, ASBGRBrowsing Service) to configure the service.

Assign the service instance to a node

After you create an instance of the service, you must assign the instance to the desired node. You must check out the instance before assigning it to a node. You must check in the instance after assigning it to a node.

Note: For the ASBGRBrowsingService, you can also specify the name of the Galaxy to assign an instance of the service.

Deploy the instance

After assigning a service instance to a node, you must deploy the instance. A deployment confirmation message appears.

For information about configuring specific AVEVA PCS Services, see the following topics:

- Configure the ASBGRBrowsing service
- Configure the ASBMxDataProvider service
- Configure the ASBAuthentication service

Configure service TCP ports

Important: As of AVEVA System Platform 2014 R2, there is no need to configure any TCP ports. All Platform Common Services communication is based on the Microsoft WCF shared port 808. This information is retained for existing applications built with a version of AVEVA System Platform prior to the 2014 R2 release that have been assigned custom port numbers.

On a node running an OPC UA server, you need to open two ports, one for the OPC UA server and one for the OPC UA discovery service, if configured. See the following table for more information about OPC UA port configuration.

The ASB Core Services, installed with PCS and started by the Watchdog Service, also use specific ports. The following table lists the default TCP ports assigned to the user-configurable PCS Services and the ports in use for the PCS Core Services.

- The PCS Core Services port numbers should not be used to configure instances of user services.
- The firewall must allow incoming connections to the ports in the following table for their corresponding services to function.

Service	Port Numbers in Use
User-configurable Services	

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Service	Port Numbers in Use
ASBGRBrowsing Service	7500, 7501 (default) See note following this table.
ASBMxDataProvider Service	3572 (default)
ASBAuthentication Service	7779 (default)
Event Service	3575 (default, configurable)
EventHistorization Service	3586 (default, configurable)
OPC UA Service	48031 (default, configurable) Endpoint for Application Server OPC UA Server instance
OPC UA Discovery Service	4840 (OPC Foundation service - not an ASB service)
PCS Core Services-Not configurable	
Local Discovery Server	9111
Primary Local Galaxy Server	9110
Secondary Local Galaxy Server	9210
Primary Cross Galaxy Server	9310
Secondary Cross Galaxy Server	9410
Galaxy Pairing	7085
Configuration Service	6332
Content Provider Service	6011
Deploy Agent Service	6533, 6633
Service Manager Service	6111, 6113
System Authentication Service	9876
aaServiceASBSoftwareUpdate	7587

Important: If a Galaxy Repository (GR) has more than one Galaxy, two additional ports must be opened to enable a remote GR to browse to each additional galaxy. For example, two galaxies would require ports 7500, 7501, 7502, and 7503 to be open. Three galaxies would require ports 7500-7505 to be open.

Configure the ASBGRBrowsing service

The ASBGRBrowsingService enables you to browse Application Server namespaces.

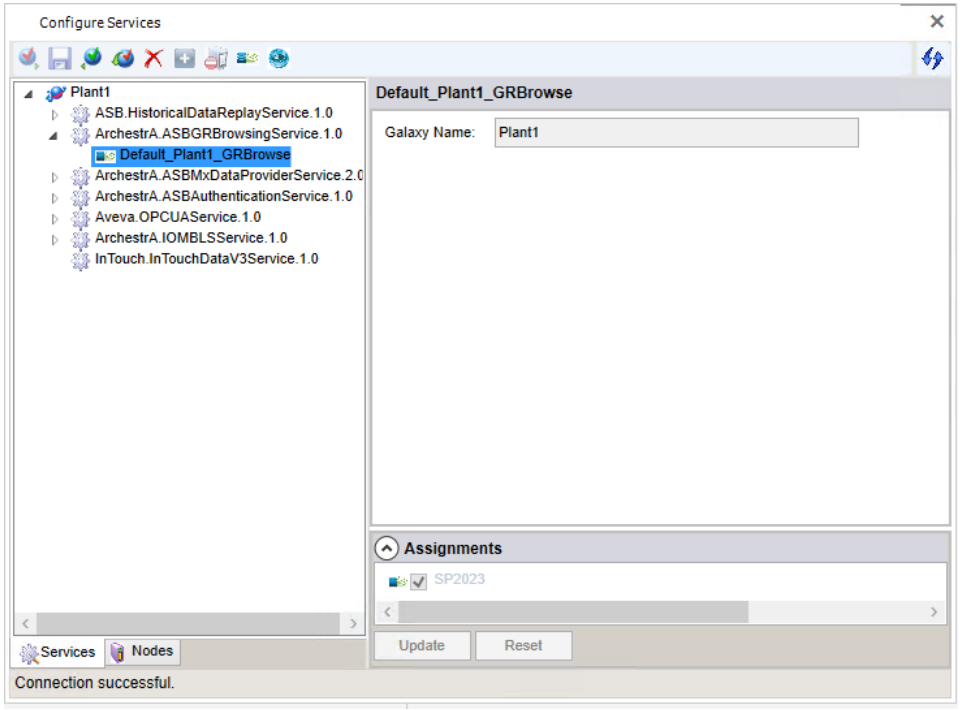
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Important: The ASBGRBrowsing service is configured with one default service instance named "Default_<GalaxyName>_GRBrowse". Do not change the name of this default service instance or delete this service instance. The GRBrowsing service will stop working when you undeploy the GR Platform. If the default service instance has been renamed, revert the service instance name back to the default format. If the default service instance has been deleted, create a new instance with the correctly formatted default name.

To configure the ASBGRBrowsing Service

1. Select the **Galaxy** tab, then select **Configure.**
2. Select **System**, then **Services**.

   The **Configure Services** utility opens.



3. Expand the tree view. Right-click **ASBGRBrowsingService**, and then select **Create.** You can also select **ASBGRBrowsingService**, then type **CTRL+N**. The new instance appears in the tree structure.
4. Right-click the instance name and select **Check-out**
5. Enter the Galaxy name in the **Galaxy Name** box.
6. In the **Assignments** area, select the node you want to assign to the instance, and then select **Update**.

   Note: The **Update** button is enabled only when you select a node. You cannot delete a node that is already assigned to a service instance. To delete a node, you must first unassign the node from the service instance and then delete it.
7. On the left-pane, right-click the instance, and then select **Check-in**.
8. On the left-pane, right-click the instance, and then select **Deploy**. A message appears indicating whether the node is successfully deployed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure the ASBMxDataProvider service

The ASBMxDataProvider Service acts as a gateway providing access to Application Server data.

To configure the ASBMxDataProvider Service

1. On the ribbon, select **Galaxy**, then select **Configure.**
2. Select **System**, then **Services**.

   The **Configure Services** utility opens.
3. Right-click **ASBMxDataProviderService**, and then click **Create.** You can also press CTRL+N. The new instance appears in the tree structure.
4. Right-click the instance name and click **Check-out.**
5. In the **Assignments** area, select the node you want to assign to the instance, and then click **Update**.
6. On the left-pane, right-click the instance, and then click **Check-in**.
7. On the left-pane, right-click the instance, and then select **Deploy**. A message appears indicating whether the node is successfully deployed.

Configure the ASBAuthentication service

The ASBAuthenticationService provides user authentication.

To configure the ASBAuthentication Service

1. On the ribbon, select **Galaxy**, then select **Configure.**
2. Select **System**, then **Services**.

   The **Configure Services** utility opens.
3. Right-click **ASBAuthenticationService**, and then click **Create.** You can also press CTRL+N. The new instance appears in the tree structure.
4. Right-click the instance name and click **Check-out**
5. In the **Assignments** area, select the node you want to assign to the instance, and then click **Update**.
6. On the left-pane, right-click the instance, and then click **Check-in**.
7. On the left-pane, right-click the instance, and then select **Deploy**. A message at the bottom of the window appears indicating whether the node is successfully deployed.

Note: A runtime node will be offline if its hosted WinPlatform object is undeployed or if there is a pending software upgrade (SUP state). To change the status of the node to online and deploy new services, apply any required software upgrades and redeploy the WinPlatform object.

IOMBLSService

The IOMBLSService provides services for browsing, configuring, and deploying objects and visual elements. All the services provided by IOMBLSService are hosted in a single executable (IOMBLS.exe). IOMBLService is not user-configurable, and runs in the background. Do not delete, duplicate, close, or modify this service in any way.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure and deploy the OPC UA service

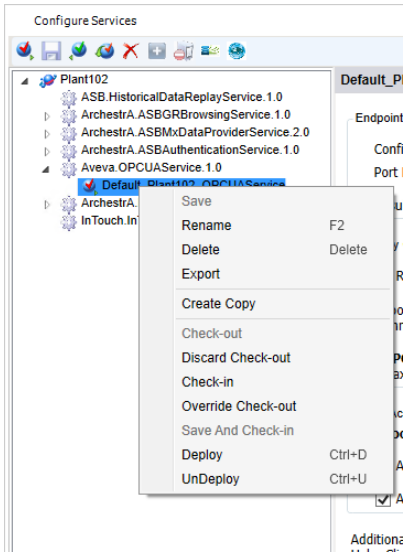
The AVEVA OPC UA Service provides access from an OPC UA client to Application Server data, without the need for the Galaxy Browser, a gateway, or other protocol translation mechanism.

To configure and deploy the OPC UA Server Service

1. On the IDE ribbon, select **Galaxy** then select **Configure**
2. Select **System**, then select **Services**.
3. The **Configure Services** utility opens.
4. Expand the items in the left-hand pane as needed. Right-click the OPC UA service name and select **Check-out** from the context menu.
5. Edit the **port number** for the OPC UA instance if necessary. The default port is **48031**.
6. **Security configuration**
   - **Require encrypted communication:** It is strongly recommended that you enable this option, as this will encrypt the payloads across the connection. Note that the client must match this configuration.

     Important! An OPC UA connection cannot be established if you do not enable this option while OS security is enabled, even if the option "Allow authenticated Galaxy Users to write attributes" (under Step 7) is enabled.
7. **Client access rules**
   - **Anonymous client connection:** Allowing anonymous client connections is recommended ONLY for initial setup configurations and testing. Anonymous client connections should be disabled for production environments. Refer to Client access rules and galaxy security to see the effect that this option has on data access.
   - **Allow authenticated Galaxy Users to write attributes:** Enabling this setting will only take effect when encrypted communication is also enabled (see Step 6, above). The OPC UA client must match this configuration.
8. The **Assignments** section (below the right pane) shows both the local node and deployed remote runtime nodes to which the OPC UA service can be assigned. Select the checkmark next to the runtime node where you want to deploy the OPC Server service, and then click **Update**.
9. In the left pane, right-click the OPC UA service instance name and select **Check-in** from the context menu.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



10. Right-click the instance again to open the context menu and select **Deploy**, or press CTRL+D. A message appears indicating whether the service has been successfully deployed to the OPC UA client node. If deployment is successful, the icon next to the instance name changes to indicate that the instance has deployed.
11. If you have deployed the service to a remote node, you can verify functionality through the Operations Control Management Console.
    a. Under Operations Integration Server Manager, select the remote node name.
    b. Expand Operations Integration Supervisory Servers and select the OPCUA Client (under OI.GATEWAY.3). Check the following settings:
        - Server Node: localhost
        - OPCUA Server: opc.tcp://localhost:48031

To add additional OPC UA services

Each OPC UA service is dedicated to a single OPC UA client node. To add additional OPC UA services:

1. Right-click **AVEVA.OPCService**, and then select **Create** from the context menu, or press CTRL+N. The new instance appears in the tree structure.

   Note: Each instance must have a unique port number. Enter the port number in the **Base Address** field. The default port number is **48031**. See Configure service TCP ports for a list of port numbers used by ASB services.

2. Rename the OPC UA service as needed. Right-click on the service name and select **Rename** from the context menu, or press F2. Then, enter the new name.
3. Repeat the steps above for configuring and deploying each additional OPC UA service.

To change a deployed OPC UA service

1. Check out the service instance.
2. Make any needed changes.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **Port Number:** If you are creating multiple services, each service instance should have a unique port number. If more than one service has the same port number, an error is generated in the logger. Multiple instances of the service can be deployed, as long as each service has a unique port number. A new URI (uniform resource identifier) is automatically generated when a port number is changed.

  Note: You may need to open the inbound port in the firewall to allow communication with the remote node.

- **Security Configuration:** When enabled (default), communication between OPC UA clients and the OPC UA server is encrypted. This is the recommended setting. If this setting is unchecked (disabled), communication is not encrypted.

- **Client access rules**

  Allow anonymous client connection (enabled): When enabled (default), an anonymous OPC UA client is allowed to connect to the OPC UA server. This is recommended only for testing and initial set up configurations. Once you have completed configuration and/or testing, be sure to disable this setting to provide protection against possible unwanted intrusions and to ensure that only authenticated users have access. Anonymous client connection should not be enabled in a production environment.

  Galaxy Security settings do not have any affect on these behaviors. See Working with Security for more information.

  Allow authenticated Galaxy user to write to attributes (enabled): When enabled (default), an authenticated Galaxy user can change attribute values in runtime, if their security role allows them to do so. See About Roles for more information.

  Allow authenticated Galaxy user to write to attributes (disabled): When disabled (unchecked), an authenticated Galaxy user is not permitted to change attribute values in runtime, even if their security role allows them to do so.

  See Client access rules and galaxy security for more information about user permissions for each setting combination.

3. Check in the service or services.
4. Undeploy and then redeploy the service or services.

Client access rules and galaxy security

Client Access Rules configured for the OPC UA Service interact with the Galaxy security authentication mode to allow or deny different levels of access for authorized users.

There are two configurable Client Access Rules in the OPC UA Service dialog. By default, both rules are enabled:

- Allow anonymous client connection (no username/password)
- Allow authenticated Galaxy Users to write to attributes, depending on their security role

The following table defines the level of data access users are allowed under different combinations of Client Access Rule configurations, Galaxy security authentication mode, and the type of OPC UA credentials (anonymous or authenticated user with username/password).

If security for the Galaxy is enabled (Galaxy security = **Secured**, column **1**), encrypted communication between the OPC UA clients and OPC UA service must also be enabled. See Configure and deploy the OPC UA service.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Galaxy security	OPC UA	Client access rules		Level of data access		
Authentication mode	Client credentials	Allow anonymous connection	Allow authenticated Galaxy users	Connect	Read	Write (see below)
Secured	Authenticated	Enabled	Enabled	YES	YES	YES
Secured	Authenticated	Enabled	Disabled	YES	YES	NO
Secured	Authenticated	Disabled	Enabled	YES	YES	YES
Secured	Authenticated	Disabled	Disabled	YES	YES	NO
Secured	Anonymous	Enabled	Enabled	YES	YES	NO
Secured	Anonymous	Enabled	Disabled	YES	YES	NO
Secured	Anonymous	Disabled	Enabled	NO	N/A	N/A
Secured	Anonymous	Disabled	Disabled	NO	N/A	N/A
None	Authenticated	Enabled	Enabled	NO	N/A	N/A
None	Authenticated	Enabled	Disabled	NO	N/A	N/A
None	Authenticated	Disabled	Enabled	NO	N/A	N/A
None	Authenticated	Disabled	Disabled	NO	N/A	N/A
None	Anonymous	Enabled	Enabled	YES	YES	NO
None	Anonymous	Enabled	Disabled	YES	YES	NO
None	Anonymous	Disabled	Enabled	NO	N/A	N/A
None	Anonymous	Disabled	Disabled	NO	N/A	N/A

Important: Whenever Client Access Rules and Galaxy Security allow a user to write data, this permission is always conditioned by whether or not the user's configured security role also allows them to write data to a specific attribute. This means that when Galaxy security is enabled, the user's security role must explicitly allow them to write to attributes, regardless of the OPC UA client access rule setting. If their security role does not allow them to write to attributes, they cannot, even if the level of data access in the above table shows that they can.

Automation objects

Before building a ViewApp, you must understand templates and object instances.

Templates are objects whose properties can be configured to represent a variety of process equipment types in a ViewApp.

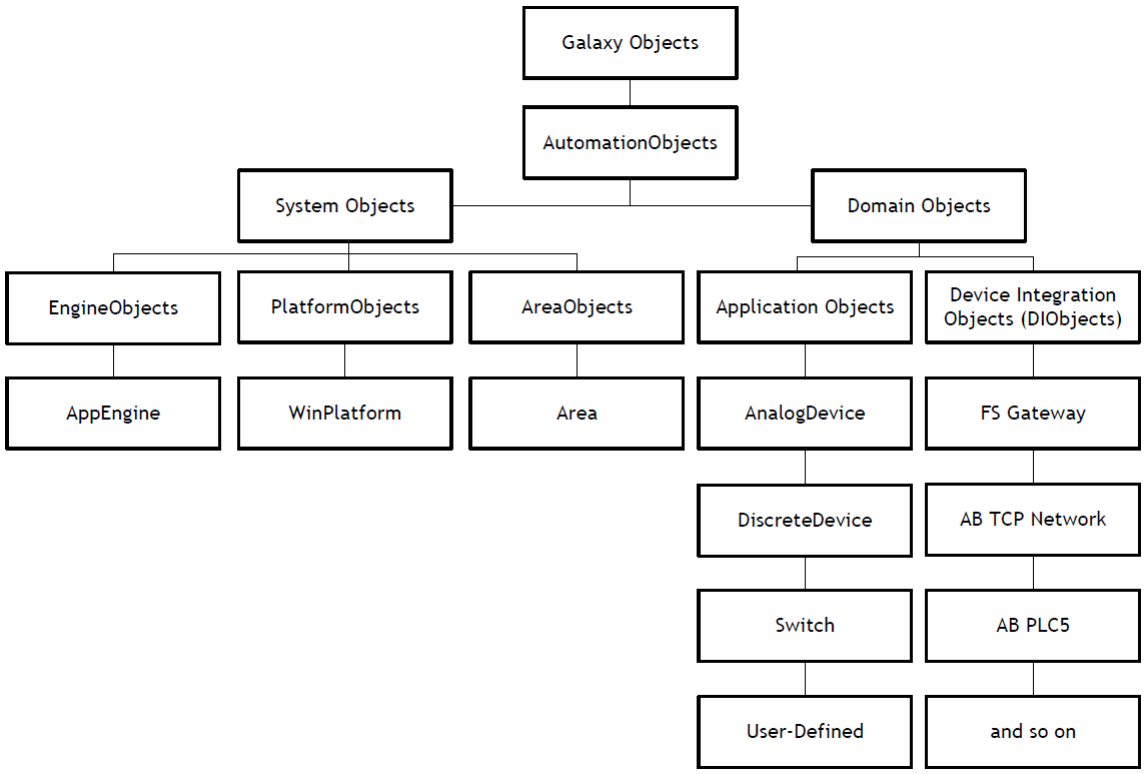
For example, you might create a template for valves. You configure the template with all the unique attributes for valves. You use that template to make object instances of valves. You can further configure and customize each object instance to represent a specific valve.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Object instances are the specific devices in your environment, such as diaphragm valves or very complex devices, like a reactor. You create an instance from a template and then customize the specific instance as needed.

Instances are deployed to the runtime environment. Templates exist in the development environment and cannot be deployed.

The following graphic shows the different kinds of objects and how they are organized.



View Object Properties

You can view the properties of an object by right-clicking and clicking **Properties**. Object properties vary, depending on the type of selected object and whether it is a base template, a derived template, or an instance.

The **Properties** window contains the tabs that allow you to examine different object properties. Only tabs applicable to the object are displayed.

Tab Name	Displayed for Object Type:
General	All objects including GalaxyObject
Attributes	All objects except GalaxyObject
References	Instances only
Cross References	Instances only
Change Log	All objects including GalaxyObject

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Tab Name	Displayed for Object Type:
Operational Limits	All objects including GalaxyObject
Errors/Warnings	All objects except GalaxyObject

The **General** tab contains basic information for the object, such as the codebase (major version), what template it is derived from, and any current errors and warnings.

The **Attributes** tab lists the object's attributes and values. In addition to attributes that have been added to the object, attributes for Errors and InAlarm are also listed, as are attributes for CodeBase (major version of the object), MinorVersion ("dot" version), and ConfigVersion. The ConfigVersion will increment each time the object has been checked in after an edit.

The **References** tab lists I/O references for the object.

The **Cross References** tab lists any attributes that are cross referenced for the object.

The **Change Log** tab is a historical list of changes, including deploy/undeploy operations, that have been made to the object.

The **Operational Limits** tab lists prohibited actions and the reason for the prohibition. For example, Check In is not allowed for objects that are already checked in.

The **Errors/Warnings** tab lists any errors or warnings for the object.

For more information about specifying the properties of objects, see Working with Objects.

About Templates and Instances

Before you start modeling your application using the Application Server, you must understand templates and object instances.

Templates are elements in Application Server that contain common configuration parameters for object instances that you use multiple times in your application.

For example, you might create a template for valves. You configure the template with all the unique attributes for valves. You use that template to make object instances of valves. You can further configure and customize each object instance to represent a specific valve.

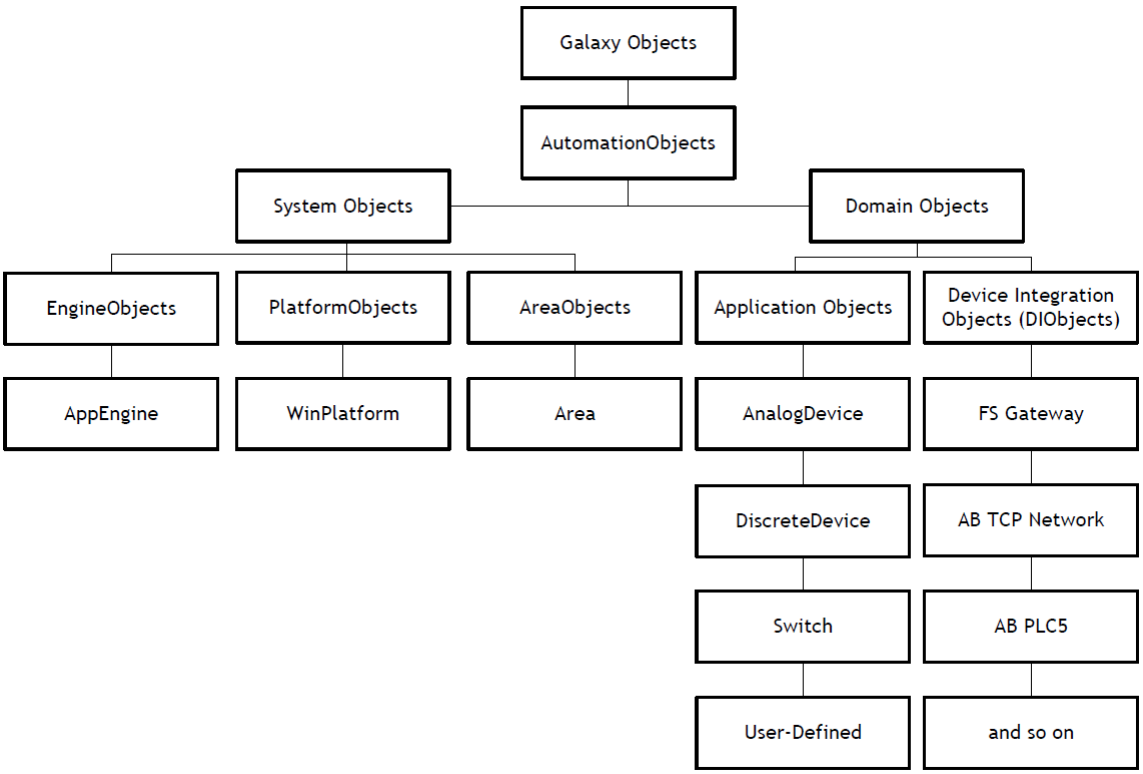
Object instances are the specific devices in your environment, such as diaphragm valves or very complex devices, like a reactor. You create an instance from a template and then customize the specific instance as needed.

Instances are deployed to the run-time environment. Templates exist in the development environment and cannot be deployed.

Creating templates and instances is very similar to object-oriented programming. For example, templates and instances have a parent/child relationship that involves inheriting attributes. There are differences, however, between object-oriented programming and creating templates and instances in Application Server.

Collectively, templates and instances are called objects. The following graphic shows the different kinds of objects and how they are organized.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



If you are new to this kind of programming, the next section explains the basic concepts you need to know before you start. If you are familiar with object-oriented programming, the concepts in the next section may be familiar to you, but notice the important differences between object-oriented programming and Application Server.

Instances

Instances are the run-time objects created from templates in Application Server. Instances are the specific things in your environment like processes, valves, conveyor belts, holding tanks, and sensors. Instances can get information from sensors on the real-world device or from application logic in Application Server. Instances exist during run time.

In your environment, you may have a few instances or several thousand. Many of these instances may be similar or identical, such as valves or holding tanks. Creating a new valve object from scratch when you have several thousand identical valves is time-consuming. That’s where templates come in.

Templates

Templates are high-level definitions of the devices in your environment. Templates are like a cookie cutter from which you can make many identical cookies.

You define a template for an object, like a valve, one time and then use that template when you need to define another instance of that item. Template names have a dollar sign ($) as the first character of their name.

A template can specify application logic, alarms, security, and historical data for an object.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

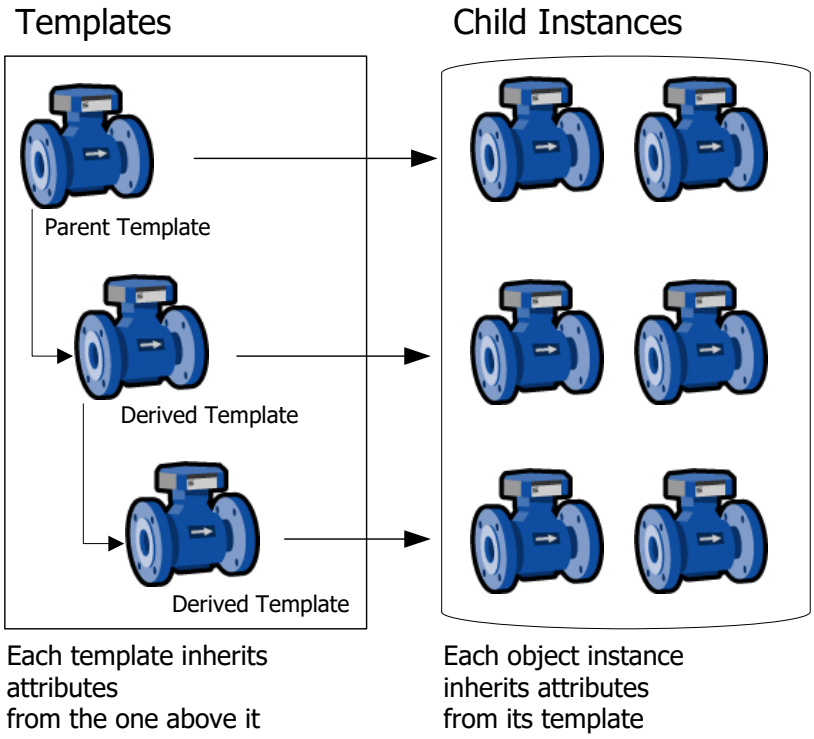
A template can also define an area of your environment. You can extend and customize a template by adding attributes, scripts, or features to meet the specific needs of your environment. Objects inherit attributes, scripts and features from their parents.

Application Server comes with predefined templates, called base templates. You cannot modify base templates. All templates you create are derived from base templates.

You can also nest templates, or contain them. Contained templates consist of nested object templates that represent complex devices consisting of smaller, simpler devices, including valves. A reactor is a good candidate for containment.

Templates only exist in the development environment.

Using the Diaphragm valve template, you can quickly create a Diaphragm valve instance when you need another Diaphragm valve in your application.



Propagation

If you need to change something about all diaphragm valves, you can change the template for the Diaphragm valve and all diaphragm valves in your application inherit the changes, assuming the attributes are locked in the parent template. This makes it easy to maintain and update your application.

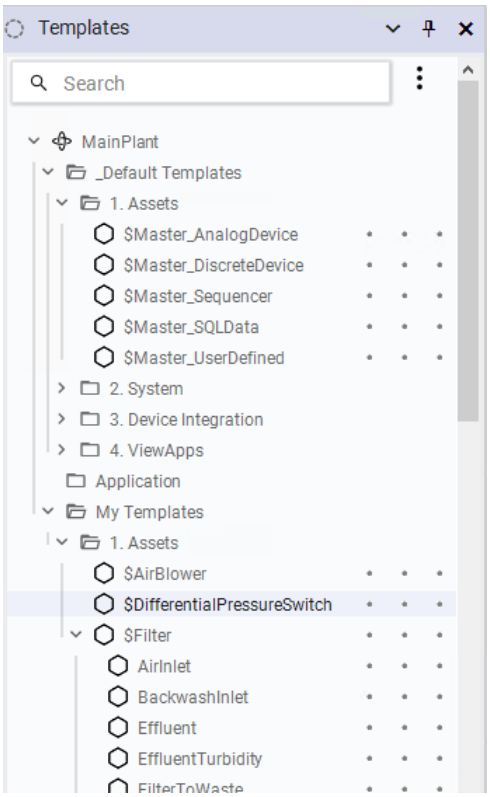
About base templates

When you first open the IDE, you see a set of read-only base templates, which are indicated by an orange lock symbol. If you based your Galaxy on the Default.cab, you will also see a set of first-level derived templates, separated into subfolders, in the **Template folder**.

You cannot modify base templates, but you can use them to create derived templates, which are copies of the base templates. We recommend that you use the first-level derived templates that are provided under the

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Default Templates folder to create additional templates. Most of these first-level derived templates have the prefix **Master**. You can modify derived templates and use them to create instances of them for your applications.



The Template folder includes:

- Asset templates

  Use these templates to represent real devices in your Galaxy, such as pumps and valves.

- Device Integration templates

  Use these templates to create instances that communicate with external devices, such as PLCs.

- System templates

  Use these templates to define system instances, like other computers.

Application Templates

These base templates let you easily create devices in your Galaxy. They contain the properties you need to set for each kind of device. For example, a DiscreteDevice device contains all the settings you need to specify for an on/off device. Of course, you can extend and customize any device by using attributes, scripts, and features.

Device Integration Templates

These base templates provide communication with external devices. External devices run on the application engine.

For example:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- DINetwork object – Refers to the object that represents the network interface port to the device through the Data Access Server. The object provides diagnostics, and configuration for that specific card.
- DIDevice object – Refers to the object that represents the actual external device (such as a PLC or RTU), which is associated to the DINetwork Object.

System Templates

These objects represent the parts of a Galaxy and not the domain they are monitoring/controlling. These base templates let you create more system level grouping and computers, such as areas you add objects to or another host AppEngine.

WinPlatform Object

The WinPlatform platform object is a key base object because you need a platform to host the objects you are modeling. This object:

- Calculates various statistics for the node it is deployed to. These statistics are published in attributes.
- Monitors various statistics related to the node it is deployed to. These monitored attributes can be alarmed and historized.
- Starts and stops engines which are deployed to it, based on each engines' startup type .
- Monitors the running state of engines deployed to it. If the platform detects an engine failed, it can, optionally based on the value of the engine’s restart attribute, restart the engine.

AppEngine Object

The AppEngine object must have a Platform on which to run. This object:

- Hosts ApplicationObjects, Device Integration objects and areas.
- Contains the logic to set up and initialize objects when they are deployed.
- Contains the logic to remove objects from the engine when they are undeployed.
- Determines the scan time which all objects within that particular engine run.

Area Object

All application objects belong to an area. Areas can contain sub-Areas. Areas provide a key organizational role in grouping alarm information and providing that information to those who use alarm/event clients to monitor their areas of responsibility.

The values of three Area object alarm attributes can be saved to the historian:

- Active alarm counter
- Unacknowledged alarm counter
- Disabled (or silenced) alarm counter

ViewEngine Object

The ViewEngine object must have a Platform on which to run. This object:

- Hosts ViewApp objects which will be used through the desktop client.
- Hosts InTouchViewApp objects.
- Contains the logic to set up and initialize objects when they are deployed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Contains the logic to remove objects when they are undeployed.
- Determines the scan time which all objects within that particular engine run.
- Determines if the ViewApp objects that it hosts will be read/write or read-only.

Note: InTouch applications are not affected by the "ViewApps are Read Only" setting of the ViewEngine. This setting is used for the ViewApp object only.

WebViewEngine Object

The WebViewEngine object must have a Platform on which to run. This object:

- Hosts OMI ViewApp objects which will be used through the OMI web client.
- Sets up a web server and content server to serve the OMI ViewApp and its content files to the clients.
- Sets up a telemetry endpoint allowing OMI clients running in browsers to subscribe and receive telemetry values from the Galaxy.
- Sets up authentication mechanisms allowing users to log in to the OMI ViewApp based on the Galaxy configuration.

There are no configurable attributes or other settings for WebViewEngine objects. No data is historized for them and no alarms can be configured for any of their attributes.

InTouchViewApp Object

The InTouchViewApp object must have a ViewEngine on which to run. This object:

- Manages the synchronization and delivery of files required by the associated InTouch® application.
- Provides run-time access to tags on the associated InTouch application.
- Starts WindowMaker for the associated InTouch application when edited.

Note: InTouch applications are not affected by the "ViewApps are Read Only" setting of the ViewEngine that hosts the InTouchViewApp object. This setting is used for the ViewApp object only.

ViewApp Object

The ViewApp object must have a ViewEngine (for access from the Application Manager) or WebViewEngine (for OMI web client access) on which to run. This object:

- Provides visualization at run time for an AVEVA OMI application.
- Incorporates a Screen Profile and the content defined in an associated Layout.
- Provides run-time access to your production environment.
- A ViewApp deployed on a ViewEngine (as opposed to a WebViewEngine) can be read/write or read-only, depending on whether the ViewEngine hosting the ViewApp object has its "ViewApp are Read Only" setting enabled (ViewAppsReadOnly attribute set to true).

About Derived Templates

All templates you create within the IDE are derived templates.

When creating your Galaxy application, plan ahead and create derived templates for devices of a certain type so you can use the templates to create the individual instances.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

A new derived template is an exact copy of its parent template with the possible exceptions of locking and security and modified attribute values. You can lock attributes to prevent them from being modified in child templates.

After you create a new derived template, you can customize it. For more information about customizing and extending templates, see Create Derived Templates.

Every template has a set of attributes and default values. When you create an instance, attributes are inherited by the instance. In the instance, you can reconfigure many of the attributes inherited from the parent template if they are not locked on the parent template.

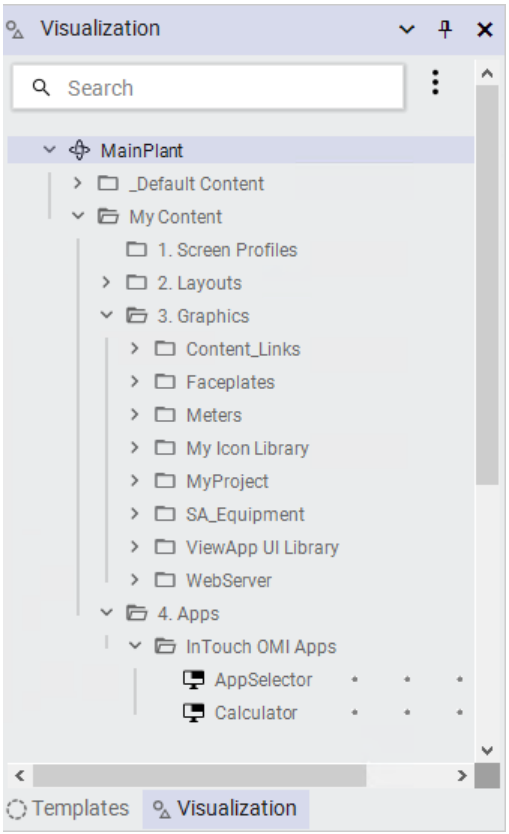
For more information about customizing instances, see Edit Objects.

Working with Objects

The System Platform Integrated Development Environment is the tool that you will use to work with Application Server objects. Both templates and instances are collectively referred to as objects. For more information about what templates and instances are, see About Templates and Instances.

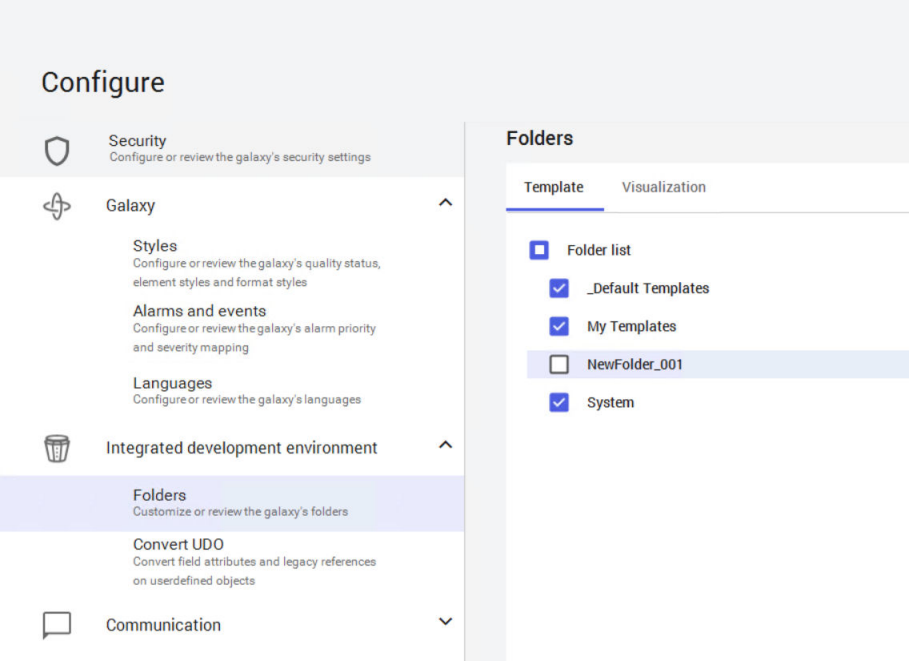
Manage Folders

Folders are used to organize objects used to build the Galaxy, and broadly categorized into the Template and Visualization folders. Each folder is indicated by the folder icon. The Template folder contains application objects templates, used for creating a digital twin of the physical environment. The Visualization folder contains objects used for creating run-time displays, such as graphics, apps, controls, widgets, screen profiles, and layouts. The Template and Visualization folders each contain subfolders that further categorize the objects that they contain. These folders are initially created during the Galaxy creation process.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

You can move items to different folders within the Template and Visualization folders, but you cannot move a Visualization object to the Template folder and visa versa.You can create additional folders and subfolders to store the templates and graphics used in your application. All folders contained within a Galaxy are displayed by default, but you can customize which top-level folders are shown to make the workspace less cluttered. To set the visibility of a folder, go to the **Galaxy** backstage menu and select **Configure**, then **Integrated development environment**. Then, select **Folders** and deselect any folders listed under the Template and Visualization folders that you want to hide. Click **Save** to preserve any changes. Subfolders do not appear in this configuration option and cannot be selected/deselected.



Create Folders

When you create your own folder, it must have a unique name. Folder names are not case sensitive, so `Valves` is the same name as `valves`. A folder name can be up to a maximum of 64 alphanumeric and special characters, including spaces, except $.

To create a new top-level folder in the Template or Visualization folder

1. On the **Home** ribbon,in the **Create** area, select **Folder**, then **Template folder** or **Graphic folder**.
2. Type a name for the new folder.

   A new folder appears and is selected. You can now drag templates into the new Template folder, or you can drag graphics into the Graphics folder.

Create Child Folders

Folderscan be created within existing folders. Nested folders help in further organizing templates and graphics. You can create a maximum of ten levels of folders.

To create a child folder

1. Select the parent folder.
2. On the **Home** ribbon,in the **Create** area, select **Folder**, then **Template Folder** or **Visualization Folder**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3. Type a name for the new folder.

   A new folder appears beneath the parent folder and is selected.

   You can now drag templates into the new folder.

Delete Folders

You can delete folders you no longer want or need. Before you start, make sure you move or delete all content from the folder. The folder you want to delete must be empty, or it cannot deleted.

To delete a folder

1. Select the folder to delete.
2. On the **Home** ribbon, in the **Edit** area, select **Delete**.
3. Click **Yes** to delete the folder.

Using Industrial Graphics in AVEVA Connect

AVEVA Connect is the common cloud repository allowing you to to manage common contents, such as, Industrial Graphics, Controls, and Widgets in the Cloud.

Graphics can be downloaded and uploaded on demand. Graphics are stored in 'Stores' in AVEVA Connect; there are three types of stores – global, tenant and user specific. Within each store users can configure multiple drives. You must have an AVEVA Connect user account to manage graphics in the cloud. Each drive can be configured with different access levels for different users.

Users will have read-only or read-write access depending on the access granted by the administrator.

- Users with Read-Only access can view graphics in the cloud. They can download the graphics to the cloud, but these graphics would be available as read-only. They cannot modify or upload the graphics.
- Users with Read-Write access can view, download, modify, and upload graphics to the cloud.

Multiple users can access the graphics in the same drive, but only one user can edit or save a graphic at one time.

Logging in to CONNECT

You must have a CONNECT account and authorization from your administration to access CONNECT.

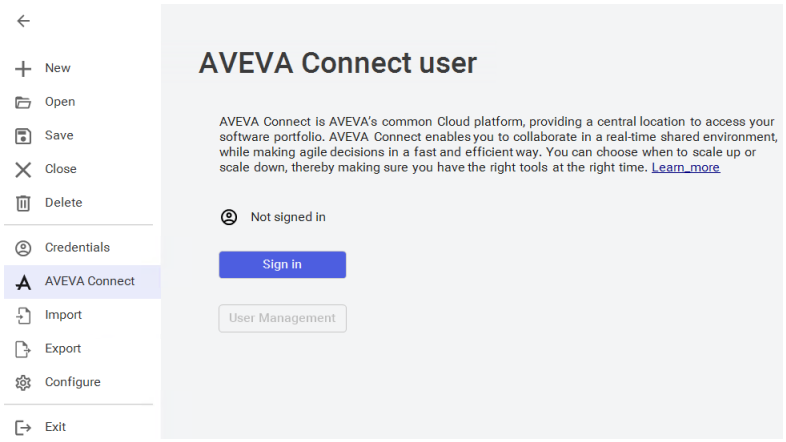
1. If you are using InTouch, launch WindowMaker, then, from the **File** menu, select **AVEVA Connect**.

   If you are using the System Platform IDE, select the **Galaxy** menu, then select **AVEVA Connect**.

   Note: The **AVEVA Connect** option will not be available if you have selected **connected experience** mode under the **Licensing Mode** tab of the Configurator.

   The CONNECT user configuration screen appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. Click **Sign In**.
3. Enter your CONNECT email address.
4. Enter your password.

   If the credentials are verified, you are signed in.

   Note: You must have the Content Contributor role in order to access the CONNECT Industrial Graphics drive. This is required for both Operations Control connected experience and non-connected experience use.
5. Select the **AVEVA Cloud Integration Studio solution.** For example: Pym Technologies.
6. The AVEVA Drive appears as a separate repository within the Visualization folder.

Note: Once you have selected an Integration Studio solution, it is cached in the browser memory. To switch the Integration Studio solution, you must clear the browser cache.

Navigating between Shared Drives

Once you are logged into a specific Store you can move between different drives and upload and download graphics.

1. On the **View** menu, in the **Cloud drive** group, click **Shared Drive**.
2. Select a drive from the available options.

   The Visualization folder will refresh to display the selected drive.

Note: You cannot move a graphic which is locked by another user.

Uploading/Downloading Graphics to the Cloud

The AVEVA Drive functions similar to your local Visualization folder. You can upload or create folders on the AVEVA Drive to organize your graphics. With Read-Write access, you can download graphics to the local Visualization folder and use it for your applications. When you download or upload graphics, the contents will be overwritten according to the selected overwrite option

- If the contents already exist in the destination drive, the contents will remain in the same toolset location. Otherwise, the contents will be added in the same toolset path as the source

For the selected embedded contents

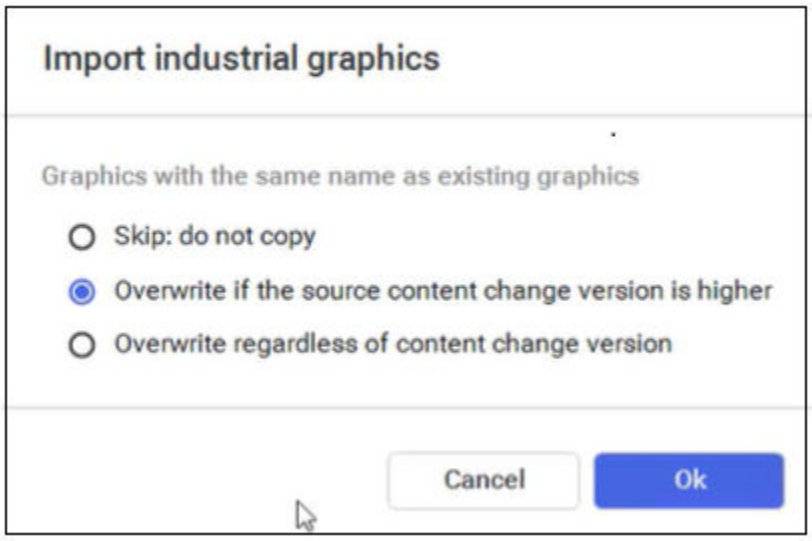
- If the embedded contents already exist in the destination drive, the contents will remain in the same toolset location in destination.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- If the embedded content resides in the same toolset location as its parent content or within the same folder hierarchy, then the embedded contents will be added in the same toolset path as the parent. Otherwise, the embedded contents will be added in the same toolset path as the source.

Upload Graphics to CONNECT

1. Drag the graphic(s) from the local Visualization folder and drop it to the AVEVA Drive folder.
2. If the graphic already exists in the drive then a pop up dialog appears, requesting to overwrite or skip the upload.



3. Select the appropriate option, and click **OK**.
4. The **Upload Content** dialog appears and displays the progress of the operation. Click **View Details** to view the progress.
5. Click **Close**.

You can also upload folder(s) that contains multiple graphics.

Download Graphics to the Local Visualization Folder

Users with Read-Write access can download graphics from the AVEVA Drive to their local Visualization folder.

1. Drag the graphic(s) or folder from the AVEVA Drive and drop it to the local Visualization folder.

   The **Download Content** dialog box requests confirmation of the download.

   - Click **Yes** to continue with the download.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Click **No** to cancel the operation.
- Click **View Details** to view the list of controls and namespaces used. Any errors during downloading are also displayed.

2. If the graphic already exists, the Import industrial graphic dialog box appears.
3. Select the appropriate option.
4. The graphic(s) are downloaded to the local Visualization folder. Embedded graphics are also copied.

Upload/Download Support for Various Cloud Content

The upload/download support of various cloud content is listed in the table below.

Cloud Content	Upload Support	Download Support
Embedded Client Control and HTML5 Widget	Upload will package the embedded graphics contents including Client Control and HTML5 Widgets as aaPKG format and copy/overwrite to the Cloud.	Download will extract the embedded graphics contents including Client Control and HTML5 Widgets and import/overwrite the control into the local application
Embedded application object graphic	Not Supported	Not Supported
Custom script library	Not Supported	Not Supported
Application Style Library	Not Supported	Not Supported

Managing Graphics in CONNECT

You can also create, rename, update, duplicate and delete graphics directly on AVEVA Drive, just like managing local graphics.

- Right-click the AVEVA Drive or any of the folders to create new graphic or toolset.
- Double-click to edit graphics using the graphic editor.
- Right-click and select the corresponding options to rename, duplicate or delete the graphic, and update thumbnails.

  The context menu options that are not supported for the Cloud appear greyed out. For example, Set Web Client Home Symbol, Set Web Client Root Folder, Export Symbol, and Export Localization.

  Note: Updating thumbnails for embedded graphics directly in the AVEVA Drive causes a lot of cloud traffic. Hence, we recommend that you update the thumbnails locally.

Managing Graphics with Multiple Users

Multiple users can use the graphics available on AVEVA Drive at a time. However, a graphic can be edited by only one user at a time. When a graphic in the Cloud repository is being edited by one user, the graphic is checked-out and locked, preventing other users from making any edits. All other users can only view the read-only unmodified graphic when it is checked-out. The latest changes reflect only when it is checked-in the the modifying user.

- A lock icon against the graphic icon in the folder indicates that the graphic is checked out by the logged in user.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- An edit icon against the graphic icon in the folder indicates that the graphic is being edited by another user.

After the first user saves and checks in the graphic, the second user can check it out and make changes to it. Once the graphic is saved and checked-in, all users can view the modified graphic.

Cloud Graphic Versioning

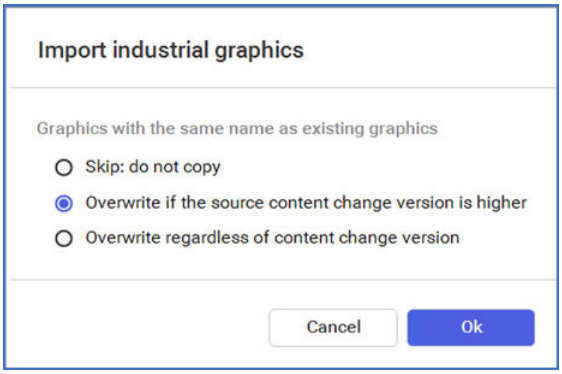
The AVEVA Drive graphics supports Industrial Graphics, Client Controls, and Widgets in the Cloud, published from different products with different versions of the graphic subsystems..

Key Concepts of Cloud Graphic Version Support

- Industrial Graphics are stored locally and in the Cloud in a specific format that supports all the features and capabilities available in the Graphic subsystem.
- When new features and functionality are added to the Graphic subsystem, the graphics need to be saved and checked-in to be available for other users.
- Different versions of the Industrial Graphics subsystem are identified by a unique, internal version number, not visible to the user. At the time of their release, each product will consume the most current version of the Industrial Graphics subsystem.
- The version number of the Industrial Graphics subsystem is included as part of each stored Industrial Graphic, and is used by the Industrial Graphic subsystem to ensure only a product with a matching or higher version of the Industrial Graphic subsystem is allowed to edit, embed, or download that Industrial Graphic.
- All users connecting to the same Cloud repository will see the same set of graphics. The ability to edit, embed, or download a Cloud graphic will depend on the version of the Industrial Graphics subsystem in the product through which the user is connecting.
- The Industrial Graphics saved to the Cloud repository by a product with a newer version of the Industrial Graphic subsystem will appear disabled in the Visualization folder of the products with older version accessing the same Cloud. The user from the old product version will not be able to edit, embed, or download the graphic repository.
- Just like the local Visualization folder, graphic names must be unique within any single Cloud repository. This holds true even if the graphics are stored by products with different versions of the Industrial Graphics subsystem.

Overwriting Graphic Contents in the Cloud

If the graphic content being uploaded or downloaded already exists, the system prompts to overwrite the graphic.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The overwriting depends on the graphic subsystem versions of the source and the destination, and the change log version of the graphic.

- Option 1 will skip the overwrite regardless of the graphic subsystem versions or the change log version of the source and the destination.
- If the graphic subsystem version is different between source and destination, both options 2 and 3 will overwrite the graphic, regardless of change log version.
- If the graphic subsystem version is same between source and destination:
  - Option 2 will overwrite only if source has higher change log version.
  - Option 3 will overwrite regardless of the change log version.

Overwriting Custom Client Controls and Widgets in the Cloud

The guidelines of Cloud Graphic Version Management do not apply to the Custom Clients Controls and Widgets at the time of this release. Client Controls and Widgets are supported for both download or upload operation. However, overwrite is not supported. If a Custom Client Controls or Widgets is already available at the target location, irrespective of the client control internal version, the overwrite will be skipped. To overwrite the source repository client control, you must first delete the client control in the source repository. After that, upload/ download the client control will be successful.

Note: If a new Custom Client Controls or Widget is uploaded to the Cloud, all other users connected to the same Cloud repository must restart the WindowMaker for the latest Custom Client Controls or Widget to be available.

Cloud Graphic Versioning - Scenarios and Examples

This section provides a non-exclusive list of high level scenarios and expectation for Cloud graphic version handling.

Let's consider two products P1_Old and P2_New, of different versions, which connect to the same tenant in AVEVA Cloud.

- P1_Old has the older version of Graphic Subsystem and P2_New has the newer version of Graphic Subsystem.
- To begin with, the cloud repository has graphics S1 and S2 uploaded by P1_Old.

Scenario / Steps	Expectation
Scenario 1: P2_New creates S3. P1_Old attempts to edit S3.	S3 is saved in newer graphic version and hence it cannot be edit, embed, or download in older version. P1_Old receives a message stating that S3 is saved in newer version and cannot be opened.
Scenario 2: P2_New opens S2. P2_New saves S2 in Cloud. Then, P1_Old attempts to open S2.	S2 created in older graphic version can be opened and saved in the newer graphic version. S2 is saved in newer graphic version and hence it cannot be edit, embed, or download in older version. P1_Old receives a message stating that S3 is saved in newer version and cannot be opened.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

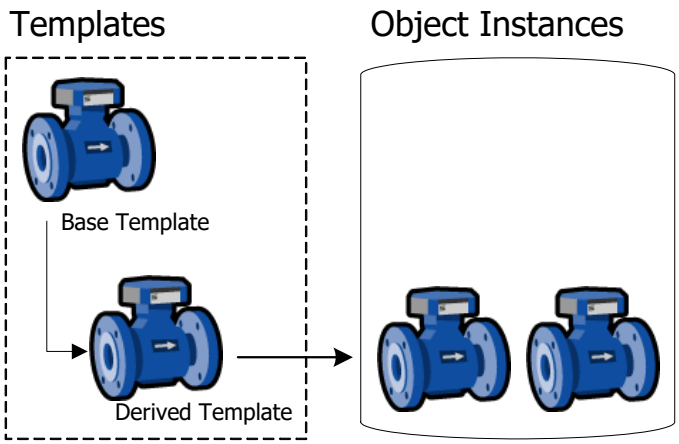
Scenario 3: P1_Old saves S1 and S2, where S2 is embedded in S1 P2_New saves and updates only S2. P1_Old attempts to edit S1.	S1 is opened because it is still the same version in P1_Old. The embedded S2 will fail to load.
Scenario 4: P2_New creates a new S4 in Cloud. P2_New embeds S1, S2, and S3 in S4.	S1 is in older graphic version but still can be loaded in S4 (newer version). S2 and S3 can be embedded because they are the same graphic version as S4.
Scenario 5: P1_Old creates a new S5 in Cloud. P1_Old embeds S1, S2, and S3 in S5.	S1 can be embedded because they are the same graphic version. S2 and S3 cannot be embedded due to newer version.
Scenario 6: P1_Old download S3.	S3 cannot be downloaded because it is in newer graphic version.
Scenario 7: Cloud repository has a graphic S3 which is in newer graphic version. P1_Old attempts to upload a graphic with same name S3.	By selecting Option 3, the graphic can be overwritten.
Scenario 8: P2_New has a graphic S8. P1_Old creates locally S8, S9, and embed S8 in S9. P1_Old uploads S9 to Cloud.	S9 is uploaded to Cloud. S8 is prevented upload because a newer version exists in the Cloud. By selecting Option 3, the content can be overwritten.

Create Derived Templates

All templates you create are derived templates. A derived template inherits attributes and behaviors from the parent template. You cannot change the attributes in a base template.

After you create a derived template, you can customize and modify its attributes. If you change locked attributes in the parent template, the changes propagate to the derived template.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



After you create derived templates, you can customize them, derive other templates from them, and create instances of them. You can change and modify unlocked attributes in the instances, making adjustments to meet the needs of the specific object you are modeling.

For example, your plant processes can use several models of a pump made by a single vendor. Each model has unique characteristics that map to different attribute values of the DiscreteDevice base template.

To derive a template from another template

1. In the **Template folder** or **Derivation** view pane, select the template to use as the parent template
2. On the **Home** ribbon, in the **Create** area, select **Template**. A derived template is created in the same folder as its parent and placed in name edit mode. The default name is the same as the parent template followed by a numeric sequence.
3. Rename the derived template, if needed. Template names can be up to 32 alphanumeric or special characters, including the required $ as the first character. The second character cannot be $ and the name must include at least one letter. Template names cannot contain spaces.

Note: The following reserved names cannot be used as template names: Me, MyContainer, MyArea, MyHost, MyPlatform, MyEngine and System.

---

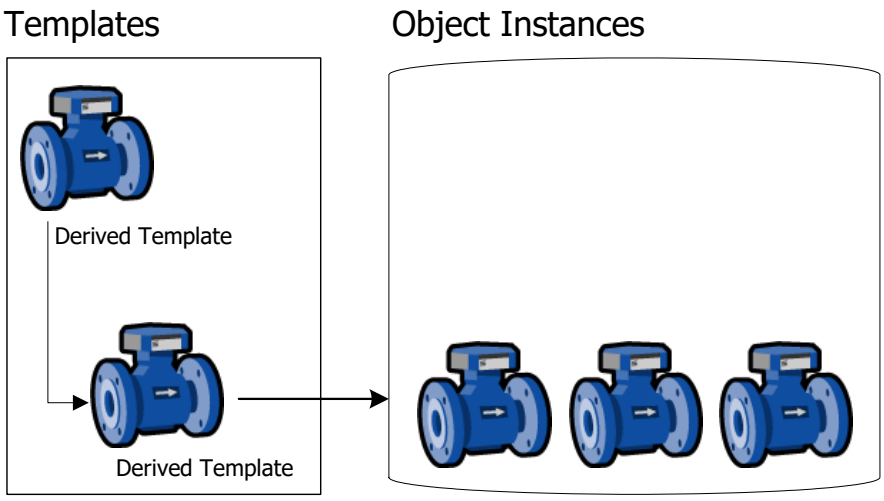
1. You are ready to customize your new template. For more information, see Edit Objects.

Derive Templates from Another Derived Template

You can create derived templates from other derived templates. The child template inherits attributes from all parent templates. Any changed attributes in the immediate parent overrides attributes in grandparent levels.

If you change locked attributes in the parent template, the locked attributes propagate to the derived template.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



A derived template is an exact copy of its parent with the exceptions of locking, security, and the unlocked attributes that have been edited. If you create a new derived template from an existing container template, the new derived template has the same contained templates.

A good practice is to create a hierarchy of derived templates until you reach logical endpoints. Then create instances from each unique derived template.

To create a derived template from a derived template

1. In the **Template folder** or **Derivation** view pane, select the template to use as the parent template
2. On the **Home** ribbon, in the **Create** area, select **Template**. A derived template is created in the same folder as its parent and placed in name edit mode. The default name is the same as the parent template followed by a numeric sequence.

   Template names can be up to 32 alphanumeric or special characters, including the required $ as the first character. The second character cannot be $ and the name must include at least one letter. Template names cannot contain spaces.

   Note: The following reserved names cannot be used as template names: Me, MyContainer, MyArea, MyHost, MyPlatform, MyEngine and System.
3. You can create another derived template by repeating the previous steps, or you can customize your new derived template. For more information about customizing your template, see Edit Objects.

Create Contained Templates

Containment is the relationship in which one object includes another. Containment relationships organize objects in a hierarchy. You can build objects that represent complex devices consisting of smaller, simpler devices.

In scripts, these objects can be referred to by the name that derives from the containment relationship. This name is called a hierarchical name.

An object can have three kinds of names if it is contained by another object. The three names include:

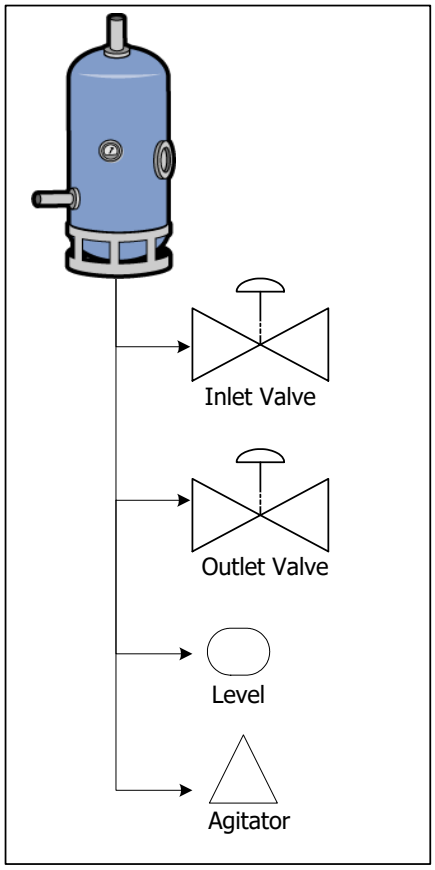
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Name	Description
Tag Name	The unique name of the individual object. For example, `Valve1`.
Contained Name	The name of the object within the context of its container object. For example, the object whose Tag name is Valve1 may also be referred to as `Tank1.Outlet`, if `Tank1` contains it and it has the contained name "Outlet".
Hierarchical Name	Hierarchical names that are fully-qualified names of a contained object include the name of the objects that contain it. Because the object that contains it may also be contained, there are potentially multiple hierarchical names that refer to the same object. For example, if: "Reactor1" contains Tank1 (also known within Reactor1 by its contained name "SurgeTank"). "Tank1" contains Valve1 (also known within Tank1 by its contained name "Outlet"). Valve1 could be referred to as: "Valve1" "Tank1.Outlet" "Reactor1.SurgeTank.Outlet".

Note: Base templates cannot be contained by another template, either as the container or as the template being contained. You can only use containment with derived templates.

Higher level objects contain lower level objects. This allows you to more closely model complex plant equipment, like tank systems. You can nest templates to 10 levels.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: Objects can only contain objects like themselves. For example, ApplicationObjects can only be contained by other ApplicationObjects. Areas can only contain other Areas.

ApplicationObject Containment

ApplicationObjects can be contained by other ApplicationObjects. This provides context for the contained object and a naming hierarchy that provides a powerful tool for referencing objects.

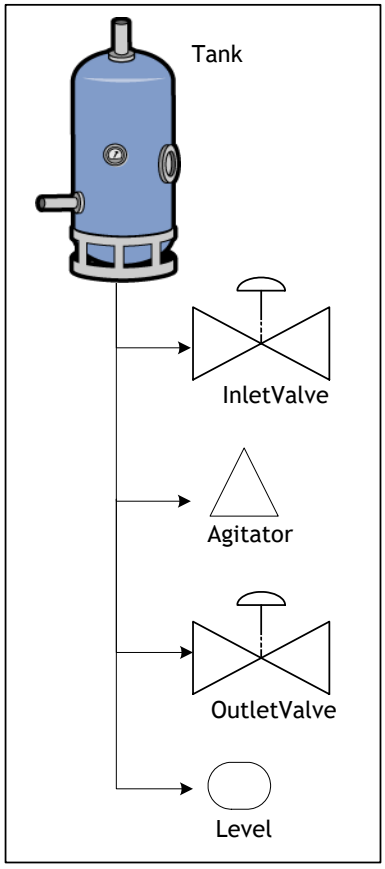
Note: Base templates cannot be contained by another template, either as the container or as the template being contained. You can only use containment with derived templates.

An example of a containment hierarchy is a tank that contains the following objects:

- Inlet Valve
- Agitator
- Outlet Valve
- Level

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

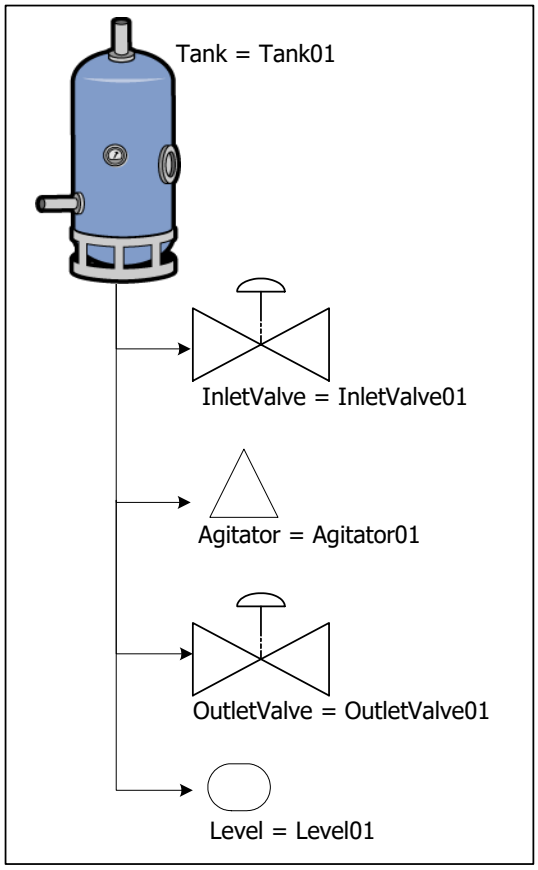
Tank Template



To enable referencing and flexibility within scripting, these objects can be referenced in several different ways. Each object has a unique tag name, such as:

- Inlet Valve = InletValve01
- Agitator = Agitator01
- Outlet Valve = OutletValve01
- Level = Level01

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

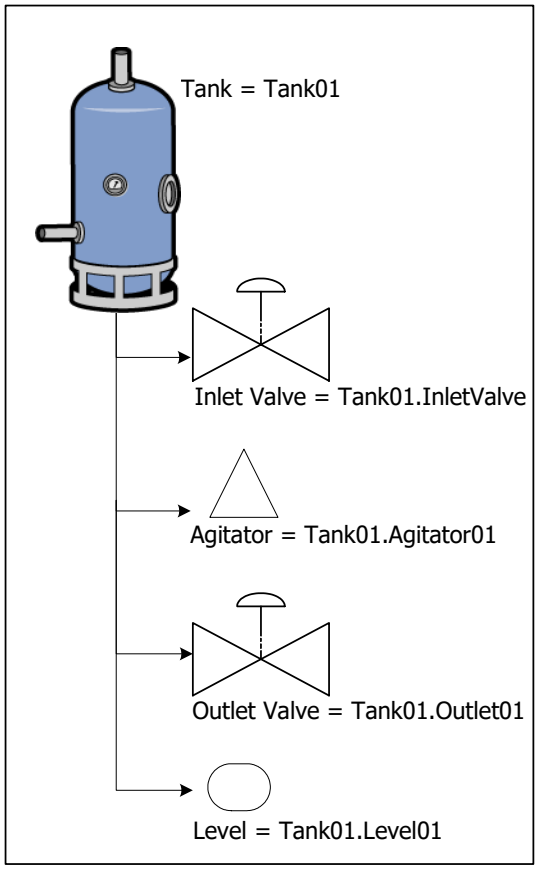


Within the context of each hierarchy, the contained names are unique, in that the names only refer to this tank system and the contained objects.

So if the tank is named `Tank01`, the contained names are:

- `Tank01.InletValve`
- `Tank01.Agitator01`
- `Tank01.Outlet01`
- `Tank01.Level01`

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



This naming convention adds context to the instances contained by `Tank01`.

You can see any object's hierarchical name by opening it and looking at its **Object Information** page.

Additionally, you can use containment references in scripts such as:

- `Me.Outlet`: Allows a script running within the parent object to generically reference its child outlet instance.
- `MyContainer.Inlet`: Allows a script running in any of the children instances to reference another child instance named Inlet that belongs to the same parent.

Using Contained Names

The contained name of a contained object only has to be unique in the context of its container.

An object can have three kinds of names, depending on whether it is contained by another object. The three names include:

Name	Description
Tag name	The unique name of the individual object. For example, `Valve1`.
Contained name	The name of the object within the context of its container object. For example, the object whose tag name is Valve1 may also be referred to as `Tank1.Outlet`, if `Tank1` contains it and it has the contained name "Outlet".

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Name	Description
Hierarchical name	Hierarchical names that are fully-qualified names of a contained object include the names of the objects that contain it. Because the object that contains it may also be contained, there are potentially multiple hierarchical names that refer to the same object. For example, if: "Reactor1" contains Tank1 (also known within Reactor1 by its contained name "SurgeTank"). "Tank1" contains Valve1 (also known within Tank1 by its contained name "Outlet"). Valve1 could be referred to as: "Valve1" "Tank1.Outlet" "Reactor1.SurgeTank.Outlet".

For example, an instance of a $Tank is named Tank01. An instance of $Valve called Valve01 is contained within the instance Tank01.

Change the contained name of Valve01 to InletValve. Now Valve01 can also be referred to by its hierarchical name Tank01.InletValve. The name of the contained object can be changed, though, within the scope of the hierarchy.

Contained names can be up to 32 alphanumeric or special characters. The second character cannot be $ and the name must include at least one letter. You cannot use spaces.

Containment Examples

You can have a Tank object that contains two DiscreteDevice objects that represent its inlet and outlet valves.

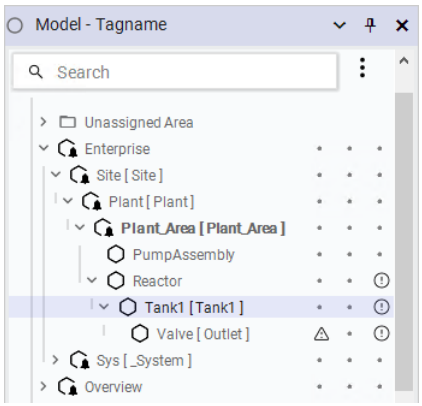
Note: Base templates cannot be contained by another template, either as the container or as the template being contained. You can only use containment with derived templates.

To implement containment

1. Create the following instances: `Tank1` from $UserDefined and `Valve` from $DiscreteDevice. `Valve` has only one name, `Valve`.
2. In the **Model** or **Deployment view**, drag `Valve` on to `Tank1`.

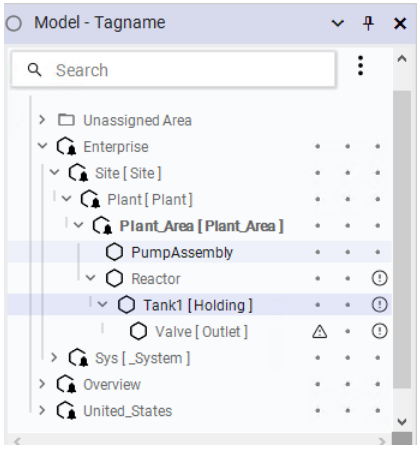
   Note: If `Tank1` already contains an object with a contained name of `Valve`, the Galaxy generates a unique contained name for the newly contained object, such as `Valve_001`.
3. Change the contained name of `Valve` within `Tank1` to `Outlet`. `Valve` can now be referred to by its tagname, `Valve`, as well as its hierarchical name, `Tank1.Outlet`.
4. Create an instance called `Reactor1` from $UserDefined.
5. In the **Model** or **Deployment view**, drag `Tank1` onto `Reactor1`.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



6. Change the contained name of `Tank1` to `Holding`. `Tank1` now has two names, `Tank1` and `Reactor1.Holding`. Also, `Valve` has a three-part hierarchical name: `Reactor1.Holding.Outlet`.

   For the three objects in this example (`Reactor1` containing `Tank1` containing `Valve`), the following naming hierarchy exists:



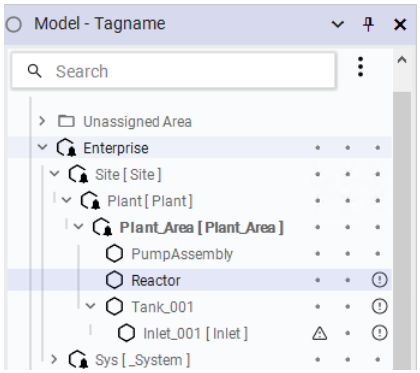
To implement template-level containment

Note: Contained Templates do not have tagnames. When an instance hierarchy is created from a template and its contained children, unique tagnames will be created for the instances based on their contained names.

1. Create the following derived templates: `$Tank` from $UserDefined and `$Valve` from $DiscreteDevice.
2. Derive `$Inlet` from `$Valve`.
3. In the **Template folder**, drag `$Inlet` on to `$Tank`. If `$Tank` already contains a template named `Inlet`, the Galaxy generates a unique tagname for the new contained template, such as `Inlet_001`.

   The contained template now has a hierarchical name `$Tank.Inlet`.
4. Create an instance (`Tank001`) of `$Tank`.
5. The **Model** and **Deployment views** show an instance `Tank001` that contains an instance called `Inlet`.

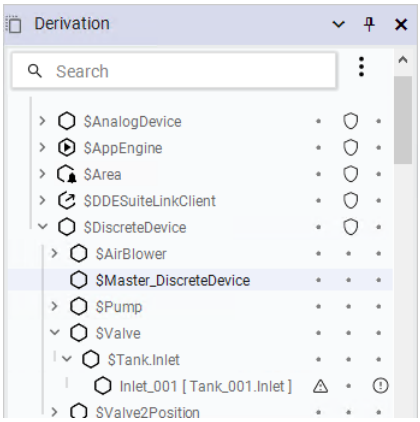
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



View Containment Relationships

Containment relationships appear for templates in the **Template folder**. For instances, the relationship appears in both the **Model** and **Deployment views**.

In the **Derivation** view, if a template contains other templates, you can expand it to show the containment under that template.



The **Derivation view** shows templates and instances with regard to containment in the following ways:

- Non-contained instances show their tagnames.
- Contained instances show their tagnames and hierarchical names.
- Non-contained templates show their template name.
- Contained templates show their hierarchical name.

Rename Contained Objects

Before you rename a contained name of an object, make sure that the object is not checked out to another user or currently deployed.

The new contained name must comply with naming restrictions. Template names can be up to 32 alphanumeric or special characters, including the required $ as the first character. The second character cannot be $ and the name must include at least one letter. You cannot use spaces.

Contained names also cannot be the same contained name as an existing contained object within the same level of hierarchy in the containment relationship.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

WARNING! Be careful when renaming contained objects. References from other objects to the object being renamed are not automatically updated with the new name. You must update the references. Objects with broken references receive bad quality data at run-time.

To rename an object's contained name

1. Select the object in an Application view.
2. On the **Home** ribbon, in the **Edit** area, select **Contained Name**.
3. Type a new contained name.

   All IDEs connected to the Galaxy show the object's new contained name.

Edit Objects

With the Object Editor, you define attributes specific to an object. When you open the Object Editor in non-ReadOnly mode, the object is checked out. No one else can edit an object while you are working with it. If someone else is already working on the object, you can open it to view but you cannot make changes.

The Object Editor contains pages that can be used to modify objects. To view a page in the editor, click its tab. Three pages – **Attributes**, **Scripts**, and **Object Information** – are common to all objects, while other pages are unique to certain object types. If you import a Galaxy or objects with field attributes, the **Field Attributes** page will be present for those objects; the Field Attributes page can also be enabled through a Galaxy user configuration option. See Configure User Preferences for more information.

When you save an object, the configuration data for the object is validated. If errors or warnings are identified during validation, a message appears. You can cancel the save or save the object configuration as it is.

- If you cancel, the Object Editor remains open so you can correct the problems.
- If you save the configuration as it is, the object is placed into a bad or warning state. The object's status is marked in the Galaxy database as Good, Warning or Error. Error means the object is undeployable.

To edit an object in the Object Editor

1. Select the object.
2. On the **Home** ribbon, select **Open**, then select **Open**. A red check mark appears next to the object's icon indicating it is checked out, and the Object Editor opens.

   Note: If you are adding I/O attributes to an application object or system object, such as an area object, the preferred way to add and edit I/O references is through the **IO Devices** view and **Mapping** view. Editing I/O references set for automatic assignment in the Object Editor disables automatic assignment. For more information about I/O auto assignment, see Using I/O Auto Assignment.
3. Make your changes. For more information about locking attributes, see About the Attributes Page. For more information about setting security, see Set Object Security.
4. To save your changes and keep the object open for more editing, on the **Home** ribbon, in the **Save** area, select **Save**.
5. When you are finished editing the object, click the **Save and Close** icon at the top left of the Object Editor.

   NOTE: To keep the object checked out, select the **Keep Checked Out** icon before closing.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Object Editor User Assistance

Tooltips are available in the Object Editor. Point to any editor option and a tooltip appears, showing the attribute name. This name is used when referring to the attribute in scripts, for example.

Each object also includes documentation about usage, configuration, run-time behavior, and attributes. For help with configuring the object, select the object in the Template folder or in an application view and press **Ctrl + F1**, or right-click on the object and select **Object Help** from the context menu. If you want, you can customize the object help file. See Customize Help for more information.

Help File Structure

The header part of the Help file contains the following information:

Tag name	The object's name.
Contained Name	The object's contained name. For more information, see Create Contained Templates.
Description	A short summary of what the object is for.
Code Base	The code version of the object.
Derived From	The immediate parent template for the object.
Object Version	The configuration version of the object.
Process Order	The run-time execution order within the host engine's scan (none, before, after) relative to the **Relative Object** element.
Relative Object	The object that runs before or after in the **Process Order**.

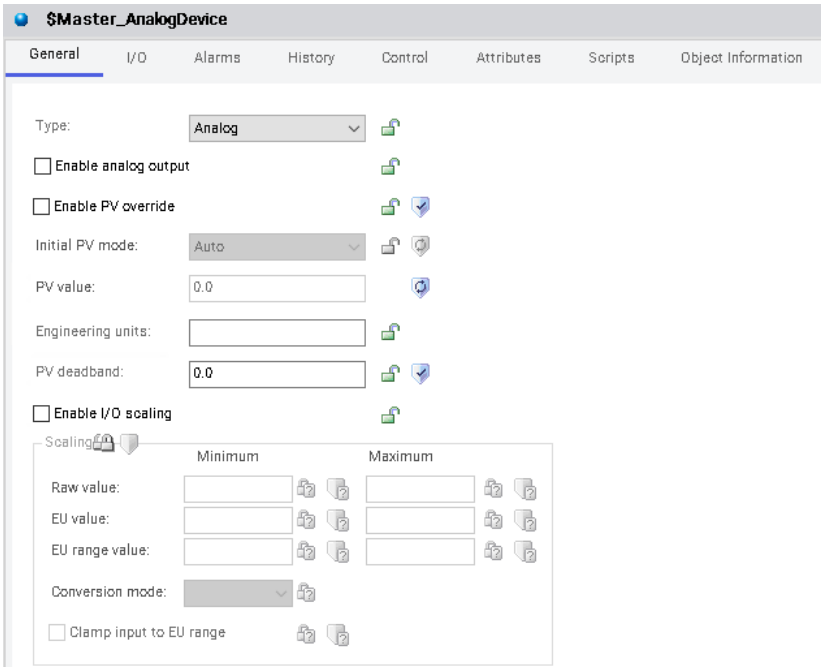
The rest of the help file shows general information about the object.

About the General Editor Layout

When you open the attributes for an instance or a template, you see the Object Editor. The Object Editor is where you configure the object's attributes and add scripts or associated graphics to the object. The Object Editor has several pages related to the type of object you select. If you are working with an instance, you see different pages than if you are working with a template.

For example, the following illustration shows an analog device template.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



When you open the Object Editor, the object is automatically checked out so no other user can work on it. When you close the Object Editor, the object is checked in to the Galaxy database, if it was automatically checked out when the editor was opened.

To keep the object checked out, click **Keep Checked Out** before closing.

To save configuration changes you made and check the object back in, click the **Save and Close** icon.

After the object is checked in, other users can edit it.

Lock and Unlock Template Attributes

When you create derived templates, you can lock or unlock some or all of the attributes and enabled features. Locking an attribute prevents it from being changed in derived templates or instances. You can only lock attributes in templates.

Locking an attribute in a template specifies that its value or setting is inherited by all derived objects, both templates and instances. Locking an attribute also makes the attribute act as a constant during run time.

Note: When you lock values and settings for an attribute, they are not visible when you use Object Wizards, and thus the locked values and settings will be inherited by all derived objects.

In the **Attributes** page, you must enable locking before an attribute or any of its features can be locked in a template. To enable locking, click the **Show/Hide Lock** icon to the right of the attribute name. When locking is enabled, lock symbols will appear next to values that can be locked. Lock symbols are not visible in the template or its derived instances unless locking is enabled.

- Attributes that are locked in a parent template are referred to as "locked in parent." This parent can be at any parent level above the selected object.
- Attributes that are locked in a template are referred to as "locked in me."

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Caution: When using I/O auto assignment, do not lock the "Read from" field (input source) or "Write to" field (output destination). If you lock these fields in the parent template, they cannot be updated with the resolved reference when the object is deployed and the runtime value will be "---Auto---".

See Group Locking/Security for information about locking or unlocking all of a feature's attributes with a single click.

Lock controls and status are shown with an icon. If the option is enabled, click the lock control to switch it between locked and unlocked. These icons mean:

Icon	Name	Description
	Locked (in me)	The associated attribute is locked (in me) and enabled. Only templates can have this kind of lock. The attribute value is read/write. Derived templates and instances do not have a unique copy of this attribute. Child objects share the locked attribute of the parent. Changing the value of a locked attribute in the parent template updates the value of that attribute in all derived templates and instances.
	Locked (in parent)	The associated attribute is locked in the parent object and cannot be unlocked or modified by the child object. Both templates and instances can have these. The attribute is read-only. The templates and objects do not have a unique copy of this attribute, but instead use the attribute value in the parent where the attribute is locked.
	Unlocked	The associated attribute is unlocked and enabled. Both templates and instances can have this kind of lock. The attribute is read/write. The object has its own copy of the attribute value and the value is not shared by derived objects.
	Indeterminate	Refers to a specified group of options. An indeterminate state indicates different lock states for individual options in the group.
	Undefined	The associated attribute doesn't exist. This indicates that another attribute must be enabled before the associated attribute is created and before its lock status can be determined.

Note: Locking an attribute during configuration makes its value a constant. You cannot write to locked attributes during run time.

Example: To lock an attribute

1. Create a derived template from the $DiscreteDevice base template. Name the derived template `$Valve`.
2. Open the `$Valve` template and on the **States** page, lock one of the attributes by clicking the **Lock** icon.
3. Save `$Valve`.
4. Create a derived template from `$Valve`. Name it `$BigValve`.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

5. Create an instance from `$Valve` named `Valve1`.

   In the editor of `$Valve`, the attribute lock icon shows the attribute is locked in `me`.

   You cannot change the attribute value in `$BigValve` and `Valve1`. The editor options for the attribute are disabled and the lock icon, if shown, indicates a lock in the parent. Also, the attribute lock icon in children derived from `$Valve` is now locked and disabled.

   If you change the attribute value in `$Valve`, the change propagates to `$BigValve` and `Valve1` after you save the changes.

Example: To unlock an attribute

1. Using the objects from the previous example, in the `$Valve` template's editor, unlock the locked attribute.
2. Save `$Valve`.

   In the editor for `$Valve`, the attribute lock icon shows it is unlocked.

   The lock type for this attribute of `$BigValve` now indicates locked in `me`. The lock type for this attribute of the `Valve1` instance shows unlocked but the locking icon is unavailable.

Set Object Security

Operators interact with objects through the individual attributes of those objects. Each attribute on the Object Editor that can be modified by operators at run time can have an associated security control, which is used to modify its run-time security classification.

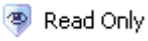
In the **Attributes** page, security icons must be enabled before you can set security for an attribute or any of its features in a template. To enable security, click the **Show/Hide Security** icon to the right of the description field.

When security is enabled, security symbols will appear next to values for which security is configurable. Security symbols are not visible in the template or its derived instances unless enabled in the parent template.

If an attribute's security classification is configurable, click the security control to select one of seven possible states:

Security Icon	Description
Free Access	Lets you change this value without restriction even if you have no defined permissions on the object. Anyone can write to these attributes to perform safety or time critical tasks that can be hampered by an untimely logon request, such as halting a failing process.
Operate	Lets you work with Operate permissions to do certain normal day-to-day tasks. These include writing to attributes like Setpoint or Command for a Discrete Device object. This level of security requires you to have Operate permission for the security group for the object.
Secured Write	Requires you to authenticate using your user name and password each time you want to write to the attribute. You also need to have Operate permissions for the object.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Security Icon	Description
Verified Write	Requires you to have Operate permissions to log on again and a second, different user to also log on as the verifier before writing to the attribute. The verifier needs to have Can Verify Writes permission for the object.
Tune	Allows end users with Tune Operational permissions to tune the attribute in the run-time environment. Examples of tuning are attributes that adjust alarm setpoints and PID sensitivity.
Configure	Allows end users with Configure Operational permissions to configure the attribute's value. Requires that the user first put the object off scan. Writing to these attributes is considered a significant configuration change. For example, a PLC register that defines a Discrete Device input.
 Read Only	Only allows users to read this attribute's value in the run-time environment. This attribute is never written to at run time, regardless of the user's permissions.

If an attribute's security is shown in gray, its security classification is locked in its parent object and cannot be changed, or it requires the enabling of a group attribute.

Group Locking/Security

The lock and security controls associated with option groups and features quickly set those conditions for all options in the group.

The group control typically reflects the setting for all options in the group or feature set. But, if at least one option in the group has a lock or security control that is different from the other options, the group control shows an indeterminate icon.

In addition to the undefined controls, the group controls for locking and security are the same as those for individual options.

You can lock or unlock all of the attributes associated with a feature by selecting the lock icon next to the feature name, after you activate the feature. This will lock or unlock all of the attributes for the feature, unless an attribute was locked at a higher level. For example, if you locked an attribute in a template, and then created another derived template, the attribute that was locked in the original template cannot be unlocked in the derived template (it is locked in parent).

If an attribute is locked in the template, you can change the value in that template, but not in the derived children. If you change the value in the parent template, the change propagates to all child objects. For more information, see Configuring Objects.

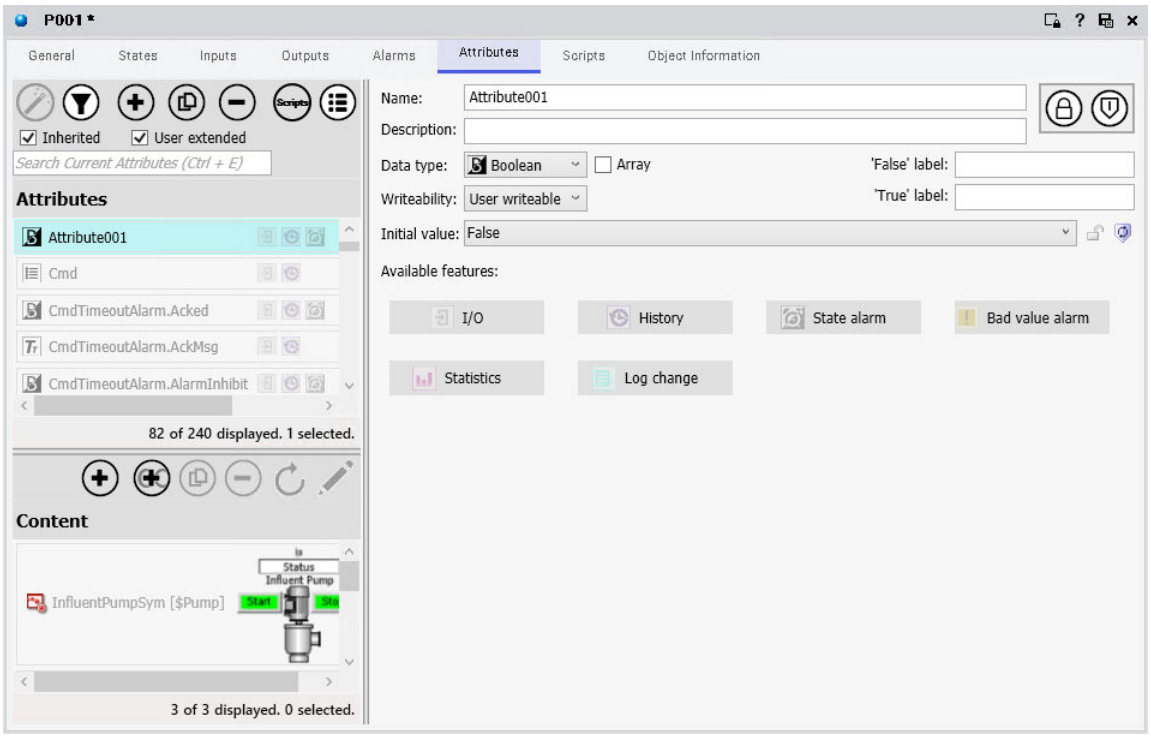
About the Attributes Page

The **Attributes** page divides into two or more panes when you initially open it. The number of panes the **Attributes** page divides in depends on whether or not attributes or graphics have been defined for the object, whether the object is an instance or a template, and if it is an instance, whether or not it is derived from a template that includes an Object Wizard.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

If the object you are editing does not have any attributes defined, the **Attributes** page will be empty. The **Attributes** page will also show a pane for graphics, and depending on the type of object (template or instance) and its contents, may show additional Object Wizard configuration panes.

You can activate various features, such as I/O, History, State Alarm and Statistics. See Configuring Objects for more information about attributes and features. When you add an attribute to an object, information about the attribute is shown.



Opens the **Object Wizard** editor.

To filter attributes, click the **Filter** button. You can then select filtering criteria by checking source type, enabled feature type, writeability type, lock status, data type, visibility type (hidden or not hidden), and diagnostic type (configuration, run time, or both).

Adds a new attribute.

Duplicates the selected attributes.

Removes the selected attributes.

To display the **Scripts** pane used to associate scripts with an Object Wizard, click the **Scripts** button.

Shows Details about an attribute. The name, description, and icons representing activated features for the object's attributes will be shown.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When you add an attribute to an object, the **Attributes** page divides into three sections. The left side of the page lists attributes, the top right shows information about the currently selected attribute, and the bottom of the right side contains fields for configuring features.

You can search attributes by entering the characters you are trying to find in the **Search Current Attributes** text box. This will display attributes that contain the characters you enter.

Note: The search function finds attributes that contain the search term anywhere within the attribute name. For example, if you want to locate all attributes associated with Pump125, you could enter "125" in the text box. All attributes with 125 anywhere in their names will be shown.

You can also search for attributes by typing the character you are searching for when the cursor is pointed at the attribute list. The first attribute that contains the character, anywhere within its name, will be highlighted.

Note: The search begins at the attribute that is currently selected. Selection will move to the next attribute that contains the character entered.

- On the left side of the page, all attribute names for the object are listed. Pressing the **Show Details** button reveals additional information about the listed attributes. Check boxes are provided for **Inherited** and **User extended**.
  - **Inherited**: Checking this displays attributes that were added in the parent template used to create the object.
  - **User extended**: Checking this displays details about any features you added to the attribute, for example, for I/O.
- At the top of the right side, information for the current attribute is listed. Additional fields may be listed, depending on the data type of the attribute. However, all data types include the following properties:

Property	Description
Name	Attribute name (for example, Attribute001).
Description	User defined; a description of the attribute.
Data type	Can be set to Boolean, Integer, Float, Double, String, Time, ElapsedTime, or InternationalizedString. The data type determines which features can be added to an attribute. For example, when configuring a Boolean data type, only two alarm features are available: **State Alarm** and **Bad Value Alarm**. When configuring an Integer data type, there are four available alarm features: **Limit Alarms**, **ROC Alarms**, **Deviation Alarms** and **Bad Value Alarms**. To create an array, check the **Array** option and specify the array's length in the **# of elements** box. You can create an array for each data type except **InternationalizedString**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Property	Description
Writeability	Can be set to Calculated, Calculated retentive, Object writeable, or User writeable. See the *Object Viewer User Guide* for additional information about writeability categories. • Calculated: Permits only scripts within the same object to write to the attribute. Calculated attributes are not saved across restarts. If you select Calculated for an attribute, only scripts running on the same object can write to the attribute. • Calculated Retentive: Permits only scripts within the same object to write to the attribute. Calculated Retentive attributes are saved across engine restarts. • Object Writeable: Permits other objects to write to this attribute in addition to being set by scripts within this object. Object Writeable attributes are saved across restarts. This category is not user writeable. • User Writeable: Permits other users to write to this attribute in addition to being set by scripts and objects throughout the system. User Writeable attributes are saved across restarts. They can be locked at configuration time. If locked, they are read-only.
Initial value	Specifies the initial value for the attribute when the object is deployed. Enter value data for each data type. In the case of a non-arrayed **Boolean**, select **True** or **False** in the list box. For an arrayed Boolean, select the appropriate checkbox.

Some attribute text boxes may show --- (dash dash dash). The --- is a placeholder reference that does not cause the associated object to be placed in a warning configuration status when it is validated. You may also see attribute text boxes showing a ---.--- (dash dash dash dot dash dash dash). You need to provide a valid reference in the text box. The ---.--- placeholder causes the associated object to be placed in a warning configuration status when the associated object is validated.

Important: Edit the I/O auto assignment placeholder in the Object Editor ONLY if you do NOT want to use I/O auto assignment for the object. In most cases, I/O auto assignment is the preferred method.

- At the bottom of the right side, fields for configuring activated features are displayed. Available features include:

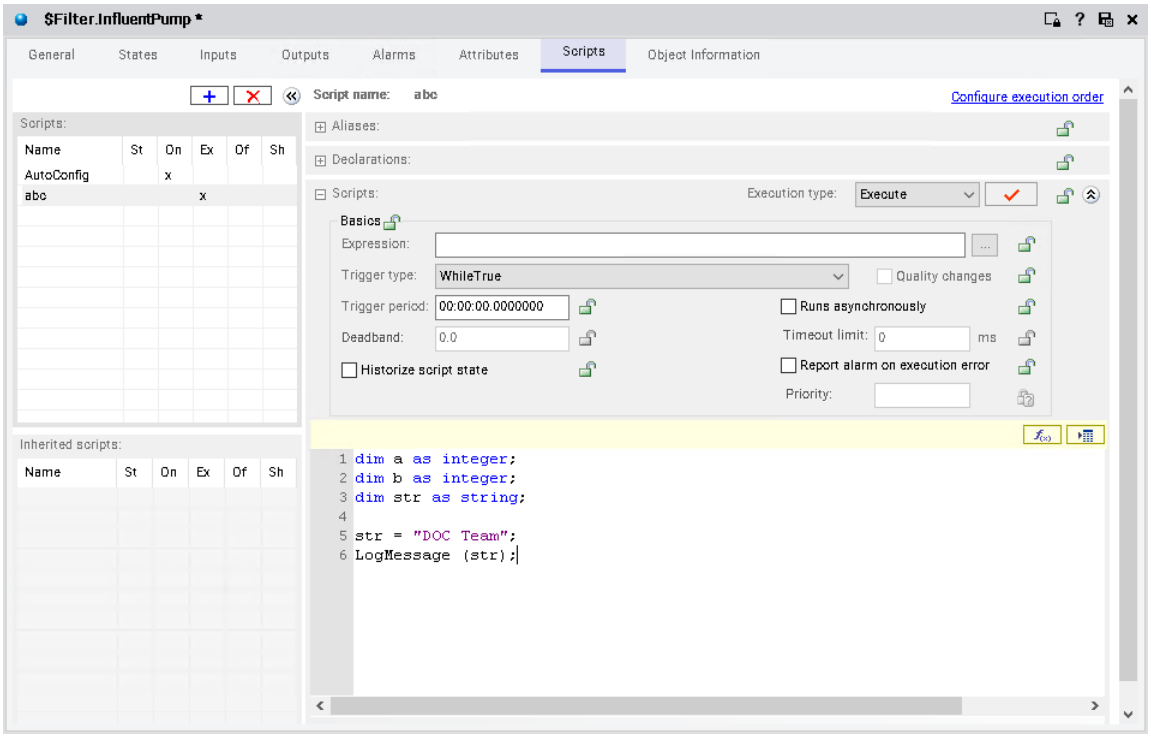
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Feature	Description
I/O	The I/O feature is used to activate the attribute for I/O. You can select **Read** (input), **Read/Write** (input and output), or **Write** (output). When you select the I/O feature for application objects and system objects, such as areas, the "Read from / Write to attribute text box contains the placeholder reference `<IODevice>. [HierarchicalName].[AttributeName].InputSource` or `<IODevice>.[HierarchicalName].[AttributeName]. OutputDest`. You should not edit these placeholders in the Object Editor, but instead use the **IO Devices view** to assign the object to a scan group. This will allow the placeholder to automatically resolve to the correct I/O reference. See Using I/O Auto Assignment for additional information.
History	The History feature enables historization to the Historian. **Note**: The AppEngine hosting the object must be configured for historization.
State alarm	The State Alarm feature is available for Boolean data types. You can set alarm priorities, the alarm state, an alarm message, and time deadband.
Limit alarms	The Limit Alarms feature is available for integer, float, and double data types. With it, you can set value alarms (Hi, HiHi, Lo, LoLo), and the limits, priority, and messages to apply to each alarm limit.
ROC alarms	The Rate of Change (ROC) Alarm feature is available for integer, float, and double data types. With it, you can set alarms if changes within a time period exceed specified values, and the limits, priority, messages, and time period to apply to each ROC alarm.
Deviation alarms	The Deviation Alarm feature is available for integer, float, and double data types. With it, you can set deviation alarms (major and minor), and the tolerance, priority, messages, target value, deadband, and settling period.
Bad value alarm	The Bad Value Alarm feature adds an alarm if the value returned from the attribute is determined to be bad quality.
Statistics	The Statistics feature monitors statistics associated with the object.
Log change	The Log Change feature will cause the attribute to generate an event each time the attribute value changes.

About the Scripts Page

The **Scripts** page is divided into several areas. To learn more about using scripts, see Writing and Editing Scripts.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The main areas of the **Scripts** page include:

- **Scripts** list: Shows all scripts currently associated with the object. The columns indicate which kind of trigger the script uses: Startup, On Scan, Execute, Off Scan and Shutdown. Click the **Add** button (+) to add a new script.
- **Inherited scripts name** list: Shows all scripts associated with the object's parent. The columns indicate which kind of trigger the script uses: Startup, On Scan, Execute, Off Scan and Shutdown.
- **Aliases** area: Lets you create and modify aliases that apply to the script you are working on. Aliases are logically descriptive names for typically long reference strings that you can use in the script to make the script more readable.
- **Declarations** area: Provides a place to add variable declaration statements, such as `DIM MyArray[1] as FLOAT;`. These declared variables live from the start to the shutdown of the object and can be used to hold values that persist from one execution of the script to the next. They apply only to the script in which they are declared.
- **Basics** area: Provides a location in which you set the expression, triggering conditions, and other settings that run the script in the run-time environment. See Writing and Editing Scripts for descriptions of triggers and when they are executed. This area includes:

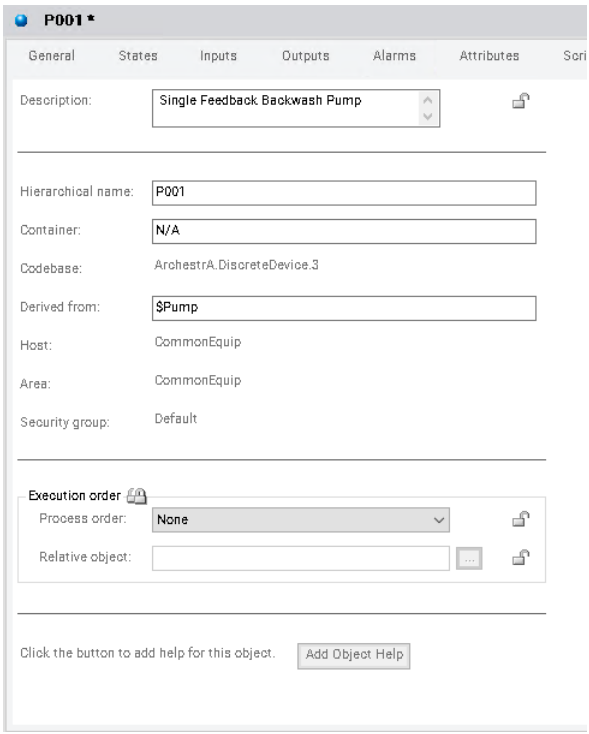
  Configure Execution Order: Sets the execution order of multiple scripts (inherited and local) associated with this object.

  Historize Script State: Select to send the state of the script to the Historian.
- **Script Creation** box: Shows the script you are writing.

About the Object Information Page

The **Object Information** page is common to all object configuration editors.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



This page includes the following fields:

- **Description**: A short summary of the object's purpose.
- **Hierarchical Name**: The fully qualified name of a contained object, including the container object's TagName.
- **Container**: The name of the other object that contains this object, if applicable.
- **Codebase**: The code version of the object.
- **Derived From**: The immediate parent template of the object, either a base or derived template.
- **Host**: Another object to which the object is assigned (for example, a WinPlatform hosts an AppEngine). An object's host determines where an object will run when it is deployed.
- **Area**: An object that represents a logical grouping to which this object belongs. An object's area mostly affects the way in which its alarms are reported to alarm clients.
- **Security group**: The security group the object is associated with.
- **Execution order**: If you want this object to run before or after another object within its engine's scan, select from the **Process order** list. Click the **Browse** button to specify the **Relative object** in the **Attribute Browser**. For more information about the **Attribute Browser,** see Reference Objects Using the Galaxy Browser.
- **Add Object Help**: Opens a copy of the HTML help page for the template this object is derived from. You can edit this information. This allows you to create Help about the object you are currently configuring for downstream users. This Help appears when you select an object in a view and then click **Object Help** on the **Help** menu. See Customize Help for more information.

Customize Help

Do not use Microsoft Word as an editor to create downstream object HTML help pages. Use an HTML or plain text editor instead.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

If clicking **Add Object Help** opens Word on your computer, change the program associated with editing HTM files. Open the Windows Explorer's **Folder Options** dialog box and go to the **File Types** page to make this change. For more information about associating programs with files, see your Windows help.

Locate the Help Folders

The path to each object's Help folder is unique. It depends on the path you selected when you installed the Galaxy Repository. The default path to an object's Help is:

```
\<Installation Path>\Program Files
(86)\ArchestrA\Framework\FileRepository\<YourGalaxyName>\Objects\<TheObjectID>\Help\1033
```

To add images to the Help file, place the images in the proper folder on the Galaxy Repository computer and use a relative path to those images in the HTML file.

For the previous example, place images in the `\1033` folder or create an images folder under it.

Reference Objects Using the Galaxy Browser

Use the Galaxy Browser to browse for:

- Attributes of objects. You can quickly find an object attribute or attribute property and add a reference to it when you are configuring an object.
- Industrial Graphics.
- Graphic element properties.

The Galaxy Browser shows attributes, graphics, or attributes and elements, depending on what you are doing at the time you access the browser.

Browse for Attributes

You use the Galaxy Browser to browse for:

- Attributes of objects, either instances or own relative references.
- Attributes of templates.

You can open the Galaxy Browser to browse for attributes from:

- Within an AutomationObject Editor. For example, from a script, from an attribute of type MxReference, or from a custom alarm message attribute field).
- Within an Industrial Graphics Editor. For example, to use in scripts, animation links and references, properties and custom properties.
- Within InTouch WindowMaker. For example, to use in a reference expression from an animation link or a script.

The Galaxy Browser shows objects in the left pane and the attributes associated with the current selection in the right pane. Only attributes that can be referenced at run time are shown. You can browse "Me." references for alarm messages only.

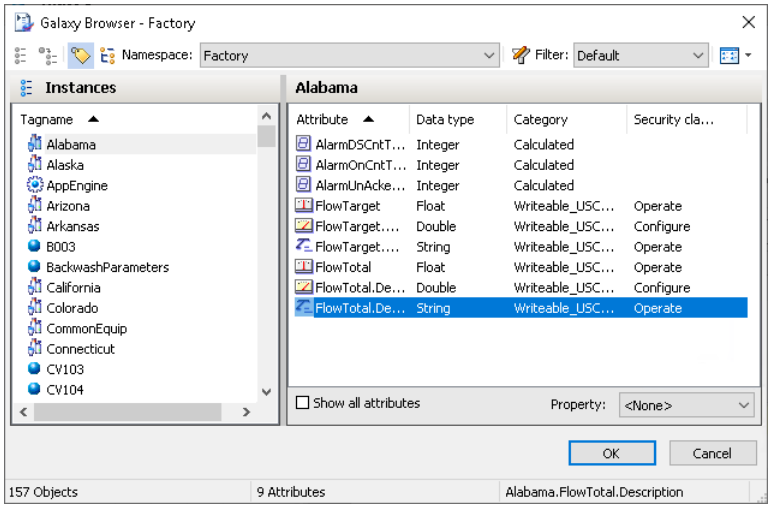
When you open the Galaxy Browser, the last browsed location for attributes is shown. The Galaxy Browser shows the object list based on the last used state (Tagname or Hierarchical name). If the last used state of the browser was Tagname and the selected editor reference is a Hierarchical name, the browser opens in Tagname mode.

The status bar displays the attribute property name and the graphic element attribute name and description.

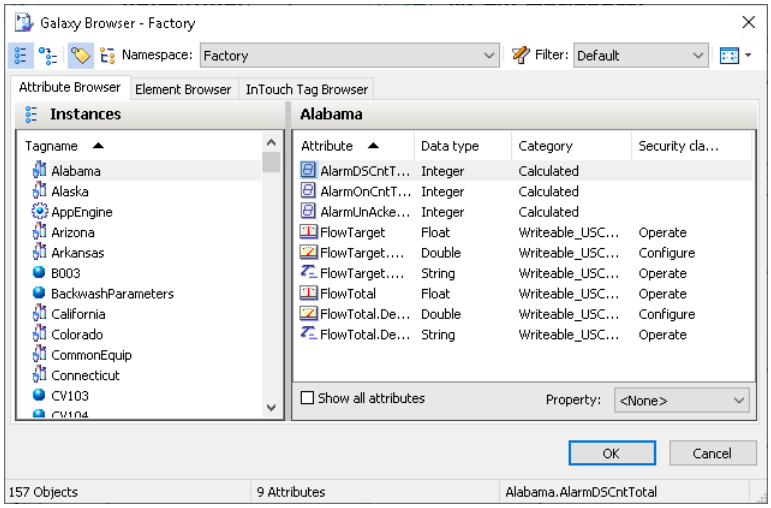
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To browse for attributes

1. In any area on a page, click the **Browse** button (...), if available. The Galaxy Browser opens.



   If you are browsing for attributes to use with an Industrial Graphic, such as for an animation or script, the Galaxy Browser shows the attributes in an **Attribute Browser** tab.



2. By default, the browser shows only those attributes that are frequently accessed. If you are viewing the attributes of an object for the first time, the right pane can be blank. Select the **Show all attributes** check box to show all of the object's attributes.
3. To filter the list of tagnames, click the **Filter** button. For information about configuring a filter, see Create a Filter for the Galaxy Browser. To switch the content of the left pane between a **Tagname** and **Hierarchical Name** list of objects, click the **Show Tag name** or **Show Hierarchical name** buttons.

4. You do not have to explicitly make a selection in the **Property** list. If you only select an attribute (leaving **Property** set to <none>), the property of the attribute defaults to **Value**.

   If the option in the Object Editor is already configured with an object reference, the **Attribute Browser** shows it or expands to the nearest matching object/attribute/property currently configured in the Galaxy.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

If you selected text in the script editor, that text is used as the initial reference string and the browser finds the nearest attribute reference to the selected text

5. When you are done selecting the attribute/property, click **OK** to place the reference into the Object Editor and close the **Galaxy Browser**.
   - The fully-qualified reference string appears in the editor option.
   - If you are working in the script editor, the selected reference appears in the script at the current cursor position and replaces text that was selected.

View Attribute Details in the Galaxy Browser

When you view attributes in the Galaxy Browser, you see two areas. The objects shown in the left area include all of the logged in user's checked-out objects plus the checked-in versions of all other objects.

Important: The Galaxy Browser shows only the Primary AppEngine and its attributes of a redundant pair. Any Backup AppEngine is not shown. See Working with AppEngine Redundancy in the Application Server help for more information.

The right area shows the attributes of the object selected in the left pane. Depending on the attribute selected, you can see these properties:

<none>	Automatically defaults to the Value property of the selected attribute.
Category	Determines when and where the attribute's data exists (for example, configuration or run time), which users can write to it, and whether the attribute is lockable or unlockable.
Dimension1 (only for arrays)	Returns the dimension of the attribute if it is an array.
Locked	Determines whether the attribute is currently locked. Valid values are: Unlocked LockedInMe LockedInParent.
Quality	The quality of the attribute as defined in the OPC Draft 3.0 quality definition. OPC quality is stored and transported as a 16-bit value. OPC quality is stored for an attribute as a current quality, and it can be historized and sent to clients.
SecurityClassification	Determines which permissions a user has with respect to the attribute when using an AVEVA application in the run-time environment. Relevant only for attributes that can be written to by users in the run-time environment. If an attribute has no security, this column is blank.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Data Type	The data type of the attribute: • Integer • Boolean • Float • Double • String • Internationalized String • Time • ElapsedTime • ReferenceType • CategorizedStatusType • DataTypeEnum • SecurityClassificationEnum • DataQualityType • CustomEnum • CustomStruct For information about each of these, see the help for the object.
Value	The primary value of the attribute. Sometimes, a list of numbers is included in the **Property** list. Those numbers map to single bits in an integer attribute's Value property. Valid bit field specifiers are: .00 (least significant bit) .01 .02 .03 .04 .05 .06 .07 .08 .09 .10 .11 .12 .13 .14 .15 .16 .17 .18 .19 .20 .21 .22 .23 .24 .25 .26 .27 .28 .29 .30 .31 (most significant bit)

Important: Bit field specifiers are not allowed for integer arrays. Although bit field access is only supported in integers, they appear to be allowed for data types besides integer because they do not cause a warning during configuration. They cause errors in the run-time environment.

Browse for Graphics

You use the Galaxy Browser to browse for graphics from:

- The Industrial Graphics Editor, when editing a graphic from the Galaxy, being either a graphic from an object (Template or Instance) or a graphic from the Visualization folder.
- InTouch WindowMaker when editing a managed InTouch application hosted by the Galaxy.

The graphic currently being edited (or browsed from) does not appear in the list of graphics.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Within the same user session, the Galaxy Browser remembers the last browsed location for graphics and presents it whenever called as the starting location so that context is kept. Initial default location for graphic browsing is the Visualization folder with the root selected (Galaxy node).

You can use the Galaxy Browser to browse graphics from:

The Visualization folder. The browser shows the Visualization folder organization on the tree in the left pane and a list of the graphics contained in the currently selected node (Galaxy node or folder node) in the right pane.

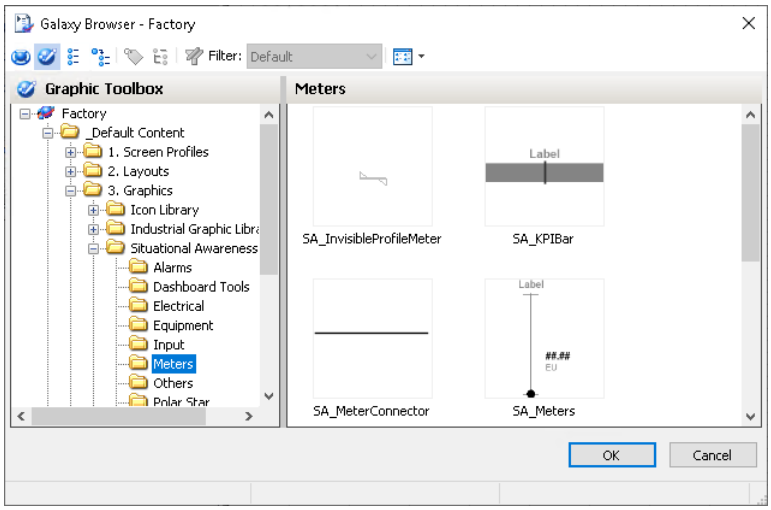
AutomationObject templates. The browser shows the Template folder in the left pane and the graphics associated with the currently selected template in the right pane.

AutomationObject instances. The browser shows a flat list of existing instances of objects in the left pane. The right pane shows the graphics associated with the currently selected instance. You create or apply filters to reduce the scope of the instances shown in the left pane.

Relative references. This is possible only when you edit a graphic belonging to an object and browse for graphics from that specific object.

To browse for graphics from the Graphic Editor

1. Click the **Embed Graphic** button in the Graphic Editor. The Galaxy Browser opens, showing the location of the graphics in the left pane and the graphics associated with the current selection in the right pane.



2. Select a graphic from the list and then click **OK**.
3. Click in the canvas to place the graphic.

Browse for Element Properties

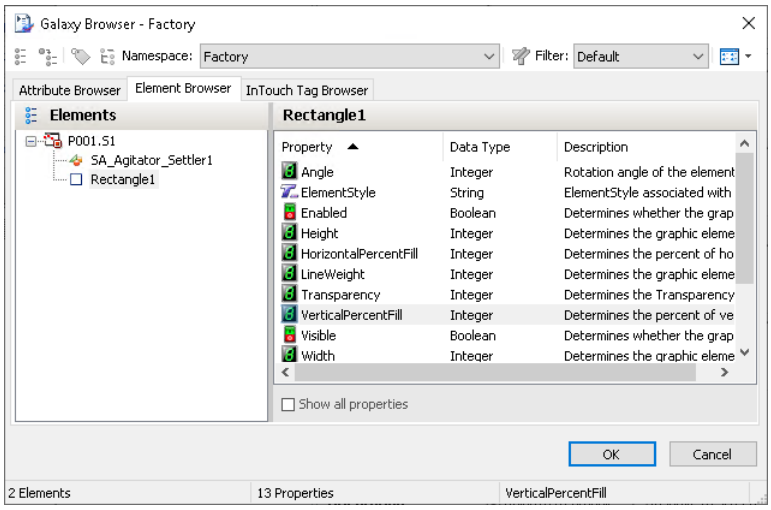
If you are working on a graphic, you can create references to properties of other graphics. For example, you can reference another graphic's properties from an animation link or script. You can browse the properties of all elements on the canvas or custom properties.

To browse for element properties

1. In any area on a page, click the **Browse** (...) button, if available. The Galaxy Browser opens.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. Click the **Element Browser** tab.



3. By default, the browser shows only those properties that are frequently accessed. If you are viewing the properties of an element for the first time, the right pane can be blank. Select the **Show all properties** check box to show all of the object's attributes.
4. Select the property and then click **OK**.
    - The fully-qualified reference string appears in the option.
    - If you are working in the script editor, the selected reference appears in the script at the current cursor position and replaces text that was selected.

For more information, see "Working with Element Styles" in the *Creating and Managing Industrial Graphics User Guide*.

Create a Filter for the Galaxy Browser

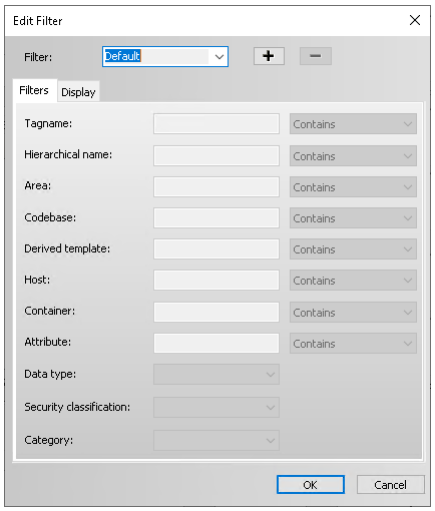
You can create one or more filters that limit the list based on the object name or common attributes. You can also configure the columns you want to show for the list.

The Default filter provides an unfiltered list of objects and attributes. It cannot be edited.

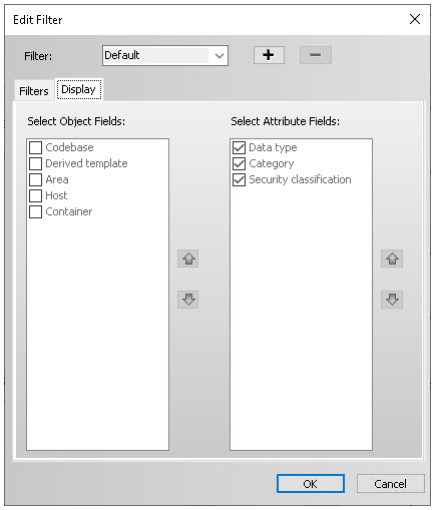
To create a filter

1. Click the **Filter** icon. The **Edit Filter** dialog box appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. Click the **Plus** button and type a name for your new filter.
3. Click the **Filter** tab.
4. Configure the filter details.
5. Click the **Display** tab.



6. Configure the columns to show in the right pane of the **Galaxy Browser**. Use the up and down arrows to set the order for the columns.
7. Click **OK**. The new filter appears in the **Filter** list.

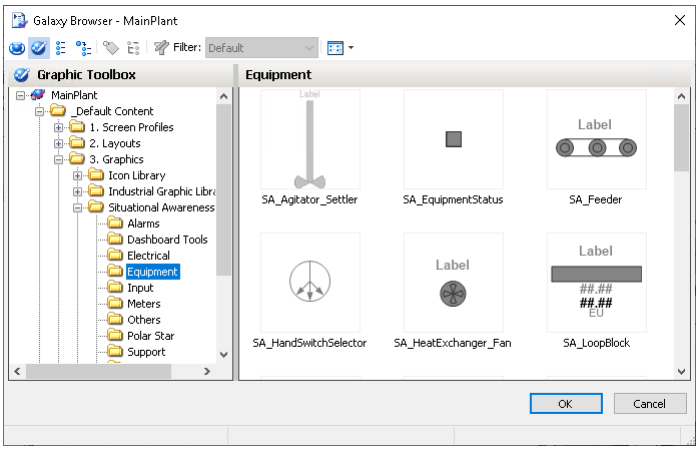
Change How Information is Shown in the Galaxy Browser

You can change how information shown in the right pane of the Galaxy Browser. You can view:

- A list of only the attribute or graphic names.
- A list of details for attributes or graphics.
- Named graphic icons.
- Thumbnails for graphics.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The following figure shows graphic thumbnails.



Manage objects

After you create several objects, like templates, you need to manage them. For example, you need to check objects in and out, you need to validate objects, and you may need to rename objects. You can also export and import objects, allowing you to reuse objects created in one Galaxy in another Galaxy.

Check out objects

To make changes to an object, you must check out the object. Then, you can modify the object and save private versions of it before checking the object in for other users to use. You can select more than one object for checking out at the same time.

The Galaxy marks the objects as checked out to you and it updates the object's **Change Log** that you can view in the **Properties** dialog box. A check mark is shown next to an object's icon in the IDE. No one else can check out the object until you check it back in or until you perform an Undo **Checkout operation**. However, others can open the object for read-only viewing.

To find objects that have been checked out, use the **Find** dialog box.

To check objects out

1. In the **Template folder** or **Application views**, select the objects you want to work with.
2. Do one of the following:
   - To check out the object without opening the Object Editor, on the **Home** ribbon, select **Check-in**, then select **Check Out**.
   - To check out and edit the object, open it in the Object Editor. An object is automatically checked out to you when you open its editor.

Check in objects

After you finish making your changes, you check the object back into the Galaxy. When you check the object back in, a dialog box prompts you to enter comments about changes you made.

You can turn this dialog box off if you do not want to enter information about your changes. For more information, see Customize Your Workspace.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: If the object was automatically checked out when the editor was started and you close the editor without making any changes to the object's configuration, an undo-checkout is automatically performed.

To check in an object to the Galaxy database

If the object is open in the Object Editor:

- Click the **Save and Close** icon at the top right of the Object Editor window.

If the object is not open in the Object Editor:

1. In an **Application** view or the **Template folder**, select the object after you are finished making your changes.
2. On the **Home** ribbon, select **Check-in,** then select **Check-in**. The **Check-in** dialog box appears.
3. Type any comments you want and click **Check-in**.

Validate objects

Objects need to be validated before they can be deployed. An object validates its configuration either when you are configuring it, or typically when you save that configuration to the Galaxy database.

Validating an object's configuration includes:

- Checking allowable attribute value ranges.
- Compiling its scripts.
- Verifying its attribute references.
- Validating its extensions.
- Validating other configuration parameters that are unique to the object.

Important: Script validation on a template does not resolve references used in the script. For example, references to attributes that do not exist will not be discovered.

Typically, each option on the Object Editor that requires a string or numeric input has an allowable range of inputs. If you type an input outside the allowable range and then try to change the Object Editor page, close the Object Editor or save the object's configuration, a message appears about the input error, showing the allowable range.

To open the Validation area

- On the **View** ribbon, select **Operations**. The **Operations** pane, where validation information will be displayed, opens.

Validate scripts and other external components

Some objects depend on external components to run, such as script function libraries and references to other objects' attributes. The status of these external components can change, perhaps disabling some capability of the object.

For example, an object refers to a value of an attribute of another object, which is subsequently deleted. This will result in the remaining object going to a Warning status.

Normally, the system will update the validation status of an object when the missing script function or object/attribute is later added to the system. But there are a few cases where the status of an object needs to be manually validated by the user.

For example:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- When importing scripts and script libraries, there are cases when the script will import before the associated library and validate incorrectly, and
- When graphics associated with an object are imported along with a graphic they embed, the containing graphic may be imported first and validated incorrectly.

In each of these situations, the object may incorrectly have a status of either Bad or Warning. In this case, you may want to manually validate the object to update its status, especially if the status is preventing the object from being deployed. For more information, see Validate an object manually.

Two kinds of indicators are shown in the object icons:

- Deployment status for instances only.
- Configuration status for templates and instances.

Validate an object manually

After you check an object in, you can verify that an object's configuration is valid and update its status by manually validating it. You can use the **Template folder**, the **Application views** or the **Find** dialog box to find objects that need to be validated.

To validate all objects in the Galaxy, validate the Galaxy object.

Note: For a large Galaxy this is potentially a time consuming operation, and should be used only when necessary.

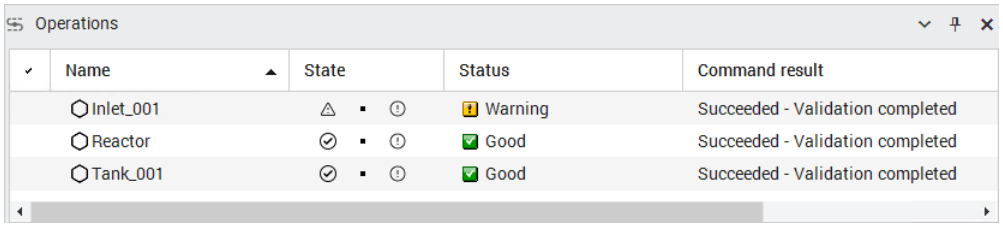
You can select more than one object for validation.

If an object is being edited, validation may not be performed. Also, if validation is in process on an object, other operations you start on the object will fail.

Note: You cannot cancel validation operations.

To manually validate one or more objects

1. In the **Template folder**, an **Application** view, or the **Find** dialog box, select the objects to validate.
2. On the **Home** ribbon, in the **Save** area, select **Validate**. The **Operations** view in the IDE opens and shows the status of the selected objects.



3. Continue using the IDE to perform other operations, if needed, while validation is going on, including work on other objects in the Galaxy. If you are validating a Galaxy, then you must leave the Galaxy alone until the validation process is complete.
4. When the validation process is complete, you see the results of the validation in the **Operations** view. If the validation failed on an object, you see a message. Correct the problem and validate again.

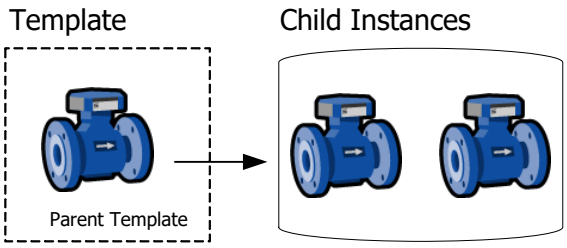
Note: You can also see the errors or warnings that led to an object having a status that is not Good by looking at the Error/Warnings tab within the object's Properties.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Create an instance

After you create templates, you can create instances. Creating instances makes a specific object from a template, with all the characteristics and attributes of the template.

After you have created an instance of an object, it can be deployed.



You can also customize an instance, if needed. For example, you can have a valve template. When you create an instance of that valve, you can specify the inputs and outputs for that specific valve on your factory floor.

To create an instance

1. Select the template you want to use for the instance. For example, to create a valve instance, select a valve template.
2. On the **Home** ribbon, in the **Create** area, select **Instance**. An instance is created.
3. Type the name for the instance. Instance names can be up to 32 alphanumeric characters. You cannot use $ as the first character. The name must include at least one letter. Instance names cannot include spaces.
4. To move the new instance in the **Deployment** view or **Model** view, drag the instance to the new location.

   You are ready to configure the instance, if needed. For more information, see Edit Objects.

Object names

Object names must be unique within each namespace, not within the Galaxy.

- Template names can be up to 32 alphanumeric characters, including the required $ as the first character. The second character cannot be $ and the name must include at least one letter. You cannot use spaces in an object name.
- Instance names can be up to 32 alphanumeric characters. You cannot use $ as the first character. The name must include at least one letter. You cannot use spaces.
- The object name extension "_New" cannot be used if another object in the Galaxy uses the same name without the "_New" extension. For example, if you have an object named "Pump," you cannot have a second object named "Pump_New." You could, however, use "Pump_Old" and "Pump_New."

  Note that this restriction applies to templates as well as instances. For example, you cannot have a derived template named "$Area_New" or "$ViewApp_New" since these add the "_New" extension to a base template name.

Note: You cannot use the following reserved names for objects: Me, MyContainer, MyArea, MyHost, MyPlatform, MyEngine and System.

An object can have three kinds of names if it is contained by another object. The three names include:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Name	Description
Tagname	The unique name of the individual object. For example, `Valve1`.
Contained name	The name of the object within the context of its container object. For example, the object whose Tagname is Valve1 may also be referred to as `Tank1.Outlet`, if `Tank1` contains it and it has the contained name "Outlet".
Hierarchical Name	Hierarchical names are fully-qualified names of contained objects that include the name of the objects that contain it. Because the object that contains it may also be contained, there are potentially multiple hierarchical names that refer to the same object. For example, if: "Reactor1" contains Tank1 (also known within Reactor1 by its contained name "SurgeTank"). "Tank1" contains Valve1 (also known within Tank1 by its contained name "Outlet"). Valve1 could be referred to as: "Valve1" "Tank1.Outlet" "Reactor1.SurgeTank.Outlet".

Delete an object

You can delete both templates and instances with the following exceptions. You cannot delete:

- Deployed instances
- Containers for other objects
- Objects checked out by other users
- Templates that have children (derived templates or instances)

Note: Make sure you correctly select the objects you want to delete. After you delete an object, you cannot undelete it. You must recreate it.

To delete an object from the Galaxy

1. In the **Template folder** or an **Application** view, select the object to delete. Select multiple objects by using **Shift+click** or **Ctrl+click**.
2. On the **Home** ribbon, in the **Edit** area, select **Delete**.
3. A message appears asking you to confirm that you want the object deleted. Select **Delete**.

Export objects

You can export some or all of your Galaxy objects. When you export, you are exporting the objects' associated templates, configuration states, and containment states. The information is saved in an .aaPKG file.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

After the Galaxy objects are exported, you can import them into the same or another Galaxy.

If your objects have scripts associated with them, you need to export the script library separately. For more information about exporting script libraries, see Export script function libraries. For more information about scripts and script libraries, see the *Application Server Scripting Guide*.

If you are exporting an InTouchViewApp object and want to include graphics that are not embedded in the application, such as graphics called from a script, you need to embed the specific desired graphics in the InTouchViewApp.

Note: You can associate all Galaxy graphics with an InTouchViewApp for backup, restore, and deployment operations, but not for import and export operations.

Before you start, make sure all objects you want to export are checked in. If an object selected for export is checked out, the checked in version of that object is exported instead. This can lead to old versions of objects being exported.

Exporting an entire Galaxy is different than backing up the database. Unlike the case with backups, change logs for the objects are not exported. When you export objects, only the related security information for the specific object is exported.

To export an object

1. In the **Template folder** or an **Application** view, select one or more objects to export.
2. On the ribbon, select **Galaxy**, then select **Export.**
3. Select **Objects**, then **Selected objects**. The **Export Automation Object(s)** dialog box appears.

   To export all of the objects in the Galaxy,select **All objects** instead of **Selected objects**.
4. In the **Export** dialog box, browse to a path and type a name for the exported file.
5. Click **Save**. The file is saved with the specified name and an .aaPKG extension.
6. When the export is complete, select **Close**.

   Now you can import the .aaPKG file into another existing Galaxy.

Protect objects on export

Protect graphics and derived templates by flagging the objects as protected in the galaxy database. Protected graphic and template behavior is similar to that of a base template. This option is available through a specialized form of the **Export Object(s)** operation.

System Integrators and other system designers can use this functionality to protect objects designed in a master galaxy that are intended for use in production galaxies or galaxies on runtime nodes.

About protecting objects on export

Protecting an object on export does not change the object in the galaxy from which it was exported. Protection is effective only on import of protected objects. Specific behavior is described as follows:

Element or Function	Description
Base Templates	Are protected by default and cannot be checked out, edited or renamed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Element or Function	Description
Protected templates and graphics	Cannot be checked out, edited, or renamed, but can be deleted. A template derived from a protected template is unprotected. Protected templates and graphics are marked with a lock icon.
Ancestor objects	Are protected in the exported .aaPKG file when a child object is protected.
Protection effective	Protection is effective on import of protected objects. A protected object retains its protected status even when exported using the standard **Export Object(s)** workflow.
Instances	Cannot be protected. Only templates and graphics can be protected.
Select both template and instance for export	The option to export objects as protected is disabled. Only templates and graphics can be protected.
Export workflow	The export workflow is the same as for all objects except that you can select the option to export **As Protected Object(s)**.
Graphics	Graphics and client controls directly or indirectly embedded in a protected graphic are also protected. Protected graphics can be opened in the Graphic Editor as read-only.

Protected objects are marked in the **Template folder, Visualization folder**, or Application views (**Model**, **Deployment**, or **Derivation View**) with a shield icon to the right of the object name..

Export objects as protected

The export objects workflow is the same as for all objects with the exception of specifying the export of selected objects as protected.

To protect objects on export

1. In the **Template folder** or an **Application** view, select one or more objects to export.
2. On the ribbon, select **Galaxy**, then select **Export.**
3. Select **Objects**, then **Selected as protected**. The **Export Automation Object(s)** dialog box appears.
4. In the **Export** dialog box, browse to a path and type a name for the exported file.
5. Click **Save**. The file is saved with the specified name and an .aaPKG extension.
6. When the export is complete, select **Close**.

   Now you can import the .aaPKG file into another existing Galaxy.

Exporting Objects with I/O Auto Assignment

Objects configured for I/O auto assignment can be exported the same way as other objects. See Configuring Objects for additional information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Application objects configured for I/O auto assignment are linked to DI objects and scan groups. Therefore, when you import these objects into a different Galaxy, the objects will look for DI object and scan group names that match the linkages that were made in their originating Galaxy. If a matching DI object and scan group are not found, all I/O auto assignment information is discarded, and the application objects are placed under the "Unassigned IO Device" folder in the **IO Devices** view.

- When exporting application objects, some I/O mapping assignments may be unavailable when importing the objects into a different Galaxy. Application objects that do not find a DI object and associated scan group that match their I/O references will lose their I/O mapping assignments and will have to be remapped.
- When exporting an application object with overrides in the I/O mapping table, overrides are preserved if a DI object and scan group that match its assignment from the original Galaxy are present in the new Galaxy.

  Note: The I/O mapping references are permanently deleted if matching DI objects and scan groups are not found. The I/O references will not reconstitute, even if you later add the scan groups and DI objects to which the application objects were formerly mapped.
- When exporting DI objects, there are no special considerations related to I/O assignments. I/O references for application objects will not change when the DI objects are imported into another Galaxy.

  Note: When importing a DI object into a different Galaxy, an existing DI object with the same name as the imported DI object may be overwritten. This can break device linkages to application objects previously configured within the Galaxy.

Export script function libraries

If you want to export objects that use scripts, the scripts are exported with the object.

Some scripts include functions that depend on external files called script function libraries. In this case, you must export the script function libraries separately.

To export a script function library

1. On the ribbon, select **Galaxy**, then select **Export.**
2. Select **Script**. The **Export script function library** dialog box appears.
3. If needed, browse to folder where you keep your script libraries. In the **Script function library** list, select the library or libraries you want to export.
4. Click **Export**. The selected script library is exported. Each script is named with the name of the script and an .aaSLIB file name extension.
5. When the export is complete, click **Close**. Now you can import the .aaSLIB file into another existing Galaxy.

Import objects

You can reuse objects from another Galaxy in your Galaxy. This saves you a lot of time if the objects are already set up in another Galaxy.

Importing instances previously exported from a Galaxy retains previous associations, when possible, such as assignment, containment, derivation, and area.

You can import objects from exported .aaPKG files or from an .aaPDF file. An .aaPDF file contains the configuration data and implementation code for one or more base templates. It is created by a developer using the Application Object Toolkit.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

You cannot have two objects with the same name or more than one copy of the same version of an object in the same Galaxy. When you import an object, you can choose how you want naming and version conflicts handled.

You should perform an "Upload Runtime Changes" before importing a new version base template if instances of the template are deployed. This saves changes made at Runtime to the Galaxy database.

Objects that were created in Application Server 2014 R2 or later cannot be imported into a Galaxy running an older version. For example, you cannot import an object created in Application Server 2017 into a Galaxy running Application Server 2012.

Note: The Application Server version is not the same as the Codebase version. Objects with newer Codebases can be imported.

You cannot directly import script libraries, Client Controls and OMI Apps compiled with .NET 8 Framework unless the node is upgraded first. We recommend using NET 4.8.1.

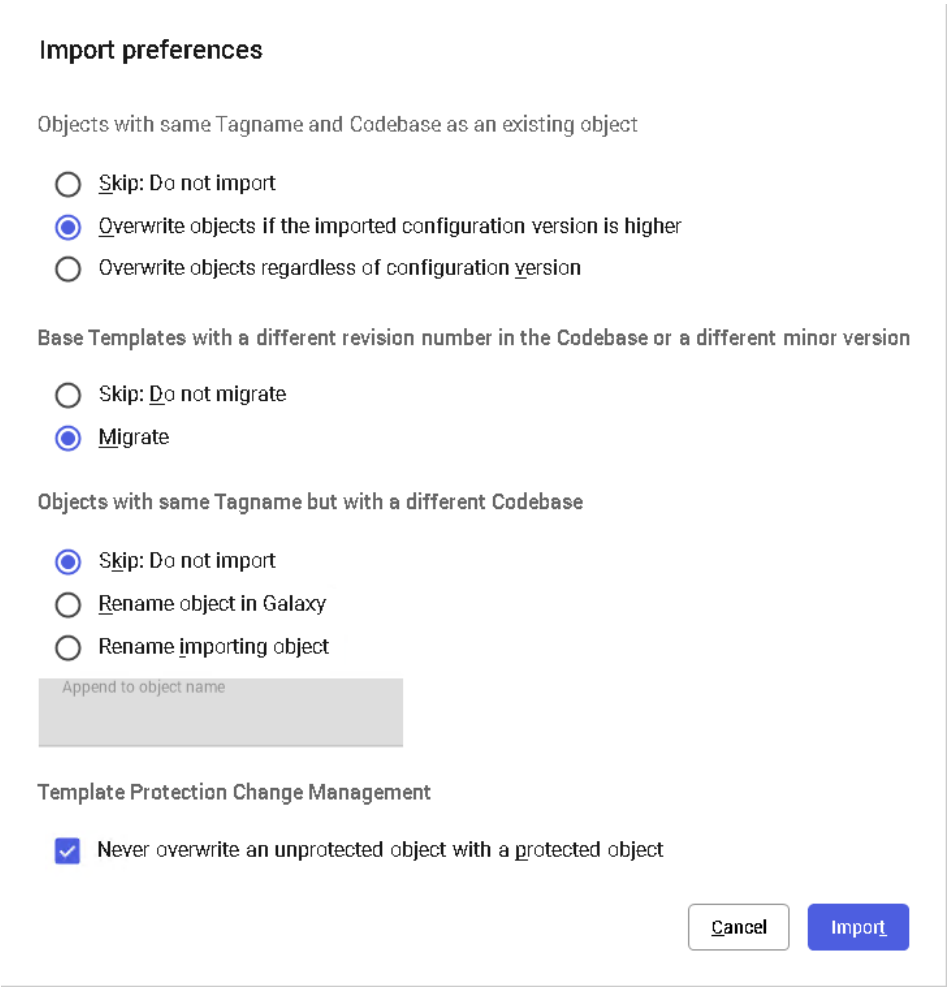
If you import the .NET 8 WPF/Winform controls, currently no runtime can utilize them even when you import them.

For the OMI web client, there is an internal workaround to import a .NET 8 Blazor component along with the original WPF 4.8 app, so the WPF will be used in the IDE/Layout Editor and deployed to the desktop, and the .NET 8 Blazor app will be deployed to the web client.

To import objects

1. On the ribbon, select **Galaxy**, then select click **Import.**
2. Select **Objects**, then select **Objects from package**. The **Importing objects** dialog box appears.
3. Browse for the file with either an .aaPKG or an .aaPDF extension. You can select more than one file. Click **Open**. The **Import Preferences** dialog box appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



4. In the **Objects with same Tagname and Codebase as an existing object** area**,** select one of the following:
   - **Skip: Do not import** leaves the existing object unchanged.
   - **Overwrite objects if the imported configuration version is higher** replaces the existing object with the object being imported if the imported object has a newer configuration. This is the default.
   - **Overwrite objects regardless of configuration version** replaces the existing object regardless of whether the existing object has an older configuration or the same configuration.
5. In the **Base Templates with a different revision number in the Codebase or a different minor version** area, select one of the following:
   - **Skip: Do not migrate -** objects with an older codebase are not migrated when a newer codebase exists in the Galaxy.
   - **Migrate -** objects with an older codebase are migrated when a newer the replacement object.

     For more information about migrating, see After you import.
6. In the **Objects with same Tagname but with a different Codebase** area, select one of the following:
   - **Skip: Do not import** leaves the existing object unchanged.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **Rename object in Galaxy** imports an object with a matching tagname but a different codebase from the existing one. The existing object is not overwritten but is renamed.
- **Rename importing object** imports an object with a matching tagname but a different codebase from the existing one. The existing object is not overwritten. The imported object is renamed.

7. In the **Template Protection Change Management** area, select the checkbox to prohibit overwriting an unprotected object with a protected object (default). Clear the checkbox to allow overwrites.
8. Click **Import**. The import process starts.
9. When the import process is complete, you can start using the objects you imported.

Import protected objects

The import objects procedure is the same for all objects, protected or unprotected. The following describes import preferences and results specific to importing protected objects:

Import Preferences	Result for Protected Objects
Objects with same Tagname and Codebase as an existing object: • **Skip: Do not import** • **Overwrite objects if the imported configuration version is higher (default)** • **Overwrite objects regardless of configuration version**	Overwrite preferences are governed by the **Template Protection Change Management** option, "Never overwrite an unprotected object with a protected object" (default). • Overwrites can occur if both the packaged and galaxy objects have the same protection status. • A protected object can overwrite an unprotected object only if specifically permitted by clearing the "Never overwrite ..." option. • When overwriting a protected parent object, protection is not cascaded to child objects. Unprotected child objects remain unprotected.
Base Templates with newer or older Codebases (or minor version updates): • **Skip: Do not migrate** • **Migrate**	Import and migration of base templates is unchanged by protected object import functionality.
Objects with same Tagname but with a different Codebase: • **Skip: Do not import** • **Rename object in Galaxy** • **Rename importing object**	Object renaming on import can occur only if the object to be renamed is not protected. • If the object in the Galaxy is protected, and the importing object is not, and **Rename importing object** is selected, the importing object is renamed to have the suffix "_new". • If the object in the Galaxy is unprotected, and the importing object is protected, and **Rename object in Galaxy** is selected, and the **Never overwrite an unprotected object with a protected object option** is not selected, the object in the Galaxy is renamed with the suffix "_old".

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Importing Objects with I/O Auto Assignment

Objects configured for I/O auto assignments can be imported the same way as other objects. See Import objects for additional information.

When importing objects that utilize I/O auto assignment, be aware of the following:

- If you are importing DI objects, there are no special considerations related to I/O auto assignment. I/O device mapping is not affected. An imported DI object will overwrite an existing DI object with the same name.
- If you are importing application objects, different outcomes can result, depending on the DI objects and scan groups that already exist in the Galaxy.
  - I/O device mapping is recreated, if possible, for each application object that is successfully imported and for which a DI object and scan group are found that match the existing I/O assignment. This includes all user-configured I/O attribute overrides.
  - Objects for which a DI object and scan group with matching names cannot be found are moved to the unassigned area. All I/O information for those objects is discarded. This includes default I/O auto assignment information as well as any custom configured I/O attribute overrides. You will need to assign these objects to a DI object and scan group in the new Galaxy and recreate all user-configured overrides. None of this information is preserved for application objects that do not find a matching DI object and scan group.

Import a SQLData object

The SQLData object is used to store and retrieve data from a SQL Server database. Typical applications for this object include basic recipe management and database capture of attribute values.

The SQLData object can be configured to use one of the following authentication modes:

- Windows Integrated Security
- Windows Account
- SQL Server Authentication

When you import an object from another galaxy that is configured with SQL Server Authentication, the encrypted password does not follow the object, and the object will be in a warning state in the new galaxy. You must open the object in the Object Editor and enter a new password to remove the warning state.

Import script function libraries

You can enhance an object's functionality by attaching a script to it. Some scripts include functions that depend on external files called script function libraries. Scripts are included in the object import operation, but you must import the script function libraries separately.

Note: You cannot directly import script libraries, Client Controls and OMI Apps compiled with .NET 8 Framework unless the node is upgraded first. If you attempt to import these elements, the system will say that the task was successful when in fact it was not. The solution is to upgrade the node. We recommend using NET 4.8.1.

If you import the .NET 8 WPF/Winform controls, currently no runtime can utilize them even when you import them.

For the OMI web client, there is an internal workaround to import a .NET 8 Blazor component along with the original WPF 4.8 app, so the WPF will be used in the IDE/Layout Editor and deployed to the desktop, and the .NET 8 Blazor app will be deployed to the web client.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To import a script function library

1. From the ribbon, select **Galaxy**, then **Import**, then **Scripts**.
2. Browse to the directory that contains the script function library you want to import.
3. Select the script function library to be imported, and then select **Open** to start importing the selected library. Acceptable file types and file name extensions for script function libraries are:
   - Script Library Files (.aaSLIB file extension)
   - .NET or .COM files (.dll, .tlb, .olb, or .exe file extensions)
   - InTouch Script Extension Files (.wdf file extensions)

   An information window opens when the import successfully completes.
4. Validate any graphics that use the custom script functions contained in the imported library. See Validate objects for additional information.
5. Redeploy any InTouchViewApp instances containing graphics that use the script functions in the imported library.

Note: When you import a new version of an existing script library into a Galaxy, you must stop and restart the engine that hosts the script's owning object. The new script version does not replace the old version until after the hosting engine has restarted and you have redeployed the object.

If you import an object whose script references a script function library that is not resident in the Galaxy, the imported object is set to Bad state and cannot be deployed. To correct this, import the script function library and validate the object. For more information about scripts, see the *Application Server Scripting Guide*. For more information about validating scripts, see Validate objects.

If you import a script function library that is a different version than the current library, a message is displayed to notify you that there are dependent objects. The message indicates how many objects will need to be redeployed if you continue with the import. If you continue with the import, the dependent objects are marked for redeployment.

Script function libraries that are COM libraries developed using Visual Studio 6 or earlier are not automatically deployed. To place this COM library on the target platform, you can either:

- Install it directly on the target platform and register it.
- Import it to the Galaxy, and then export it as an aaSLIB. Modify the aaSLIB xml to designate that the library is to be registered as a COM object. Reimport the aaSLIB so that it is automatically deployed and registered.

Import client controls

If you import a client control that is a different version than the current object, a message is displayed to notify you that there are dependent objects. The message indicates how many objects will need to be redeployed if you continue with the import. If you continue with the import, the dependent objects are marked for redeployment.

When the import process is complete, restart the IDE to apply the changes made by the import process. After the IDE has restarted, you can start using the objects you imported.

Import an app from the AVEVA digital exchange

A number of pre-built AVEVA OMI apps are available for download from the AVEVA Digital Exchange.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To access the Digital Exchange

1. From the ribbon, select **Galaxy**, then **Import.**
2. Select **Objects**, then **Explore product hub**.

   Your browser will open and connect to the AVEVA Digital Exchange.
3. Log in, or sign up for new account.
4. Click on the link for the **System Platform Developers Community** and browse for apps. Select the app you want and follow the online instructions.

Import AVEVA apps and client controls

You can import AVEVA apps developed with Windows Presentation Foundation (WPF), or Winforms/.NET controls through the IDE. Once you import the app or control, you can configure its properties and deploy it in runtime.

An AVEVA App is collection of one or more controls primarily developed with Windows Presentation Foundation (WPF) for use in AVEVA OMI ViewApps. Other technologies such as Windows Forms (WinForms) and HTML5, can also be used. You can create your own AVEVA Apps, which can be imported via WPF interoperability. For more information about creating your own AVEVA Apps, see the AVEVA OMI SDK Help. The AVEVA OMI SDK is installed automatically when you install the IDE.

You can enhance a Galaxy's functionality by adding an embedded control from an AVEVA App to a pane in a layout for inclusion in a ViewApp.

A control developed with Winform or .NET can be imported in much the same way, and like an AVEVA app, you can configure the control within a layout for inclusion in a ViewApp.

In order to qualify for import as an AVEVA App, the App should contain at least one WPF FrameworkElement Class control (System.Windows.FrameworkElement).

AVEVA Apps should be organized in a single folder which can optionally include subfolders. These subfolders typically contain locale files.

To import an AVEVA App

1. Assemble the App, together with all required dependencies and related files, into a folder.
2. From the ribbon, select **Galaxy**, then **Import.**
3. Select **Visualization**, then **OMI Apps**.
4. Select the folder that contains the App to be imported.

   The contained App, along with all dependent files, including dependent files contained in subfolders, are imported when you select **OK**.

   If the App does not import successfully, see Troubleshoot AVEVA or WPF apps that fail to import.

After the import process completes successfully, a new object appears in the Visualization folder named DisplayModule_00*n*, where *n* is 1 or the next available integer. The new object can be renamed, deleted, or exported as an .aaPKG file for use in another Galaxy. See Export objects for additional information.

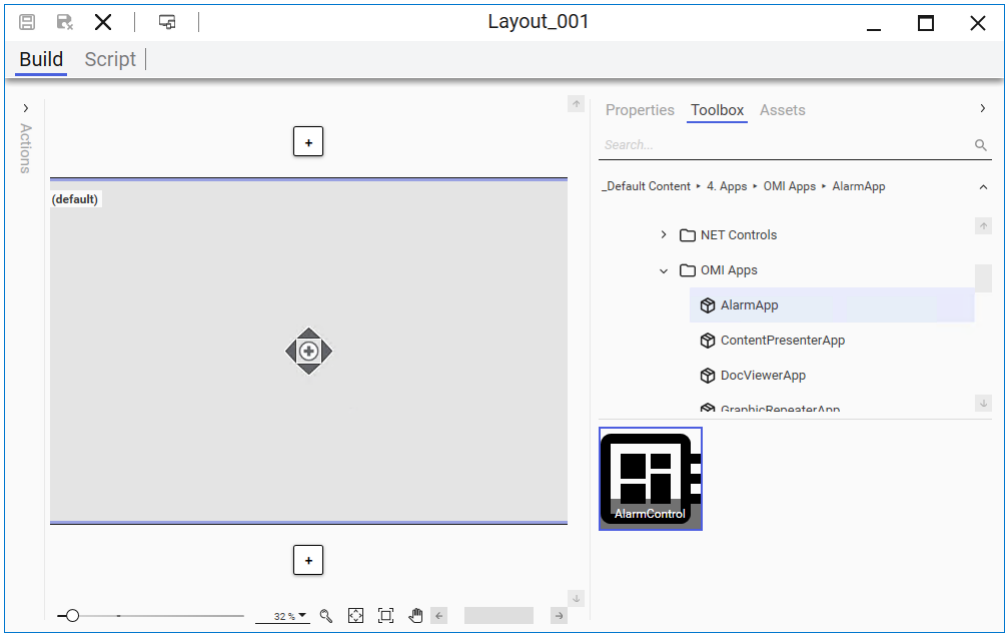
Add a control from an AVEVA app to a ViewApp

Controls from the imported App appear in the **Toolbox** tab of the **Layout** or **ViewApp** editors.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To assign a control from an App to a ViewApp

1. Open a layout in the **Layout** Editor and select the **Build** tab at the left of the editor.
2. Select the **Toolbox** tab on the right side of the editor and select a folder that contains the AVEVA Apps you want to add.
3. Select the AVEVA App to be added. The controls available in the App are shown.
4. Select a control and drag it onto a pane.
5. To view or override the control's public properties, select the **Properties** grid. Enter overrides as needed.
6. Save the layout. You can then use the layout that contains the embedded control in a ViewApp.



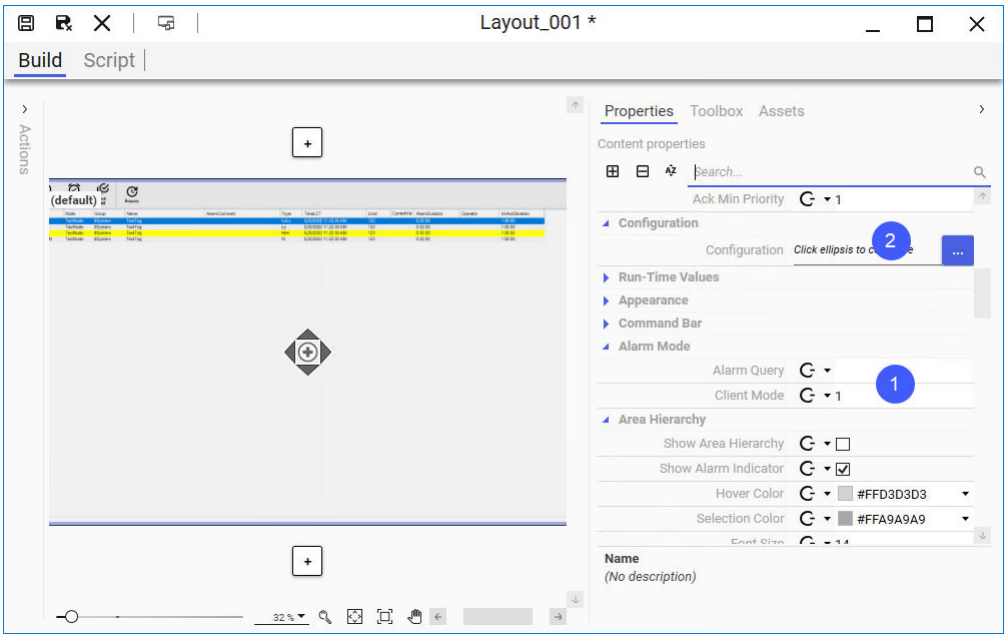
Configurable properties in AVEVA apps

AVEVA apps can contain two types of configurable properties:

1 — Dependency properties

2 — CLR properties (.NET framework properties)

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Configurable Property Data Types

- System.Boolean
- System.Byte
- System.Char
- System.Decimal
- System.Double
- System.Int16
- System.Int32
- System.Int64
- System.SByte
- System.Single
- System.String
- System.UInt16
- System.UInt32
- System.UInt64
- System.DateTime
- System.Drawing.Color
- System.Windows.Media.Color
- System.Windows.Media.Brush
- Any property that can be converted to or from string

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

CLR properties can only be set at configuration (design) time, while dependency properties can be bound to any attribute and can be set at runtime. If a dependency property is changed at runtime, the new value is propagated to the attribute. Dependency properties are data-bindable at design time, and you can set the binding direction to in (read), out (write), or both. To change the binding direction, click on the arrow to toggle the binding direction.

For dependency properties, you must specify the attribute to which you are binding the property, as well as the direction of the binding (in (read only), out (write only) or both. The attribute naming convention is the same you use, for example, when configuring an animation in the Graphic Editor. Note that the ViewApp Editor does not provide syntax validation for attribute names, but the Logger will list any configuration errors.

Binding is only supported for the following basic data types.

DataBindable Properties

- System.Boolean
- System.Byte
- System.Char
- System.Decimal
- System.Double
- System.Int16
- System.Int32
- System.Int64
- System.SByte
- System.Single
- System.String
- System.UInt16
- System.UInt32
- System.UInt64

Any data type can be converted to and from the System.String type property.

AVEVA app development guidelines

The following types of properties are available for configuring WPF-based displays:

- Public writable CLR properties on the display that are not hidden using the browsable attribute
- Dependency Properties
- Nested Properties which satisfy the criteria of configurable properties
- Enum and Flag properties
- Any property that can be converted to or from string

The following types of properties are NOT available for configuring WPF-displays:

- Any property defined by the System.Windows.UIElement class
- Any property defined by the System.Windows.FrameworkElement class

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

    - The Tooltip property is an exception to this and can be configured
- The following properties defined by the System.Windows.Controls.Control class
    - IsTabStop
    - TabIndex
    - Template
- Any property defined by the System.Windows.Controls.ContentControl class

Optional Filter File: AppManifest.xml

You can include an optional file called **AppManifest.xml** in your imported AVEVA App. See the *AVEVA OMI SDK Help*, located in the **AVEVA Documentation** folder, for additional information about AppManifest.xml. This file will allow you to filter properties when you are configuring the app. The basic file structure is:

- DLL name
    - Control name
        - Property

The following is a sample AppManifest.xml file, from the Hamburger App. The files specifies controls and properties that are exposed to System Platform IDE users as they configure a layout or ViewApp.

```
<?xml version="1.0"?>
<AppManifest AppVersion="1.0"?>
   <Filters>
      <!--List all the Controls that need to be exposed to the Systm Platform IDE user to
      utilize them in a View Application.
      Only the controls listed here will be available for the IDE users to place them on
      panes within a Layout or a ViewApp.-->
      <Control AssemblyName="AVEVA.Apps.HamburgerApp"
      ControlFullName="AVEVA.Apps.HamburgerApp.HamburgerButton">
         <!--List the public properties on this control that need to be exposed when user
         is configuring this control in a Layout or
         a ViewApp. The Property Editor within the Layout Editor and ViewApp Editor
         includes these properties when this control is
         selected and allows the user configure new default values.-->
         <Property Name="HamburgerColor" />
         <Property Name="SlideInPanePosition" />
         <Property Name="Background" />
      </Control>
   </Filters>
</AppManifest>
```

AVEVA app limitations and restrictions

- Events and scripting are not supported
- Only basic datatype properties configuration
- Apps cannot be imported from a network share location
- Collections and arrays are not supported
- Upgrading Apps is not supported
- Nested properties can be viewed and configured, but they cannot be configured with data binding

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- To host the .NET controls, use the WPF element **WindowsFormHost** to wrap the control into WPF.

```
<UserControl x:Class="AVEVA.Visualization.TrendApp.TrendControl"
      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
      xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
      mc:Ignorable="d"
      d:DesignHeight="300" d:DesignWidth="300">
   <Grid>
      <WindowsFormsHost x:Name="FormsHost" />
   </Grid>
</UserControl>
```

Troubleshoot AVEVA or WPF apps that fail to import

There are two major factors that may stop an AVEVA or WPF app from successfully importing.

Expose controls and properties: Make sure that you have included an AppManifest.xml file with the app, and that all controls and properties that are needed in your app are explicitly exposed in it. For additional information about the AppManifest.xml file, see AVEVA app development guidelines and the *AVEVA OMI SDK Online Help*.

Block business logic from executing during design time: If you have business logic executing in the control constructor, or in static properties or static constructors, the import operation will not be successful. The control constructor is invoked when the control is instantiated. Therefore, do not include business logic within the control or static constructors, or in static properties.

- If you cannot remove business logic from a control or static constructor, or a static property, make sure the property is not invoked or that the property does not execute while in design mode.
- To block execution of logic within a constructor while in design time, add the following line to the constructor:

```
   !DesignerProperties.GetsInDesignMode
```

After you import

Imported templates are listed in the proper folder in the Template folder as defined in the object. Imported instances are shown in the Application views.

The following post-import rules apply:

- If a folder does not exist, it is created.
- If the object belongs to a security group that does not exist, it is associated with the Default security group.
- If the object belongs to an area that does not exist, it is associated with the Unassigned Area.
- If the host to which the object is assigned does not exist, it is assigned to the Unassigned Host.
- If you selected **Migrate** from the **Import Preferences** dialog box, the migrated objects are marked with "software upgrade required" if they are deployed. These objects will be upgraded when the objects are redeployed.
- If you import a new version of an existing instance, the new version is marked as requiring deployment if the existing object is already deployed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configuring Objects

Enhance and extend derived templates and instances by adding and configuring features, attributes, scripts, and graphics. These elements can be associated with an Object Wizard in a derived template to simplify the task of creating instances.

The **Object Editor** includes a separate page for adding scripts. All other elements (graphics, attributes, Object Wizards, and external content) are added and configured from the Object Editor's main page. For an overview of the **Attributes** page, see About the Attributes Page.

Add Attributes to an Object

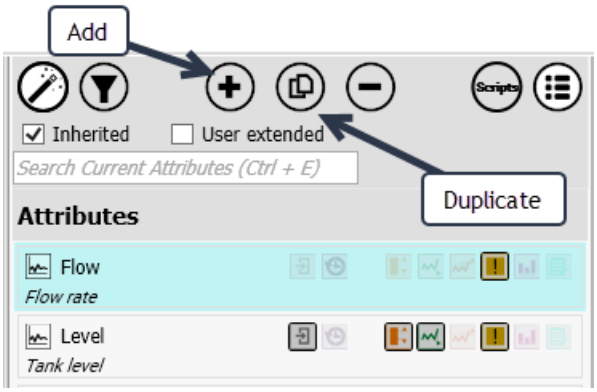
There are two ways to add attributes to an object. These are:

1. Select the **Add Attribute** button. The first attribute you add to an object has the default name "Attribute001" and the default data type "Boolean.

   Subsequent attributes added with the **Add Attribute** button use the previous attribute's name and increments the number at the end of the name, or appends "1" at the end of the name if the previous name did not end in a number. It also retains the previous attribute's description, Data type, and Writeability.

2. To duplicate an existing attribute, select the **Duplicate Attribute** button. Selecting **Duplicate Attribute** retains any features that were configured for the selected attribute, in addition to name, description, Data type and Writeability.

Note: When you add an attribute to a derived template, the attribute, its data type, and writeability are automatically locked in derived objects.



If attribute parameters such as initial values and security classifications are locked in the template, they cannot be changed in child instances. If these parameters are unlocked in the template, the initial value and security are editable and lockable in derived templates. When unlocked in either the base or derived template, the value is editable in instances.

After you add an attribute to an instance, it appears in the **Attribute Browser** list for use with the scripting and attribute configuration functions. For more information about using the **Attribute Browser**, see Reference Objects Using the Galaxy Browser.

In the **Attributes** page, you can define the following initial information and parameters for the attribute:

- Add a new attribute to an object.
- Name the attribute and provide a description.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Configure its data type.

  For a Boolean data type, you can specify different text strings for the **'False' label** and **'True' label**. For example, if a Boolean attribute is associated with the status of a motor, you can specify the states as "**Stopped**" and "**Running**". Text boxes appear for you to enter these strings when you select a Boolean data type.

  These labels are also shown in the **Value** and **Limit** columns of the Alarm and Event database and InTouch AlarmView control.

- Specify the attribute writeability.
- Set initial values if the attribute is user writeable.
- Enable and set locks and security on the new attribute.
- Set whether the new attribute is an array and how many elements are in the array.

You can then add Features to the attribute. For more information, see Adding Features to Attributes.

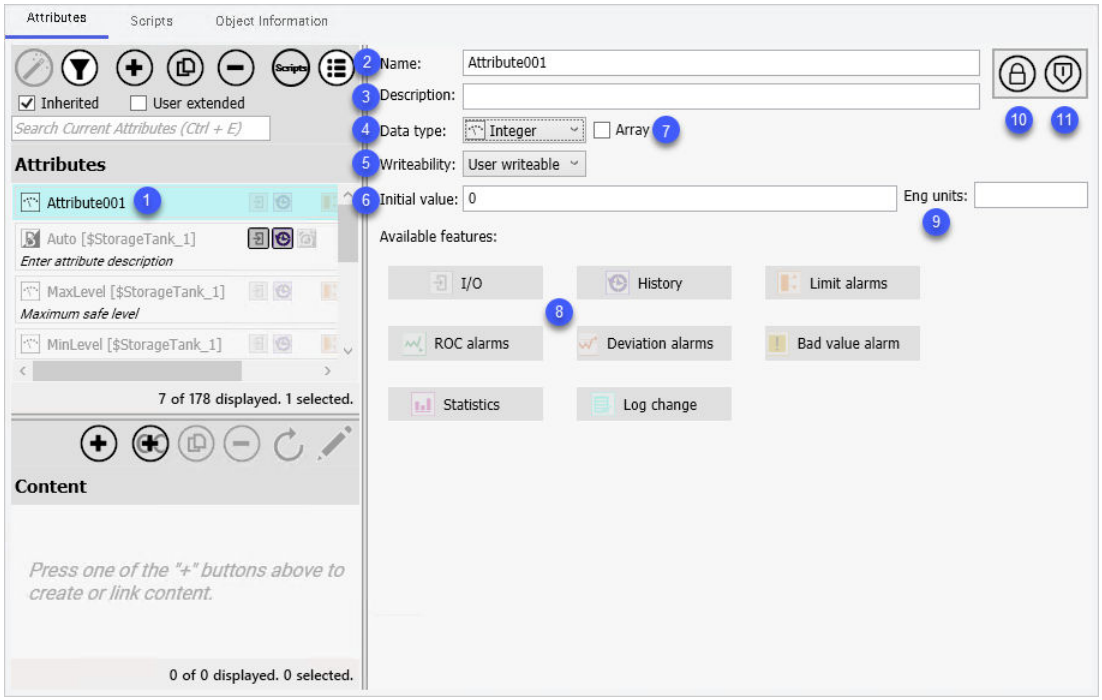
Configure an attribute

To configure attributes

1. Highlight the attribute you want to configure.
2. Edit the **Name** of the attribute.
3. Enter a **Description**.
4. Select a **Data type.** See Data type for additional information.
   - Boolean: Enter False and True labels (if not False and True).
   - Integer: Enter Initial value and Engineering units.
   - Float: Enter Initial value and Engineering units.
   - Double: Enter Initial value and Engineering units.
   - String: Enter Initial value.
   - Time: Enter Initial value.
   - ElapsedTime: Enter Initial value.
   - InternationalizedString: Enter Initial value.
   - BigString: Enter Initial value.
5. Select Writeability type:
   - Calculated
   - Calculated retentive
   - Object writeable
   - User writeable
6. Enter the Initial value for the attribute.
7. Select the checkbox if the attribute is part of an array.
8. Enter the Engineering unit type (numeric data types only).

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

9. Select Features and configure them. See Adding Features to Attributes for additional information.
10. Lock the Initial value, if you do not want users to be able to modify the value in derived objects. You can show or hide the lock icon by toggling the **Show/Hide Lock icons** button. See Lock and Unlock Template Attributes for additional information.
11. Configure the attribute's security state. You can show or hide the security icon by toggling the **Show/Hide Security icons** button. See Set Object Security for additional information.

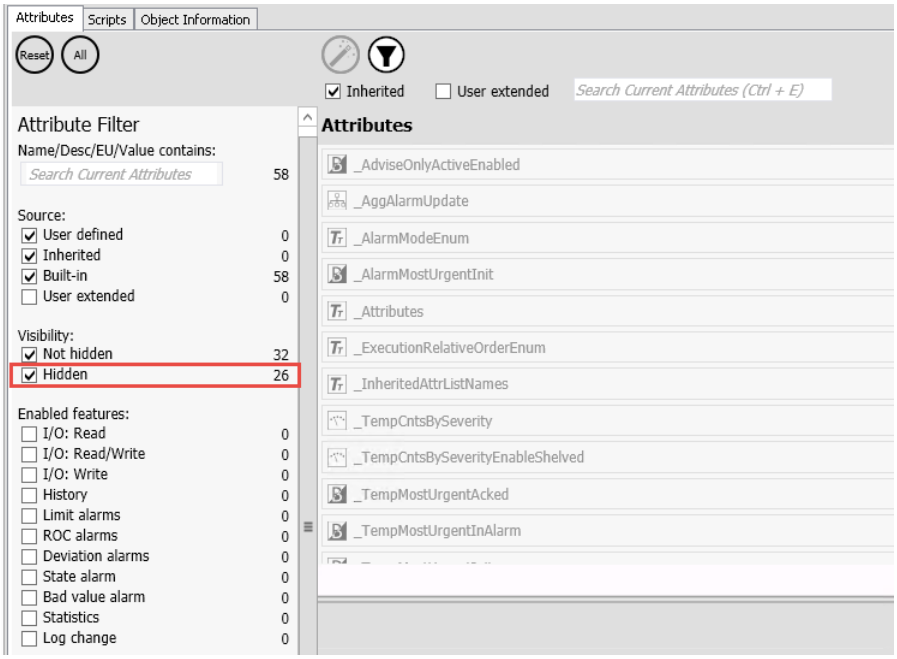


Attribute Naming Conventions

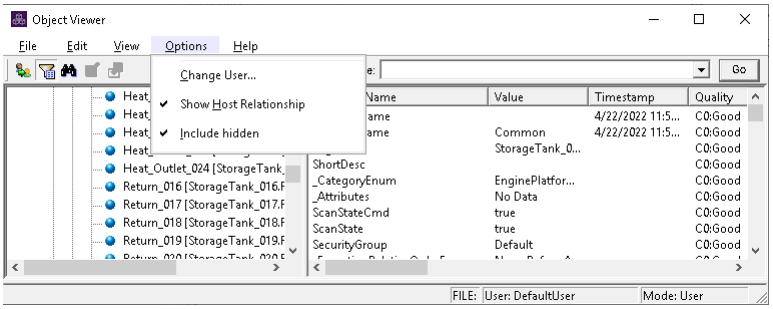
Attribute names can be from one to 329 characters long. Valid characters are alphanumeric characters and the following special characters: . (period), _ (underscore), $, and #. The maximum name length includes extensions to an attribute that result from the addition of features, such as historization, alarms, and I/O that can be associated with an attribute. These additions are categorized as "User extended" attributes. See Adding Features to Attributes for more information.

- An attribute name that starts with an underscore (_) as the first character of the name is a hidden attribute. By default, hidden attributes are not shown in the **Attribute Editor**, the **Properties>Attributes** dialog box, or the **Object Viewer**.
  - To view hidden attributes in the **Attribute Editor**, select the **Hidden** filter option.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



- To view hidden attributes in the **Properties** dialog box, select **Include Hidden**.
- To view hidden attributes in the **Object Viewer**, select **Include Hidden**.



Note: We recommend that you do not use hidden attributes in scripts.

- Using the word "quality" as an attribute name is not supported. "Quality" is used by InTouch HMI in a set of dotfields to show the reliability of data values assigned to an I/O tag. An attribute named "Quality" cannot be accessed through FS Gateway or InTouch due to a naming conflict.

Important: After creating an attribute, it is available for adding Features or additional attributes. If you extend an attribute you have added to an object or another attribute, and then delete or rename that attribute, all associated Features and additional attributes added to the object are lost.

To create and associate a attribute with an object

1. On the **Attributes** page of the Object Editor, click the **Add** (**+**) button. An attribute is added to the **Attributes** list.
2. Type the new attribute name and add an optional description.
3. In the **Data type** list, select the **Data type** for the new attribute. The available features and basic options change depending on your selection in the **Data type** list.
4. Set the remaining parameters as needed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: For detailed information about each item on the **Attributes** page, see About the Attributes Page.

5. Lock the values, if needed. The lock is available only when you are working with a template. If you are working with an instance, it shows the lock status for the value in the parent object.
6. Set any security you need. For more information about setting security, see Set Object Security.
7. Save and close the Object Editor when you are done.

Attributes and scripting

Follow these guidelines and best practices when using attributes in scripts:

- If you use **Calculated** and **Calculated retentive** attributes as counters, they must be manually initialized. For example, if you use `me.Attribute=me.Attribute+1` as a counter in a script, you must also initialize the attribute with something like `me.Attribute=1` or `me.Attribute=<some attribute value>`.
- **Calculated** attributes can be initialized in scripts with **Execution type** triggers of On Scan and Execute, but not initialized in Startup scripts.
- If you use an attribute **array** in a **startup script**, value and quality may become bad when the array is initialized in the startup script. To prevent this, set the dimension after initializing the array as shown below. This will ensure that the value is set and the attribute quality is good.

```
Me.ArrayAttribute[]="0,0,0,0,0,0,0,0,0,0";
Me.ArrayAttribute.Dimension1 = 10;
```

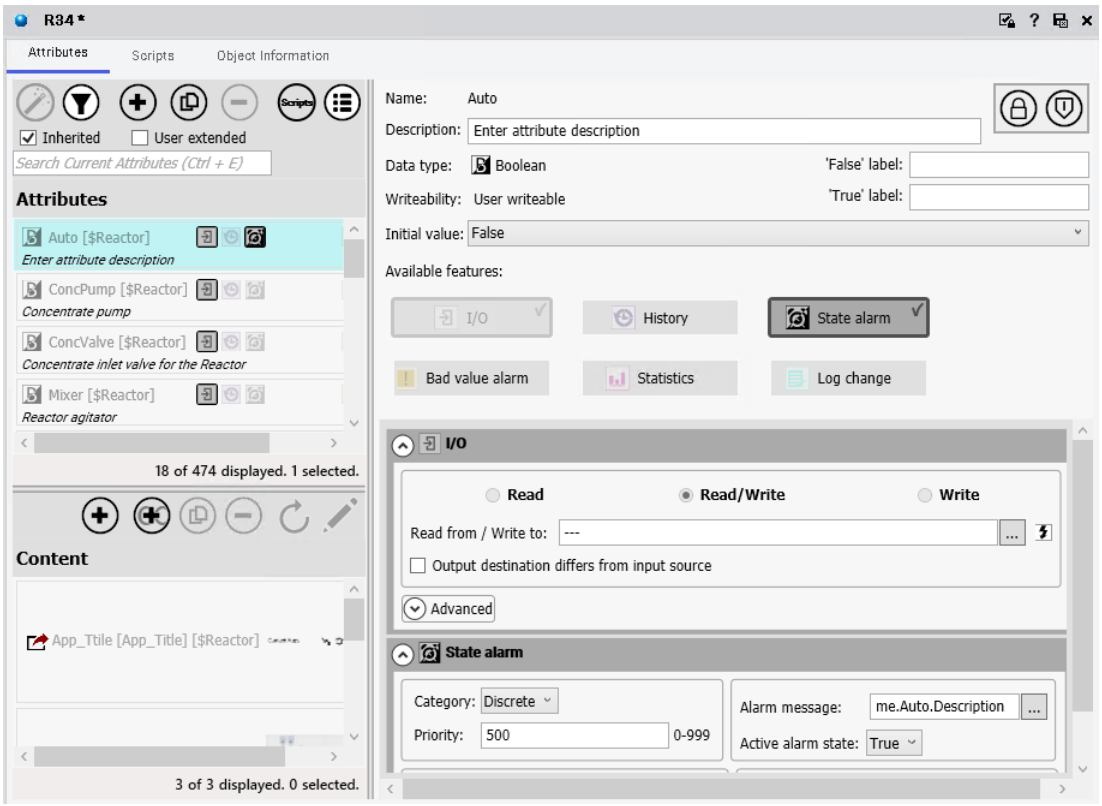
- You must initialize **Calculated retentive** attributes in Startup scripts and you can initialize these attributes in On Scan and Execute scripts. A **Calculated retentive** attribute retains the attribute's current value after a computer restart, redundancy-related failover, or similar situation in which valid checkpoint data is present.

  Your Startup script should contain a statement testing the Boolean value of the Engine.StartingFromCheckpoint attribute on the object's AppEngine. If the value is TRUE, do not initialize the attribute. If the value is FALSE, initialize the attribute. For more information about Engine.StartingFromCheckpoint, see the help for the AppEngine object, available from the AppEngine object editor.
- We recommend that you **do not use hidden attributes** (attribute names that begin with an underscore) in scripts.

Adding Features to Attributes

The **Attributes** page allows you to configure an existing attribute for I/O, alarms, and history functionality not embedded in the original object.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



About Features Inheritance

You can add Features to attributes that are in either derived templates or in instances. You cannot add Features to attributes in Base templates. The following parent-child object characteristics also apply to Features added to objects:

- If you add a Feature to an attribute in a derived template that has objects derived from it, all child objects inherit the Feature.
- You cannot add a Feature to attributes on derived objects that duplicate parent object Features in name and type.
- You cannot add a Feature with the same name as an existing Feature.
- Renaming a Feature on an attribute in the template to which it was originally added renames all other objects derived from the template. This change happens when the template is checked in.
- You can check in a template with an attribute configured with a new Feature with the same name as an existing Feature on an attribute in a derived object. The template definition of the Feature overrides the Feature in the derived object.
- If you remove a Feature on an attribute from a template, that Feature is removed from any child object. You see the change when you check in the template.

To create and associate a Feature with an object

1. On the **Attributes** page, select an attribute from the **Attributes List**. The available Features change to the Features allowed for the attribute.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. Click the button for the Feature you want to apply to the selected attribute. The associated parameters for each kind of Feature become available. For detailed information about each item on the **Attributes** page, see About the Attributes Page.
3. Select the parameters for the Feature you have added. Available Features, which vary depending on the attribute's Data Type and Writeability, are the following:
   - **I/O**: For information about adding the I/O Feature, see Using the I/O Feature.
   - **History**: For information about adding the History Feature, see Using the History Feature.
   - **Limit alarms**: For information about adding a Limit alarm Feature, see Using the Limit Alarms Feature.
   - **ROC alarms**: For information about adding a Rate of Change (ROC) alarm Feature, see Using the ROC Alarms Feature.
   - **Deviation alarms**: For information about adding a Deviation alarm Feature, see Using the Deviation Alarms Feature.
   - **State alarm**: For information about adding a State alarm, see Using the State Alarm Feature.
   - **Bad Value alarm**: For information about adding a Bad Value alarm, see Using the Bad Value Alarms Feature.
   - **Statistics**: For information about adding a Statistics Feature, see Using the Statistics Feature.
   - **Log change**: For information about adding a Log Change Feature, see Using the Log Change Feature.
4. Lock the values, if needed. The lock symbol is available only when you are working with a template. If you are working with an instance, it shows the lock condition of the value in the parent object.
5. Set any security for the attribute. For more information about setting security, see Set Object Security.
6. Save and close the Object Editor to include the new Features in the configured object.

Using the I/O Feature

The I/O Feature allows you to configure all aspects of data input and output for an attribute.

You can configure I/O type and you can specify input sources and output destinations. The I/O types you can specify are:

- Read (Input): See Configure I/O as read-only.
- Read/Write (InputOutput): See Configure I/O as Read/Write.
- Write (Output): See Configure I/O as Write-only.

You can also configure advanced properties, described in the following table. The attribute's data type and I/O type determine what Advanced I/O properties are available.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

I/O Feature Advanced Property	Description
Buffered	Enable buffered data to propagate to data subscribers the entire subset of values accumulated within a single scan cycle. Buffering data ensures that if a given attribute changes its value several times during a single scan cycle, there is no folding of data, which occurs when an accumulation of values of multiple data changes within a single scan are overwritten and only the latest value is stored. **Note:** If Galaxy-wide buffering is enabled, you cannot enable (or disable) buffering for individual attributes. With Galaxy-wide buffering, all attributes are buffered. See Enable Galaxy-wide Buffering for additional information.
Deadband	Specify the minimum amount by which a value must vary in order for the attribute to register a change, for example, by triggering an alarm, historizing an alarm or event, or triggering a script. The Deadband property is not available for string data types.
Reflect input to output	Enable to propagate an input value to an output destination. Enabling automatically disables the **Output destination differs from input source** option. Typically, set this option when you want to read an input from one source, manipulate its value in a script, and send the manipulated value to a different destination address, all during a single scan of an object. For more information, see Using Read/Write I/O in Scripts. Available only when the I/O type is Read/Write. You must set separate **Read from** and **Write to** properties.
Output every scan	Available only when **Reflect input to output** is selected. Write to the specified output destination occurs even when there has been no state or data change since the previous scan. The timestamp is not updated if there has been no state or data change since the previous scan.
Enable I/O scaling	Enables scaling between the raw value and the Engineering Units (EU) value. Scaling is the process of taking raw data from a device and presenting it as an appropriate value for your application. Application Server supports two types of data scaling: linear and square root, described in this table. Available only for integer, float, and double data types.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

I/O Feature Advanced Property	Description
Maximum	Available only when I/O scaling is enabled. Raw: The raw input maximum to be used in the scaling equation. EU value: The maximum value in engineering units to be used in the scaling equation. Extended EU range: The highest value allowed for the attribute before it is clamped, if clamping is enabled, or set to NaN. This value must be greater than or equal to the specified maximum EU value.
Minimum	Available only when I/O scaling is enabled. Raw: The raw input minimum to be used in the scaling equation. EU value: The minimum value in engineering units to be used in the scaling equation. Extended EU range: The lowest value allowed for the attribute before it is clamped, if clamping is enabled, or set to NaN. This value must be less than or equal to the specified minimum EU value.
Conversion mode	Available only when I/O scaling is enabled. Select Linear or Square Root conversion mode from the list. Linear: Typically converts directly from one value scale and type (raw) to another value scale and type (EU). *ScaledValue = ((RawValue-RawMin)/(RawMax-RawMin) * (EngUnitsMax-EngUnitsMin)) + EngUnitsMin* Square Root: Provides more precise scaling, typically used when the raw value is too large to be usefully scaled to an EU. *ScaledValue = sqrt((RawValue-RawMin)/(RawMax-RawMin))* *If RawValue < RawMin, then ScaledValue = (sqrt(RawMin) * (EngUnitsMax-EngUnitsMin)) + EngUnitsMin*
Clamp input to EU range	Available only when I/O scaling is enabled. If enabled, the scaling calculation result is clamped at either the maximum EU range value or the minimum EU range value, and the attribute quality is set to Uncertain. If not enabled, and the attribute value exceeds the EU range, then the value will continue to scale out of range, and the quality is set to Bad.

Configure I/O as read-only

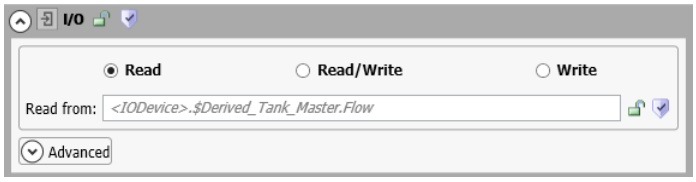
Select **Read** in the I/O parameters area. Define an input **source** by using I/O auto-assignment, by typing in the reference string, or by clicking the **Attribute Browser** button at the right.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Use I/O auto-assignment to prepare the input source for automatic assignment to a configured Device Integration (DI) object or other data source. For information about using I/O auto-assignment, see Using I/O Auto Assignment.

  Important: If the InputSource attribute is locked in the parent template, the attribute cannot be updated with the resolved reference when the object is deployed, and the run-time value will be "---Auto---".

- Use the **Attribute Browser** to select an attribute and automatically insert the correct reference string for that attribute. For more information about using the Attribute Browser, see Reference Objects Using the Galaxy Browser.
- **Telemetry Server:** See Configure I/O for Use with AVEVA Telemetry Server Communication Drivers for additional information.



You can add multiple Read (input) I/O Features to an object. However, you cannot add a Read/Write (InputOutput) I/O Feature to an attribute that already has either a Read or a Write (output) I/O Feature. Arrays are not supported.

Note: Lockable attributes can be configured with a Read I/O Feature, but they only function correctly during run time if the configured attribute is unlocked.

If the data types of the attribute and its I/O source attributes are the same, they are set to equal values according to the object's execution rate. If the two attributes are different data types, coercion rules are applied.

If coercion fails or the input value is out of the attribute's range, quality for the configured attribute is set to Bad. Otherwise, the configured attribute quality matches the **source** attribute. When the object is Off Scan, quality is always Bad and user sets are accepted.

Attributes configured with a Read I/O Feature are not protected by their security classification. The only enforced security specifies if an IDE user can edit or add features to the object. A Read I/O Feature can be added to a template or instance. If added to a template, the existence of the Read I/O Feature is automatically locked in derived objects.

Configure I/O as Read/Write

Select **Read/Write** in the I/O parameters area. Define an input **source** by using I/O auto-assignment, by typing in the reference string, or by clicking the **Attribute Browser** button at the right.

- Use I/O auto-assignment to prepare the input source for automatic assignment to a configured Device Integration object or other data source. For information about using I/O auto-assignment, see Using I/O Auto Assignment.

  If InputSource or OutputDest attributes, or both, are locked in the parent template, the attributes cannot be updated with the resolved reference when the object is deployed, and the run-time value will be "---Auto---".

- Use the **Attribute Browser** to select an attribute and automatically insert the correct reference string for that attribute. For more information about using the Attribute Browser, see Reference Objects Using the Galaxy Browser.

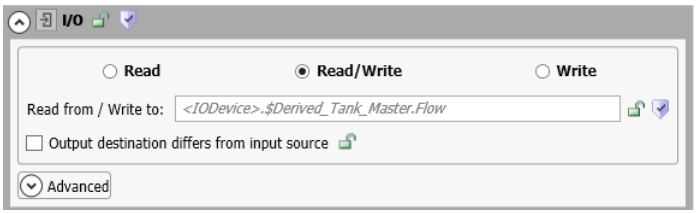
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- If you are configuring an I/O attribute for use with the Telemetry Server, see Configure I/O for Use with AVEVA Telemetry Server Communication Drivers for additional information.

If the output destination and the input source are not the same, click **Output destination differs from input source**. Enter a **Destination** attribute by using I/O auto-assignment, by typing in the reference string, or by using the **Attribute Browser** to search for the reference string in an object.

Important: If you clear the **Output destination differs from input source** check box, the **Write to** text box automatically shows "---". In the run-time environment, "---" is the same reference as the **Read from** value entered during configuration time. During run time, you can change the **Read from** reference. During configuration, do not lock the **Write to** parameter if you clear the **Output destination differs from input source** check box.

A Read/Write I/O Feature allows an attribute in a template or an instance to be configured so that its value is both read from and written to an external reference. The Read/Write I/O Feature monitors the value/quality of an input and sends outputs on state change.



The **Write to** (output) destination can be the same or different from the **Read from** (source). The references are always to another acceptable attribute type in the Galaxy.

You can add multiple Read/Write I/O Features to an object. However, you cannot add a Read/Write I/O Feature to an attribute that already has a Read or Write I/O Feature.

Note: You can add a Read/Write I/O Feature to lockable attributes, but they only function correctly during run time if the configured attribute is unlocked.

When Objects Are On Scan

When an object is On Scan, the value and quality of the attribute configured with a Read/Write I/O Feature mirrors the quality of the externally referenced attribute during a successful read. The data quality of the attribute is set to Bad when reads fail. Reads can fail because of communication errors or datatype conversion failures.

While the object is On Scan, the data can change quality. If an external set (for example, from a user) to an attribute changes either the value or quality, then a write of the attribute's value to the destination occurs during the next execute phase. The quality must be Good or Uncertain for a write to occur. For writes to occur because of a quality change, the quality change must be a transition from Bad or Initializing to Good or Uncertain.

The attribute called WriteValue is publicly exposed and plays an important role in driving outputs. When the object is Off Scan, quality is always Bad and user sets are accepted.

Using Read/Write I/O in Scripts

Two common types of scripts can be written on an attribute configured with a Read/Write I/O Feature: One can look at the input side and one can look at the output side.

The input side script uses the current value coming from the input source location and performs logic or calculations on it. This script refers directly to the attribute in its expressions. For example, if the attribute is

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

"me.attribute1", the script refers directly to "me.attribute1" for data change conditions and for expressions within the script.

The output side script can modify output or validate a new requested output value. This script refers to the "WriteValue" attribute configured on the attribute: "me.attribute1.WriteValue".

To validate a new requested value to the attribute1, for example, a data change condition expression is written on "me.attribute1.WriteValue". In addition, if the script wants to do clamping or validation, it can manipulate the "me.attribute1.WriteValue" directly to clamp the output value. For example:

```
If (me.attribute1.WriteValue > 100.0 ) then
  Me.attribute1.WriteValue = 100.0;
Endif;
```

The data change expression for this script is "me.attribute1.WriteValue" because this value changes when a new value is about to be written to the field.

The script can intercept this value just before output and manipulate it. To prevent WriteValue from being written out, its data quality can be set to Bad with the SetBad() function.

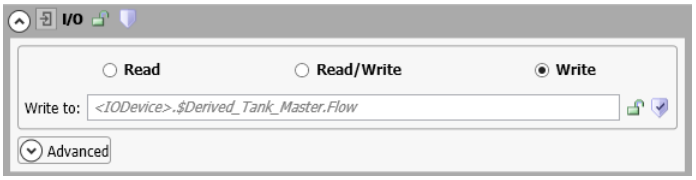
For more information, see Working with Outputs.

Configure I/O as Write-only

Select **Write** in the I/O parameters area. Define a **Write to** (output) **destination** by using I/O auto-assignment, by typing in the reference string, or by clicking the **Attribute Browser** button at the right.

- Use I/O auto-assignment to prepare the output destination for automatic assignment to a configured Device Integration (DI) object or other data source. For information about using I/O auto-assignment, see Using I/O Auto Assignment.
- If the OutputDest attribute is locked in the parent template, the attribute cannot be updated with the resolved reference when the object is deployed, and the run-time value will be "---Auto---".
- Use the **Attribute Browser** to select an attribute and automatically insert the correct reference string for that attribute. For more information about using the Attribute Browser, see Reference Objects Using the Galaxy Browser.

Select the **Output Every Scan** check box if you want the attribute to write to the **Destination** attribute every scan period of the object. Otherwise, the write executes only when the value is modified or when quality changes from Bad or Initializing to Good or Uncertain.



Writeable and Calculated attributes can be configured with a Write I/O Feature. Arrays are not supported.

A Write I/O Feature can be added to a derived template or to an instance. If added to a template, the existence of the Write I/O Feature is automatically locked in derived objects. The output **Destination** attribute is separately lockable in templates.

If the data types of the configured and destination attributes are the same and only when the quality of the extended attribute is good, the two attributes are set to equal values according to the configured object's execution rate. If the two attributes are different data types, coercion rules are applied. If coercion fails, the extended attribute is placed into a configuration error and type mismatch state.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

An attribute that is enhanced with a Write I/O Feature has the following characteristics:

- A value can be output only when quality is Good or Uncertain. The quality is not output, only the value is output, because quality is not output on sets.
- When the quality changes from Bad or Initializing to Good or Uncertain, the value is output, even if the value is not modified.
- When the quality changes from Good to Uncertain, with no value modification, the value is not output.
- When the object goes Off Scan, no output is done.
- When the extended object is Off Scan, quality is always Good and user sets are accepted.

Working with Outputs

The following information applies to the functionality of Read/Write and Write Features as well as to the output function of AnalogDevice objects.

If a single set request is made to a destination attribute during a single scan cycle, that value is sent to the destination. During a single scan cycle, though, more than one set request to the same destination is possible. In that case, folding occurs and the last value is sent to the destination.

During a single scan cycle, only the last value requested during a scan cycle is sent to its destination when the object executes. Its status is marked as Pending as it waits for write confirmation from the destination object. All other set requests during that scan cycle are marked as successfully completed.

If one or more new sets are requested during the next scan cycle, then the second scan cycle's value is determined as described in the preceding paragraphs. It is then sent to the destination when the object executes again and the value sent to the destination during the previous scan cycle is marked with successful completion status even if write confirmation is not received.

Within a single scan cycle, data is folded and only the last set requested is sent to the destination, unless buffering is enabled. For example, an {11,24,35,35,22,36,40} sequence of set requests results in a value of 40 being sent to the destination object. All other values result in successful completion status.

The exception to this behavior is when you enable buffering, either on a Galaxy-wide basis or for an individual attribute. For more information, see I/O Feature advanced properties information in Using the I/O Feature.

Boolean data types are an exception to folding behavior when buffering is not enabled. This allows an unknown user input rate (for example, repeated button pushes) with a consistent object scan rate for outputs, and creates reproducible results.

In this case, a combination of folding as described plus maintenance of a queue of one element deep better meets the expectation of users. To begin with, the first value set after the object is deployed (the default True or False) is always written to its destination.

Subsequently, the following occurs during a single scan cycle: A two-tiered caching scheme of a Value to be Sent and a Next Value to be Sent is implemented. The Value to be Sent is based on data change as compared to the last value sent to the destination object. The Next Value to be Sent is based on data change as compared to the Value to be Sent value.

When the first data change occurs, the new value is cached in the Value to be Sent queue. Folding occurs if the same value is requested again. If another value change occurs, this second value is cached in the Next Value to be Sent queue. Again, folding occurs if the same value is requested again.

The Value to be Sent value is sent during the next scan cycle, and the Next Value to be Sent value is sent during the following scan cycle.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: In the case of Boolean data types used in Supervisory sets (sets between ApplicationObjects) or a mixture of Supervisory and User sets during a single scan cycle, the behavior is the same as the other data types.

For Boolean data types and User sets, the following examples apply:

Previous Scan Cycle Value Sent	Scan Cycle Set Requests	Value to be Sent	Next Value to be Sent
0	1,0,0,1,1	1	none
1	1,0,0,1,1	0	1
0	1,1,0,0	1	0
1	1,1,0,0	0	none

When the same attribute is extended with an Input extension and an Output extension, writes to the Output extension's **Destination** occur every scan regardless of whether the extended attribute has changed.

This behavior occurs even when the **Output Every Scan** check box is cleared, which may add more network traffic. The behavior does not apply to an Input extension.

Quality of Read, Read/Write, and Write I/O

When the object is On Scan, the value and quality of an attribute configured with a Read I/O Feature mirrors the quality of the externally referenced attribute in the case of successful reads. The data quality of the configured attribute is set to Bad when reads fail because of communication errors or datatype conversion failures.

While the extended object is On Scan, it behaves as follows: If an external set (for example, from a user) to the configured attribute causes either the value or quality to change, then a write of the configured attribute's value to the destination occurs during the next execute phase.

The quality must be Good or Uncertain for a write to occur. For writes to occur because of a quality change, the quality change must be a transition from Bad or Initializing to Good or Uncertain. The attribute called WriteValue is publicly exposed.

When the configured object is Off Scan, quality is always Bad and user sets are accepted.

Configure I/O for Use with AVEVA Telemetry Server Communication Drivers

If you are using AVEVA Telemetry Server Communication Drivers to send and receive data from Application Server, you will need to configure I/O attributes, in accordance with the applicable communications protocol. The Telemetry Server communicates with outstations via the DNP3, IEC-60870, and/or Modbus protocols. Use the reference syntax that corresponds with the protocol you are using. Since some outstations may send data to the Telemetry Server only once per day, enable buffering for the Galaxy to ensure that data is not lost. See Auto-Configure Data Buffering for information about enabling buffering on a Galaxy-wide basis.

Note: If multiple attributes are configured for buffered data from the same data point in the Telemetry Server, or if auto-buffering is enabled, only one attribute will receive the buffered data from that data point at run time.

Dynamic References

You can configure Telemetry Server data points using dynamic references. When you create and deploy an attribute with a dynamic reference in Application Server, the point is automatically generated on the Telemetry Server, and eliminates the need to manually create the same reference in two places.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

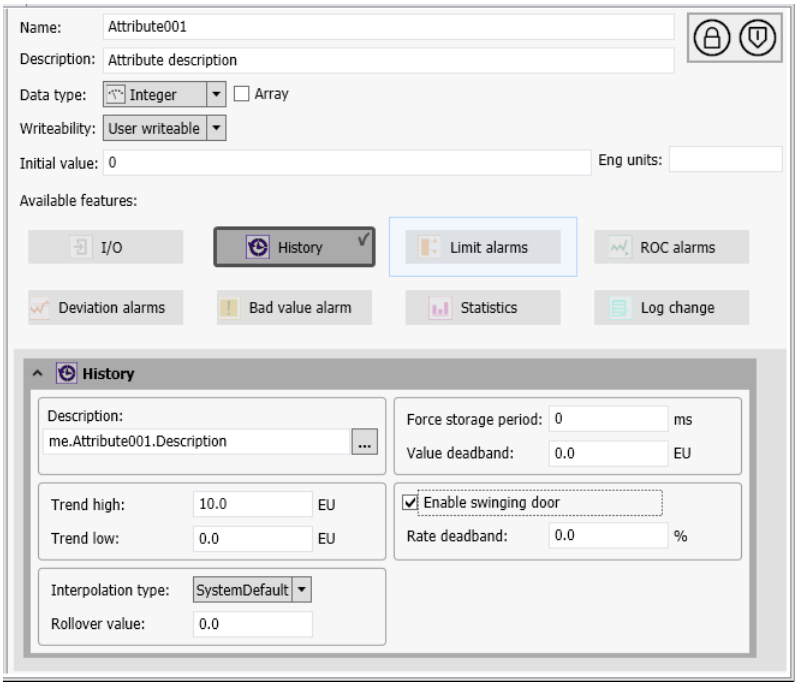
Additional Information

For more information about enabling buffered data and creating Telemetry Server references, see the following sections:

- Auto-Configure Data Buffering
- Working with Buffered Data
- Telemetry Server Data References
- Telemetry Server Dynamic Tags

Using the History Feature

Any attribute that exists at run time and is not already historized can be configured with a history Feature.



A history Feature can be added to a template or an instance attribute. If added to a template attribute, the existence of the history Feature is automatically locked in derived objects.

You can configure Writeable and Calculated attributes of the following data types with a history Feature:

- Float, Double (stored as a Float)
- Integer
- Boolean
- String stored as Unicode, 512 character limit
- Custom Enumeration stored as an Integer
- ElapsedTime stored as seconds

You can configure the following attributes for a history Feature:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

History Feature Attributes	Description
Description	The attribute containing the description string, if any, associated with the attribute being configured. Enter the attribute manually, or use the browse button to open the Galaxy Browser to find a specific attribute. Example: For the description associated with Attribute001 that has been added to object Reactor31, you would enter or browse to "Reactor31.Attribute001.description". The description also can be a simple string without reference to an attribute.
Force storage period	The Historian normally stores values by exception. The force storage period is designed for situations where data is expected to change very infrequently, compared to the engine scan rate, for examples, setpoints and values which only toggle in abnormal cases. The force storage period sets a length of time, in milliseconds, after which the value must be historized, even if the value has not changed, or has not changed more the the **Value deadband**. Setting the value to 0 disables the parameter. Example: A setting of 3600000 (60 seconds/minute x 60 minutes/hour x 1000) will force the value to be stored once per hour (if the value remains unchanged), as measured from the time the object was last put onscan. The force storage period should be long relative to the AppEngine scan cycle. If the force storage period is shorter than the scan cycle, forced storage will occur at every scan period, as the value change is not recorded until the scan takes place. When the storage period is set to a value greater than the scan period and there is no change in value, or if the change in value is less than the **Value deadband**, the forced storage period artificially updates the timestamp on the previous (unchanged) value, and allows the Historian to store that value again with the artificial timestamp. In most cases, that is, in cases where a change in value occurs, values are associated with the timestamp from the source system (e.g., PLC), and that time is applied to the historized values. On the other hand, if a value is historized because of the force storage period, the value is instead stored with the system time of the engine scan when the force storage period was exceeded. Factors such as communications latency, clock differences, and detailed engine timing can result in force storage values being historized out of order or overlapping with those values historized due to value changes. It is also possible that the timeout for the forced storage period may coincide exactly with the timestamp of an actual exception update from the instrumentation. If this occurs, both values are stored with the associated timestamp. To avoid this, disable (set to 0) the force storage period in cases where the value normally updates frequently. In cases where the value rarely

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

History Feature Attributes	Description
	changes, set the force storage period to a much longer time period, for example, to a value between 1 and 48 hours (3600000 to 172800000 ms).
Value deadband	The threshold value, measured in engineering units, that the absolute value of the difference between the new and last-stored values must differ before storing the new value to history. The value deadband is intended to filter out low-level noise in the signal so that only significant changes are historized. When a value deadband is set, value changes that are within the value deadband will not be historized. Therefore, you should only specify a value deadband when the reduction of processing and storage overhead is more important than completeness of the data. If you choose to set a value deadband, the value should be set to as small a value as possible. The value should be based on how large (or small) the changes in value that are expected. A value of zero (0) is valid and means that any level of change results in the new value being stored. A change in Quality always causes a new record to be stored, regardless of whether the Value has changed. When combined with the **Force storage period**, new values are compared with the last historized value, regardless of whether the the previous value was historized because it exceeded the value deadband or exceeded the force storage period.
Trend high	The default top of a trend scale. This value must be greater than or equal to the low value for the trend. If this value is changed at run time, the maximum engineering unit change is not reflected in the HIstorian until you redeploy the object. This attribute applies only to numeric data types.
Trend low	The default bottom of a trend scale. This value must be less than or equal to the high value for the trend. If this value is changed at run time, the minimum engineering unit change is not reflected in the HIstorian until you redeploy the object. This attribute only applies to numeric data types.

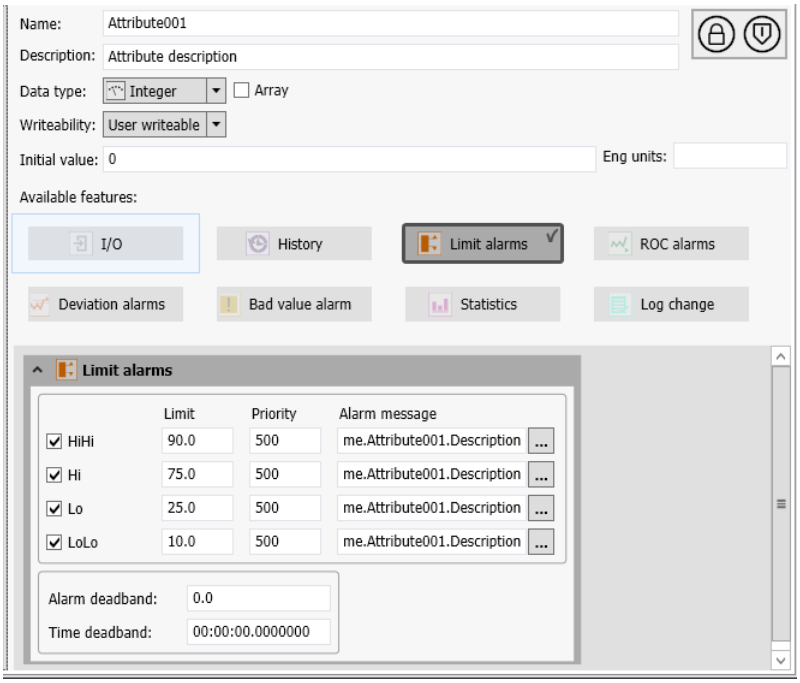
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

History Feature Attributes	Description
Enable swinging door	Enable and provide a valid swinging door rate deadband and the force storage period becomes the deadband override period. Boolean or string data types cannot be configured with a swinging door deadband.
Rate deadband	Available only if swinging door is enabled. The percentage rate of change deadband based on the change in the slope of incoming data values to the HIstorian. Example: A swinging door rate deadband of 10 percent means that data is saved to the HIstorian if the percentage change in slope of consecutive data values exceeds 10 percent. Default is 0.0, which indicates a swinging door rate deadband is not applied. Any percentage greater than 0.0 can be assigned to the rate deadband.
Interpolation type	The method used by the Historian to interpolate analog historical data. The interpolation type determines which analog value is selected during a Historian ata retrieval cycle. Select an Interpolation type: **System Default**: The HIstorian system-wide interpolation setting is used. The system-wide setting must be either stairstep or linear interpolated. **Stairstep**: The last known value is returned with the given cycle time. If no valid value can be found, a NULL is returned to the HIstorian. **Linear**: HIstorian calculates a new value at the given cycle time by interpolating between the last known value prior to the cycle time and the first value after the cycle time.
Rollover value	A non-negative numeric value that represents a tag's reset limit when the Historian operates in counter retrieval mode. In counter retrieval mode the Historian uses a tag's rollover value to calculate and return the delta change between consecutive retrieval cycles. The default value is 0.0. Boolean and string data types cannot be configured with a rollover value.

Using the Limit Alarms Feature

Select the **Limit alarms** Feature to add and configure a Limit Alarm on an attribute of Integer, Float, or Double data type. You can add a Limit alarm Feature to a template or instance. If added to a template attribute, the Limit alarm Feature is automatically locked in derived objects. Limit alarm Features cannot be added to attribute arrays.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



You can enable up to four categories of Limit alarms. When enabled, alarms will be triggered when the value reaches the configured limit.

- HiHi
- Hi
- Lo
- LoLo

For each Limit alarm, specify the following parameters:

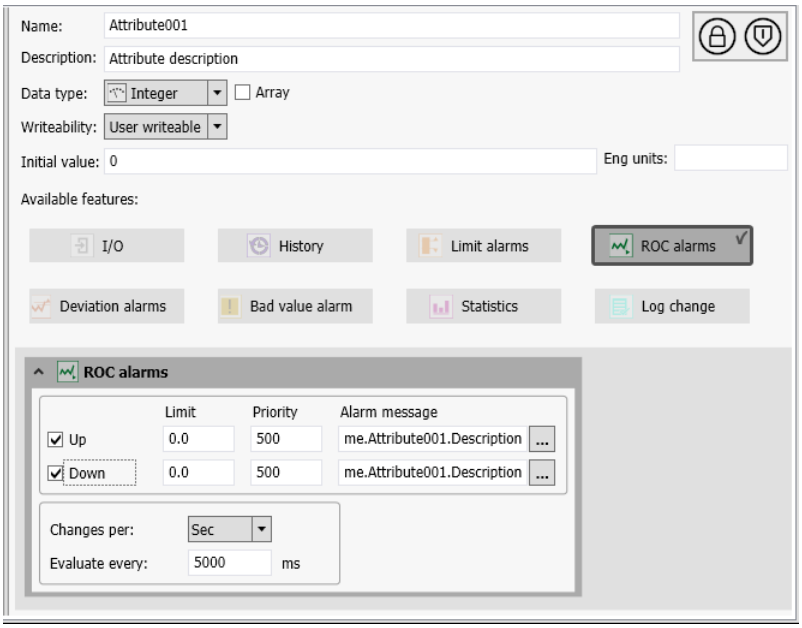
Limit Alarm Feature Parameters	Description
Limit	The limit that the attribute value must exceed to trigger an alarm. Enter the limit value for each enabled Limit alarm category.
Priority	A numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent. For more information about pre-configuring alarm priorities, see Configuring Alarm Historization, Mapping, and Shelving Priority Ranges.
Alarm message	Browse and select an existing attribute or type a text string as an alarm message. This text string appears in the InTouch alarm view.
Alarm deadband	The amount, in engineering units, that the attribute value must exceed the configured HiHi, Hi, Lo, or LoLo limit before an alarm is triggered.
Time deadband	The time, in seconds, that must elapse after the attribute value exceeds an alarm limit before the alarm is triggered.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For additional information, see Limit Alarms.

Using the ROC Alarms Feature

Select the **ROC alarms** Feature to add and configure a Rate of Change (ROC) alarm on an attribute of Integer, Float, or Double data type. You can add a ROC alarm Feature to a template or instance. If added to a template attribute, the ROC alarm Feature is automatically locked in derived objects. ROC alarm Features cannot be added to attribute arrays.



You can enable up to two categories of ROC alarms. When enabled, alarms will be triggered when the value reaches the configured limit.

- Up: Rate of increase
- Down: Rate of decrease.

For each ROC alarm, specify the following parameters:

ROC Alarms Feature Parameters	Description
Limit	The limit that the attribute rate of change value must exceed to trigger an alarm. Enter the limit value for each enabled alarm category. The down limit is a positive value, but indicates a negative rate of change, or a descending rate Hi limit.
Priority	A numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent. For more information about pre-configuring alarm priorities, see Configuring Alarm Historization, Mapping, and Shelving Priority Ranges.
Alarm message	Browse and select an existing attribute or type a text string as an alarm message. This text string appears in the InTouch alarm view.

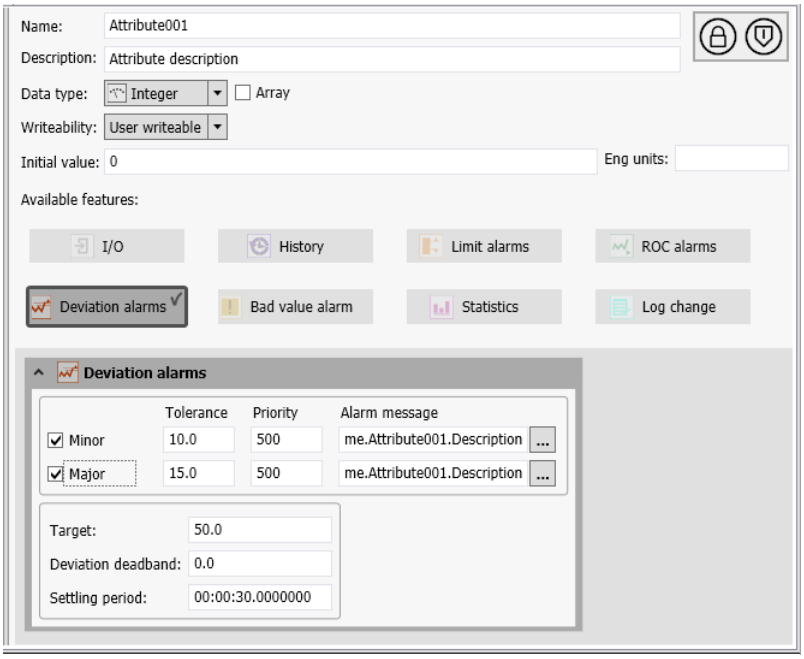
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

ROC Alarms Feature Parameters	Description
Changes per	Select a unit of time for the rate of change calculation – seconds, minutes, hours, or days.
Evaluate every	The time interval, in milliseconds, at which the rate of change calculation and detection is performed.

For additional information, see Rate of Change Alarms.

Using the Deviation Alarms Feature

Select the **Deviation alarms** Feature to add and configure a Deviation alarm on an attribute of Integer, Float, or Double data type. You can add a Deviation alarm Feature to a template or instance. If added to a template attribute, the Deviation alarm Feature is automatically locked in derived objects. Deviation alarm Features cannot be added to attribute arrays.



You can enable up to two categories of Deviation alarms. When enabled, an alarm will be triggered when the attribute level deviates from a target value by a configured amount.

- Major: The major alarm deviation tolerance.
- Minor: The minor alarm deviation tolerance.

For each Deviation alarm, specify the following parameters:

Deviation Alarms Feature Parameters	Description
Tolerance	The major or minor deviation tolerance that the attribute value must exceed to trigger an alarm.

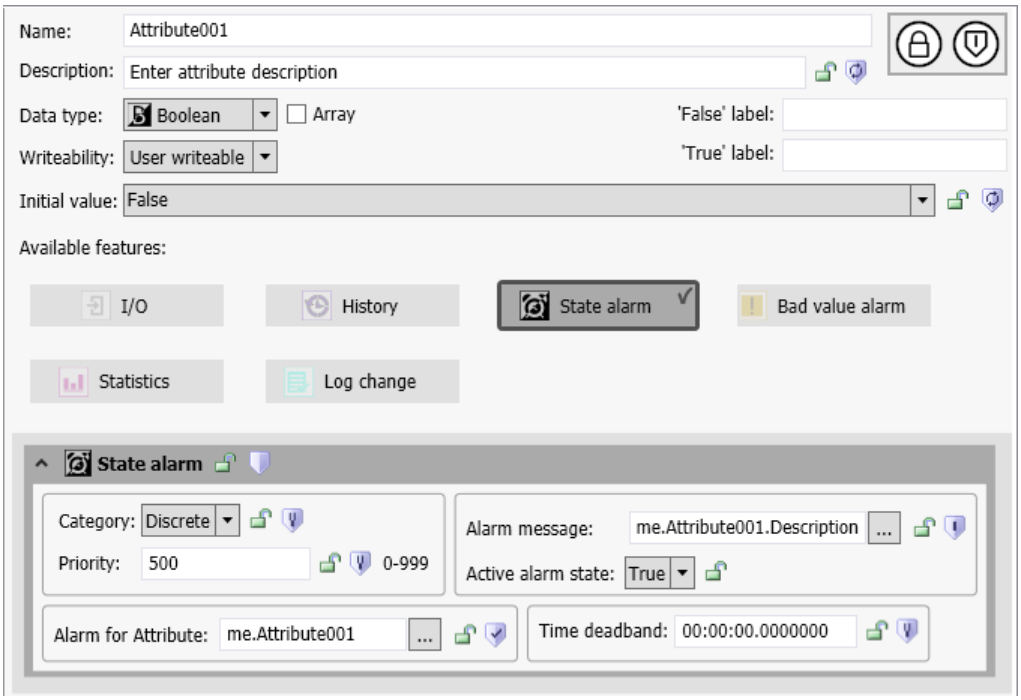
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Deviation Alarms Feature Parameters	Description
Priority	A numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent. For more information about pre-configuring alarm priorities, see Configuring Alarm Historization, Mapping, and Shelving Priority Ranges.
Alarm message	Browse and select an existing attribute or type a text string as an alarm message. This text string appears in the InTouch alarm view.
Target	The setpoint value for deviation calculation.
Deviation deadband	The amount the attribute value must drop (measured in engineering units) below any of the deviation limits before the respective condition is set to FALSE.
Settling period	The time, in seconds, allowed for the attribute value to return within allowable range after a Target change is made before the deviation alarm is triggered. A deviation alarm cannot occur again until the alarm clears. The value you specify will be converted to the following format: days hh:mm:ss.fffffff.

Using the State Alarm Feature

Select the **State alarm** Feature to add and configure a State alarm on an attribute of Boolean data type. You can add a State alarm Feature to a template or instance. If added to a template attribute, the State alarm Feature is automatically locked in derived objects. State alarm Features cannot be added to attribute arrays.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



You can set an alarm message and Priority for a state alarm. The time deadband is used to filter out rapid, transitory value spikes, and the active alarm state determines the trigger value (True or False) that will activate the alarm. You can also link the State alarm to an external attribute to capture alarm conditions detected in the PLC, rather than in the Application Server object.

For each State alarm, specify the following parameters:

State Alarms Feature Parameters	Description
Category	The alarm category. Select a category from the list box. See State Alarms for additional information.
Priority	A numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent. For more information about pre-configuring alarm priorities, see Configuring Alarm Historization, Mapping, and Shelving Priority Ranges.
Alarm message	Browse and select an existing attribute or type a text string as an alarm message. This text string appears in the InTouch alarm view.
Active alarm state	The state of the active alarm, True or False, when the alarm is triggered.
Alarm for Attribute	This lets you link to an external attribute, such as a PV attribute, to provide historical state alarm data from the PLC. This allows the PLC to pass alarm conditions it detects directly to Application Server. You can use the Attribute Browser to select which attribute to link.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

State Alarms Feature Parameters	Description
	The Alarm for Attribute parameter is for use only with Historian Clients and is not compatible with InTouch HMI clients. For more information, see How to Configure an External Alarm Source.
Time deadband	The time, in seconds, that must elapse after the attribute value exceeds an alarm limit before the alarm is triggered.

Using the Bad Value Alarms Feature

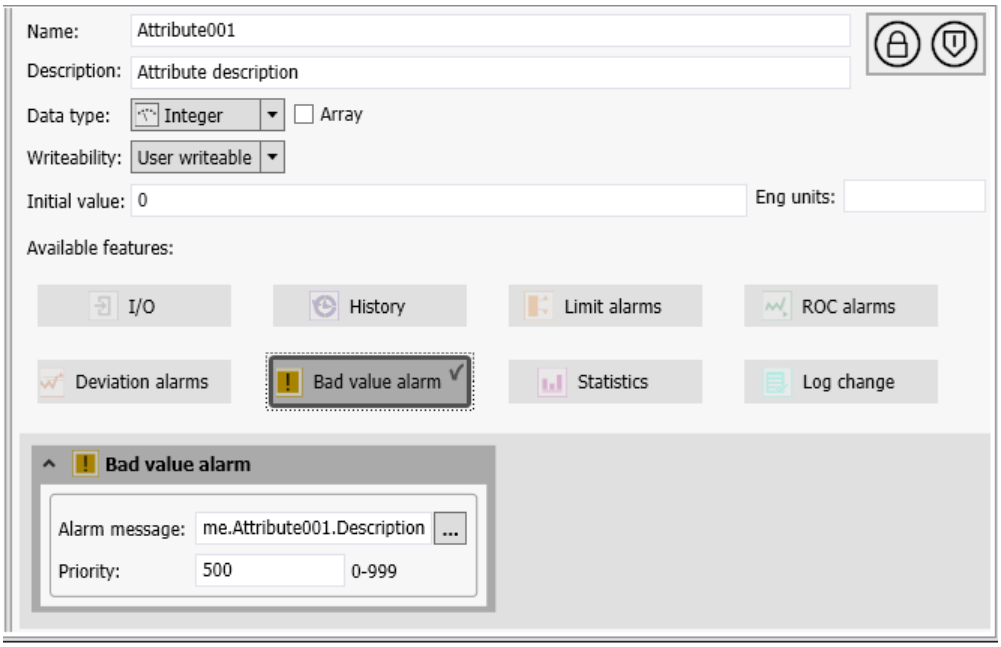
Select the **Bad value alarms** Feature to add and configure a Bad Value alarm on an attribute of Boolean, Integer, Float, or Double data type. If enabled, alarms will be triggered when the attribute has bad data value or data quality.

A bad value results from bad data quality. Bad quality can occur when:

- The PLC indicates that quality is bad (for example, if communications with a physical device is lost).
- There is a communications error (for example, an OPC or I/O server failure).
- There is a configuration error (for example, a broken I/O reference).
- An out of range value is recorded.
- A script sets data quality to bad.

You can configure graphic elements to highlight when a bad data quality state occurs, and its underlying cause. See Use Quality and Status Indicators to Provide Run-Time Feedback for more information.

You can add a Bad Value alarm Feature to a template or instance. If added to a template attribute, the Bad Value alarm Feature is automatically locked in derived objects. Bad Value alarm Features cannot be added to attribute arrays.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

In the **Alarm message** box, you can browse and select an existing attribute or you can type a text string as an alarm message. This text string appears in the InTouch alarm view.

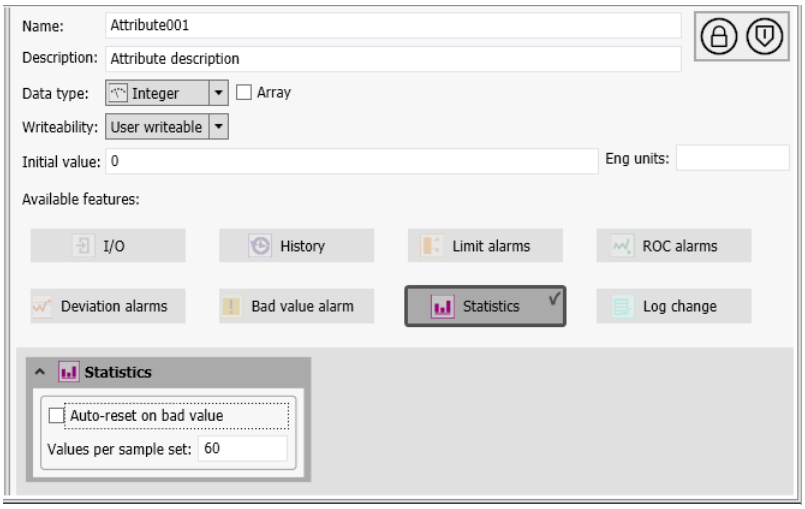
Specify a **Priority** for this alarm, a numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent.

For more information about pre-configuring alarm priorities, see Configuring Alarm Historization, Mapping, and Shelving Priority Ranges.

Using the Statistics Feature

Select the **Statistics** Feature to add and configure calculation of statistics on an attribute of Boolean, Integer, Float, or Double data type. If enabled, the following statistics are calculated:

- Time since last transition: If no change in the input value, the elapsed time value increments to show the total elapsed time since the last transition. A change in input value resets the elapsed time since last transition.
- Average: Defined as the sum of the sample size divided by the number of sample size attributes in the data set.
- Standard deviation: Defined as the amount of variation or "scatter" from the average.



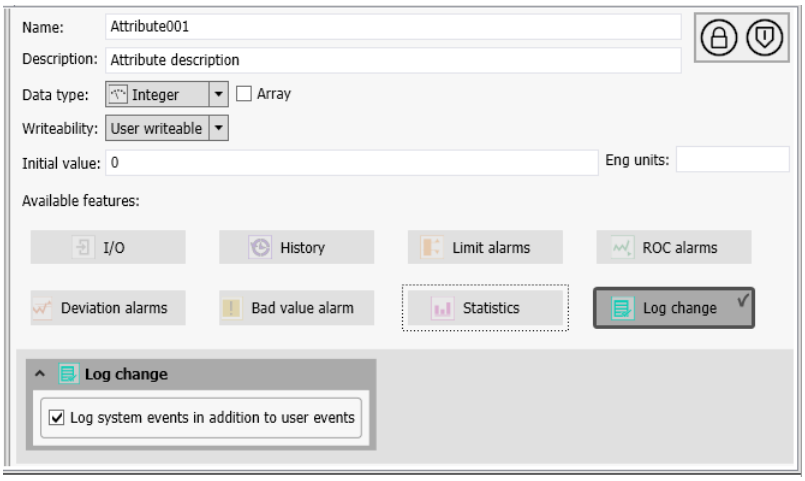
You can configure the following parameters for the Statistics Feature:

Statistics Feature Parameters	Description
Auto reset on bad value	If enabled, the input automatically triggers a reset command when the quality of an input goes from BAD or UNCERTAIN to GOOD.
Values per sample set	The maximum number of samples to collect in calculating the mean and standard deviation. The minimum number of samples is 2. A large sample set will impact run-time performance. This attribute is unavailable for a Boolean data type.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Using the Log Change Feature

Select the **Log Change** Feature to log system events for an attribute of Boolean, Integer, Float, or Double data type.



When enabled, the Log Change feature logs data change events as well as user events.

Using I/O Auto Assignment

Instead of configuring I/O references manually, or writing scripts to set them at run time, you can use I/O auto assignment. Manual configuration of I/O references can be time consuming. Scripting these references eliminates the issues of manual configuration, but can significantly increase the time needed for deploying objects. With I/O auto assignment, you do not need to check out individual objects to configure I/O references, and you do not experience the run-time penalties associated with scripting.

Note: I/O auto assignment is the default setting for application and other system objects, such as area objects. Device Integration objects, such as the OPCClient object, must be manually configured.

When you add input or output attributes to an area or application object in the **Attributes** page of the Object Editor, the default setting prepares these attributes for I/O automatic assignment. The auto assignment reference appears in the I/O section of the **Attributes** page if you have enabled the I/O attribute feature. The default string for an input reference is:

```
<IODevice>.[ObjectName].[AttributeName].InputSource
```

where *[ObjectName]* is the hierarchical name of the application or system object, and [*AttributeName]* is the name of the attribute.

The default string for an output reference is:

```
<IODevice>.[ObjectName].[AttributeName].OutputDest
```

The string *<IODevice>* is a placeholder that indicates the I/O reference will be built automatically in a process called auto binding. The reference resolves automatically as an auto-bound reference when you link the object to a scan group and DI object, without having to manually enter or script the reference.

The following is an example of an I/O reference string before the object has been assigned to a scan group and DI object:

```
<IODevice>.Mixer.Tank.Inlet.InputSource
```

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Once you assign the object to a scan group, the reference resolves to include the assigned Device Integration object and scan group. For example, assigning the object to the scan group "Fast" under DI object "OPC001" will change the reference to:

```
OPC001.Fast.Mixer.Tank.Inlet.InputSource
```

Important: Do not lock InputSource or OutputDest attributes when using I/O auto assignment. If either InputSource or OutputDest attributes, or both, are locked in the parent template, the attributes cannot be updated with the resolved auto-bound reference when the object is deployed, and the run-time value will be "---Auto---".

I/O auto assignment is configured in the **IO Devices view**. Use this view to associate application and system objects with DI objects and scan groups.

Scan groups are contained by DI objects and help categorize devices that are associated with them on the basis of how often their I/O points are scanned. For basic information about DI objects, refer to Device Integration Templates. For basic information about scan groups, refer to the OPCClient Object help file.

Note: A DI object will not appear in the **IO Devices view** unless at least one custom scan group has been defined for it.

Assign objects that need to be closely monitored or that have a faster rate of change to a scan group with a faster scan rate. Objects with slower rates of change can be assigned to a scan group with a slower scan rate. The scan rate is the rate at which the AVEVA run time will perform scans. This may be different than the scan rate a PLC uses to monitor the hardware attached to it.

Note: Fast scan groups consume more network and computing resources than slow scan groups. In a large Galaxy, load balancing may require you to use multiple scan groups with similar scan rates.

The I/O references for the objects' attributes set to use I/O automatic assignment are dynamically generated when you assign an object to a scan group. The auto-bound reference replaces the placeholder string and is displayed in the IDE. It is generally in the following format:

```
<DIObject>.<ScanGroup>.<Object>.<Attribute>
```

`<DIObject>` and `<ScanGroup>` correspond to the name of the DI object and scan group to which you assign the object. For example, if the DI object name is `DI01` and the scan group name is `FastSG`, then the auto-bound reference will start with `DI01.FastSG`.

`<Object>` may consist of multiple elements if there are contained objects. The following is an example of an object string:

```
Tank3.Valve1.Attr2
```

Here, `Tank3.Valve1` corresponds to <object>. The complete auto-bound I/O reference would be:

```
DI01.FastSG.Tank3.Valve1.Attr2
```

Although you assign objects, references are generated for each I/O attribute associated with the object, and not for the object itself. Therefore, you may have to change the I/O references for a subset of an object's attributes. Once objects are associated with DI objects and scan groups, you can validate the auto-bound assignment for each of the object's attributes in the **Mapping view**, and if necessary, you can enter override values in this view.

Assigning Areas and Objects to Scan Groups

The **IO Devices view** displays all application and system objects, as well as DI objects that have custom scan groups associated with them. Templates are not shown in this view, nor are DI objects that do not have custom scan groups defined for them (DI objects with only the default scan group are not displayed in the IO Devices view).

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: You cannot assign application and system objects to a default scan group; these assignments can only be made for custom scan groups.

In the **IO Devices view**, objects and areas are initially unassigned and placed in a flat view under the "Unassigned IO Devices" folder. You then select one or more of these unassigned objects or areas and drag and drop them onto a scan group. Deployed application and system objects cannot be assigned to a scan group. Undeploy them first.

Note: DI objects can remain deployed during the assignment process.

Once an object is attached to a scan group, the I/O references for the objects' attributes set to use I/O automatic assignment are automatically generated for you by concatenating the DI object, scan group, object (including contained objects), and attribute names.

You can assign multiple objects for auto-binding at once. Hold down the **shift** or **ctrl** key to select multiple objects and drag the selected items to a scan group. Object assignment is based on hierarchy and area inheritance, as follows:

- If all the objects you select are at the same hierarchical level, all contained (subordinate) objects that have not been assigned, except sub-areas, are also selected and will move to the same scan group.
  - To assign a sub-area, it must be specifically selected.
  - If some contained objects were previously assigned to a different scan group, these will retain their original assignments. To change an existing assignment, you must specifically select the assigned object and move it to a different scan group.
- If you select objects at different hierarchical levels for assignment to a scan group – for example, if you select an area, an object within the same area, and an object from another area – then only the selected items move to the scan group.
- If you select an area that contains another area, only the area selected will move. You must specifically select an area to assign it, regardless of its hierarchical level.

When an area or containing object is assigned to a scan group, the objects it contains are assigned as follows:

- Objects within the area that share the area's device linkage will be reassigned to the new device along with the area. This remains true even in the case where both the area and the contained objects are not currently linked to a device. Unassigned objects within the area will use the area's assignment.
- Objects within the area that are already assigned to a scan group that is different from that of the hosting area will retain their original assignments.
- If necessary, you can reassign contained objects to a different scan group after the initial assignment, or override assignments in the **Mapping view**.

Note: You cannot change scan group or DI object assignments for deployed objects in the **IO Devices view**. You must undeploy the object first.

Deleting a scan group will delete the I/O assignments of any objects that were previously assigned to it, including override values.

Caution: Creating or deleting scan groups within a derived DI object template may overwrite I/O assignments when these changes are propagated to instances of the DI object template. If this happens, all I/O auto assignment information for objects linked to the updated DI objects, including override values, will be lost. The objects will move to the "Unassigned I/O Devices" folder.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To assign an object or area to a scan group

1. In the **IO Devices view** or **Model view**, select application objects and areas for assignment.
2. Drag and drop the objects and areas to a scan group in the **IO Devices view**.
3. The object or area and all contained objects will be moved to the selected scan group.

   Note: If the area or object has contained objects that were previously assigned to a different scan group, these will retain their original assignments. Contained areas will not move and must be overtly selected.

4. The I/O references for each attribute set for I/O auto assignment will be dynamically configured and displayed as auto-bound in the **Mapping view**.

Renaming Application, System, and DI Objects

You can rename application, system and DI objects in the **IO Devices**, **Model** and **Deployment** views. If an area or object has already been assigned to a scan group, its I/O references are updated automatically.

Renaming Scan Groups

Changes to scan groups (renaming, adding, or deleting them) can only be done by editing the DI object that contains the scan group in the Object Editor. Use the DI object's **Scan Group** page to make any necessary changes.

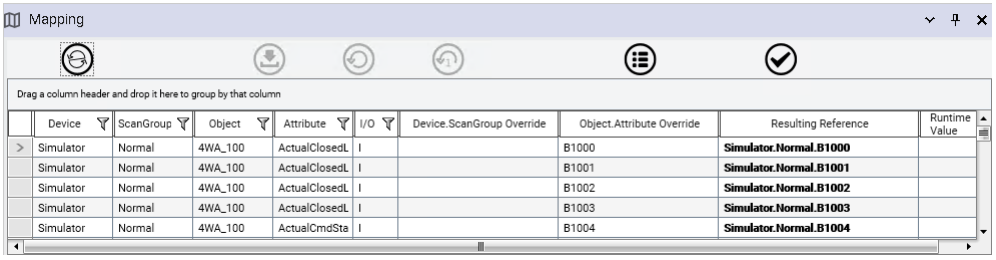
If scan groups are renamed at the template level, the changes will propagate to the DI object instances and I/O assignments will be preserved.

Note: You cannot rename or delete the default scan group.

Validating and Editing I/O Assignments

The **Mapping** view is a table that displays I/O auto assignment references for application and system objects that are linked to DI object scan groups. Only auto-bound references are displayed in the **Mapping** view. Application and system objects not yet assigned to a scan group, and manually configured references and are not shown. To be visible in the **Mapping** view, the object must be selected in the **IO Devices** view.

When you initially open the **Mapping** view after starting the IDE, the table will scroll so the far right column is in view.



Device	ScanGroup	Object	Attribute	I/O	Device.ScanGroup Override	Object.Attribute Override	Resulting Reference	Runtime Value
Simulator	Normal	4WA_100	ActualClosedL	I		B1000	Simulator.Normal.B1000	
Simulator	Normal	4WA_100	ActualClosedL	I		B1001	Simulator.Normal.B1001	
Simulator	Normal	4WA_100	ActualClosedL	I		B1002	Simulator.Normal.B1002	
Simulator	Normal	4WA_100	ActualClosedL	I		B1003	Simulator.Normal.B1003	
Simulator	Normal	4WA_100	ActualCmdSta	I		B1004	Simulator.Normal.B1004	

- Selecting a DI object in the **IO Devices** view lists I/O auto assignment attributes for all auto-bound objects linked to all scan groups under it.
- Selecting individual scan groups restricts the scope of the information displayed in the **Mapping** view to the auto-bound objects that have been linked to the selected scan groups.
- Selecting individual application or system object further restricts the scope of attributes displayed to only those associated with the selected object.

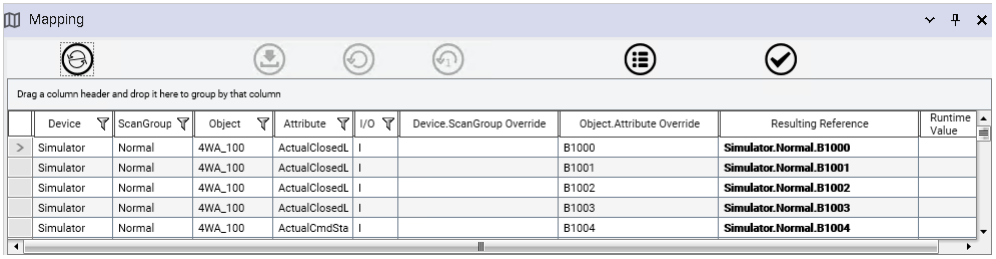
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: You can select multiple items in the **IO Devices** view. Selected items can be at different hierarchical levels. Selecting a subordinate object will exclude non-selected objects within the device hierarchy, even though the parent object is selected.

In the **Mapping** view, you can view and validate I/O references for each automatically generated attribute, and you can override the automatically generated I/O references. As is the case in the **IO Devices** view, you do not have to check out objects to change their I/O assignments.

Using the Mapping View

The **Mapping** view is divided into columns, each of which displays information for an I/O attribute that has been auto assigned.



Column Heading	Definition
Device	Device Integration Object (DIO) name
ScanGroup	Scan group name
Object	Application object or area object name
Attribute	Attribute name
I/O	Input or Output
Device.ScanGroup Override	Override setting: enter an override value to replace the DIO and scan group that was automatically assigned
Object.Attribute Override	Override setting: enter an override value to replace the object and attribute names
Resulting Reference	Concatenated I/O reference string (for example, "`DIO1.SG2.Mixer3.Attr5`" or "`DIO1.SG2.Conveyor.40001`")
Runtime Value	Data returned from run time when the **Validate References** button has been clicked

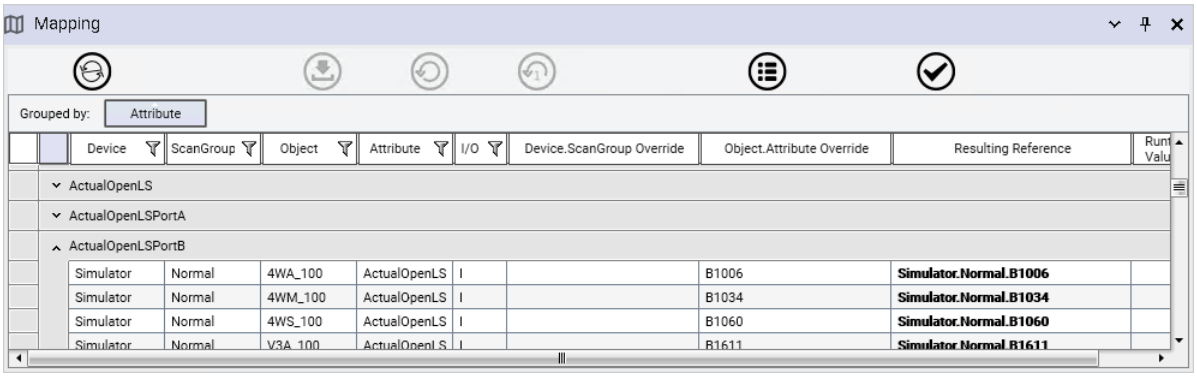
You can change how I/O attributes are displayed by sorting, grouping, or filtering the attributes.

Attributes are initially sorted by Device name (DI object). To sort attributes using a different column, click on the column heading once to sort in ascending order. Click a second time on the same column heading to sort in descending order. A third click on the column heading turns off sorting for that column.

To group objects by I/O assignment criteria, drag a column heading to the top of the table in the **Mapping** view. For example, if you drag the heading "Attribute" to the top of the table, I/O assignments will be grouped by

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

attribute name. In the figure below, I/O assignments are grouped by attribute name. Note each attribute grouping has been expanded.



Note: Grouping is not available for the **Device.ScanGroup Override**, **Object.Attribute Override**, or **Resulting Reference** columns.

You can use more than one column heading to group I/O assignments. For example, you could group IO assignments first by attribute name and then by I/O type.

Columns can also be filtered to display only certain entries by clicking the filter symbol and then selecting the filtering criteria. When a filter is applied, the color of the filter symbol changes to orange.

Note: Filtering is not available for the **Device.ScanGroup Override**, **Object.Attribute Override**, or **Resulting Reference** columns.

The table contains additional information that you can reveal by clicking the **Show Details** button.

Details Column Heading	Definition
Attribute Data Type	Attribute type as defined by the attribute definition set in the Object Editor
Target Data Type	Attribute type of the target, as determined by the run time
Runtime Quality	Data quality from run time

Validating I/O References

Use the **IO Devices** pane and the **Mapping** pane to populate the **Resulting References** column in the **Mapping** grid. You can then validate auto-bound I/O attributes that have been assigned in **Mapping**. To begin I/O reference validation, press the **Validate References** button, available in the **Mapping** pane.

Validating I/O references is separate and distinct from the **Validate** function available in the **Template folder** and in the application views. That **Validate** function validates the object's references configuration against the the Galaxy Repository. It is a configuration validation.

In contrast, validating I/O references is a run-time validation, and performs a one-time "get" from the run-time subsystem to retrieve the value and quality of the resulting references. A reference string in the **Resulting References** column is validated if a valid value and good quality are returned for that string. Objects should be deployed for an I/O reference validation to be meaningful.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: As a best practice, it is more efficient to validate each reference while performing I/O auto-assignment than it is to perform a single validation after configuring and deploying the entire application.

An estimate of the time needed for validation is displayed. The time is updated as validation progresses. When validation completes, the total number of unresolved I/O references is displayed along with the total number of auto-bound references checked, for example, "2 out of 5000 reference(s) failed to validate." To view references that did not validate during the validation period, click the "Show Advanced Columns" button. This will reveal the **Runtime Quality** column.

You can sort the **Runtime Quality** column to aggregate unresolved I/O references at the top of the list of references. See Using the Mapping View for additional information on sorting. You can then move the associated objects to a different scan group or apply overrides. Recheck the references after completing any required reassignments and overrides.

Overriding I/O Auto Assignments

The **Mapping** view contains two override columns. To change an I/O assignment, enter the new parameter in one or both of the two override columns.

The information in other columns cannot be edited directly. You can also reassign an object to a different scan group in the **IO Devices** view, in which case the new assignment will be reflected in the **Mapping** view.

Note: If an object is already deployed, changing any of its I/O assignments in the **Mapping** view will change the object's status from "Deployed" to "Deployed with pending configuration changes."

To override the DI object and scan group assignment that was made in the **IO Devices** view, enter the value you want in the column **Device.ScanGroup Override.** Naming restrictions for the Device.ScanGroup Override value are as follows:

- The value must contain one DI object name and one scan group name contained by the DI object, separated by a period (.).
- The value cannot start or end with a period.
- The value cannot have any leading spaces.
- The I/O reference cannot be longer than 329 characters.

To override the object and attribute name, enter the value you want in the column **Object.Attribute Override**. Naming restrictions for the Object.Attribute Override value are as follows:

- The value cannot have any leading spaces.
- The I/O reference cannot be longer than 329 characters.

You can change attribute information using the Object Editor (after checking out the object), instead of overriding values in the **Mapping** view. However, if you enter the specific I/O reference for an attribute in the Object Editor (manual configuration), it will no longer appear in the **Mapping** view. Only attributes configured for I/O automatic assignment appear in this view.

Overriding Large Numbers of Attributes

If you have a large number of overrides to enter, information can be copied and pasted between the **Mapping** view and Microsoft Excel. However, you must be careful when moving data to and from the **Mapping** view. Observe the following conditions:

- Only data in the two override columns is editable. All other columns are read only.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Even though you can only change the override columns, always copy all columns. This will help you verify that data is being copied correctly when you paste back into the **Mapping** view.
- When you copy data from the **Mapping** view, column headers are automatically included to help you while editing in Excel. When you copy your edited data back to the **Mapping** view, do not copy this header row, only copy the data.
- If you need to sort attributes, sort them in the **Mapping** view before you copy them into Excel.
- Do not change the sort order of the attributes in the **Mapping** view until you have pasted the edits made in Excel and applied them.
- Do not sort the attributes in Excel. Attributes can only be sorted in the **Mapping** view and the sort order cannot change until your edits have been applied.

WARNING! The Mapping view does not verify data that is copied to it. Therefore, if the order of attributes is different in Excel than it is in the Mapping view, the I/O references will be broken. Once changes are applied, there is no way to undo them.

To validate and edit I/O assignments

1. Click the **Validate References** icon to check that the I/O assignments in the **Mapping** table are valid. This action retrieves data and quality information from the run-time data acquisition subsystem.

   Important: I/O assignments can only be validated if the Platform, AppEngine, and associated DI object have been successfully configured and deployed.

2. Check the data returned in the **Runtime Value** column. If the assignment is not valid or if the quality is bad, no data will be returned (the **Value** column will be empty), and "Not Connected" will be displayed in the **Runtime Quality** column (click the **Show Details** button to reveal this column).
3. To isolate invalid assignments, select the filter icon at the top of the **Runtime Quality** column filter for "Not Connected" messages.
4. Edit the I/O assignment of objects as needed.
   - To override a scan group and DI object assignment, enter the new value in the **Device.ScanGroup Override** column. The override value for this column cannot contain more than one period (.) to separate the DI object name from the scan group name.
   - To override an object and attribute name, enter the new value in the **Object.Attribute Override** column.
5. Click **Apply Pending Overrides** to save your changes.
   - To discard the last unsaved override setting, click **Undo Last Pending Override**.
   - To discard the all unsaved override settings, click **Undo All Pending Overrides**.

Uploading Run-Time Configuration Changes for Auto Assigned Objects

If changes are made during run time to an auto assigned I/O reference, you can upload these changes as overrides to the original assignments. Invalid changes are ignored, and an error message is sent to the OCMC Logger.

- Run-time changes to scan groups or DI objects are saved as **Device.ScanGroup** overrides.
- Run-time changes to objects or attributes are saved as **Object.Attribute** overrides.

When you look at the object in the **IO Devices view**, you will see that the original I/O auto assignment setting is retained and the object remains auto assigned to its original DI object and scan group.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The changes uploaded from run time will appear in the **Resulting References** column of the **Mapping view**. The same information from run time will appear when you examine the object's I/O attributes in the **Object Editor**. These changes are preserved across subsequent redeployments.

To upload run-time configuration changes

1. From the **Model** or **Derivation** view, select the object(s) with run-time changes you want to preserve.
2. Click **Upload Runtime Changes** from the **Object** menu. The **Upload Runtime Changes** dialog box will appear and provide status of the upload process.

I/O Auto Assignment Workflow Example

This section describes the basic steps for implementing I/O auto assignment, from start to finish. The basic steps are:

1. Use the **Attributes** page of the **Object Editor** to prepare system objects and application objects for I/O auto assignment.
2. Use the **Object Editor** to configure DI objects and scan groups.
3. Use the **IO Devices** view to assign system and application objects to scan groups (objects must be undeployed).
4. Use the **Mapping** view to validate I/O references. You can enter overrides in this view, if necessary.

To prepare objects for I/O auto assignment

1. Create a derived template from the $UserDefined base template and open it in the Object Editor.
2. Add attributes as needed. For each attribute that requires input or output, click the I/O button.

   The default I/O mode is **Read/Write**, and the default string indicating that the attribute is primed for I/O auto assignment appears in the **Read from / Write to** field.
3. Change the I/O mode as needed.

   Important: Do not change the default string if you want to use auto assignment for the attribute. The default string indicates that auto assignment is not yet complete and the object is not linked to a PLC.
4. Save and check in the template, then close the Object Editor.
5. In the **Template folder**, create instances from the UDO derived template. The instances appear in the **Model view**, under Unassigned Area.

To configure DI objects and scan groups

1. In the **Template folder**, create a Device Integration (DI) object from the $OPCClient, $DDESuiteLinkClient, or $Redundant DIObject template. The DI object appears under "Unassigned Area" in the **Model view** and under "Unassigned IO Devices" in the **IO Devices** view.
2. Add one more scan groups to the DI object you created in the previous step. The naming convention for the scan groups must conform to the naming convention of the PLC with which you are connecting.

   The scan groups will show as belonging to the DI object in the **IO Devices view**.

To assign system and application (user created) objects to scan groups - Method 1

1. Create a new area and set it as the default. Do not deploy the area yet.
2. In the **IO Devices view**, drag and drop the area to a scan group.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3. Create new application objects. As you create new objects, they will appear in the **IO Devices view** under the default area.

   The IDE will complete the references for each I/O attribute for you.

4. If you need to change the scan group assignment for an object, drag and drop it to the new scan group.

To assign system and application (user created) objects to scan groups - Method 2

1. Create a new area and assign it to a scan group.
2. Create new application objects. These will appear under the Unassigned folder in the **Model view**.
3. Assign the application objects to the area assigned to the scan group in step 1. All application objects assigned to the area will inherit the area's linkage to the DI object and scan group.
4. If you need to change the scan group assignment for an object in the area, drag and drop it to the new scan group.

To validate I/O references (and override auto assigned values, if necessary)

1. Open the **Mapping** view. This view opens automatically if you select an object, scan group, or DI object in the **IO Devices** view.

   Each row in the **Mapping** table contains the I/O reference for one attribute.

2. Enter an override value for the attribute in one of the override columns.

   Note: The **DI Object.Scangroup Override** column can only contain the DI object name and scan group name, separated by a period (.).

3. Deploy the platform and associated DI objects.
4. Click the **Validate References** button to check that communication is established with the PLC.
5. Deploy the objects.

Using I/O Auto Assignment with OPC Clients

Using I/O auto assignment with the OPC communication protocol may require that you add overrides to allow communication between the IDE and certain PLCs, such as Allen-Bradley Logix and Siemens S7 controllers.

If you are using auto assignment to communicate with a device connected to an OPC Server on a different node (for example, a Communication Driver on a remote Galaxy), you will use overrides to build valid I/O references with all required parameters to fulfill the OPC hierarchical path. If bulk edits are required, you can copy and paste between the **Mapping** view and Microsoft Excel.

I/O auto assignment uses the syntax: *<DI Object.ScanGroup>. <Object.Attribute>*. It is the first part of the syntax that limits how the OPCClient builds the I/O auto assignment reference, since additional parameters in the device hierarchy, such as Ethernet ports and backplanes are not included. While these parameters may be logically viewed as part of the *DI Object.ScanGroup* reference, you must add the parameters to the **Object.Attribute override** column (not the **DI Object.ScanGroup override** column) to create the fully qualified I/O reference.

Using DDESuiteLink instead of OPC will accommodate the automatically generated I/O reference to these PLCs, and in most cases, overrides will not be needed.

Caution: Do not change the order of references, either in Excel or in the **Mapping** view by sorting, grouping, or filtering the reference list while editing references.

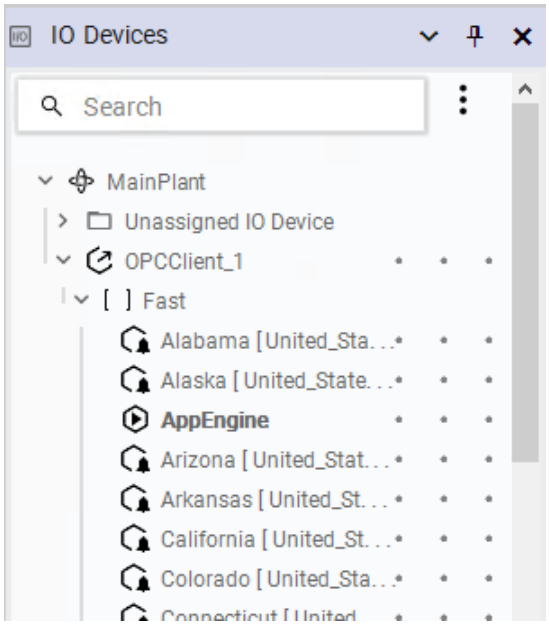
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Override Scenario 1 — Siemens S7 Controllers with an OPCClient object

The following scenario illustrates using I/O auto assignment with a Siemens S7 PLC. The Siemens S7 PLC in this example scenario includes a TCP/IP port that is in addition to the OPCClient object and scan group names captured by I/O auto assignment in the **IO Devices view**.

To configure the Siemens S7 PLC with the OPCClient object

1. Select the OPCClient object or scan group that contains the PLC in the **IO Devices view**.



   I/O attributes for each object associated with the OPCClient or scan group (depending on which is selected in the **IO Devices** view) are shown in the **Mapping** view.

2. In the **Mapping** view, the resulting reference will be in the form:
   `OPCClientName.ScanGroupName.ObjectName.AttributeName`

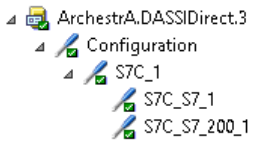
   Example (**Resulting Reference** column):

   `OPCClient_01.SG1.UDO_01.Attribute_001`

3. Do not edit the **Device.ScanGroup Override** column (leave it blank). This leaves the OPCClient Object and scan group assignment as defined in the **IO Devices** view.

4. Enter the fully qualified field reference in the **Object.Attribute Override** column. For example:

   `S7C_1.S7C_200_1.UDO_01.Attribute001`



5. Apply the override. The resulting reference will be displayed. For example:

   `OPCClient_01.SG1.S7C_1.S7C_200_1.UDO_01.Attribute001`

Override Scenario 2 — Allen-Bradley CIP Controllers with an OPCClient object

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The following scenario illustrates using I/O auto assignment with a Allen Bradley Logix PLC. The Allen Bradley Logix PLC in this example scenario includes a device hierarchy with a port object, ethernet port, backplane, and controller. These are in addition to the OPCClient object and scan group names captured by I/O auto assignment in the **IO Devices view**.

To configure the Allen-Bradley Logix PLC with the OPCClient object

1. Select the OPCClient object or scan group that contains the PLC in the **IO Devices** view.

   I/O attributes for each object associated with the OPCClient or scan group (depending on which is selected in the **IO Devices** view) are shown in the **Mapping** view.

2. In the **Mapping** view, the resulting reference will be in the form:
   `OPCClientName.ScanGroupName.ObjectName.AttributeName`

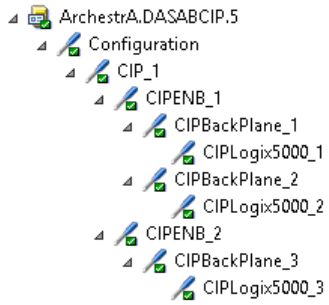
   Example (**Resulting Reference** column):

   `OPCClient_01.SG1.UDO_01.Attribute_001`

3. Do not edit the **Device.ScanGroup Override** column (leave it blank). This leaves the DI Object and scan group assignment as defined in the **IO Devices** view.

4. Enter the fully qualified field reference in the **Object.Attribute Override** column. For example:

   `CIP_1.CIPENB_1.CIPBackPlane_1.CIPLogix5000_1.UDO_01. Attribute001`



5. Apply the override. The resulting reference will be displayed. For example:

   `OPCClient_01.SG1.CIP_1.CIPENB_1.CIPBackPlane_1. CIPLogix5000_1.UDO_01.Attribute001`

Writing and Editing Scripts

You can write scripts that run commands and logical operations based on specified criteria being met to extend and customize your objects. For example, a script can be configured to start running when a key is pressed, a window is opened, or the value of an attribute changes.

With scripts, you can create a variety of customized and automated system functions. A script adds behavior that runs when the object that contains the script is deployed and the object is either:

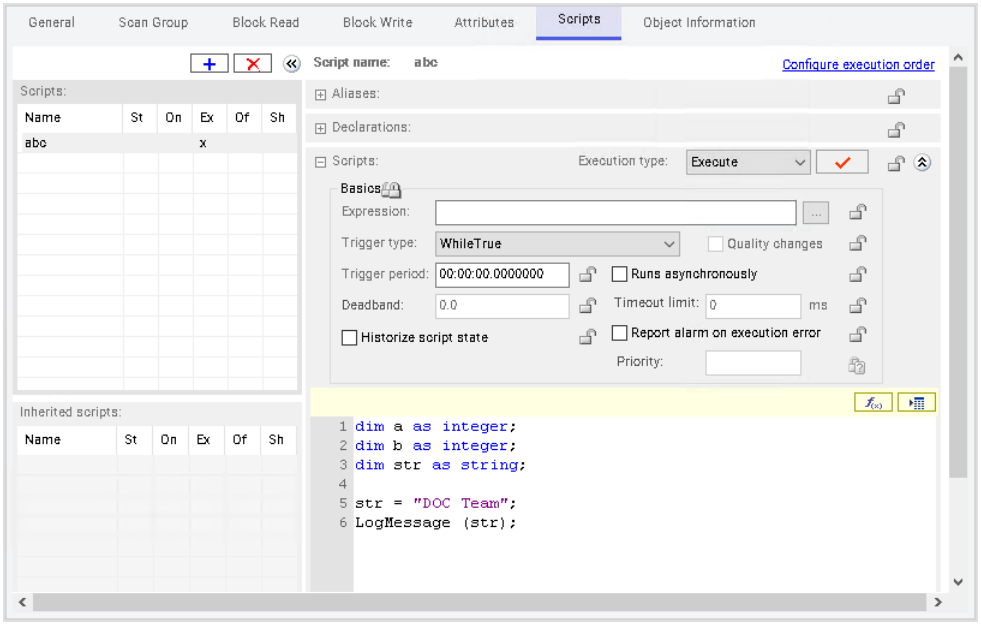
- On scan in the run-time environment or
- Changes scan or start/shutdown state

A script typically runs based on attributes of the object that contains it, but can be started by another script based on changing values of attributes of more than one object.

When a script condition is true, the script runs at least once immediately. The maximum length of a script trigger period is 49-days. A script never runs if the trigger period exceeds 49-days.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For specific information about writing scripts, including the scripting language, syntax, commands, and using .NET, see the *Application Server Scripting Guide*. For more information about the scripts page, see About the Scripts Page.



Important: You cannot pass attributes as parameters for system objects. Instead, use a local variable as an intermediary or explicitly convert the attribute to a string using an appropriate function call when calling the system object.

About Scripts

The following characteristics apply to the scripting environment:

- Script text has no length limitations.
- Selecting a script function from the **Script Function Library** dialog box adds it and its syntax to the script text where you can edit it.
- You can save a script with syntax errors, but the object cannot be deployed until you correct the script syntax errors.
- You can validate your scripts before using them. This helps you avoid syntactically correct but semantically incorrect combinations such as two statements declaring the same variable. Variables can be declared only one time in a single block.
- You can change the name of a script at any time by renaming it in the Object Editor.
- In the run-time environment, a script execution error stops the script's current execution. Script execution is retried on the next AppEngine scan.

Script Execution

The existence and execution order of scripts associated with an object are inherently locked at each stage of development in the template, derived template, and instance. For example, a set of scripts associated with a template are treated as an ordered block in the **Configure Execution Order** dialog box when configuring execution order in a next-generation derived template.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

New scripts in the derived template can be run in any order before and after the template's block of scripts. The derived template's execution order is treated as a block in any downstream derived templates or instances. Scripts cannot trigger any faster than the scan period of the AppEngine the script is associated with or faster than the scan period of the AppEngine that hosts the object that the script is associated with.

Scripts run in one of two modes:

- Synchronous scripting mode is the default for running scripts in the run-time environment. This mode runs scripts in order while an object is running on scan.
- Asynchronous scripting mode is a group of scripts running on the same, lower priority execution thread. These scripts only support Execute triggering and run independently from each other. Set the maximum number of independent threads in the AppEngine configuration editor. To use either scripting mode, you must select **Execute** as the **Execution Type** in the **Scripts** area on the **Scripts** page.

To create a script and associate it with an object

1. Add a script. On the **Scripts** page of the Object Editor, click the **Add** (**+**) button. A script is added to the **Scripts Name** list.
2. Type a name for the script and press Enter. Script names can be up to 32 alphanumeric characters, including periods. At least one character must be a letter.

Note: For detailed information about each item on the **Scripts** page, see About the Scripts Page.

1. Select a trigger that starts the script in the run-time environment.

   Execution Type triggers include: Startup, On Scan, Execute, Off Scan and Shutdown.

   - If you select **Startup**, **On Scan**, **Off Scan**, or **Shutdown**, the **Basics** group is unavailable. The script is triggered when the object starts, goes on scan, goes off scan, or shuts down.

     If you select **Execute**, the **Basics** group is available.

     If you selected **Execute** as the script trigger, select a **Trigger Type**. Depending on the type selected, you are required to enter an **Expression** and/or **Trigger Period** and **Deadband** values. When the combination of **Expression**, **Trigger Type**, **Trigger Period** and/or **Deadband** is satisfied in the run-time environment, the script starts running. See the following table for more information.

     The **Trigger Period** format is as follows:
     Hours:Minutes:Seconds:Milliseconds

     For example, 3 hours, 5 minutes, and 10.5 seconds is:
     03:05:10.5000000

     > Expressions are limited to one language statement in length and calling only synchronous mode script functions. Avoid using script functions with side effects in expressions because subtle behaviors can occur.

Trigger Type	Description
Periodic	When the object containing the script is going On Scan, a Periodic script evaluates its expression at the next scheduled scan period of the AppEngine. The script then runs periodically at the trigger interval specified in the **Trigger Period** box. A time interval of zero (0) starts the script during every scan. This trigger does not require an expression.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Trigger Type	Description
While True	When the object containing the script is going On Scan, a While True script evaluates its expression at the next scheduled scan period of the AppEngine. The script runs if true and then periodically thereafter at the trigger interval. The script continues running as long as the **Expression** value evaluates to true. A Trigger Period is required. Zero (0) evaluates the expression at the AppEngine scan period and non-zero means the expression is evaluated at the specified time interval.
On True	When the object containing the script is going On Scan, an On True script evaluates its expression at the next scheduled scan period. The script starts at the transition between the expression going from false to true.
On False	When the object containing the script is going On Scan, an On False script evaluates its expression at the next scheduled scan period. The script starts at the transition between the expression going from true to false.
Data Change	Scripts run when the value or quality of the expression changes. The expression must evaluate to a single, non-arrayed value of the following types: integer, real, time, elapsedtime, string, double, Boolean, custom enumeration and quality. To allow execution based on quality, select the **Quality changes** check box. A deadband can be specified for all data types. Deadband units for time and elapsedtime types are milliseconds. Deadband is always ignored for strings because any change (even from "ABC" to "abc") is considered a change. Only major changes in quality (from Good/Uncertain to Bad/Initializing or vice versa) are considered changes. After the object is put on scan, Data Change-triggered scripts start running at the AppEngine's next scan period and then at each subsequent scan period in which the value or quality changes.
While False	When the object containing the script is going On Scan, a While False script evaluates its expression at the next scheduled scan period, runs if false, and then periodically thereafter at the trigger interval. The script runs as long as the **Expression** value evaluates to be false. A Trigger Period is required. Zero (0) evaluates the expression at the AppEngine scan period and non-zero means the expression is evaluated at the specified time interval.

2. Select one or more of the following:
   - Set the **Runs Asynchronously** and associated **Timeout Limit** parameters, as needed. The Timeout Limit is set in milliseconds. Note that if a script exceeds the Timeout Limit, it continues to execute all sequential statements, but generates a warning in the logger when it completes. If the script contains a loop, it will exit the loop and complete execution of the script, when the timeout limit is reached.
   - Select **Report Alarm on Execution Error** and set a **Priority** for the alarm if you want the alarming function to alert you if a script execution failure occurs.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Select **Historize Script State** to store the state of the script in your application's Historian.

3. In the **Declarations** area, type variable declarations about the script you are writing.
4. Set aliases for the reference strings in the **Aliases** area. This can simplify the script code and allows script code to be created and locked at a template level using alias names. When an individual instance of that template is created, you can link external attributes to the alias names.

   In the **Aliases** area, click the **Add** button to add a new alias. An alias is added to the list. The name is shown in edit mode. Double-click the **Reference** entry, and enter a reference string for the alias. You can also click the **Browse** button at the end of the **Reference** block to open the Attribute Browser for easy selection of an object's attributes.
5. Write the script in the **Script Creation** box. Use the **Display Script Function Browser** and **Display Attribute Browser** buttons to help you insert script functions and object attribute references in your script. For help with the specific commands and syntax, see "QuickScript .NET Functions" in the *Application Server Scripting Guide*.

   Click the **Validate Script** button to validate if your script contains any syntax errors.
6. Order the scripts. If you have more than one script associated with a single object, click **Configure Execution Order**. Ordering does not apply to asynchronous scripts. If a script is added to an instance derived from a template that contains scripts, the new script automatically defaults to running after the derived scripts.
7. When you are done creating your script and setting its execution triggering parameters, save and close the Object Editor.

Script Locking and Change Propagation

When you lock a script in a template, the following rules apply:

- The name of a script and its existence is implicitly always locked. This means:
  - You cannot delete the script in derived objects.
  - You cannot change the name of the script in derived objects.
  - If you rename the script in the template, the name changes in all derived objects.
  - If you delete a script from the template, the script is deleted from all derived objects.
  - If you add a script to the template, the script is added to all derived objects.
  - You can add scripts to derived objects. Adding scripts to derived objects does not affect parent object scripts.
- You can lock or unlock the script text in a template. There is script text for **Declarations, Execute, Startup, Shutdown, On Scan** and **Off Scan**.
- You cannot separately lock a script in the script editor. A single group lock is used to lock or unlock all scripts at once. Once a script is locked, derived templates and instances cannot modify any of the script text.
- You can lock alias names and alias references separately.
  - When you create aliases for a script, an AliasReference attribute is created for each script with aliases. To lock alias references, you must lock the AliasReference attribute.
  - Changes to alias names only propagate to derived objects when aliases are locked in the **Scripts** page.
  - Changes to alias references only propagate to derived objects when the AliasReference attribute is locked in the **Attributes** page.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

    - If the AliasReference attribute is not locked, the alias references will be editable in derived objects, but changes made in the parent template will not propagate to derived objects.
- The script description, runs asynchronous flag, expression, trigger type, trigger period, deadband and execution error alarm are individually lockable and can be locked separately from the script text. A group lock is provided for this group of attributes.
- When you add a script to a template, all properties of the script are editable.
- When you add a script to an instance, all properties of the script are editable, except the lock properties (locks do not serve any purpose in instances since you cannot derive child objects from them).

  Important: An expression typically uses attribute references. If you lock the expression and the associated script in a template, use aliases in both the expression and the script. This allows you to specify the attributes that the aliases point to on a per-instance basis while the script code is locked.

The following rules apply to the derivation behavior of locked script attributes:

- If an attribute is locked in a template, then all templates and instances derived from the template share the copy of the value of the locked attribute. A change to the value is only allowed in the template that locked it. The change propagates to all derived templates and instances.

  For scripts, locking an attribute of the script, such as its script text or execution type, in a template means all derived templates and instances point to that locked attribute. Future changes to that locked attribute's value, such as modifying the script text, propagate and appear in all derived templates and instances.

  If instances are deployed, they are marked pending update status. After they are redeployed, the change to the locked attribute in the template exists in the deployed instance.
- If an attribute is not locked in a template, then all templates and instances derived from that template receive their own copy of the value of that unlocked attribute. A change to that unlocked value is allowed in derived templates and instances because they own their own copy. Any change to the unlocked attribute value in the template does not propagate to any derived template or instance.

  An unlocked attribute in a script (such as expression or script order) in a template means that all derived templates and instances have their own copy and the value of that unlocked attribute can change. Future changes to that locked attribute's value (for example, modified expression) in the template does not propagate to any derived template or instance. If instances are deployed, their status does not change to pending update. Redeploying them does not cause the value to change in the deployed instance.

Creating and Working with Content

The **Content** pane of the **Attributes** page lets you:

- Add, modify, rename or delete object-owned graphics. Object-owned means that the graphic exists only for that particular object. Note that object-owned graphics are propagated to child objects.
- Link the object to content (graphics, layouts, External Content) in the Visualization folder. This is shared content, that can be linked to multiple objects. Links to content are propagated to child objects.
- View content inherited from parent templates. However, you cannot add, modify, or delete inherited content. To edit an inherited content item, you must check out and edit the derived template that owns the content.

You must have the derived template or object instance checked out in order to add, modify, or delete content associated with the object; otherwise, the content is read-only and you can only view it.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

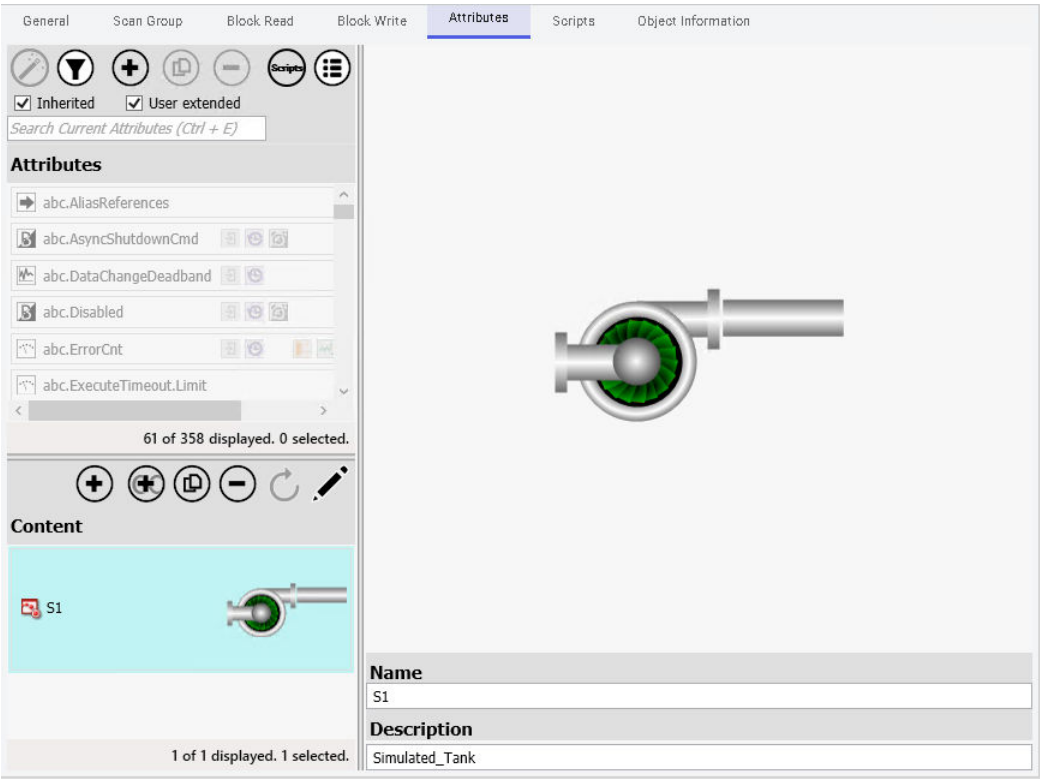
You can only add content to derived templates or instances, not to a base template.

To add or link to content:

Linked content can be a graphic, an external content item, or a layout.

- To link to a content item in the Visualization folder to the object, select the **Link Content** button. This opens the **Galaxy Browser**. Then, select the graphic, layout, or External Content item to be linked to the object. Linked content can be shared by multiple objects. Relative references within the linked content will resolve correctly.
- To create an **object-owned graphic**, select the **Add Content** button.
- To edit an owned-graphic, select the graphic and click the **Edit** button.

Note: If you edit a linked graphic, layout or other shared content, the changes will appear in all objects that link to that content.



Adding Graphics Items

To add a graphic to the object

This procedure adds an owned graphic to the object. To link a graphic or external content item, see Link to shared content in the visualization folder.

1. From the **Content** pane in the **Attributes** page of the Object Editor, click the **Add** (+) button. A new graphic name and graphic icon are added to the pane.
2. Select the new graphic in the **Content** pane.
3. In the **Name** field, edit the new local graphic name.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

4. In the **Description** box, type a description for the graphic being added.
5. Click the **Edit** icon. The Graphic Editor opens. For instructions on using the Graphic Editor, see "About Creating and Managing Industrial Graphics" in the *Creating and Managing Industrial Graphics User Guide*.

Errors and Warnings in an Owned Graphic

When you save an owned graphic that you have edited in the **Graphics Editor**, the owning object is also saved. If there are any errors and warnings, they will be shown in the **Save Confirmation** dialog. However, the errors and warnings include not just the graphic you are saving, but because the owning object is also being saved, the dialog lists all errors and warning related to the owning object and other graphics owned by the object.

To determine if the warning or error is related to the graphic you are saving, or if it is related to the owning object (or other graphic owned by the object), check the prefix in the warning/error message. The first (left-most) entry in each error/warning message lists the graphic or object name.

Modifying Graphics

All modifications made to graphics referenced by other objects are visible in the referenced objects.

To modify a graphic

1. On the **Attributes** page of the Object Editor, in the **Content** area, select the graphic to modify.
2. Click the **Edit Content** icon. The Graphic Editor opens, showing the selected graphic.
3. Make changes. See "About Creating and Managing Industrial Graphics" in the *Creating and Managing Industrial Graphics User Guide*.

Set Graphics Properties with a Galaxy Style Library

To view and manage the settings for the Galaxy Style Library, select **Configure** on the the **Galaxy** menu. A Galaxy Style Library lets you set the appearance and behaviors of different graphics properties within a ViewApp at run time.

- Use Element Styles to Modify Graphics — define graphics properties such as font size, font color, fill color, line weight, and line color.
- Use Quality and Status Indicators to Provide Run-Time Feedback — provide visual cues about different quality and status states.
- Use UI Themes to Set Appearance of WPF Controls — define graphics properties for WPF and navigation controls used in ViewApps.

Use Element Styles to Modify Graphics

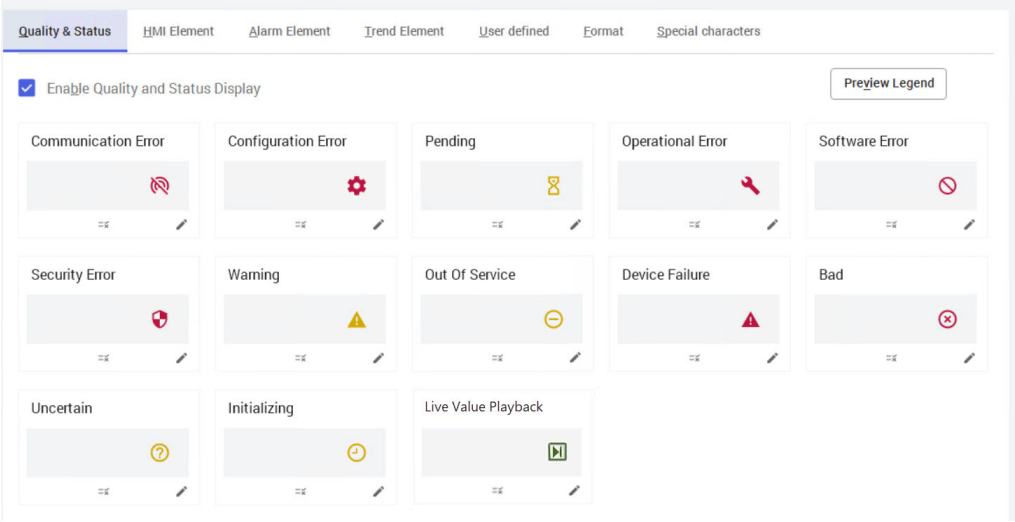
Element Styles define one or more visual properties such as fill, line, text, blink, outline, and status properties of graphic elements. You can apply an Element Style to a graphic element to set preconfigured visual properties defined in that Element Style. Element Styles establish consistent visual standards for graphics.

For more information about Element Styles, see Understanding Element Styles.

Use Quality and Status Indicators to Provide Run-Time Feedback

In the same way that Element Styles can be used to preconfigure consistent visual properties of graphic elements, Quality and Status (QS) indicators can be preconfigured to provide visual elements specific to quality and status states at run time. Only the highest priority QS indicator is shown at run time. Therefore, if two or more indicators are active, only the highest-priority indicator is shown. For example, if there are both operational and software errors, only the operational error is shown.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The indicators are listed in order of priority, with the highest priority indicator listed by row from left to right, and then top to bottom. You can configure text, fill, line, outline, blink, and associated display icons for the following QS indicators:

- Communication Error
- Configuration Error
- Pending
- Operational Error
- Software Error
- Security Error
- Warning
- Out of Service
- Device Failure
- Bad
- Uncertain
- Initiating
- Live Value Playback (used for the AVEVA OMI Historical Playback feature)

Use UI Themes to Set Appearance of WPF Controls

UI Themes define the color settings that are applied to WPF controls used in ViewApps. In order to set the colors for a WPF control, the control must comply with the Material Design Spec. See the MSDN WPF Community Projects website for additional information.

The UI Theme includes the following configurable parameters:

- **Theme** (light or dark): default – **light**
- **Background** color
  - Default **light** background color: grey, lightest setting

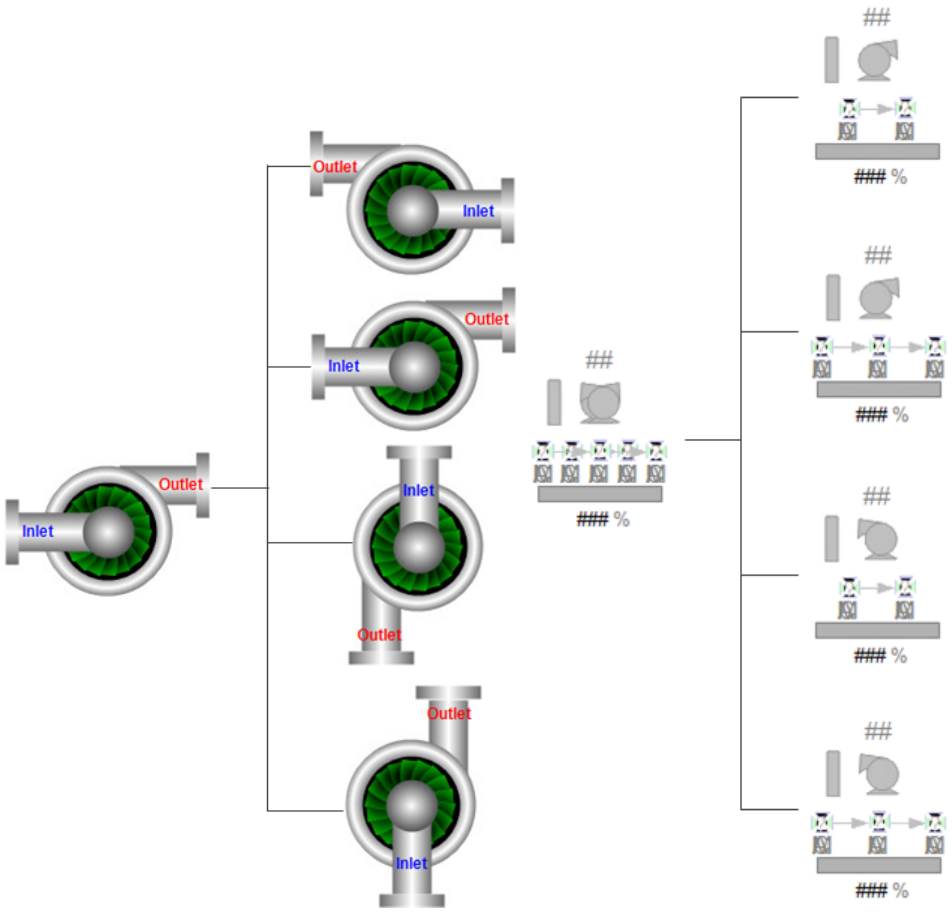
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

  - Default **dark** background color: blue-grey, second darkest setting
- **Primary** color – default **teal**
  - **Low.** This setting is only used by WPF controls that explicitly include the Primary Low property. The **low** setting is always darker than the **mid** setting. Default is the second darkest setting (second from left).
  - **Mid**. This setting is used by all WPF controls that use the Material Design Spec. Default is the middle setting.
  - **High**. This setting is only used by WPF controls that explicitly include the Primary High property. The **high** setting is always lighter than the **mid** setting. Default is the second lightest setting (second from right)
- **Accent** color – default **purple** (darkest setting)

Using Symbol Wizards

The Industrial Graphics Editor includes the Symbol Wizard Editor, which can be used to create Symbol Wizards, which contain multiple visual or functional configurations. A Symbol Wizard can be embedded into managed InTouch applications like standard Industrial Graphics.

A Symbol Wizard's configuration is selected to meet the requirements of an application. Incorporating multiple configurations in a single Symbol Wizard reduces the number of graphics needed to develop an AVEVA application.



The Symbol Wizard Editor can create Symbol Wizards from traditional Industrial Graphics and Situational Awareness Library graphics. Both types of graphics are located in the Visualization folder in separate folders.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Symbol Wizards are saved in the IDE's Visualization folder and are not associated with any specific AVEVA object template or object instance. Except for the ability to select a specific graphic configuration, Symbol Wizards behave like standard Industrial Graphics.

Situational Awareness Library graphics provide an additional benefit of including defined properties and their associated attributes to more easily create configurations. Some Situational Awareness Library graphics include a Type property to assign a specific function for a configuration. For example, a meter graphic can be configured to represent a thermometer, a pressure meter, or flow meter by simply changing the attribute assigned to the Type property.

Typically, the process of creating and embedding a Symbol Wizard in an application requires the involvement of a Designer and a Consumer. A Designer creates Symbol Wizards using the Symbol Wizard Editor. A Consumer selects a configuration of a Symbol Wizard and embeds the instance of the graphic into managed InTouch applications.

Creating Symbol Wizards

A Designer uses the Symbol Wizard Editor to define the various required graphic configurations based on a set of rules and graphic layers. A Designer defines a set of layers, which are used to group a set of graphic elements, custom properties, and named scripts. Graphic elements and other properties can be assigned to no layers or multiple layers. Graphic elements that are not assigned to any layer always appear in all configurations.

A Designer can create a rule for each layer that defines the conditions when the layer is included in a graphic configuration. Rules are assigned with Choice Groups, Choices, and Options. A Designer selects a configuration to be the graphic default that appears when the graphic is embedded in a managed InTouch application.

After creating all configurations, the Designer verifies each configuration of a graphic using the Symbol Wizard Preview. Designers set values in the **Wizard Options** view to verify that each configuration appears as designed based on the layer rules set for the graphic.

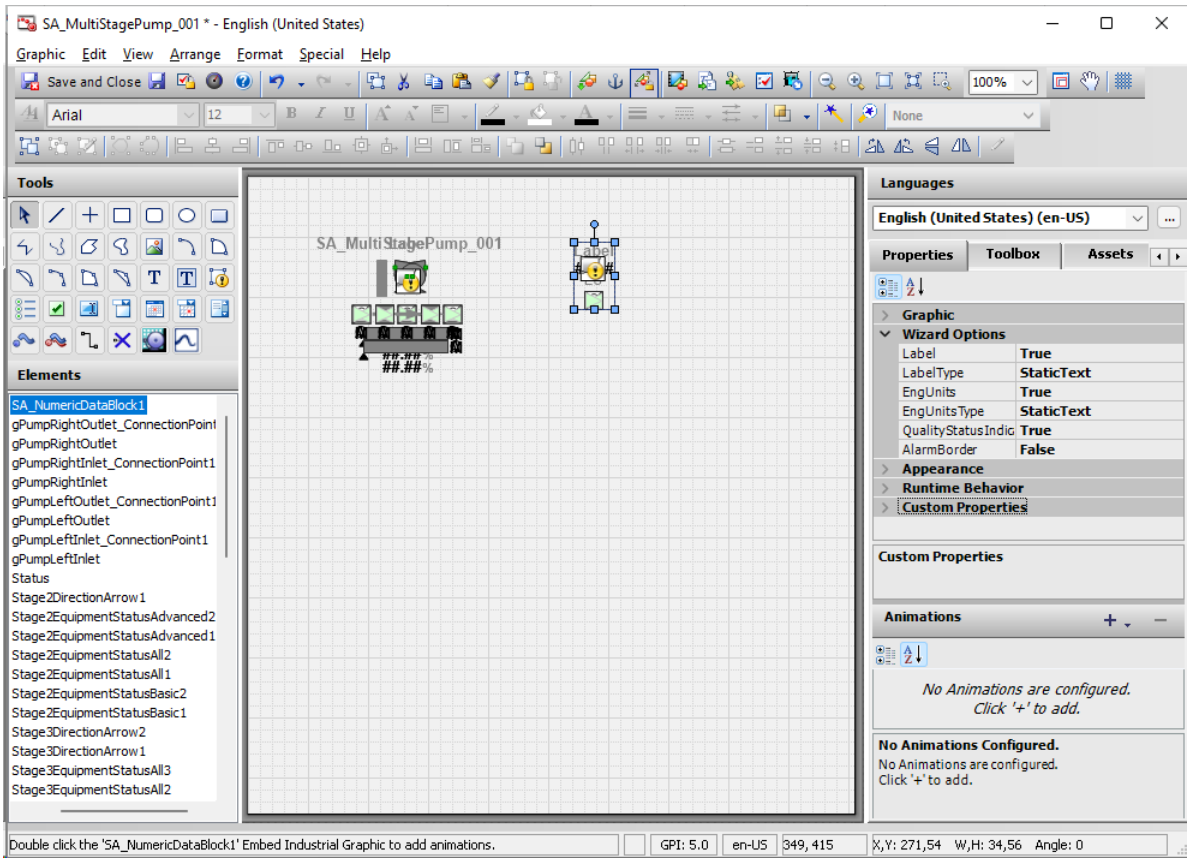
When a Symbol Wizard is finished, the Designer saves it to the Galaxy library so that it is available for use in managed InTouch applications.

For more information about creating Symbol Wizards, see Designing a Symbol Wizard in the Industrial Graphics Editor help.

Embedding Symbol Wizards into an Application

Symbol Wizards are stored in a Galaxy library just like standard Industrial Graphics. When a Consumer selects a Symbol Wizard and embeds it into a managed InTouch application, the Symbol Wizard's default configuration is selected. The Consumer can change the graphic's configuration by changing the options from the Symbol Wizard's **Wizard Options** section of the **Properties** view. Depending on the selected configuration, there can be additional configuration-related properties that can be selected by the consumer.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



After selecting a graphic configuration and changing any properties, the Consumer saves the Symbol Wizard so that it can be embedded into a window from WindowMaker.

While the InTouch application is running, the Symbol Wizard appears as the configuration selected by the Consumer. A Symbol Wizard configuration cannot be changed during run-time.

For more information about how to embed Symbol Wizards into managed InTouch applications, see Using Symbol Wizards in an Application in the Industrial Graphics Editor help.

Renaming Graphics

You can rename a graphic.

To rename a graphic

1. On the **Attributes** page of the Object Editor, in the **Content** area, select the graphic to rename.
2. In the **Name** field, type the new name and press **Enter**. The new name is saved.

Deleting Graphics

Caution: Deleting a graphic with embedded references breaks the links to their related objects. "Symbol Not Found" appears when you open objects whose embedded graphics have been deleted.

To delete a graphic

1. On the **Attributes** page of the Object Editor, in the **Content** area, select the graphic to delete.
2. Click the **Delete** (-) icon. The **Confirm Delete** dialog box appears. The left pane lists the graphic selected for deletion. The right pane shows all embedded references to the selected graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3. Click **Delete**.

Rearrange objects

By default, a running ViewApp shows objects as navigation items that appear in alphabetic order by their assigned Galaxy names in the navigation model. An alphabetic order of navigation items may not be appropriate for operators. Objects can be rearranged to more closely resemble the functional organization of different components in a production process.

You can rearrange objects by dragging them to the desired location in the **Model** view of the System Platform IDE. After objects are moved, the **Assets** listings of objects in the ViewApp and Layout editors update to show the new arrangement.

Rearrange objects: drag and drop

You drag and drop an object with your mouse to move an object to a different location within the ViewApp navigation model. You can rearrange objects from the **Model** view of the System Platform IDE.

To rearrange objects from the System Platform IDE

1. Open the System Platform IDE and show the **Model** view.
2. Make sure that all objects shown in the **Model** view are undeployed.
3. Select the object to be moved within the **Model** view.
4. Drag the object to the desired location within the hierarchy of objects in the **Model** view.

   A horizontal line shows the hierarchical level within the navigation model where the object will be placed.

   The horizontal line does not appear when you attempt to make an invalid move.

About rearranged objects

Rearranged objects maintain their location within a list of objects for the following Galaxy operations:

- Export and import of a Galaxy
- Back up and restore a Galaxy
- GRLoad and GRDump operations

Also, objects can be programmatically moved to another location within a list of objects using the GRAccess ToolKit.

Rearranged Objects in the Export and Import of a Galaxy

When rearranged objects are exported to an aaPKG file, rearranged objects are retained when the aaPKG file is imported into another Galaxy. If the imported Galaxy includes rearranged objects, but the existing Galaxy does not, the object order of the imported Galaxy takes precedence.

Rearranged Objects in Backup and Restore of a Galaxy

When rearranged objects are backed up to a CAB file, rearranged objects are retained when the CAB file is restored to another Galaxy.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Rearranged Objects in a GRLoad and GRDump

When one or more objects are selected and exported through GRDump, then GRLoad can be used to import the contents of GRDump into another Galaxy. Objects retain their rearranged locations in the GRLoad import operation in the new Galaxy.

Rearrange an Object Programmatically using GRAccess Toolkit

You can use the GRAccess object model to write programs that automate the configuration of local and remote Galaxies. You can move an object in the Model view hierarchy using the SortOrder property that programmatically moves an object after a specified predecessor object in the list of objects shown in the **Model** view.

Class
IgObject

Syntax
[C#]
```
string SortOrder { set; get; };
```

[Visual C++]
```
[propget]
HRESULT SortOrder(
[out, retval] BSTR* thePredecessorName
);
[propput]
HRESULT SortOrder(
[in] BSTR newPredecessorName
);
```

Parameters

thePredecessorName
The current predecessor of this object.

newPredecessorName
The new predecessor of this object. The object does not need to be checked out.

Remarks

SortOrder sets or returns the preceding object name in the navigation hierarchy. When the sort order is set, the tagname of the asset is displayed in the ViewApp navigation hierarchy just below the predecessor.

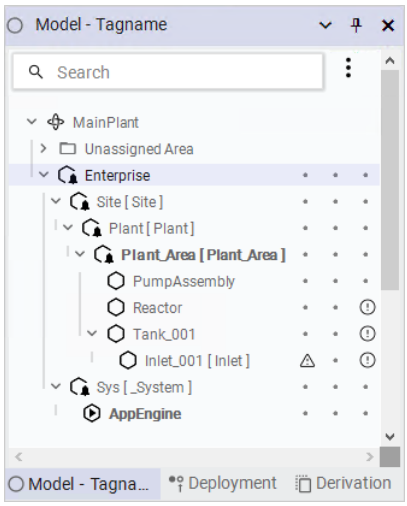
Assign alias name to an object

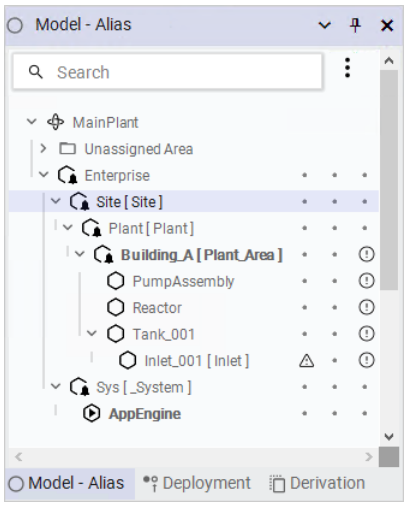
A running ViewApp can show objects as navigation items that appear by their assigned object names in the navigation model. Object names may not be intuitive and easily recognizable to operators. Objects can be assigned more understandable alias names.

Alias names are assigned from the **Model** view of the System Platform IDE. You can toggle the **Model** view to show the objects by their tagname or alias names. In the example below, the Plant_Area area object has been assigned a more specific Building_A alias name.

Objects by Tagname	Objects by Alias Names

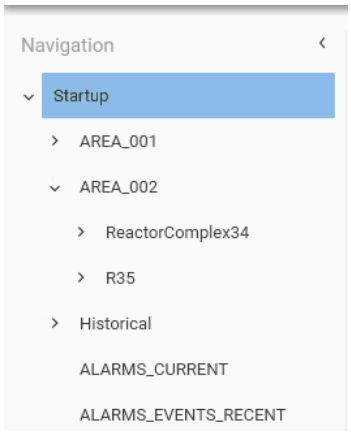
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





Alias names can be shown from the ViewApp and Layout editors but cannot be modified. Also, if an alias was assigned to an object and the ViewApp or Layout editors were open, the editors must be closed and reopened to show the new alias name.

An alias name appears in the navigation model of the ViewApp during runtime or in preview mode.



Alias names are associated with navigation attributes that have the word Title in their names. For example, the CurrentTitle attribute returns the alias name of the current navigation item in a running ViewApp. You can write scripts using alias names to change the current focus of a ViewApp. For more information about navigation attributes that reference alias names, see About ViewApp navigation.

Also, alias names can be hidden from the navigation model of a running ViewApp or in preview mode by placing an underscore (_) as the first character of the alias name. Hiding an alias name is useful for navigation items like a system node in the navigation model that does not provide any useful ViewApp information to an operator. For more information about configuring the visibility of an alias name, see Hide assets in ViewApp navigation during runtime.

Assign alias name to an object in Model view

You can assign an alias name to a derived instance of an object that appears in the **Model** view of the System Platform IDE. The System Platform IDE includes a **Show Alias** option that enables you to toggle the **Model** view to show the original tagname and alias names of objects.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

An alias name must adhere to the following rules:

- An alias name can be up to 128 characters in length.
- An alias name can consist of alphanumeric characters and special keyboard characters (!,@,#, $,%,^,&,*, (,),-,_,=,+).
- An alias name can start with any alphanumeric or special character.
- An alias name can contain spaces.
- Multiple objects can be assigned the same alias name in a Galaxy.

Note: You cannot assign an alias name to an object if it is deployed, or if it is checked out for editing.

To assign an alias name to an object

1. Open the **Model** view of the System Platform IDE to show the list of instances.
2. Select the instance to assign an alias name to
3. On the **Home** ribbon, in the **Edit** area, select **Alias name**.

   The **Rename Alias** dialog box appears with a field to assign a new Alias name.
4. Enter an alias name that meets the format rules and select **Save**.

   The object should appear in the **Model** view with its assigned alias name.
5. If you want to verify that the alias name was assigned correctly, switch the **Model** view to show alias names: on the **View** ribbon, select **Alias**. This toggles the **Model** view between tag names and alias names.

More Information About Alias Names

Objects retain their aliases when objects are exported, backed up, and loaded as part of a Galaxy operation. Also, object aliases can be programmatically set and returned using the GRAccess ToolKit.

Alias Retention in Galaxy Operations

Alias names assigned to Galaxy objects are retained for the following Galaxy operations:

- Export and import of a Galaxy
- Back up and restore a Galaxy
- Migrate a Galaxy
- Exporting instances to and importing them from a CSV file

Alias Names in the Export and Import of a Galaxy

When objects assigned alias names are exported to an aaPKG file, object Alias names are retained when the aaPKG file is imported into another Galaxy with the following exception:

AliasName is a reserved keyword in System Platform 2020 R2 and newer versions. If an import package contains previous versions of objects with name “AliasName”, the import process skips the attribute and logs a warning message in the OCMC log. Attributes with name AliasName are not imported.

Alias Names in Backup and Restore of a Galaxy

When objects assigned alias names are backed up to a CAB file, object alias names are retained when the CAB file is restored to another Galaxy.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Migrate a Galaxy

When objects assigned alias names are migrated, object Alias names are retained when migrated to another Galaxy with the following exception:

AliasName is a reserved keyword in System Platform 2020 R2 and newer versions. If a migration package contains previous versions of objects with name “AliasName”, the migration process skips the attribute and logs a warning message in the OCMC log. Attributes with name AliasName are not migrated.

Alias Names when Exporing to and Importing from a CSV File

When one or more objects are selected and exported to a CSV (Comma-Separated Value) file, you can import the contents of the CSV file into another Galaxy. Objects retain their alias names when imported to the new Galaxy.

Assign an Alias Name Programmatically using GRAccess Toolkit

You can use the GRAccess object model to write programs that automate the configuration of local and remote Galaxies. You can rename object aliases using the Alias property that programmatically gets or sets the alias name of an object.

Class
IgObject

Syntax
[C#]
```
string AliasName { set; get; };
```

[Visual C++]
```
[propget]
HRESULT Alias(
[out, retval] BSTR* theAlias
);
[propput]
HRESULT Alias(
[in] BSTR newAlias
);
```

Parameters

theAlias
The current alias of this object.

newAlias
The new alias of this object, or blank if the object does not have an alias. The object does not need to be checked out. The alias rename operation is immediate.

Object Wizards

Adding an Object Wizard to a template provides a streamlined method that lets users quickly and easily derive and configure assets and their associated graphics from the template.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Create an Object Wizard	About Object Wizards	Object wizard components	Conditional visibility expressions for choice groups, choices, and options
Add Content to an Object Wizard	Add content for use with an object wizard	Attribute and graphic overrides	Trimming
Derive a Template	Object wizard derivation	Object wizard test mode	Propagation of object wizard changes to derived objects
Configure an Instance	Configure instances	Create a new instance graphically	

About Object Wizards

Object Wizards consist of a series of user-selectable choices and options that are used to customize a deployable instance. Each choice and option may have one or more attributes, graphics, other content types and/or scripts associated with it.

An Object Wizard can be added to any derived template and provides a simplified user interface for configuring instances (assets) from the template. To access the Object Wizard editor:

1. Open a template in the **Object Editor.**
2. If the template does not open to the **Attributes** page by default, select it.
3. If necessary, select the **Configure Wizard Options** icon (1) to open the Object Wizard editor. The **Attributes** tab divides into multiple panes that are used for building the Object Wizard.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Object Wizards can be used either within the IDE, or from the **Configure New Asset** editor that also lets you configure an associated graphic as you configure the object.

Note: The **Configure New Asset** editor can only be used if 1) the template contains a graphic, and 2) your security setting allows you to edit graphics. It cannot be used with other content types, such as layouts and external content.

A single template with an Object Wizard can replace a number of derived templates for configuring a variety of similar instances. You can add an Object Wizard to any derived template.

Depending on their level of permissions, users can:

- Create and configure instances from within the IDE, and add them to the Galaxy, or modify existing instances. A user simply opens an instance and answers a series of prompts or questions. See Configure instances for additional information.
- Use the **Configure New Asset** editor to create and configure instances and a representative graphic, simply by dragging an associated (linked or embedded) graphic into another graphic, and then answering a series of prompts or questions. See Create a new instance graphically for additional information.

  Note: The drag and drop method is only valid for graphics, and not for other linked content types such as layouts and external content.

  - For more information about linking to a graphic, see Link to shared content in the visualization folder.
  - For more information about embedding a graphic, see Add a new object-owned symbol.

Important! InTouch HMI does not support directly adding assets to an InTouch HMI window if the asset is derived from an object wizard with a linked Symbol Wizard, such as the graphics in the Situational Awareness Library. **With InTouch HMI, the Symbol Wizard must be contained in an overview graphic** when deriving assets from an object wizard. In contrast, AVEVA OMI does support the the use of linked Symbol Wizards when adding assets to AVEVA OMI layouts from an object wizard.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Object wizard features

An Object Wizard contains choices and options that guide users through the process of configuring instances. You can replace many derived templates by using a single template with an Object Wizard, since the Object Wizard can contain a wide variety of configuration possibilities. Attributes and features contained in the Object Wizard that are not needed in a particular instance are removed and are not a part of the run-time object. Finally, and perhaps most importantly, an Object Wizard is easy to use and can reduce user training and system maintenance costs.

Object Wizards provide the following benefits:

- Simplified workflow for configuring instances
- Automatic propagation of changes, without having to lock templates
- Unnecessary attributes can be trimmed from instances to reduce the footprint of deployed objects
- Unused graphics can be hidden from users to reduce clutter
- Reduced system maintenance
- Reduced training requirements for plant personnel

Create a basic object wizard

An Object Wizard can include Choice Groups, each with two or more Choices, and/or Options. A Choice Group is a radio button list that lets the user select one choice. An Option contains a checkbox that a user can select or clear.

To add an Object Wizard to a template

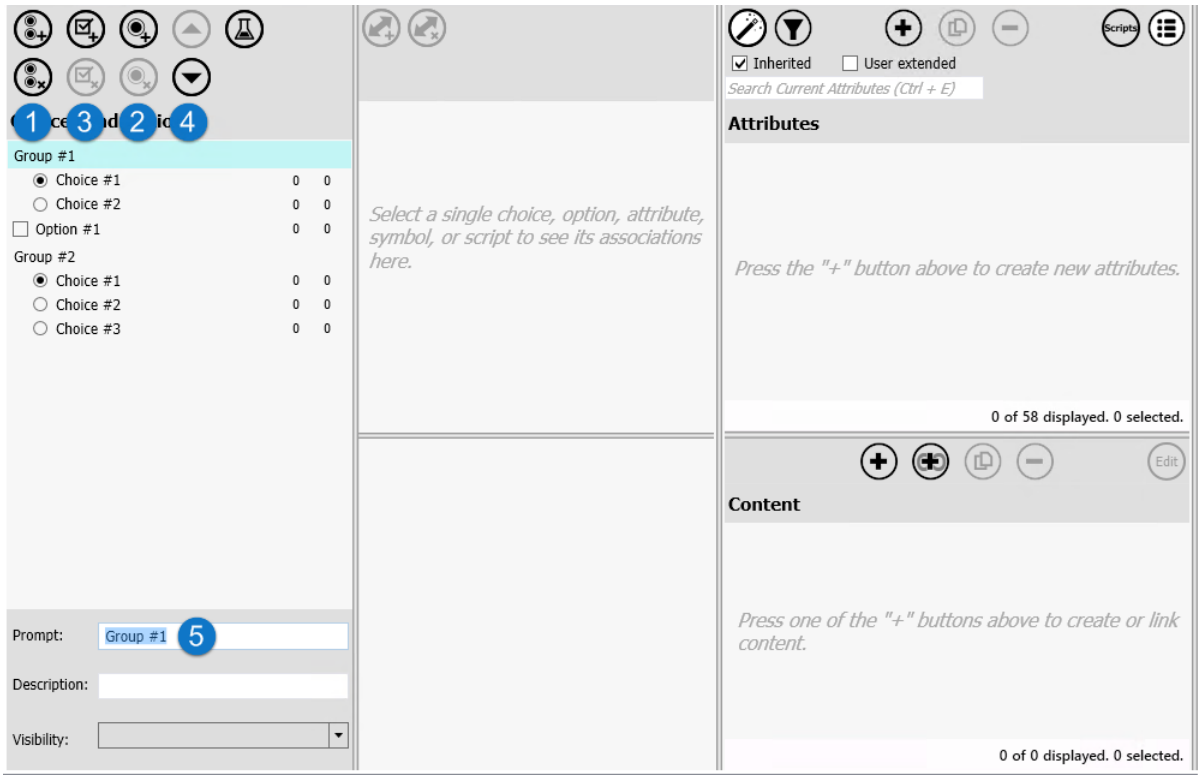
1. In the IDE, open a derived template in the Object Editor.
2. If the **Attributes** page is not already selected, select it now. See Create Derived Templates.

   Note: Only templates derived from the $UserDefined base template open in the Object Editor with the **Attributes** page selected. The Object Editor always opens the first page of the object. For example, the Object Editor opens the **General** page of the $AnalogDevice template.
3. Configure the template by adding:
   - Attributes
   - Graphics or other content such as links to documents and videos
   - Scripts

   See Add content for use with an object wizard for additional information.
4. If the Object Wizard editor is not already open, select the **Configure** W**izard Options** button.

   See Object wizard GUI panes for additional information.
5. Add a Choice Group by selecting the **Add Choice Group (1)** icon. With the Group name selected, change the name of the Group shown in the **Prompt (5)** text field. See Add a Choice Group for more information.

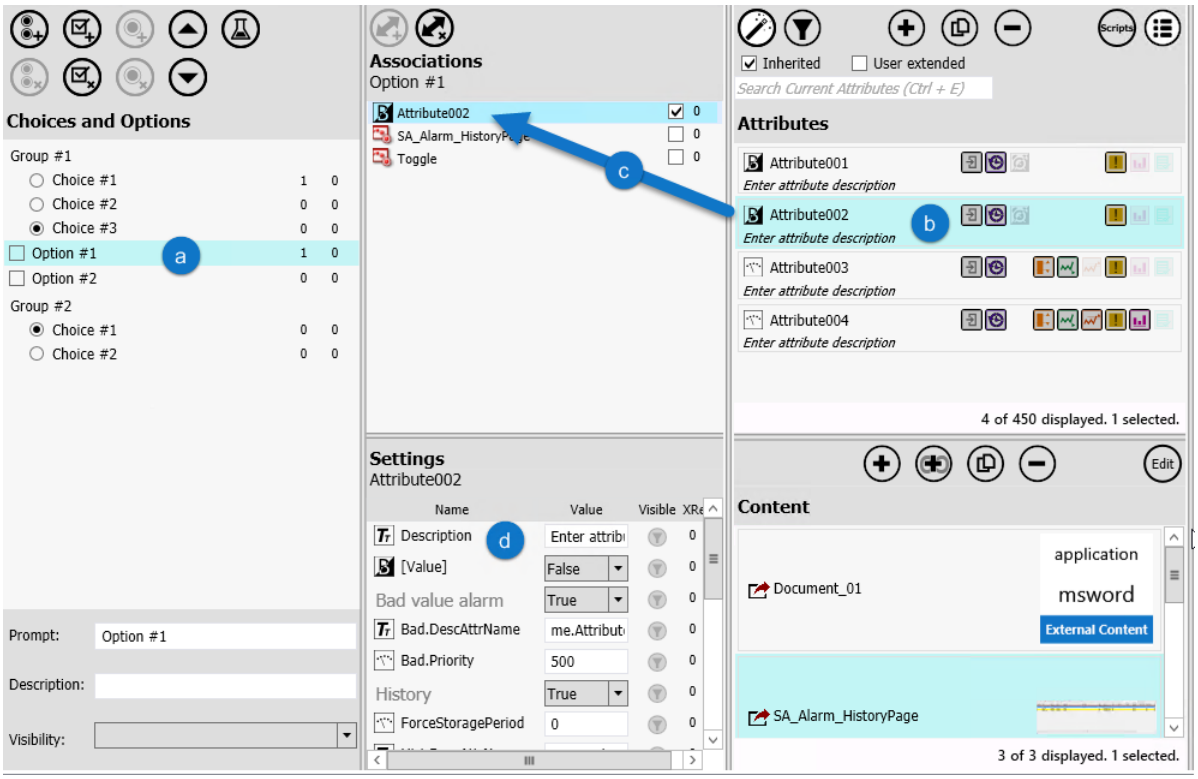
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



6. To add more Choices to a Choice Group, select the Choice Group and then select the **Add Choice (2)** icon. To change the name of a Choice, select the Choice and enter the new name in the **Prompt** field. See Add a Choice to a Choice Group for more information.
7. Add Options by selecting the **Add Option (3)** icon. Change Option names the same way that you change Choice Group and Choice names. See Add an Option for more information.
8. Use the **Up and Down arrow (4)** icons to change the order of Choice Groups, Choices (within a Choice Group), or Options.
9. To associate an attribute or content with a Choice or Option:
    a. Select the Choice or Option.
    b. Select the attribute or content (graphic or other type of content) you want to associate with the Choice or Option.
    c. Select the **Add Association** icon (or drag the attribute or content to the **Associations** pane).
    d. Settings for the selected attribute or content are shown in the **Settings** pane, once the association is made. Configure visibility and value overrides, as needed.

    Note: No settings are shown for graphics that do not contain custom properties or a Symbol Wizard.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Object wizard configuration best practices

An Object Wizard can be added to any derived template. However, to maximize the benefits of using Object Wizards, add your Object Wizard to the template as close to the top level of the derivation hierarchy as possible. To build an Object Wizard you must:

- Add and configure attributes, graphics, links to external content, and scripts before you build the Object Wizard.
- An Object Wizard requires at least one Choice Group or Option. See Object wizard components for additional information.
- Configure Choices and Options by associating attributes,graphics/content, and scripts with Choices and Options. We recommend that you limit the number of items you associate with each Choice or Option to avoid configuration mismatches.
- When configuring Choices and Options, keep the workflow as simple and direct as possible. Remember that users will be working sequentially through the Object Wizard. If you enable a feature at one level of the Wizard hierarchy, you cannot disable the same feature at a subsequent level.

  For example, if a graphic has Custom Property "X" enabled and is associated with a Choice, do not associate the same graphic with "X" disabled to an Option further down in the hierarchy. Instead, leave "X" disabled and override the setting when deriving an instance.
- Configure the Conditional visibility expressions for choice groups, choices, and options of Choices and Options in derived instances.
- Configure settings for the associated attributes and graphics. Scripts and external content do not have settings that can be configured.

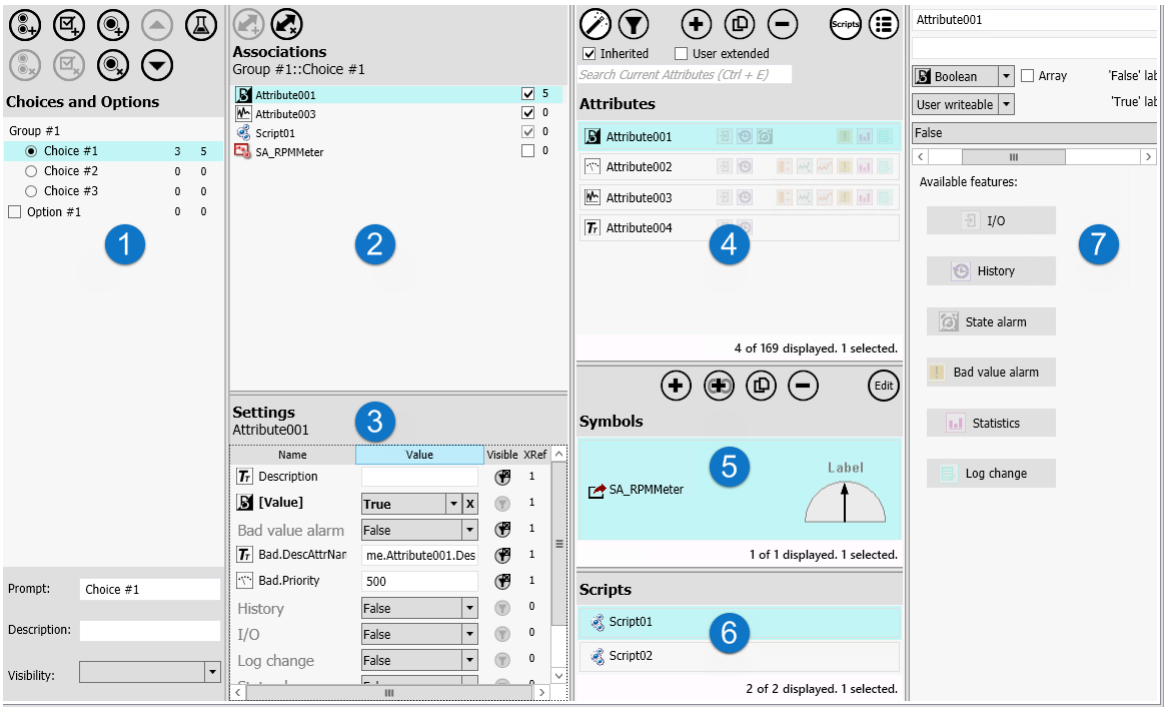
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Select whether each attribute, graphic, and other content can be Trimming from derived instances. Scripts are always configured as trimmable.

Object wizard GUI panes

The user interface for configuring Object Wizards is accessed through the **Attributes** page within the Object Editor of the Integrated Development Environment (IDE).

	Select the **Configure Wizard Options** icon to open the Object Wizard user interface. The **Attributes** page divides into multiple panes for use in creating and editing an Object Wizard. See Object wizard user interface details for additional information.
	Select the **Scripts** icon (located at the top right of the **Attributes** pane) to open the **Scripts** pane.



Ref	Name	Description
1	**Choices and Options pane**	Add, rename, reorder, and delete Choice Groups, Choices, and Options that comprise the Object Wizard in the **Choices and Options pane (1)**. You can also set the logic for displaying Choices and Options.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

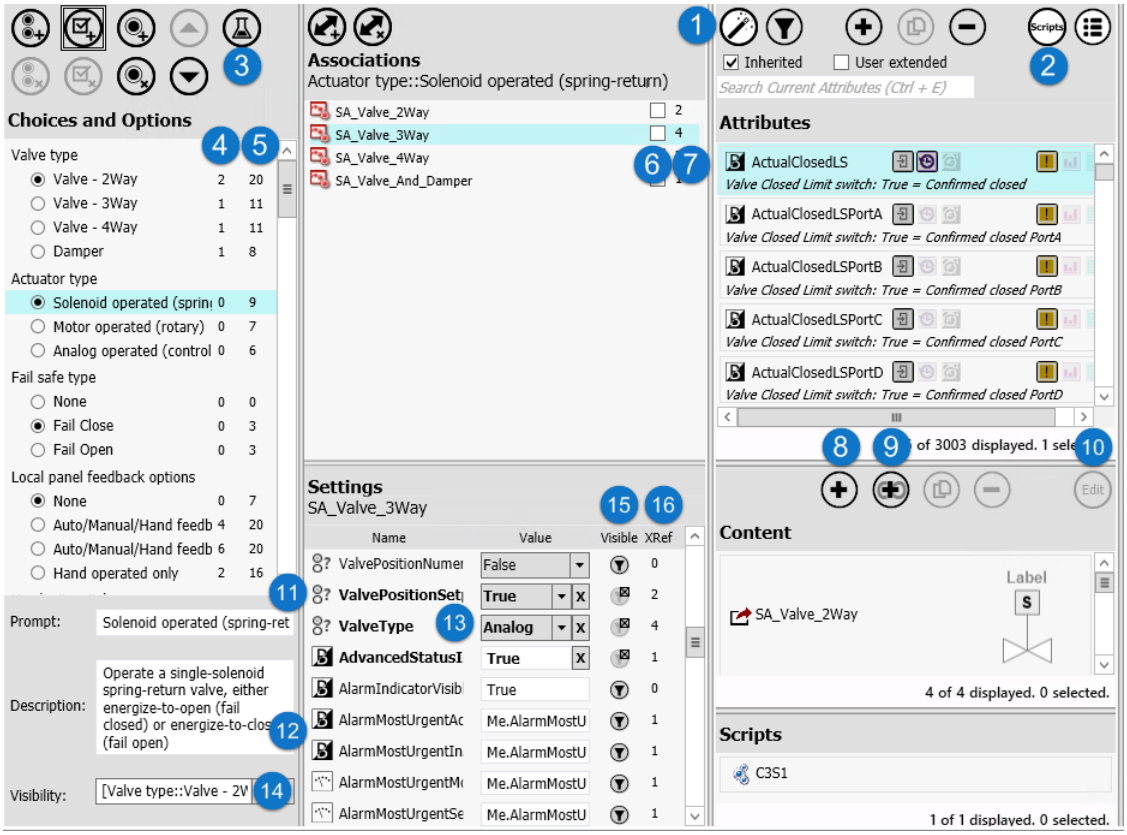
2	**Associations pane**	Drag and drop attributes, content, and scripts to the **Associations pane (2)** to associate them with a choice or option; you can use the **Add** button at the top of the pane instead of dragging and dropping. You can select multiple attributes, graphics, or scripts at once. You will also enable and disable trimming of associated attributes and graphics in this pane. See Trimming for additional information. You can view associations for individual attributes, graphics, and scripts in this pane by selecting an attribute, graphic, or script. Every association of the item is displayed. Select an association to view the item's configuration.
3	**Settings pane**	Configure attribute and graphic overrides in the **Settings pane (3)**. Overrides are set individually for each choice or option with which the attribute or graphic is associated. Default settings for attributes are set in the **Details pane (7)**. You can resize columns by selecting a column header (in this case, **Value**) and moving its edges to widen or narrow it.
4	**Attributes pane**	The **Attributes pane (4)** lists the attributes contained in the template. You can create new attributes in this area. To associate attributes with the Object Wizard, highlight a Choice or Option, and then select and drag one or more user-created attributes from the **Attributes pane** to the **Associations pane (2)**. See Associate Attributes with Choices and Options for additional information. Default configuration settings for a selected attribute are displayed to the right of this pane, in the **Details pane (7)**. See About the Attributes Page for additional information. Configured settings for the selected attribute that apply to a selected Object Wizard Choice or Option are shown in the **Settings pane (3)**. You can also enable attribute features (historization, I/O, alarms, etc. in the **Attributes pane**. Multi-select can be used to enable features for attributes, as well as to associate attributes. To use multi-select for features, select the attributes in the **Attributes pane** and enable the feature in the **Details pane**.
5	**Content pane**	The **Content (5)** pane lists graphics and external content items contained in, or linked to, the object. To associate graphics with the Object Wizard, highlight a Choice or Option, and then select and drag one or more items from the **Content pane** to the **Associations pane (2)**. You can add a new graphic to the object from this pane, or link to a graphic in the Visualization folder. See Associate content with a choice or option for additional information.
6	**Scripts pane**	To display the **Scripts pane (6)**, select the **Scripts** button. The **Scripts pane** lists scripts contained in the object. To associate a script with the Object Wizard, highlight a Choice or Option, and then select and drag one or more scripts from the **Scripts pane** to the **Associations pane (2)**. See Associate content with a choice or option for additional information. To add a script to the object, select the **Scripts tab** at the top left of the **Object Editor**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

7	**Details pane (Attribute Editor)**	The **Details pane (7)** displays details of a selected attribute, graphic, or other content. This pane lets you configure individual attributes, but is not usually used while configuring an Object Wizard. See Configuring Objects for additional information.

Object wizard user interface details

The Object Wizard user interface provides a number of visual elements to help you as you build an Object Wizard.



Ref	Icon	Name	Description
1		**Configure Wizard Options button**	Opens/closes the **Object Wizard** editor (**Choices and Options**, **Associations**, **Settings**, **Attributes**, and **Content** panes.
2		**Scripts button**	Opens/closes the **Scripts pane** so you can associate scripts with Object Wizard Choices and Options.
3		**Test mode button**	Changes the **Object Wizard** editor to test mode. This lets you verify the behavior of the Object Wizard. See Object wizard test mode for additional information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

4	<#>	**Number of associations**	Number of trimming-enabled attributes, graphics/content, and scripts associated with a Choice or Option. This can help in the maintenance of complex Object Wizards by letting you quickly identify where trimming has been enabled.
5	<#>	**Total configured settings**	Total number of configured settings for all attributes and graphics that are associated with each **Choice or Option**. This can help in the maintenance of complex Object Wizards by letting you quickly identify where overrides have been configured.
6	☐ ☑	**Trimming checkbox**	Place a check in the checkbox to enable trimming for an attribute or graphic. Trimming is always enable for scripts, and is enabled by default for attributes. See Trimming for additional information.
7	<#>	**Number of configured settings**	Number of configured settings for each **attribute** or **graphic** that is associated with the selected Object Wizard Choice or Option. Scripts and external content items do not have configurable settings.
8		**Add graphic button**	Add (create new) graphic button. The new graphic is owned by the object. See Add content for use with an object wizard for additional information.
9		**Link content button**	Link to an existing graphic or external content item in the Visualization folder. The graphic/content can be shared with other objects. See Add content for use with an object wizard for additional information.
10		**Edit graphic button**	Opens the selected graphic in the **Graphic Editor**.
11		**Symbol Wizard icon**	Symbol Wizard settings are indicated by a unique icon. See **Attribute and Graphic Icons**, below.
12	 	**Attribute / Custom Property icons**	Attribute and graphic custom property icons are displayed for Boolean, integer, float, double, string, time, elapsed time, and internationalized string. An icon is also displayed for Symbol Wizard settings. See **Attribute and Graphic Icons**, below.
13		**Setting Override indicator**	Attribute, custom property, and Symbol Wizard settings that contain an override are indicated by **bold** text and an X next to the value. To remove the override, select and clear the X. See **Override Icons**, below.
14	<text field>	**Visibility Expression**	Sets the conditional visibility of Object Wizard elements. See Conditional visibility expressions for choice groups, choices, and options for additional information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

15	**Visibility indicator icons**		When the icon is unmuted, the setting is visible to the user when configuring instances. When muted, the setting is hidden. See Attribute and graphic overrides, Attribute and graphic overrides and Configurable settings for attribute and graphic values for additional information. A small x in the top right corner of the icon indicates that the visibility setting is an override, and may differ from the setting in the parent template. **Note:** In derived templates, visibility indicators replace visibility expressions for Object Wizard choices and options. See Object wizard derivation for additional information.
		Visible	
		Visible - override	
		Hidden	
		Hidden - override	
16		**XRef**	Total number of configured overrides for the selected graphic or attribute. If the attribute or graphic is associated with multiple Object Wizard Choices or Options, hover over the XRef number to see which Choices and Options have overrides configured.

Attribute and Graphic Icons

Icon	Description / Data Type	Attribute	Graphic
	Symbol Wizard Choice or Option		X
	Boolean data type	X	X
	Integer data type	X	X
	Float data type	X	X
	Double data type	X	X
	String data type	X	X
	Time data type	X	X
	Elapsed time data type	X	X
	Internationalized string data type	X	X
[No icon]	Large type with no icon indicates that the item is an attribute feature. Attribute features are selected in the **Details pane** (see Object wizard GUI panes) and include I/O, History, State Alarms, Bad Value Alarms, Hi/Lo Alarms, etc.	X	

Note: For custom properties, string, time, and elapsed time data types can be toggled between reference/ expression and string literal. This does not apply to attributes.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Reference/expression

String literal

Override Indicators

Indicator	Description
	This is the default setting (no override). If the setting is changed in a parent template, this value will also change.
	Setting has been overridden. Changes made in a parent template will not cause the this value to change. The value is shown in **bold** and an X appears next to the value.
	No override: Setting is visible and is not overridden. If the visibility of this item is changed in a parent template, the visibility of this setting will also change.
	No override: Setting is hidden and is not overridden. If the visibility of this item is changed in a parent template, the visibility of this setting will also change.
	Override has been applied: The visibility setting ("visible") is locked in this template. If this item is hidden in a parent template, this setting will remain visible (setting will not change).
	Override has been applied: The visibility setting ("hidden") is locked in this template. If the this item is made visible in a parent template, this setting will remain hidden (setting will not change).

Object wizard components

An Object Wizard consists of Choice Groups and Options. Choice Groups contain at least two Choices, and Object Wizard users must select one Choice from every Choice Group.

Once you create Choice Groups, Choices and Options, you can associate attributes, graphics, layouts, external content, and scripts with the wizard to create an Object Wizard workflow. Typically, you create and configure the required attributes, scripts and other content prior to adding an Object Wizard to the template. See Configuring Objects for additional information.

Choices are chosen by selecting the radio button next to the Choice. Only one Choice can be selected from a Choice Group. When users select a Choice, they are also selecting any elements (attributes, etc.) associated with the Choice, even though those elements may not be visible to them.

Options are selected by placing a check in the Option's checkbox. Options can be either true (checked) or false (unchecked). Attributes, graphics, other content, and scripts can only be associated with the true state. No action is implemented when an Option is set to false.

You can rename Choice Groups, Choices, and Options as needed and add a description (the description is visible to users as a tooltip). Choice Groups and Options must be uniquely named; a Choice Group and an Option cannot have the same name. Choices within a single Choice Group must also be uniquely named.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Add content for use with an object wizard

You add attributes, graphics, layouts, scripts, and other content to a template object with an Object Wizard the same way that you add content to any other object. After adding the content, you can associate individual attributes and other content to individual Choices and Options. This association allows content that is not needed to be trimmed when you derive instances from the template. This helps to reduce the footprint of runtime objects. See Trimming for additional information.

Note: Only user-created attributes can be associated with Object Wizard Choices and Options.

When you associate content with Choices and Options, the propagation of these items to derived instances is determined by:

- User selection of Choices and Options during configuration.
- Trimming configuration for attributes and graphics, as they apply to the selected Choices and Options.

Content is trimmed from a derived instance when the instance is saved. Note that all items are propagated to derived templates, regardless of configuration and trimming settings, since unlike instances, templates are not runtime objects.

Both attributes and graphics can have unique settings for each association. The visibility of these settings, which determines if the setting can be overridden by a user deriving instances from the template, can also be configured.

For detailed information about custom properties and Symbol Wizards, see Using custom properties with Application Server and Symbol Wizard Editor.

Content can either be added or linked to an object.

When you select the **Add Content** button, a new graphic is created that is owned by the object. Any changes made to the graphic are only applicable to the owning object. Changes to the graphic in the parent object will cascade to child objects. The **Add Content** button does not let you create an object-owned external content item.

When you select the **Link Content** button, the Galaxy Browser opens the Visualization folder. Selecting a graphic, layout or external content object from the Visualization folder creates a link between the object and the content, but the object does not own the content. Note that relative references will resolve correctly for the linked graphic or layout. Changes to the content will cascade to every object with which the item is linked, regardless of the hierarchical relationship of the objects.

For information on adding scripts to a template, see Writing and Editing Scripts.

The settings configured in the **Attribute Editor**, **Graphic Editor**, and **Layout Editor** provide the default settings for each attribute and content item. You can override the default values for attributes and graphic custom properties in the **Settings pane**. Note that you can set different values for each Choice or Option with which the content is associated.

In addition, you can allow the user to override the default values in derived instances. See About the Attributes Page and Add Attributes to an Object for information about adding and configuring attributes.

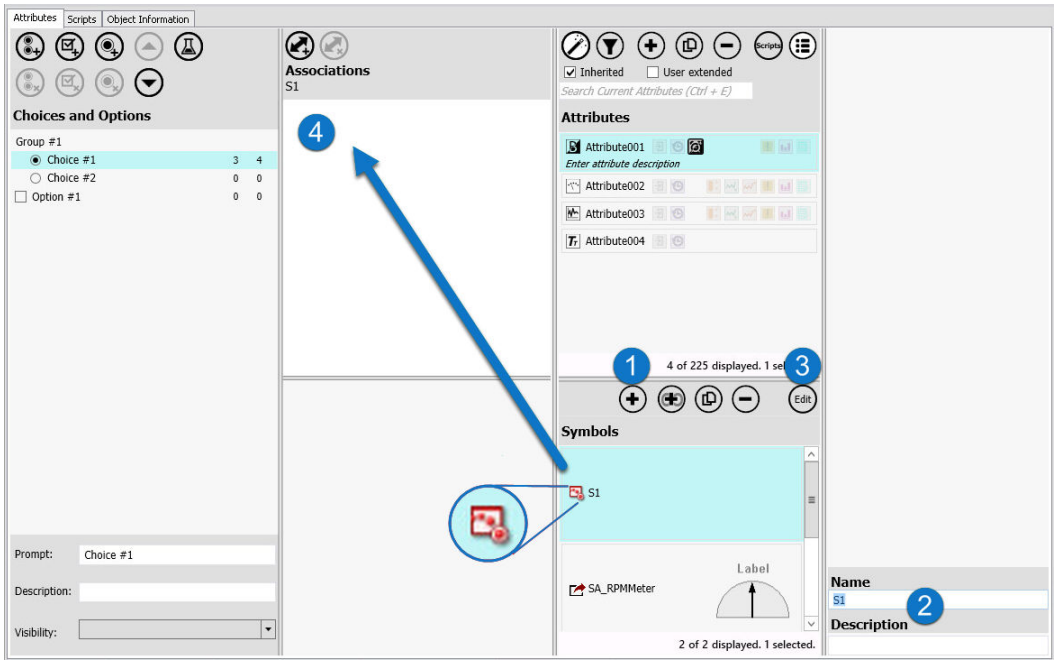
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Add a new object-owned symbol

To add a new object-owned graphic

1. In the **Content** pane, select the **Add Content (1)** button. The graphic is added with a default name (based on the previously added graphic, if any).
2. In the **Details (2)** pane, edit the name of the graphic and add a description.
3. In the **Content** pane, select the **Edit (3)** button to open the **Graphic Editor**. See *Creating and Managing Industrial Graphics User Guide* for information about using the editor to create graphics. When you are done editing the graphic, save and close the Graphic Editor.
4. Associate the graphic with one or more Choices or Options, as described in Associate content with a choice or option.

   The icon next to the graphic name in the **Content** pane indicates that the graphic is owned by the object.



Link to shared content in the visualization folder

You can link an object to content in the Visualization folder. Content can be a graphic, a layout or an external content object.

To link to an existing content item in the Visualization folder:

1. In the **Content** pane, select the **Link Content** button .

   The **Galaxy Browser** opens. Select an existing graphic, layout, or external content object from the **Visualization folder**.
2. In the **Details** pane, edit the default name of the content as needed. The edited name applies only to the object to which it is linked. The name of the content item remains unchanged in the Visualization folder. You can also add a description .

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: If you choose to link to the same content item more than once, it is automatically renamed within the object with an incremental number appended to the original name. Its name in the Visualization folder is not changed, and is displayed in brackets in the following format: `OriginalName1 [OriginalName]`. In most cases, linking to the same item more than once is not an efficient use of an Object Wizard.

3. **Optional:** To edit the linked content, select the **Edit** button to open the **Graphic Editor** (for graphics), the **Layout Editor** (for layouts), or the **External Content Editor** (for external content items)**.** Any change you make to the linked content item changes the configuration of the content, and the change affects any object that links to the item.

   Note: Editing protected graphics, such as graphics contained in the **Situational Awareness Library**, is not permitted. If you edit an item in the **Visualization folder**, the changes will affect every object that links to the item.

Associate content with a choice or option

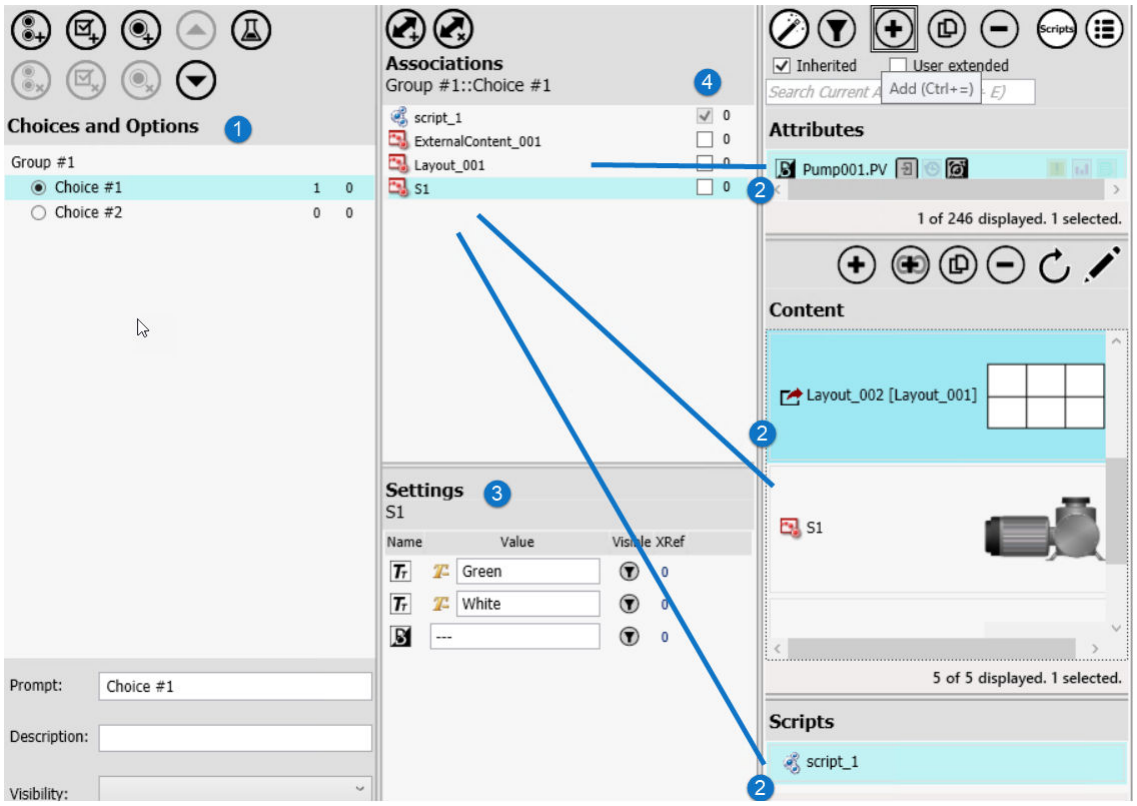
In most cases, you will have added content to the object prior to adding the Object Wizard. Once you have begun configuring the Object Wizard, you can associate the following types of content with Object Wizard Choices and Options:

- Attributes: shown in the **Attributes** pane
- Graphics: shown in the **Content** pane
- External Content Objects: shown in the **Content** pane
- .NET Controls: shown in the **Content** pane
- Scripts: shown in the **Scripts** pane

To associate content with a Choice or Option:

1. Highlight a Choice or Option in the **Choices and Options** pane.
2. Select and drag one or more item from the **Attributes**, **Content**, or **Scripts** pane to the **Associations** pane. Use the **Shift** or **Ctrl** key to select multiple items in a single pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: Individual items can be associated with more than one Choice or Option. However, a Choice or Option does not have to have any items (attributes, etc.) associated with it. For example, a Choice labeled "No" or None" typically would not have any content associated with it.

Note: A content item (attribute, graphic, layout, etc.) does not have to be associated with an Object Wizard Choice or Option. Unassociated items are simply a part of the object and are propagated to all derived objects.

3. If applicable, highlight an attribute or graphic and enter overrides for the default values. The overrides affect only the Choice or Option for which they are entered. For example, a graphic can have one label for one Choice, and a different label for another. The override applies only to the associated Choice or Option. See Attribute and graphic overrides for additional information.

   Note: You can resize columns in the **Settings** pane by selecting the column header and moving its edges to widen or narrow the column.

4. If applicable, override the default trimming setting for each associated item.
   - Trimming is disabled by default for graphics, layouts, and external content (checkbox is unchecked).
   - Trimming is enabled by default for attributes (checkbox is checked).
   - Trimming is enabled for scripts and cannot be disabled.
   - Note that the trimming defaults for graphics and attributes are different. See Trimming for additional information.

Disassociate content from a choice or option

To remove the association of a content item (attribute, graphic, etc.) with a Choice or Option:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

1. Locate the Option, or the Choice within its Choice Group and select it.
2. In the **Associations** pane, select the content item you want to remove from the Choice or Option.
3. Select the **Remove Association** button. The item is no longer associated with the Choice or Option.

Note: Removing an association does not remove the content item. The content remains a part of the template, and if the content is associated with other Choices or Options, those associations remain intact. However, removing an association may affect the trimming of content within instances (see Trimming for additional information). If the content is completely unassociated with any Object Wizard Choice or Option, it is always included in derived instances.

To completely remove an attribute from a template, see Delete content from an object.

Multiple content item associations

Attributes, graphics, and other content can be associated with more than a single Choice and/or Option. Each association can use different override values (settings) for a particular attribute or graphic. To change a setting for a particular association, enter overrides of the defaults in the settings pane.

You can use different overrides for each Choice or Option with which an attribute or graphic is associated. For example, an integer attribute could have a default value of 10, but could have an override of 5 for one Choice and an override of 20 for a different Choice.

Map attribute and graphic configurations to a choice or option

You can associate a particular attribute, layout, or graphic configuration with an Object Wizard selection (Choice or Option). In this way, pre-configured attributes and content can be added to the derived instance simply by selecting an Object Wizard Choice or Option. Different configurations can be applied to different Choices and Options.

Mapping these configurations to Object Wizard Choices and Options may require additional planning and testing. This is especially important if multiple configurations of the attribute or content are being mapped. Mismatches between configurations can result in different behaviors in the derived instance than intended. Use **Test mode** to catch potential mismatches.

Attribute and graphic overrides

Override values for an attribute or graphic (with a Symbol Wizard or custom properties) can be configured for each associated Choice or Option. When you associate an attribute or graphic, the default values are listed in the **Settings** pane. To configure values that apply to the selected Choice or Option, enter the values in the **Settings** pane. If the attribute or graphic is associated with multiple Choices or Options, different values can be used for each one. Values for the Choices and Options are saved when you save the object. If the setting is marked as visible, users can override your setting (or override the default, if you have not configured a setting) as they derive and configure instances from the Object Wizard.

Graphics with overrides that affect their shape, for example, certain Situational Awareness Library graphics, are rendered with the applicable shape-changes as defined by the overrides.

Note: If you override a Symbol Wizard setting, the only way to make the setting visible again is to remove the override. Attributes and custom properties with overrides can be made visible again by selecting the visibility checkbox.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

A configured setting is defined as any change from the default configuration of the attribute or graphic. This includes changing a value or visibility. For example, you might change the default Boolean value from true to false, or you might change an integer value from 5 to 10, or change engineering units from liters to cc. Changing the visibility of a setting also counts as a configured setting.

- The default visibility setting for attribute values is hidden (Visibility icon is muted (grayed-out) in the **Settings** pane).
- The default visibility setting for Symbol Wizard and custom property values is visible (Visibility icon is unmuted in the **Settings** pane).
- Visibility overrides are indicated by a small X in the upper right quadrant of the visibility icon.

The total number of configured settings for each individual attribute and graphic is displayed in the **Associations** pane. The total number of configured settings for ALL attributes and graphics associated with an individual Choice or Option is displayed in the **Choices and Options** pane.

Delete content from an object

You can delete (completely remove) a content item from an object. When you delete content, all settings and associations to the items are also removed.

If there are any instances or templates derived from the template that use the deleted item, the deletion of the item will propagate to all the derived objects. That is, all objects derived from the template will no longer contain the deleted content.

Delete an attribute or graphic from a template

You can delete an attribute or content (object-owned or linked) from a template that has an Object Wizard. If there are objects derived from the Object Wizard that use the deleted attribute or content, the deletion propagates to all the derived objects.

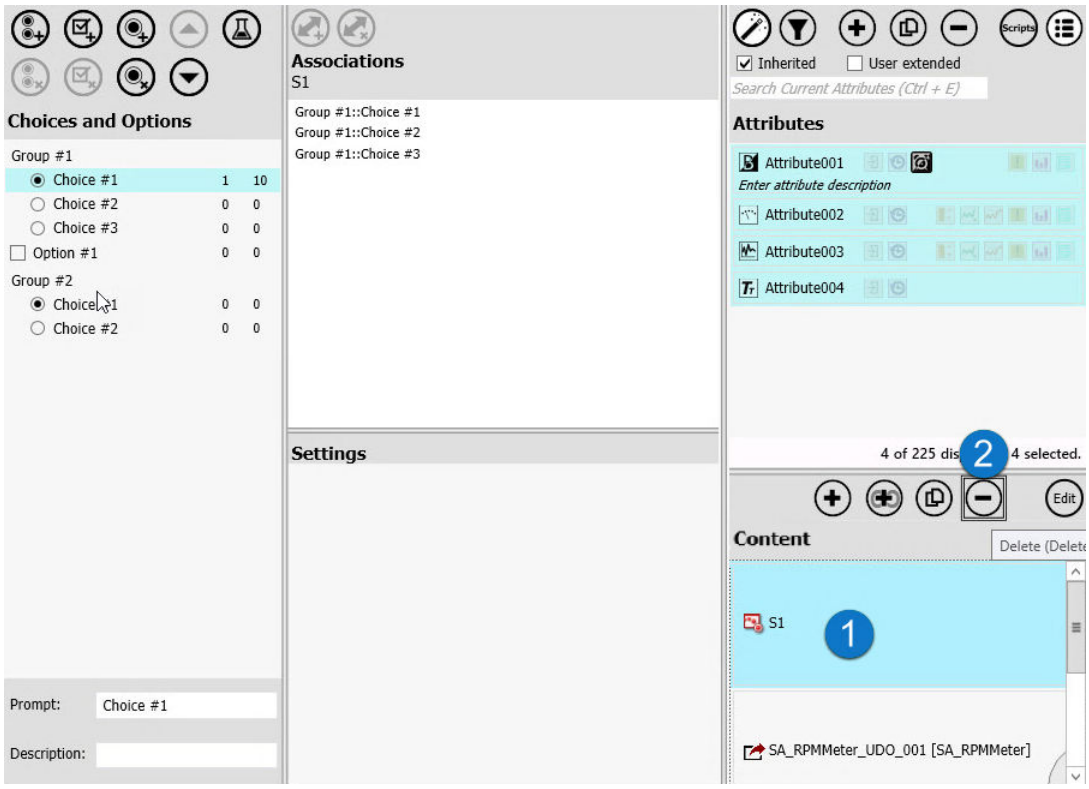
Linked graphics, layouts, and external content items will still exist in the Visualization folder. Deleting an object-owned graphic completely removes it. Once you save the template that contains the deleted object-owned graphic, the graphic cannot be retrieved.

To delete an attribute or content from a template:

1. Select the content item from the **Content** pane. To delete an attribute, select it from the **Attributes** pane. You can select more than one attribute or graphic at a time.
2. Select the **Delete** button. This removes the selected graphic from the object. Use the **Delete** button in the **Attributes** pane to delete attributes.

   Note: Added (object-owned) graphics are permanently deleted once the object is saved. Linked content is removed from the object but remains in the Visualization folder.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

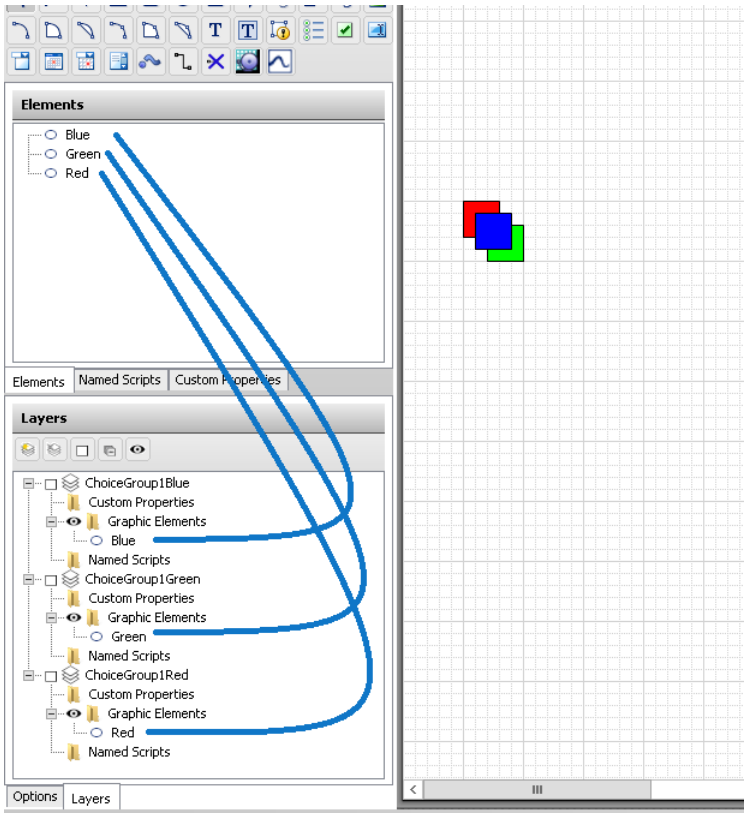


Example 1: Map a Simple Symbol Wizard to an Object Wizard

This example shows how to create a simple Symbol Wizard, set configurations, and map the configurations to Object Wizard Choices.

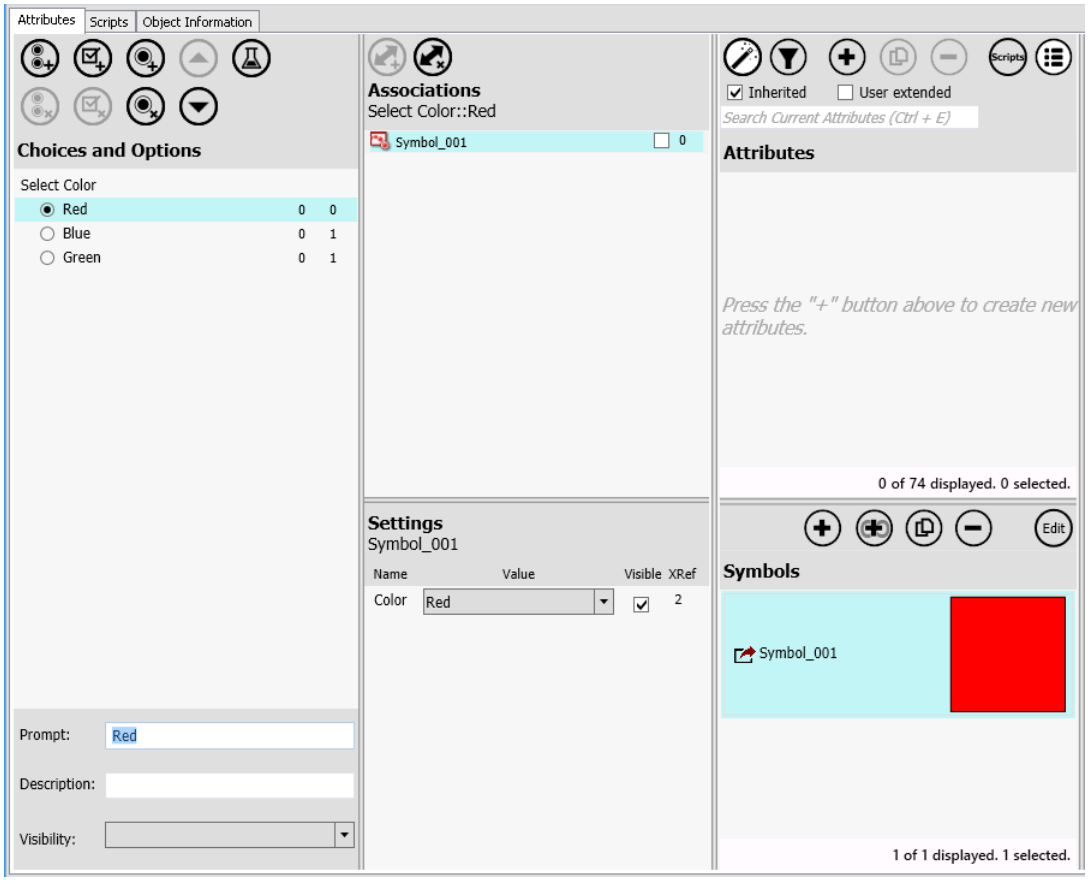
1. Create a new graphic in the Visualization folder.
2. Build a simple Symbol Wizard in the **Graphic Editor**. by adding a Choice Group with three Choices named Red, Green, and Blue.
3. Add three shapes, filled with colors that correspond to the Choices. Rename the shapes so you can identify them.
4. Assign each shape to the corresponding Choice by dragging the shape name from the **Elements tab** to the corresponding Choice in the **Layers tab**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



5. Save and close the **Graphic Editor**.
6. Derive a new template from the Template Folder, and add an Object Wizard with:
   - One Choice Group. Change the name of the Choice Group to "Select Color."
   - Three Choices: Red, Green and Blue.
7. Link the Symbol Wizard you created in step 1 to the derived template.
8. Configure each Symbol Wizard association to match the corresponding Object Wizard Choice.
   - Red Symbol Wizard Choice selected for Red Object Wizard Choice.
   - Blue Symbol Wizard Choice selected for Blue Object Wizard Choice.
   - Green Symbol Wizard Choice selected for Green Object Wizard Choice.

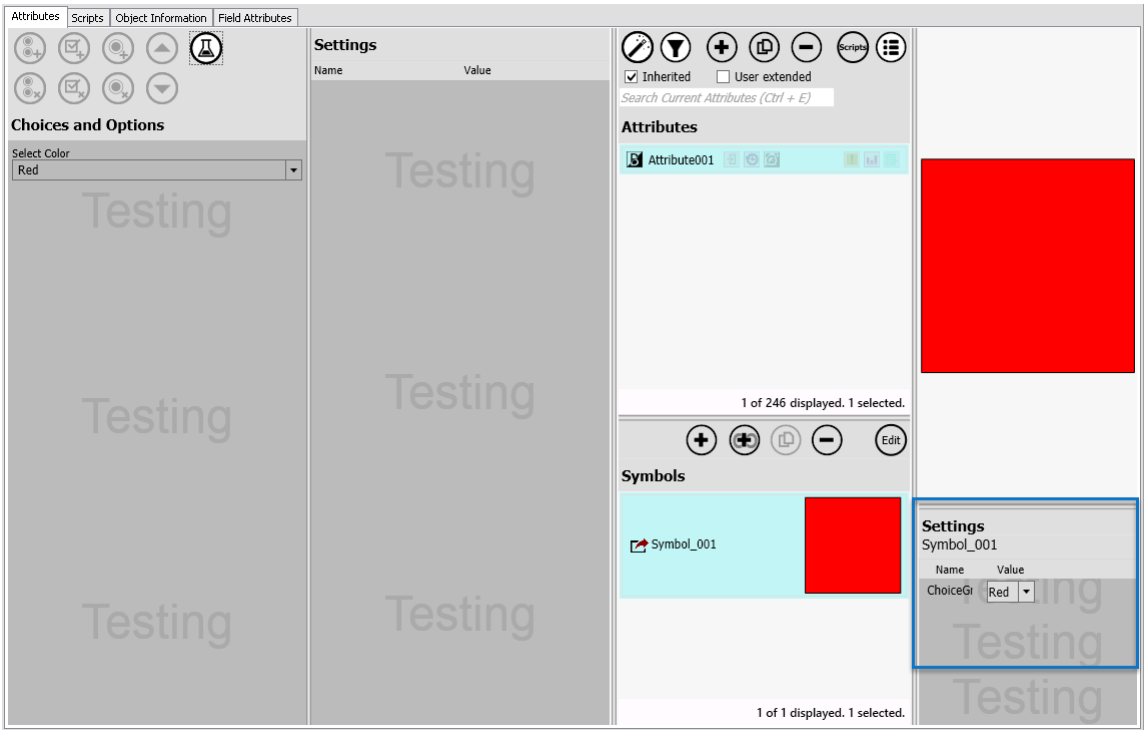
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: Only the default selection can be marked as visible (you can remove the checkmark to hide the default selection). Selections with override are not visible. Users cannot modify selections that are not visible.

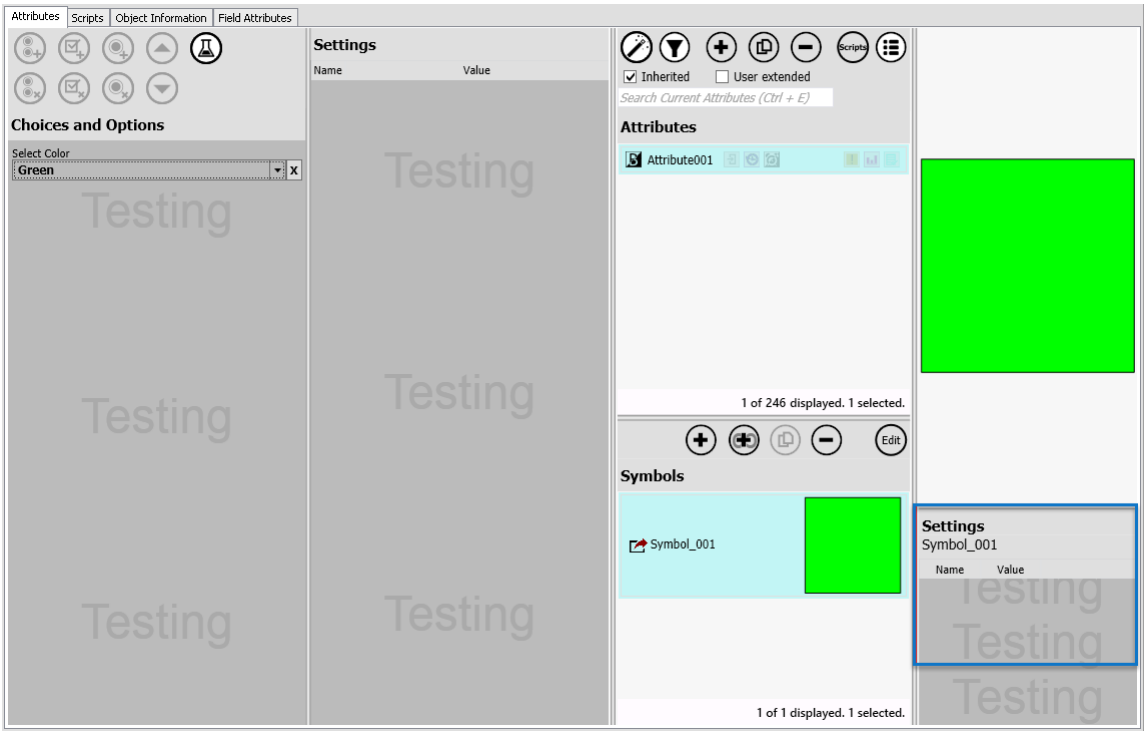
9. Switch to Test Mode.
10. With the default Option Wizard Choice selected (Red, in this example), you should see something like this:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The Symbol Wizard Choice Group can be changed in the **Symbol Settings** pane, since we did not turn off visibility for the default selection.

11. With any of the non-default Option Wizard Choices selected, you should see something like this:



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

In this case, Symbol Wizard settings cannot be changed because a non-default configuration of the Symbol Wizard is associated with the Green Object Wizard Choice. When a non-default selection is made in the Symbol Wizard, its visibility setting is turned off and disabled.

Example: Configure a Linked Graphic Associated with Multiple Choices

This example shows the steps required to link and configure a graphic from the **Visualization folder**, set configurations, and associate different configurations with Choices and Options. This example uses a graphic from the **Situational Awareness Library**. You could also link other types of content, such as layouts and external content items. However, only graphics allow you to configure and associate different configuration settings within the Object Wizard.

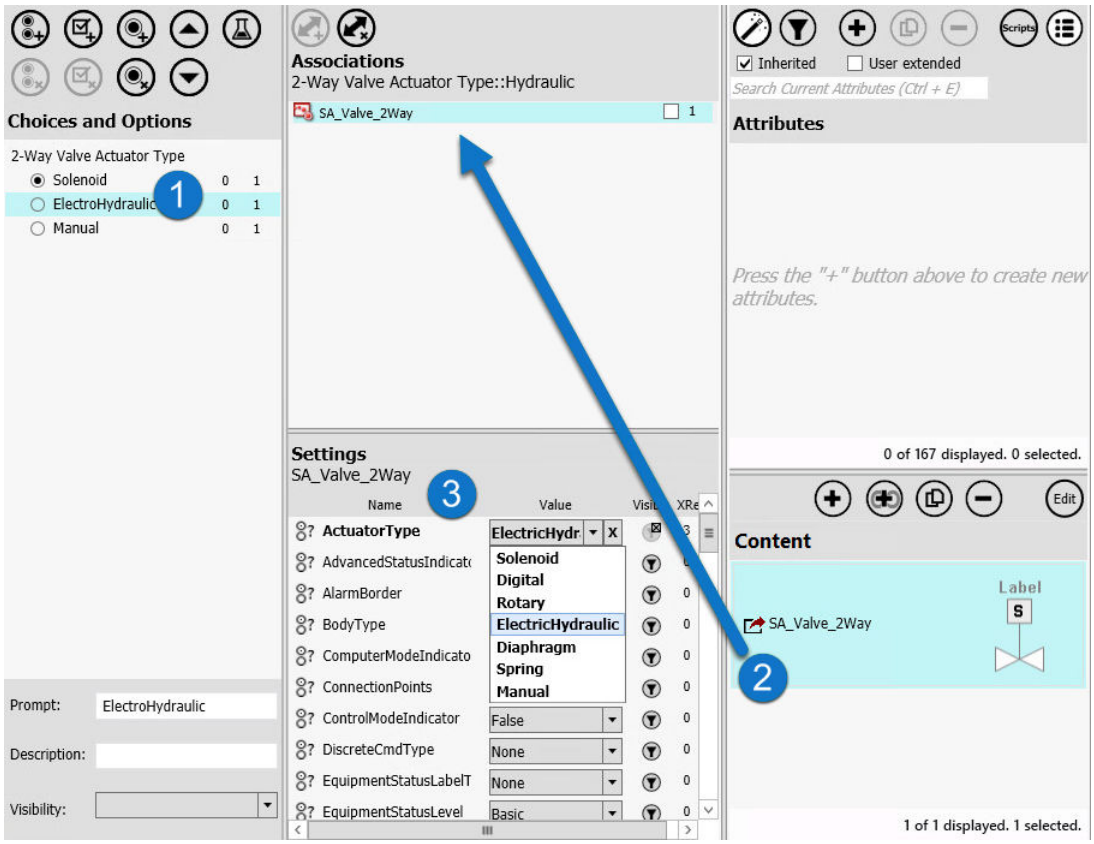
Note: Most graphics in the **Situational Awareness Library** are Symbol Wizards.

To configure a linked graphic associated with multiple choices

1. Derive a template from the $UserDefined template, then:
   a. Add an Object Wizard Choice Group and name it "2-Way Valve Configuration."
   b. Add three Choices, named: "Solenoid," "ElectroHydraulic," and "Manual."
2. Navigate to the Equipment folder of the Situational Awareness Library (in the **Visualization** folder), and link to the SA_Valve_2Way graphic. The graphic is added to the **Content** pane.

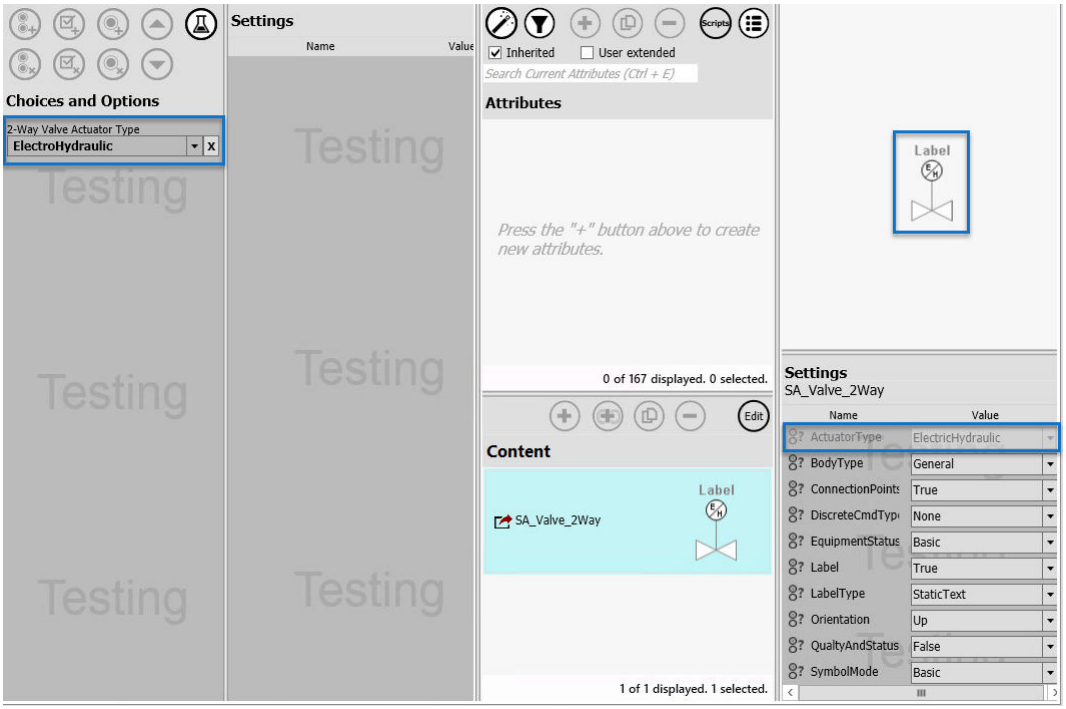
   Associate the linked graphic with each of the Choices created in step 2, and set the ActuatorType in the **Settings** pane to match the associated Object Wizard Choice.
3. Set visibility and overrides for each association:
   - For Solenoid:
     - Select the Solenoid Choice from the **Choices and Options** pane.
     - Select SA_Valve_2Way from the Associations pane.
     - In the Settings pane, make sure that ActuatorType is set to the default (Solenoid), then turn off visibility.
   - For ElectroHydraulic and Manual:
     - Select the ElectroHydraulic or Manual Choice from the **Choices and Options** pane.
     - Select SA_Valve_2Way from the Associations pane.
     - In the **Settings** pane, set the ActuatorType to match the Choice in the **Choices and Options** pane. Since both ElectroHydraulic and Manual are overrides, visibility of these settings automatically changes to hidden.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



4. Switch to **Test mode** and check each of the Choices in the **Choices and Options** pane.
   - ActuatorType under graphic **Settings** should be grayed-out to indicate it is hidden
   - The graphic setting should match what you have selected in the **Choices and Options** pane**.**
   - The graphic thumbnail should match the selected ActuatorType.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Conditional visibility expressions for choice groups, choices, and options

You can set whether or not a user will be able to see individual Choice Groups, Choices and Options as they use the Object Wizard.Visibility expressions are scripted and use simple logical expressions to determine if users will be able to see a particular Option or Choice when configuring instances. The visibility expression determines if the Object Wizard component will be visible or hidden, based on the user's previous Object Wizard selections. For example, selecting Choice #1 may hide (or show) Choice Groups, Choice or Options that follow that Choice in the Object Wizard.

Note: You cannot add a visibility expression to the first listed item (Choice Group or Option) in an Object Wizard; the first item is always visible.

When a user uses the Object Wizard to configure an instance, the default behavior of any elements hidden by visibility expressions remains in effect. For example, if Choice #1 is the default of a Choice Group, and the Choice Group is hidden, Choice #1 is still selected for the configured instance, even though it is not visible to the user. If the default state of an Option is True but the Option is hidden, the Option remains True for the configured instance.

Visibility is set through logic expressions assigned to Choices and Options. These logic expressions use the Boolean operators AND, OR, and NOT.

In derived templates that contain an Object Wizard from the parent object, icons are used to implement visibility settings that are in addition to the settings made in the parent object. Note that the visibility settings made in the parent Object cannot be changed in the derived template. See Object wizard visibility settings in a derived template for additional information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Add a basic visibility expression

To add a basic visibility expression to a Choice Group, Choice or Option:

1. Highlight the item for which you are setting visibility (Choice Group, Choice or Option). You can select any item except the initial one. That is, you cannot make the visibility conditional of the first item (Choice Group or Option) in the Object Wizard. If the first item is a Choice Group, you cannot make it or any of its Choices conditional.
2. Place the cursor in the Visibility text box of the selected item, and type " **[** " (open bracket character). A drop-down list of all preceding items (Choices and Options) is displayed.

   Note: Choices are listed in format "*Choice Group Name::Choice Name*", for example, `Choice Group #1::Choice #1`. Options are listed in the format "*Option::State*", for example `Option #1::True`.

   Note: While you can simply select an item for a simple visibility expression without typing an open bracket, we recommend that you always type the open bracket as a matter of habit. As you begin building more complex visibility expressions, you risk deleting the expression if you select an item from the drop-down without first typing an open bracket.
3. Select the item from the drop-down list that will be used to control the visibility of the selected item. For example, you can make Option #2 visible only if Choice #1 of Choice Group #1 has been selected (Group #1::Choice #1), and/or if Option #1 is set to True (Option #1::True).

   For additional information, see Operators, rules, and behavior for visibility expressions.

   Note: When a user configures an instance, the default behavior of elements remains in effect, even if they are hidden from the user. The default Choice of a hidden Choice Group is still selected, and hidden Options remain in their default states.

Add a visibility expression with logical operators

To add a visibility expression that uses logical operators to a group, choice or option:

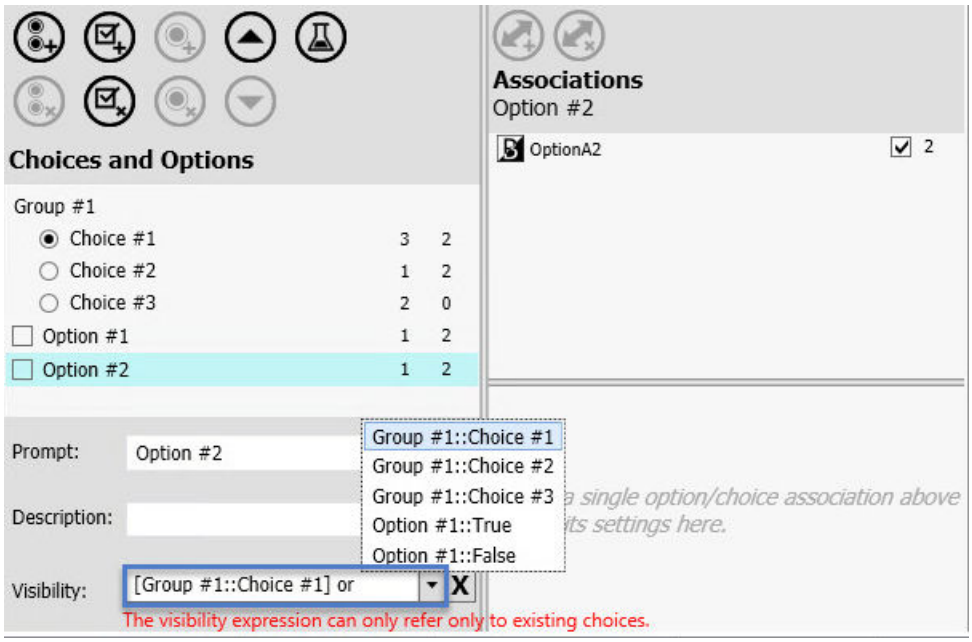
1. Highlight the item for which you are setting visibility (Choice Group, Choice or Option). You can select any item except the initial one. That is, you cannot make the visibility conditional of the first item (Choice Group or Option) in the Object Wizard. If the first item is a Choice Group, you cannot make it or any of its Choices conditional.
2. Place the cursor in the Visibility text box of the selected item, and type " **[** " (open bracket character). A drop-down list of all preceding items (Choices and Options) is displayed.

   Note: Always type an open bracket before selecting an item from the drop-down list. Simply selecting an item without first typing an open bracket will overwrite the existing expression.

   Note: Choices are listed in format "*Choice Group Name::Choice Name*", for example, `Choice Group #1::Choice #1`. Options are listed in the format "*Option::State*", for example `Option #1::True`.
3. Select the item upon which you are basing the visibility of the selected item. It will be entered in the visibility expression. To expand the expression:
   - Place the cursor at the front of the expression (before the open bracket) and enter the NOT operator, or
   - Place the cursor at the end of the expression (after the close bracket) and enter the AND or OR operator.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

4. Type " **[** " (open bracket character) and select another item from the drop-down. All Object Wizard items must be enclosed in brackets. Enclose items and operators in parentheses as needed to ensure that the expression resolves as intended.



5. Add additional operators, parentheses, and Object Wizard items as needed to complete the visibility expression. For example:

```
[ChoiceGroup1::Choice1] AND ([Option1::True] OR [Option2::True] OR [Option3::False])
```

Note: When a user configures an instance, the default behavior of elements remains in effect, even if they are hidden from the user. The default Choice of a hidden Choice Group is still selected, and hidden Options remain in their default states.

For additional information, see Operators, rules, and behavior for visibility expressions.

Operators, rules, and behavior for visibility expressions

As you build an Object Wizard, you can create visibility expressions that selectively hide Choice Groups, Choices, and Options. This can simplify the workflow by only showing Choices and Options that are applicable, based on Choices and Options that the user has already made. When an Option or Choice Group is hidden, the system applies the configured default values for attributes, graphics, and scripts associated with the hidden Option or Choice Group. You can leverage this capability to hide advanced functions from users tasked with configuring instances from templates that contain Object Wizards.

The **Visibility** prompt lets you assemble expressions to hide or show Choice Groups, Choices, or Options based on what the user has previously selected. Note that Visibility cannot be conditional for the first Object Wizard item (Choice Group or Option). The first Object Wizard item is always visible. Available logical operators are AND, OR, and NOT.

Observe these basic rules and behaviors when building visibility expressions:

- Always start by typing the " [" (open bracket character) in the visibility text box. This will bring up a list of all Choices (from other Choice Groups) and Options that precede the selected item. Do not simply highlight an element without first typing the open bracket character.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Choices are listed with their Choice Group. Each Option is listed twice, for both states (True and False). Use fully qualified Option/Choice selections surrounded by brackets. The separator between Choice Groups and Choices, and between Options and their states is "::" For example:

  [Valve Type::3-Way] OR [Valve Type::4-Way] AND [Option1::True]

- Valid logical operators:
  - Valid binary operators are "AND" and "OR".
  - Valid unary operator is "NOT".
  - "AND" has a higher precedence than "OR".
  - "NOT" has a higher precedence than "AND" or "OR".
  - You can use "&" or "&&" instead of "AND".
  - You can use "|" or "||" instead of "OR".
  - You can use "!" instead of "NOT".
- Parentheses can be used to build complex expressions. For example:
  ```
  [ChoiceGroup1::Choice1] AND ([Option1::True] OR [Option2::True] OR [Option3::False])
  ```
  Note that without parentheses, this expression would resolve very differently, since "AND" has a higher precedence than "OR". If parentheses were not used, the expression would resolve this way:
  ```
  [ChoiceGroup1::Choice1] AND [Option1::True] OR [Option2::True] OR [Option3::False]
  ```
- Literals and operators are not case sensitive.
- Spacing between literals and operators is optional.
- The visibility expression editor validates the logical expression as you enter it, but does not check if the expression is possible. For example, the expression "[Valve Type::3-Way] AND [Valve Type::4-Way]" would not be flagged since it is logically correct, even though the condition is impossible (you cannot select more than one Choice from a single Choice Group).

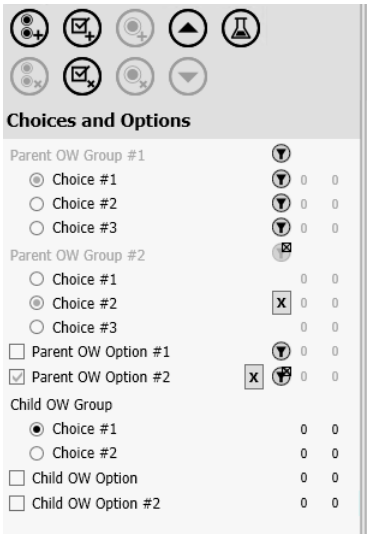
Object wizard visibility settings in a derived template

When you derive a template from a template with an Object Wizard, derived Object Wizard elements can no longer be edited, and Object Wizard group names are grayed out. Other noticeable changes are:

- A visibility icon appears next to each derived wizard item.
- You can override visibility settings by clicking on the visibility icon.
- The item counts are reset to zero; changes made in the derived object will increment the counters.
- If you change a default selection, the setting override indicator appears next to the changed setting.

See Object wizard user interface details for more information about visibility icons and the setting override indicator. For additional information about derivation and visibility, see Object wizard derivation.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The visibility expressions that were configured in the parent Object Wizard remain in effect. You can, however, hide Choice Groups, Choices, and Options. Instead of using Boolean expressions to set the visibility of Groups, Choices, and Options, use the visibility icon . When changing the visibility settings of a derived Object Wizard:

- The behavior of a hidden Choice or Option does not change when configuring an instance from the derived Object Wizard. The default state of a hidden Option remains in effect when the instance is configured. If the Option's default state is True, any attributes, graphics, or scripts associated with it are automatically configured for the instance. Similarly, the default Choice in a hidden Choice Group is automatically selected, along with its associated attributes, graphics, and scripts.
- You cannot hide (turn off the visibility of) a default Choice. You can, however, hide the entire Choice Group.
- If every Choice except the default is hidden, the entire Choice Group is hidden.
- If you change a default choice, the entire Choice Group is automatically hidden. You can turn visibility on again for the Choice Group. Once the Choice Group has been made visible, you can hide individual Choices.

Trimming

Trimming refers to the ability of the Object Wizard to remove unneeded attributes, graphics, and scripts from derived instances. By removing unneeded items, the size of instances an be minimized, thus allowing more efficient utilization of run-time system resources. This can also result in faster deployments.

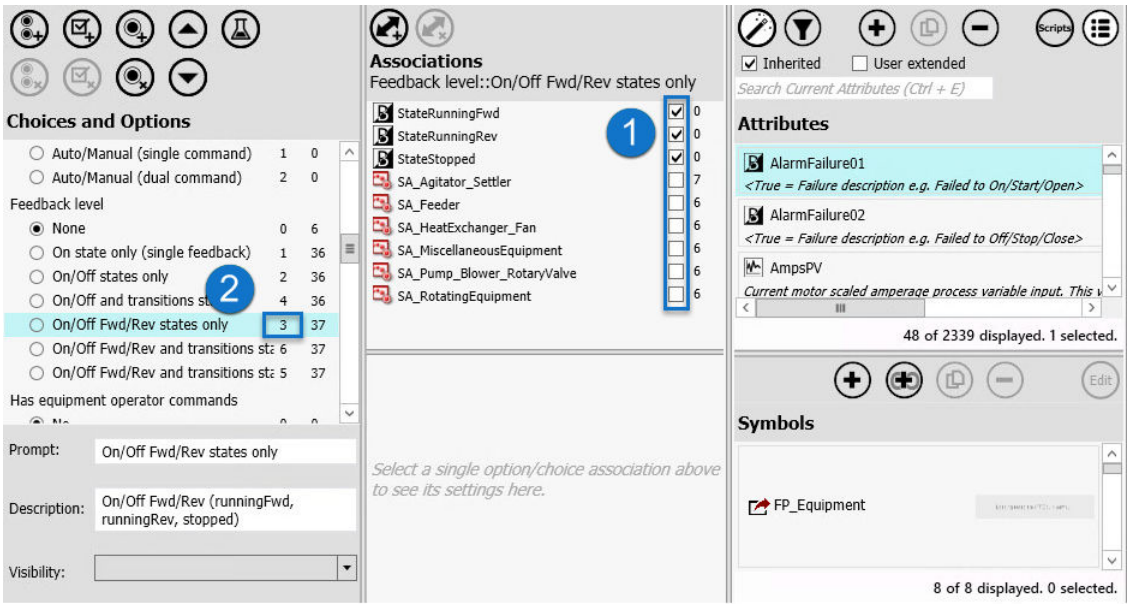
- When you associate an attribute with a Choice or Option, the **Trim** checkbox (1) is checked by default. This means that trimming is enabled, and the attribute will be evaluated for trimming.
- When you associate a graphic with a Choice or Option, the **Trim** checkbox is cleared by default. This means that trimming is DISabled, and the graphic will NOT be evaluated for trimming.
- When you associate a script with a Choice or Option, the **Trim** checkbox is checked. Trimming cannot be disabled for scripts.

Note: Graphics do not exist as run-time objects, but do exist as part of a ViewApp. Trimming unnecessary graphics can help to reduce system resource requirements.

The total number of trimmable items (attributes, graphics, and scripts) associated with each Object Wizard Choice and Option is displayed in the **Choices and Options** pane (2). Items that are not enabled for trimming are not included in this count. The default trim state of associated graphics is disabled.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: The count of trimmable items resets to 0 in derived templates.



Enable Trimming

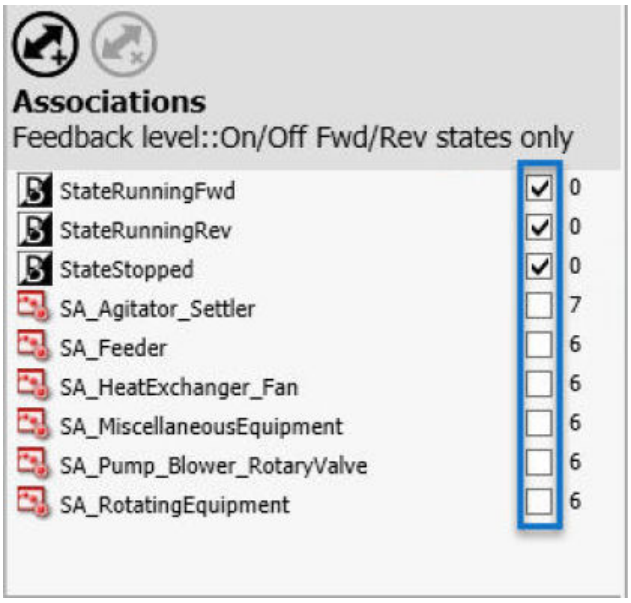
Whether or not an attribute or graphic associated with a Choice or Option will be evaluated for trimming is determined by the state of its trimming checkbox in the **Associations** pane. See Object wizard user interface details for additional information.

Note: Scripts are always evaluated for trimming.

To enable trimming:

- Select the **Trim** checkbox in the **Associations** pane.
  - When checked, trimming is enabled.
  - When unchecked, trimming is disabled.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Graphics and attributes are evaluated for trimming when an instance is derived, in accordance with the following behaviors (scripts are always evaluated for trimming):

- If none of its associations have trimming enabled, the item is not evaluated for trimming and is always propagated to derived instances.
- The item is propagated to the instance if any Choice or Option where trimming is enabled is selected.
- The item is trimmed if at least one of its associations has trimming enabled, and none of the associations with trimming enabled are selected (even if associations with trimming disabled have been selected).
- Trimming occurs only when the instance is saved.

Note: Trimming is not applicable to derived templates, since these are not run-time objects.

Attribute and graphic overrides

You can associate different configurations of graphics and attributes with different Object Wizard Choices and Options. You can also allow users to override values that you previously set as they configure derived objects. See Configurable settings for attribute and graphic values for additional information about setting overrides and visibility for Symbol Wizard and custom property values. For detailed information about Symbol Wizards and custom properties, see Using custom properties with Application Server and "Working with Symbol Wizards" in the *Creating and Managing Industrial Graphics User Guide*.

The **Settings** pane lets you configure setting overrides and visibility for associated attributes and graphics.

Attribute Overrides

For attributes, you can override default settings and set visibility for:

- Attribute description
- Data type
- Initial value
- Engineering units (if applicable)

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Activate/deactivate features, such as I/O and alarms
- Feature values

Attribute Setting Visibility

When you associate an attribute with an Object Wizard Choice or Option, its settings are **hidden by default**. You can make settings visible by selecting the checkbox next to the setting. You can also make overrides visible.

Graphic Overrides

For graphics, you can override default settings and set visibility for:

- Symbol Wizard Choices and Options
- Custom properties

Graphic Setting Visibility

When you associate a graphic with an Object Wizard Choice or Option, Symbol Wizard and custom property settings (if any) are **visible by default**. If you override a graphic custom property or Symbol Wizard setting, its visibility automatically turns off (the setting is hidden). You can make an overridden custom property setting visible again by selecting the checkbox next to the setting.

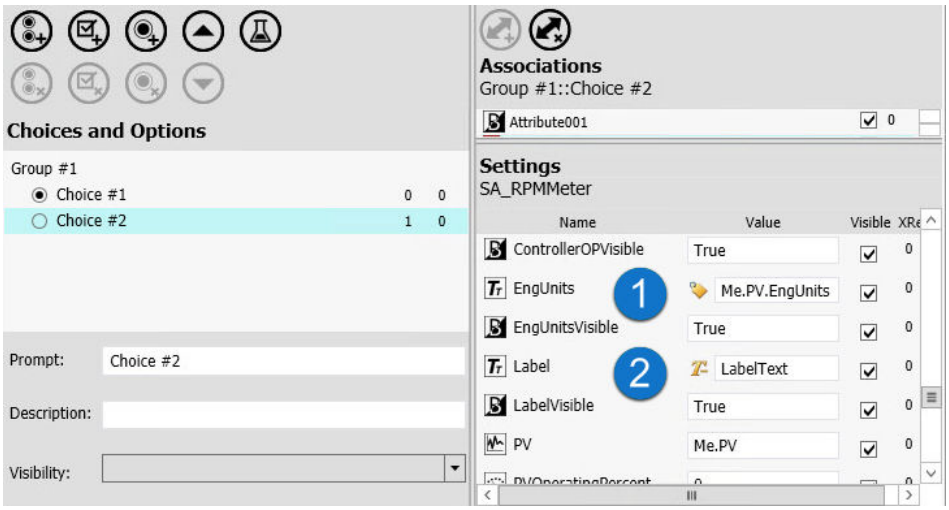
Note: If you override a Symbol Wizard setting, the only way to make the setting visible again is to remove the override.

Custom properties can be Boolean, integer, float, string, etc. See Object wizard user interface details for additional information. For detailed information about custom property data types, see Using custom properties with Application Server. When configuring string, time, and elapsed time custom properties, a toggle is provided in the **Settings** pane to configure the custom property as either:

Reference string or an expression (1). Reference strings will be evaluated to determine their meanings. For examp
`Me.Tank1.InletValve.PV`.

String literal (2). String literals are passed as static text.

To change the string type, select the icon next to the text field.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Enter overrides and set visibility for attribute and graphic values

For both attributes and graphics that contain a wizard or custom properties, you can enter overrides that apply only to the associated Object Wizard Choice or Option. While the following procedure shows how to configure a graphic, the procedure for configuring attributes is virtually identical. See also Attribute and graphic overrides, Attribute and graphic overrides.

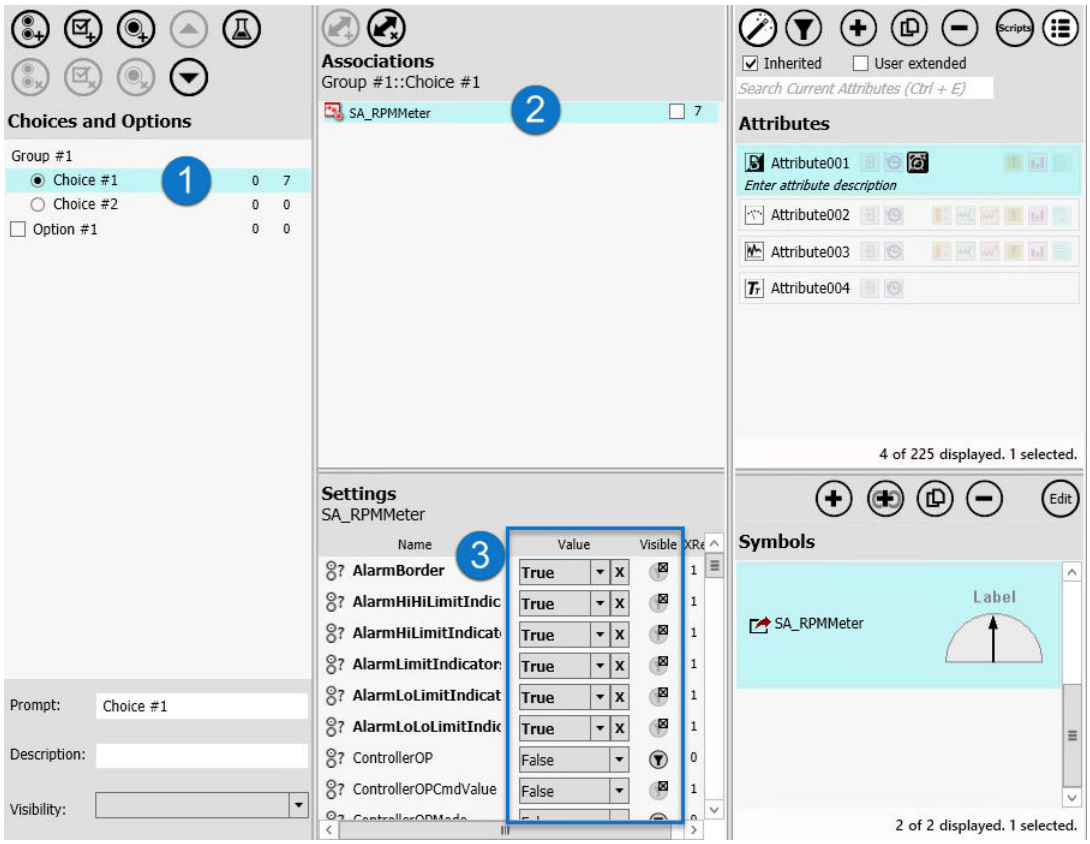
To configure an Override for a Choice or Option

1. Select the Choice or Option you want to configure (**Choices and Options** pane).
2. Select a graphic or attribute (**Associations** pane) associated with the Choice or Option. See Associate Content with a Choice or Option for information about associating graphics with Object Wizards.

   This does not apply to scripts since they can only be associated with a Choice or Option. There are no configuration or visibility settings that can be configured.
3. In the **Settings** pane, override the default values as needed. Overrides are shown in **bold** and by an **X** next to the new value. See
   - You can resize columns in the **Settings** pane as needed by selecting and moving the boundary between column headers.
   - Symbol Wizard and Custom Property settings are visible by default. Overrides are automatically hidden from users.
     - You can only restore the visibility of a Symbol Wizard setting with an override by removing the override.

       Note: This behavior is unique to Symbol Wizard overrides. You can change visibility settings for both Custom Property overrides and attribute overrides.
     - You can restore the visibility of a Custom Property with an override by selecting the Visibility icon until it is unmuted. The icon will have a small X in its upper right quadrant to indicate that Visibility has been overridden.
   - Switching a Custom Property string type between expression/reference and static text is an override (value is displayed with **bold** type with an **X** next to the value and in the Visibility setting).
   - Attribute settings are hidden by default. You can override the Visibility settings for attributes that use their default setting, as well as for attributes that have an override.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: Attribute, Symbol Wizard, and Custom Property values that are marked as visible can be overridden by users as they configure instances.

Note: The total number of configured settings (visibility settings and value settings) is shown in the **Associations** pane, and is reflected in the **Choices and Options** pane. See Attribute and graphic overrides, Attribute and graphic overrides for additional information.

Attribute and graphic overrides

Override values for an attribute or graphic (with a Symbol Wizard or custom properties) can be configured for each associated Choice or Option. When you associate an attribute or graphic, the default values are listed in the **Settings** pane. To configure values that apply to the selected Choice or Option, enter the values in the **Settings** pane. If the attribute or graphic is associated with multiple Choices or Options, different values can be used for each one. Values for the Choices and Options are saved when you save the object. If the setting is marked as visible, users can override your setting (or override the default, if you have not configured a setting) as they derive and configure instances from the Object Wizard.

Graphics with overrides that affect their shape, for example, certain Situational Awareness Library graphics, are rendered with the applicable shape-changes as defined by the overrides.

Note: If you override a Symbol Wizard setting, the only way to make the setting visible again is to remove the override. Attributes and custom properties with overrides can be made visible again by selecting the visibility checkbox.

A configured setting is defined as any change from the default configuration of the attribute or graphic. This includes changing a value or visibility. For example, you might change the default Boolean value from true to

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

false, or you might change an integer value from 5 to 10, or change engineering units from liters to cc. Changing the visibility of a setting also counts as a configured setting.

- The default visibility setting for attribute values is hidden (Visibility icon is muted (grayed-out) in the **Settings** pane).
- The default visibility setting for Symbol Wizard and custom property values is visible (Visibility icon is unmuted in the **Settings** pane).
- Visibility overrides are indicated by a small X in the upper right quadrant of the visibility icon.

The total number of configured settings for each individual attribute and graphic is displayed in the **Associations** pane. The total number of configured settings for ALL attributes and graphics associated with an individual Choice or Option is displayed in the **Choices and Options** pane.

Configurable settings for attribute and graphic values

Graphics and attributes can be configured for each Choice or Option with which they are associated. When you initially associate an attribute or graphic with a Choice or Option, the default settings of the attribute or graphic are listed in the **Settings** pane.

Note: Icons for attribute and graphic settings are defined in Object wizard user interface details.

To override default values, enter the new values in the **Settings** pane **(1)**. If the attribute or graphic is associated with multiple Choices or Options, different overrides can be used for each one. These configured (override) values are saved when you save the template.

An override is defined as any change from the default graphic configuration. For example, you might change the default Choice of a Symbol Wizard, or change a custom property or attribute value. Changing the visibility of a setting is also considered an override. By default, graphic settings are visible in derived objects, while attribute settings are hidden by default.

If the setting is marked as visible, users can enter overrides when they use the **Create New Asset** editor to configure a new instance. If the setting is hidden, it is not exposed to users and therefore the default value cannot be overridden.

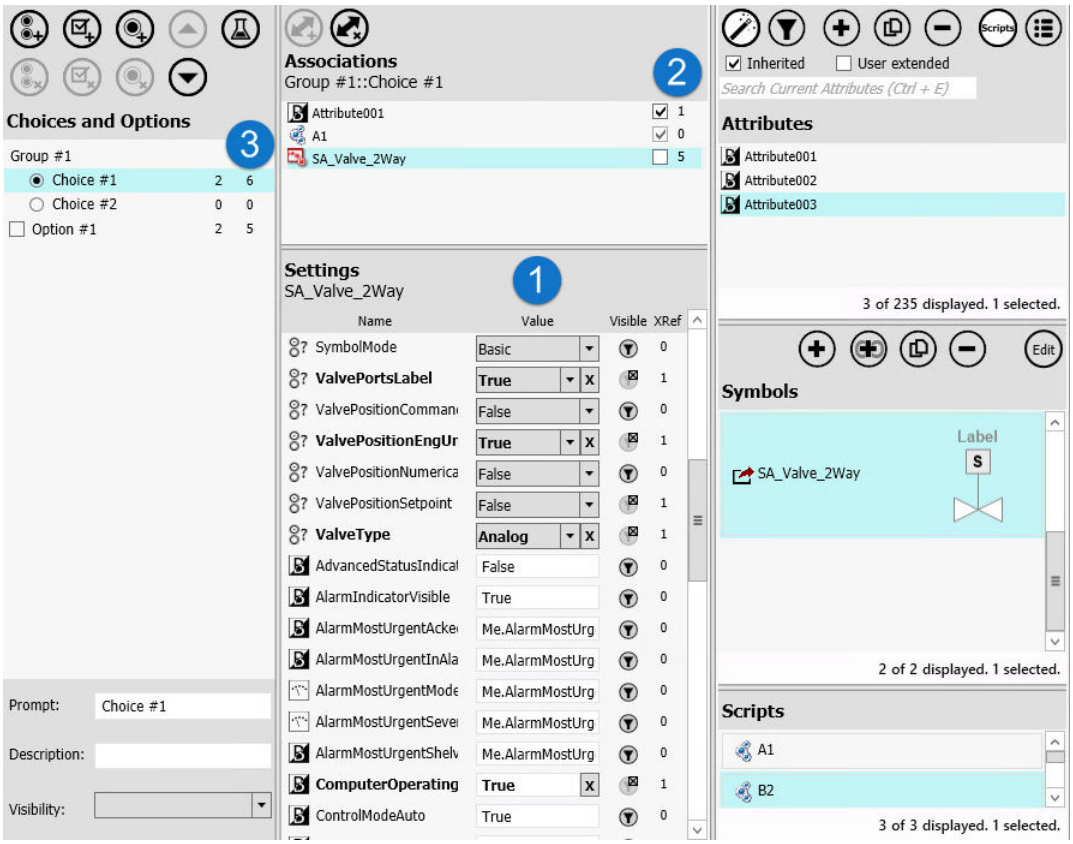
Symbol Wizard overrides are automatically hidden and cannot be changed back to visible, except by restoring the original setting. The names of configured settings are displayed in bold type, and have an "**X**" next to the setting. To restore the original setting, clear the **X**.

Configured visibility settings have a small x in the upper right quadrant of the visibility icon. To restore the original, click on the icon twice to clear the override. If the original visibility state was visible, the icon will cycle through override - hidden, override - visible, and then return to its original state of visible (no override).

Custom property and attribute overrides are also hidden automatically. However, you can override the visibility setting for a setting that has an override value and make it visible again.

The total number of configured settings for each individual attribute and graphic is displayed in the **Associations** pane **(2)**. The total number of overrides for ALL attributes and graphics associated with a Choice or Option is displayed in the **Choices and Options** pane **(3)**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Object wizard derivation

You can derive child templates from templates with Object Wizards. Creating a second (or deeper) level of derivation from a template with an Object Wizard lets you extend the functionality of the original Object Wizard, and make specialized versions for specific applications. See Reasons to derive a template from an object wizard for additional information.

Set visibility of choices and options in a derived template

The original Object Wizard cannot be changed in the child derived template. However, you can set the visibility of Choice Groups, Choices, and Options. You can also set different defaults for Choice Groups, and you can turn Options on or off.

To set Object Wizard visibility in a derived template, do any of the following:

1. Select the visibility icon next to a Choice Group, Choice, or Option.
   - When you hide a Choice Group, all its Choices are hidden as well.
   - When you hide a Choice, the remaining Choices in the Choice Group are still visible.
   - When only one Choice is visible in a Choice Group, the entire Choice Group is hidden.

   Note: The system applies the default values and actions of hidden Options and Choice Groups when a user derives an instance from the template. Visibility does not change configuration, it only affects aspects of the wizard that are visible to the user.
2. Change the default Choice in a Choice Group. This hides the Choice Group.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

You can select the visibility icon to make the Choice Group visible again. You can then set visibility of the Choices individually.

3. Change the default state of an Option. This hides the Option.

   You can select the visibility icon to make the Option visible again.

Reasons to derive a template from an object wizard

There are two main reasons for deriving a child template from a template that contains an Object Wizard:

1. To extend the existing Object Wizard and thus, make it more capable. Available actions to accommodate the specific needs of a target solution include:
   - Add Choice Groups or Options.
   - Add attributes, graphics, or associations.
2. To create specialized versions of templates that preconfigure choices, and thus simplify the task of configuring instances. For example, you can define common configuration aspects that are applicable to a set of similar instances, or modify some of the characteristics of the existing Object Wizard. Available actions to simplify configuration include:
   - Change default Choices or settings.
   - Hide Choice Groups, Choices, or Options.
   - Enter overrides for attribute or graphic settings, or change their visibility settings.

Using derived templates lets you propagate changes made to a parent template, as long as the child template does not have an override for what has changed. See Propagation of attribute and graphic settings for additional information.

If you are familiar with using Application Server, you will notice one other advantage of using derived Object Wizards: there is no need to lock objects. Changes automatically propagate to derived objects (provided the child template does not have an override for what changed in the parent), without explicitly locking attributes in the parent object.

Changes to object wizards in derived templates

After you create a derived template from a template with an Object Wizard, you can only make certain changes to the existing Object Wizard in the derived object.

- You can set overrides for attributes and graphics.
- You can hide or show settings for attributes and graphics.
- You can change visibility and defaults for Choices and Options.
- You can add new Choice Groups and Options (you cannot add new Choices to an existing Choice Group).
- You can add new attributes, graphics, and scripts, and associate them with the existing Object Wizard.
- You cannot remove existing associations from the Object Wizard.
- You cannot remove any existing Choice Groups, Choices, or Options.
- You cannot change the order of any existing Groups, Choices, or Options.
- You cannot change attribute, graphic, or script associations with any existing Choices or Options.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When a Choice or Option is changed from the default, the change is indicated by by an **X** next the modified Choice or Option. To revert to the default, select the **X**.

Similarly, when a visible setting for an associated attribute or graphic has been changed from the default, the change is indicated by an X next to the modified setting. To revert to the default, select the **X**.

Behavior of object wizard components in a derived template

The following list summarizes actions that you can and cannot perform in derived templates that contain an Object Wizard.

- Inherited Groups, Choices and Options cannot be edited.
  - Groups, Choices and Options cannot be renamed or deleted.
  - Their order cannot be changed (cannot be moved up or down).
  - Their prompts, descriptions, and visibility rules cannot be edited.
- You can change the default Choice, or the default state of an Option.
- Associations of inherited items (attributes, graphics, and scripts) with Choices and Options cannot be changed, and inherited associations cannot be removed.
- You can associate an inherited item with additional inherited Choices and Options
- Inherited attribute and graphic values and visibility settings can be overridden.
- Inherited values and visibility settings with overrides can be overridden.

  Note: Even if a value is overridden in the parent object, the value will appear as a default value in the child object. This is because the override from the parent becomes the default for the child.

Object wizard test mode

Test mode lets you simulate how an Object Wizard will behave when a user configures a derived instance. Use Test mode as you develop your Object Wizard to ensure that it behaves as expected. For complex Object Wizards, associated attributes and graphics may not always appear as you had planned. Test mode lets you verify that derived instances will behave as intended.

- Select the **Test** button to enter Test mode.

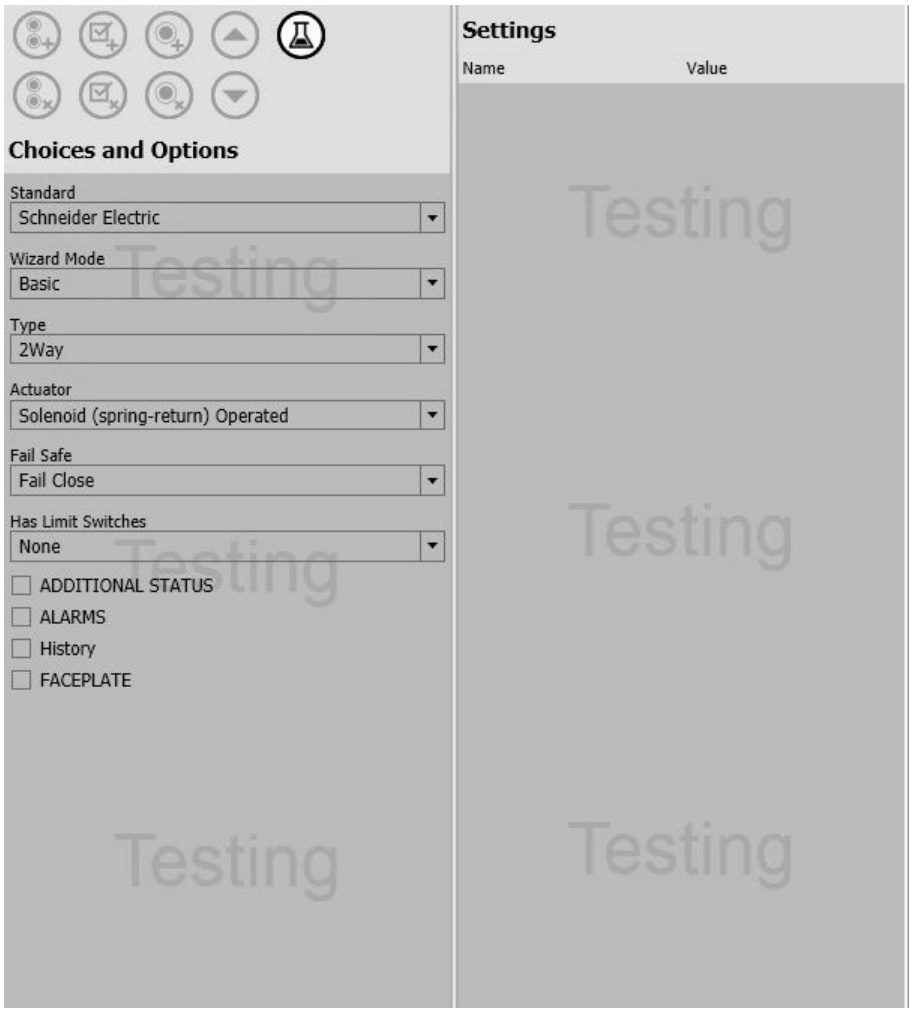
- Deselect the button to exit Test mode.

Visibility expressions determine which Choice Groups, Choices, and Options are visible or hidden (along with associated graphics, attributes, settings, and configured overrides), based on previous selections.

Verify behavior of visibility expressions

1. Open the template, containing the Object Wizard to be tested, in the **Object Editor**.
2. Select the **Test** mode button. The background color of the **Choices and Options** and **Settings** panes darkens, and the **Associations** pane is no longer shown. Choice Groups and Options are displayed, based on their visibility settings.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. Check that the Choice Groups and Options that you expect to be visible are shown.
   - Regardless of the construction of the Object Wizard, the first item (Choice Group or Option) in the Object Wizard is always visible.
   - Choice Groups are displayed as drop down lists, with only the default Choice visible.
4. Check that the expected default Choice for each visible Choice Group is shown.
5. Check that the default state (checked or unchecked) of each visible Option is correct.
6. Cycle through different Choice and Option selections, and verify that subsequent dependent items are hidden or shown as designed.
7. Verify that the settings for associated attributes and graphics are shown in the **Settings** pane, and that settings that should be visible are marked as visible. Verify that settings with visibility turned off are marked as hidden.

About object wizard behavior in test mode

At the most basic level, you can check that Object Wizard elements (Choice Groups, Choices, and Options) operate as intended. Choices and Options should appear in their default states. That is, Options that should be enabled by default should appear with their checkbox checked, and Choice Groups should show the default

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Choice. You can change these selections while in Test mode, to ensure that Choices and Options are correctly configured, including visibility settings and associated attributes, graphics, and scripts. Attribute and graphic settings that should not be changeable by consumers of the Wizard should not be visible.

- Check conditional visibility expressions. See Conditional visibility expressions for choice groups, choices, and options for additional information.
- Check that attributes and graphics are shown in their correct configuration when the corresponding Choice or Option is selected. See Enter Overrides and Set Visibility for Attribute and Graphic Values (see Enter overrides and set visibility for attribute and graphic values on page 198) for additional information.
- Check that attributes and graphics are correctly trimmed. See Trimming for additional information.

Propagation of object wizard changes to derived objects

Most changes made to an Object Wizard propagate through the object hierarchy to all objects derived from the Wizard. Examples of changes that are propagated include:

- Adding, editing, renaming, changing the order of, or deleting Choice Groups, Choices, or Options.
- Adding, removing, or modifying an association (attribute, graphic, or script) with a Choice or Option.
- Changing the default state (setting) of an Option.
- Changing the visibility setting of a Choice or Option.
- Changing the default value or visibility setting of an attribute or graphic associated with a Choice or Option.
- Changing the visibility setting of a value in the **Settings** pane.

Important: With the exception of the first two items in the above list, changes DO NOT propagate to derived templates if the setting is marked as overridden in the derived template.

Object wizard behavior when a default choice is deleted

If you remove a default Choice from a Choice Group, the topmost Choice automatically becomes the new default, unless you explicitly select a different one. The new default propagates to child objects (templates and instances).

You cannot delete a Choice if there are only two Choices in the Choice Group; you must delete the entire Choice Group (a Choice Group, by definition, must contain at least two Choices). In this case, any associated attributes or graphics that were associated with the deleted Choice Group become a part of the object; they no longer have an association with the Object Wizard. Attributes and graphics that are not associated with the Object Wizard are not trimmed from derived instances.

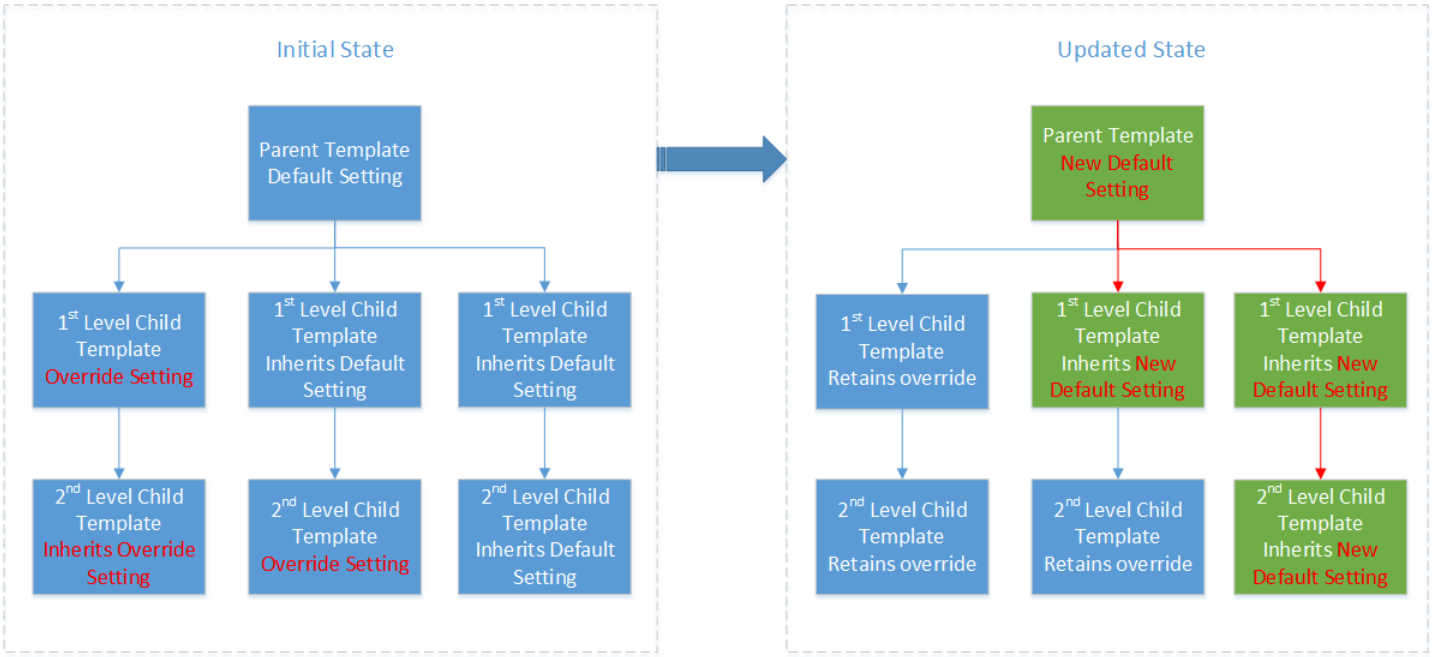
Propagation of attribute and graphic settings

Changes to the default value and visibility setting of an attribute, Symbol Wizard, or custom property value are propagated to child objects, unless the value or visibility setting has been overridden in a derived (child) object. We assume that you do not want to lose an override you entered for a child object due to a change made to the parent.

If a default value is changed in the parent object, but the value has not been overridden in a derived object, the new default value propagates from the parent to child objects where the default value has not been modified.

Likewise, if a visibility setting is changed in a parent template, the change propagates to unmodified child templates. If the visibility setting was modified in the child, the change in the parent does not propagate.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Configure instances

You can use the IDE to configure instances from a template with an Object Wizard, or you can graphically create instances using an editor that is started from the Graphic Editor. If you are using AVEVA OMI for your runtime environment, you can also access the editor through the ViewApp Editor or Layout Editor. See the AVEVA OMI online help for more information about these editors.

- From the IDE, the Object Wizard presents a series of prompts or questions, from which you select an answer to build a representation of a piece of field equipment, such as a pump, valve, PLC, etc.
- If you are using the Graphic Editor or one of the AVEVA OMI editors, select a graphic from the **Toolbox** tab and drag it onto an overview graphic. See Create a new instance graphically for more information.

Configure an instance with the IDE

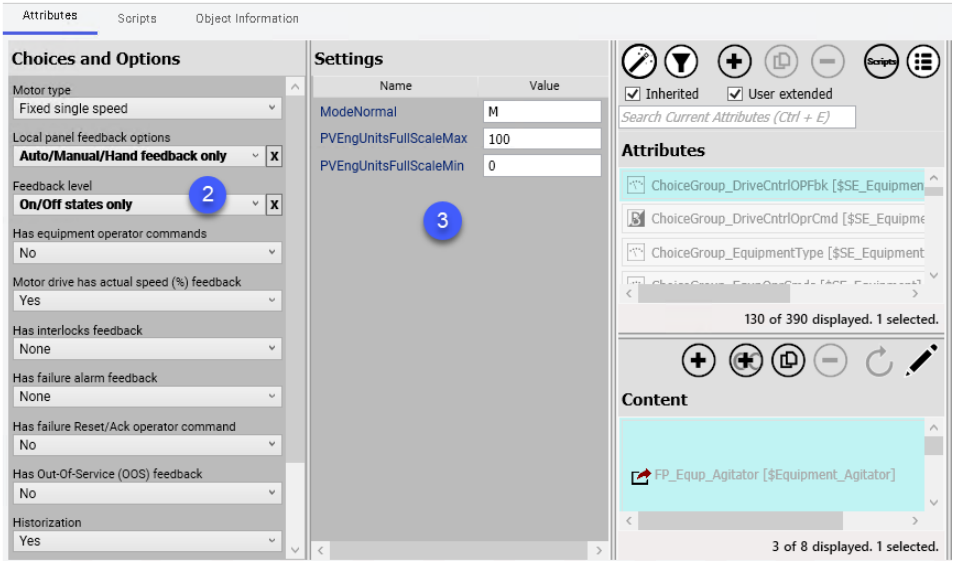
To configure an instance using an Object Wizard in the IDE:

1. In the IDE, open the instance you need to configure.
   - To create and configure a new instance:
     Select the template you want to use, and derive an instance from it. Then, open the instance. See Create an instance for additional information.
   - To change the configuration of an existing instance:
     Select the instance you want to use and open it. See Create an instance for additional information.
2. Start at the top of the **Choices and Options pane**. Either select a Choice from the first Choice Group, or select/deselect the first Option as needed to match the physical asset you are modeling. Depending on the selections you make, attributes associated with the Choice or Option may appear in the **Attributes pane**.
   - If you change from the default Choice, the override is indicated by an **X** and the override is shown in **bold** type.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- If you change the default state of an Option (selected or not selected), the override is indicated by an **X**.
- You can restore a default setting by clearing the **X**.

3. If an associated attribute contains configurable settings, these are listed in the **Settings pane**. Overrides are also indicated by an **X** next to the changed setting and are indicated by **bold** type.



4. Continue sequentially down the list of Choices and Options.
5. Save and close the instance, and then deploy it.

Guidelines for configuring instances with an object wizard

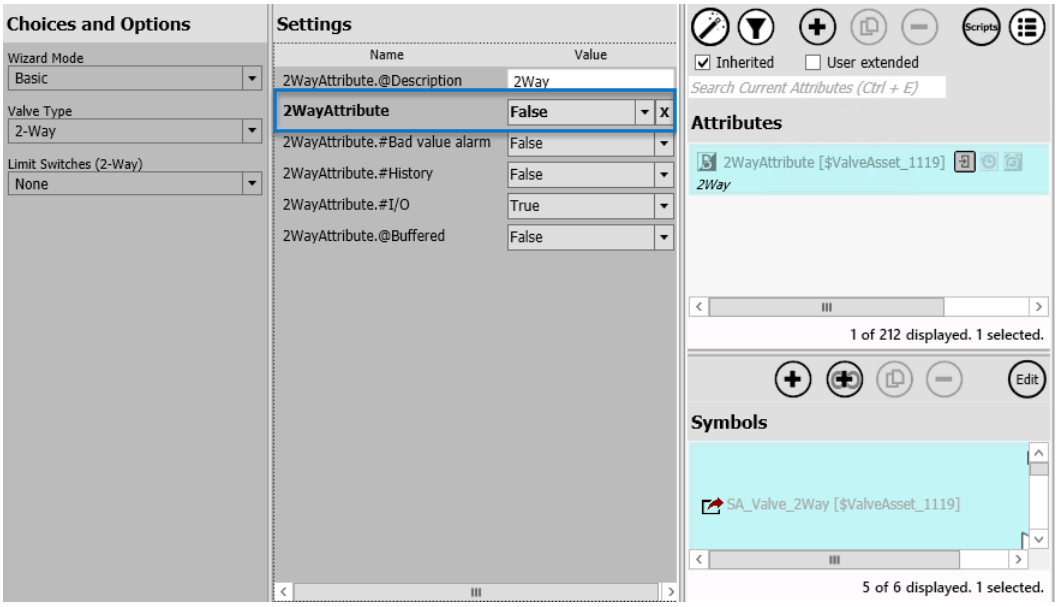
Keep in mind the following behaviors and guidelines when you use an Object Wizard to configure instances:

- Start at the top, with the first item of the Object Wizard.
- The answers you provide to each prompt may cause subsequent prompts to change. Therefore, if you skip selecting a choice or option, the subsequent selections you make could be changed or removed when you return to select a skipped step.
- Some prompts could be added or removed, based on the answers provided to previous prompts.
- If an attribute or graphic is associated with a selected Choice or Option, you may have the option to override certain settings.

Override default settings

When you override a default attribute or graphic setting in the Object Wizard, the override is marked by an X, and the setting and its description are displayed in bold.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Settings can be different types, including Boolean, integer, string, etc. Settings can be used to turn on or off a feature for the object, such as I/O, Historization, or Alarms. Changing one setting can cause other settings to be shown or hidden.

Create a new instance graphically

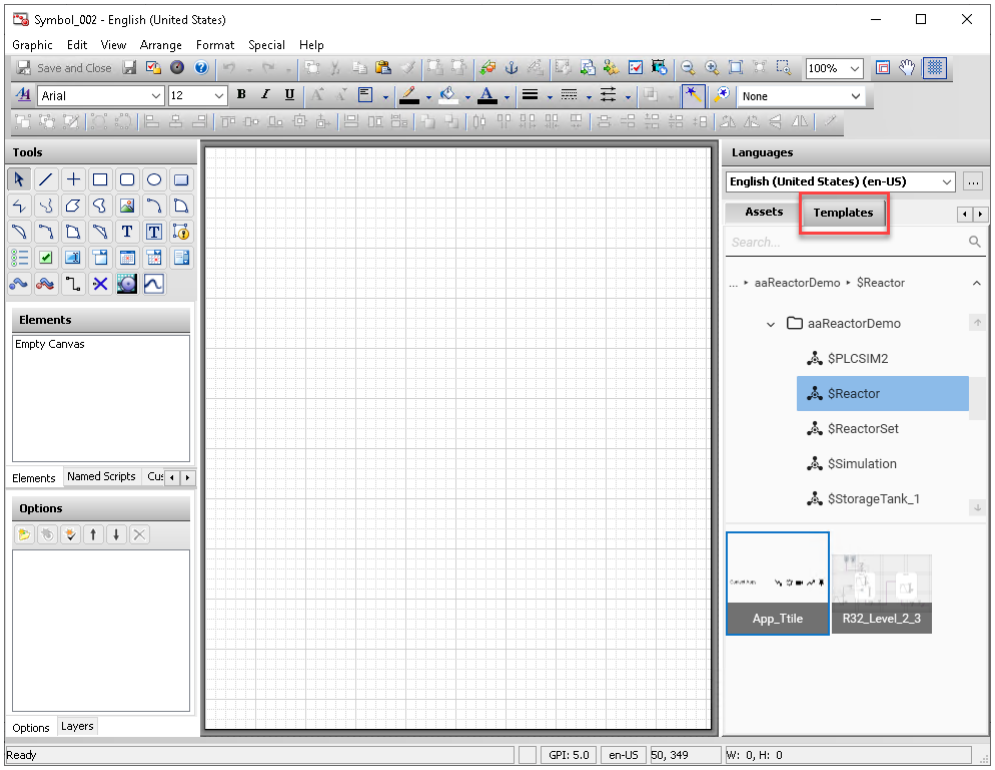
You can quickly and easily add and configure new instances from templates that contains an **Object Wizard** and *at least one graphic,* simply dragging a thumbnail image from the template onto the Graphic Editor canvas. Dropping the graphic thumbnail onto the canvas opens the **Configure New Asset** editor, where you can configure the Object Wizard settings. You can also configure Symbol Wizard and custom property settings, if applicable to the graphic you select. With this editor, you do not need to use the **System Platform IDE** to configure object properties, or spend a lot of time configuring graphic properties in the **Graphic Editor**.

Note: You can add an asset configured from an Object Wizard with a linked Symbol Wizard directly to an AVEVA OMI pane. However, InTouch HMI does not support adding and configuring an asset in an InTouch HMI window that has been derived from an Object Wizard with a linked Symbol Wizard. If you are using InTouch HMI, you must instead embed the symbol in an overview symbol. InTouch HMI does not support directly linking an Object Wizard to graphics in the Visualization folder that use a Symbol Wizard, such as those in the Situational Awareness Library.

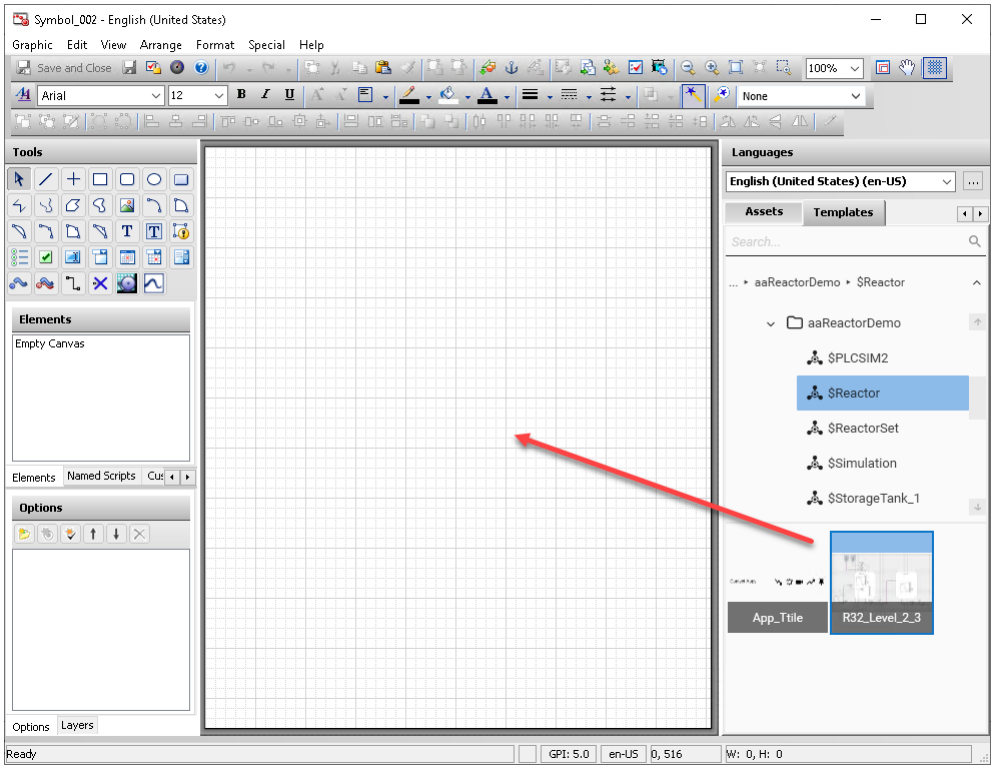
To start the **Configure New Asset** editor:

1. Select a new or existing symbol in the **Visualization** folder. The **Graphic Editor** opens.
2. If necessary, use the arrow keys until the **Templates** tab is shown, then select it.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

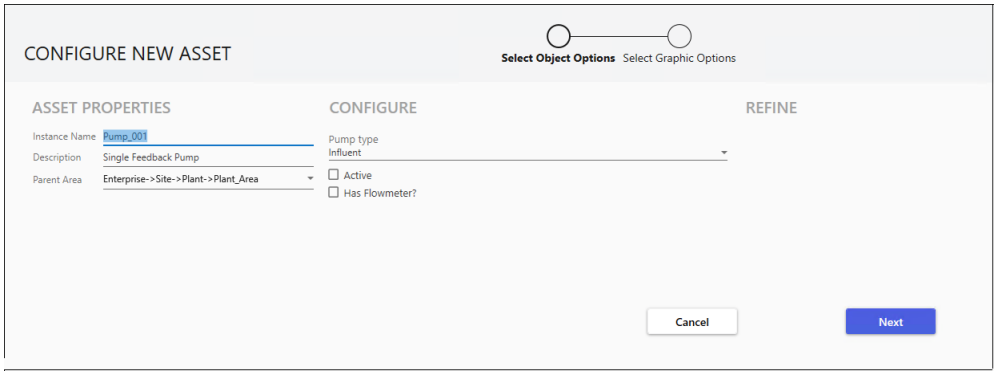


3. Navigate to the template you want to use. The template you select must contain at least one graphic.
4. Select the graphic to use and drag it onto the canvas.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

5. As soon as you drop the image onto the canvas, the **Configure New Asset** editor opens to the **Select Object Options** page.



   Under **Asset Properties** (left column) edit the following entries:

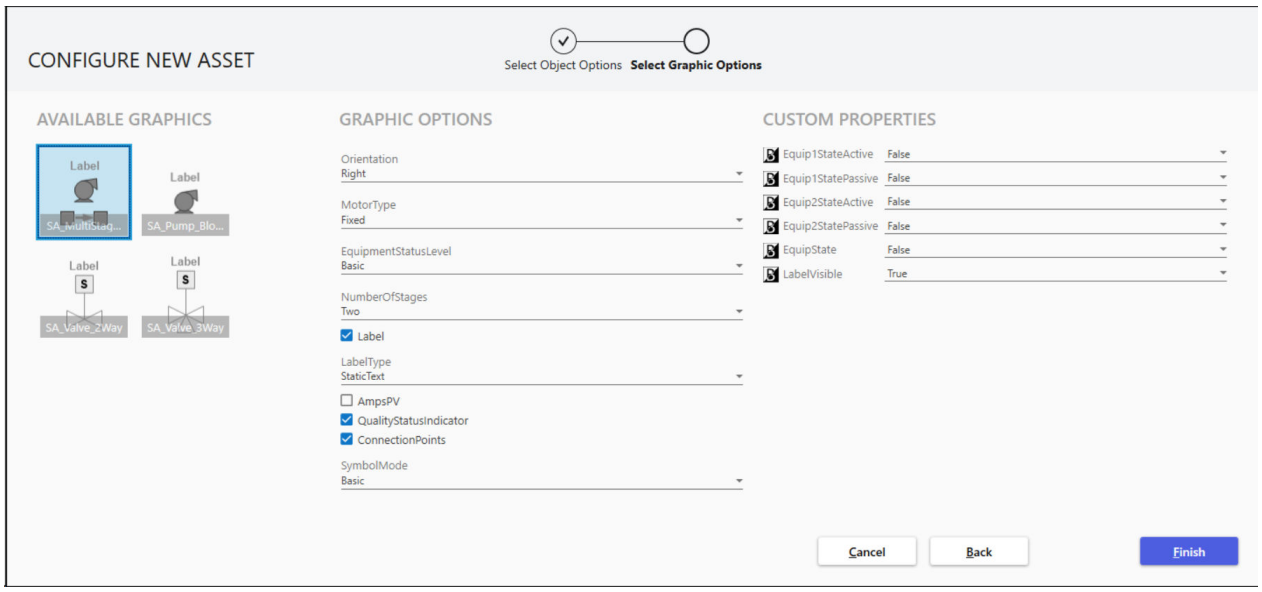
   - **Instance Name** (name of the new asset). The Instance Name is the same as the object tag name. Each Instance Name must be unique within the selected **Parent Area**.
   - **Description** (a description of the new instance).
   - **Parent Area** (select the appropriate parent area for the new instance).
   - Configure the **Object Options** (center column).
   - Configure any properties listed under **Refine** (right column—there are none shown in the example above), then click **Next.**

   Object options determine which graphic options can be used. Only the applicable graphic (or graphics) will be available. For example, if an object includes options that allow it to be configured as a pump or as a conveyor belt, you will not be able to add a pump graphic if you have selected a conveyor belt asset on the **Object Options** page.

   See VisualBuild Object options for additional information.

6. The **Select Graphic Options** page opens. If more than one graphic is displayed, select the one to use, then configure its graphic options. Click **Finish** when you are done.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

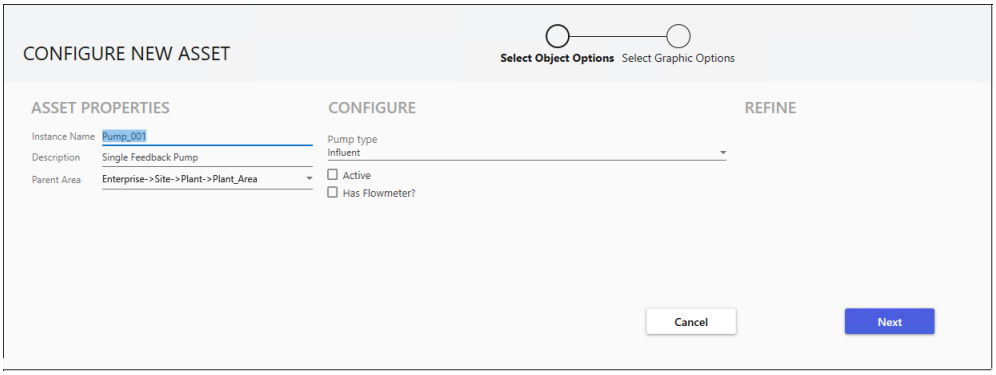


Graphics options are Symbol Wizard options. Select the Symbol Wizard options applicable to the asset you are adding to the overview symbol. Once you select the graphic options, you can configure any applicable custom properties.

7. After you click Finish, the **Configure New Asset** editor closes. Adjust the size and position of the graphic on the **Graphic Editor** canvas as needed. Add additional symbols and graphics as needed.
8. When you have finished adding symbols and other edits, select **Save and Close** to exit from the **Graphic Editor**.

VisualBuild Object options

The **Select Object Options** page consists of three columns for **Asset Properties**, configuration (**Configure**), and settings (**Refine**).



The **Asset Properties** column contains settings for:

- **Instance name** (what the instance will be named within the Galaxy). The instance name is the object's tagname, and can be up to 32 alphanumeric characters, with at least one letter and no spaces. $ cannot be used as the first character.
- **Description** (of the instance). There are no restrictions on the length or characters in the description.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

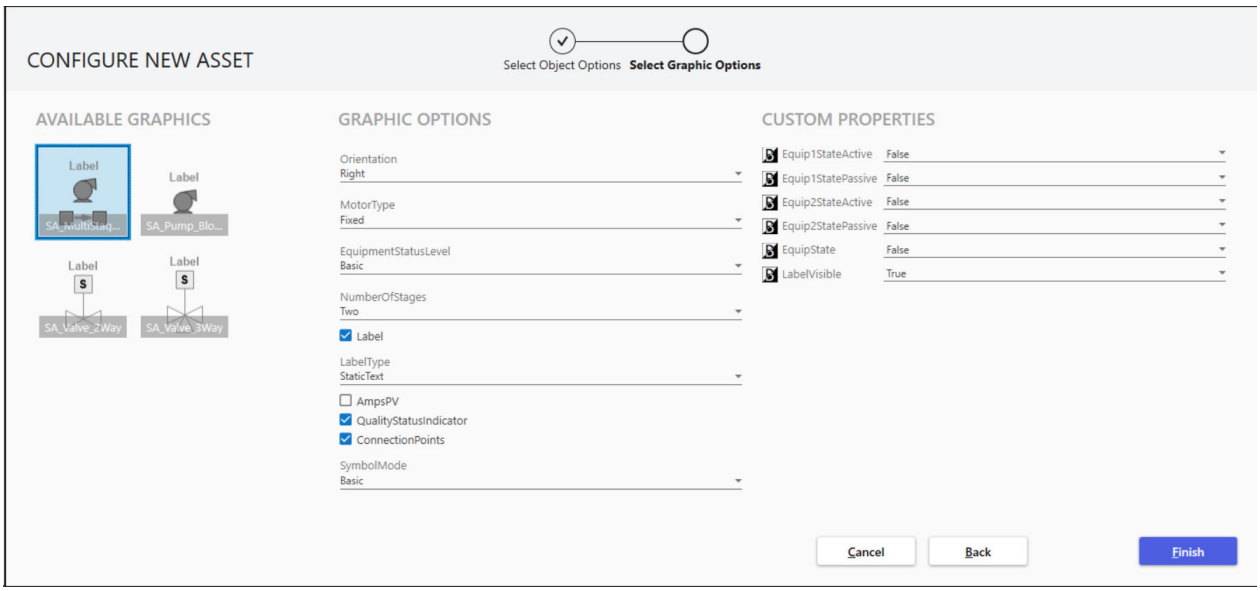
- **Parent area**: This is a drop-down selection that lets you move the instance to the appropriate area of the Galaxy.

The **Configure** column contains object property selections, arranged as a series of prompts with drop-down answers. Start at the top and work down to the end. As you make selections, additional prompts may appear in the Configure area (always below the current prompt).

The **Refine** column contains additional settings that you can set. Typically, the settings will include things like upper and lower limits for scaling and alarms, on/off settings, or other items that further define settings that were selected in the **Configure** column. Some configurations do not contain any settings under the **Refine** column.

VisualBuild graphic options

The **Select Graphic Options** page consists of three columns for **Available Graphics**, **Graphic Options**, and **Custom Properties**.



The **Available Graphics** column displays the graphic or graphics that are associated with the object as configured on the **Select Object Options** page. If more than one graphic is displayed, select the one you want to use. Only one graphic can be selected for each asset you configure.

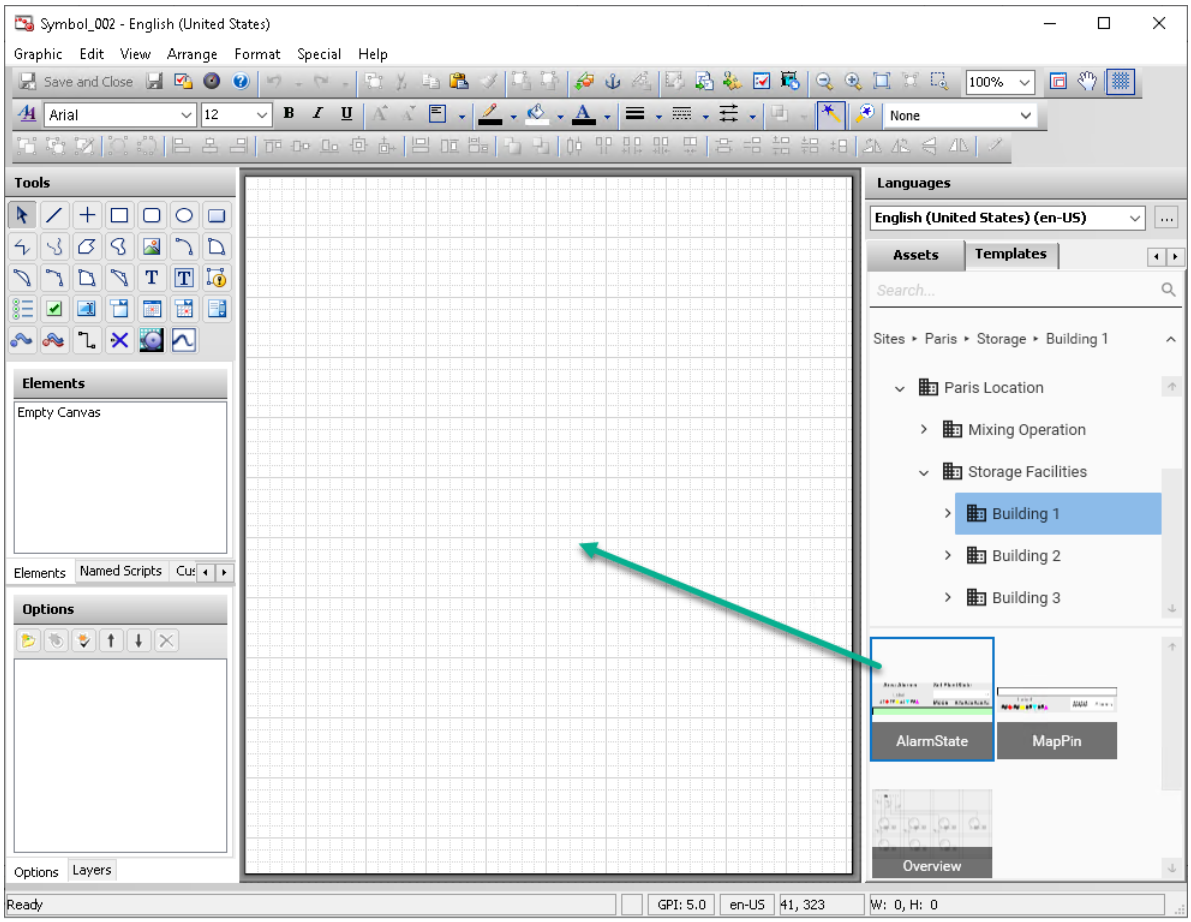
The **Graphic Options** column contains Symbol Wizard configuration selections that apply to the graphic. Typically, these are used to control things like orientation of the graphic, whether or not it has a visible label, to add alarm borders, etc.

The **Custom Properties** column contains additional refinements to set parameters for the graphic. This includes items like alarm limits and priority levels.

Add a configured asset to an overview graphic

1. To add a graphic from a configured instance, select the **Assets** tab in the **Graphic Editor.**

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. Navigate to the asset you want to add to the open graphic and drag it onto the canvas.
3. Adjust the size and position of the graphic on the **Graphic Editor** canvas as needed. Add additional graphics as needed.
4. When you have finished adding graphics and other edits, select **Save and Close** to exit from the **Graphic Editor**.

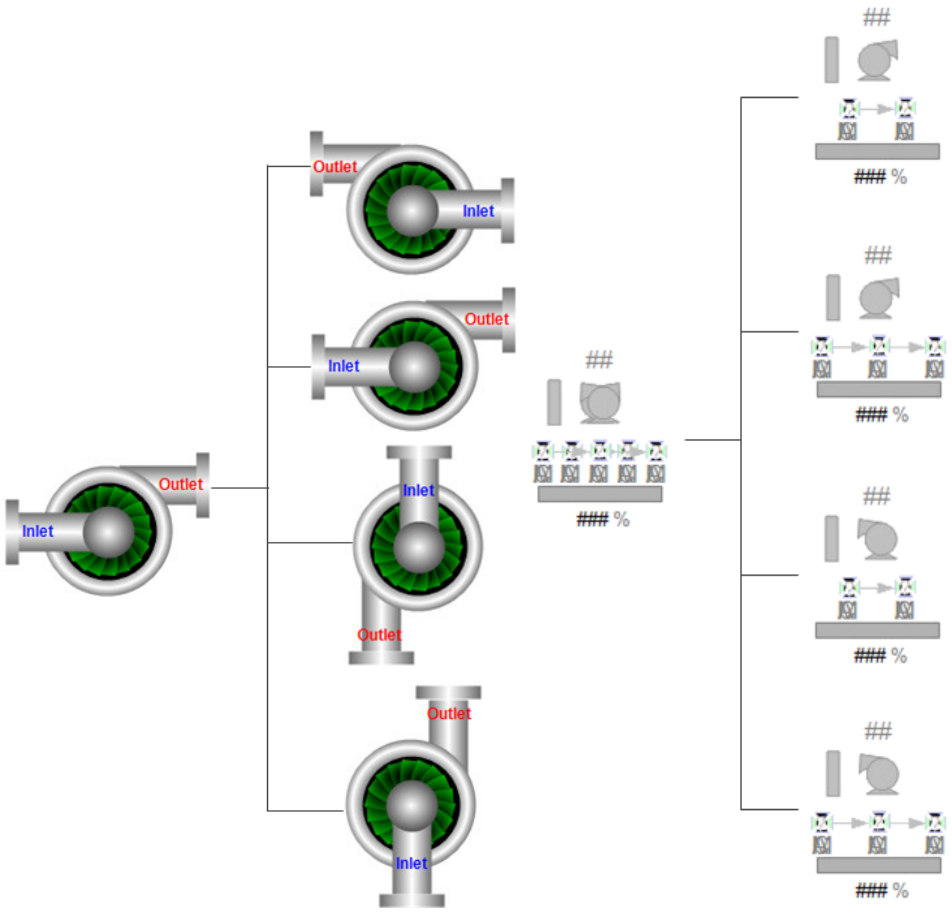
Note: InTouch HMI does not support adding a graphic derived from a Symbol Wizard linked to an object wizard directly onto an InTouch window. The graphic must be contained in an overview graphic.

Symbol Wizard Editor

The Industrial Graphic Editor includes the Symbol Wizard Editor, which can be used to create graphics containing multiple visual or functional configurations called Symbol Wizards. A Symbol Wizard can be embedded into managed InTouch applications like a standard Industrial graphic.

A Symbol Wizard's configuration is selected to meet the requirements of an application. Incorporating multiple configurations in a single Symbol Wizard reduces the number of graphics needed to develop an AVEVA application.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The Symbol Wizard Editor can create Symbol Wizards from traditional Industrial Graphics and Situational Awareness Library graphics. Both types of graphics are located in the Visualization folder in separate toolkit folders. Symbol Wizards are saved in the IDE's Visualization folder and are not associated with any specific AVEVA object template or object instance. Except for the ability to select a specific graphic configuration, Symbol Wizards behave like standard Industrial Graphics.

Situational Awareness Library graphics provide an additional benefit of including defined properties and their associated attributes to more easily create configurations. Some Situational Awareness Library graphics include a Type property to assign a specific function for a graphic configuration. For example, a meter graphic can be configured to represent a thermometer, a pressure meter, or flow meter by simply changing the attribute assigned to the Type property.

Typically, the process of creating and embedding a Symbol Wizard in an application requires the involvement of a Designer and a Consumer. A Designer creates Symbol Wizards using the Symbol Wizard Editor. A Consumer selects a configuration of a Symbol Wizard and embeds the instance of the graphic into a managed InTouch application.

The topics in this section include some general information about creating Symbol Wizards and using them with applications and Object Wizards. For more details and instructions on using Symbol Wizards, see "Working with Symbol Wizards" in the Industrial Graphic Editor help.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Creating Symbol Wizards

A Designer uses the Symbol Wizard Editor to define the various required graphic configurations based on a set of rules and graphic layers. A Designer defines a set of layers, which are used to group a set of graphic elements, custom properties, and named scripts. Graphic elements and other properties can be assigned to no layers or multiple layers. Graphic elements that are not assigned to any layer always appear in all configurations.

A Designer can create a rule for each layer that defines the conditions when the layer is included in a graphic configuration. Rules are assigned with Choice Groups, Choices, and Options. A Designer selects a configuration to be the graphic default that appears when the graphic is embedded in a managed InTouch application.

After creating all configurations, the Designer verifies each configuration of a graphic using the Symbol Wizard Preview. Designers set values in the **Wizard Options** view to verify that each configuration appears as designed based on the layer rules set for the graphic.

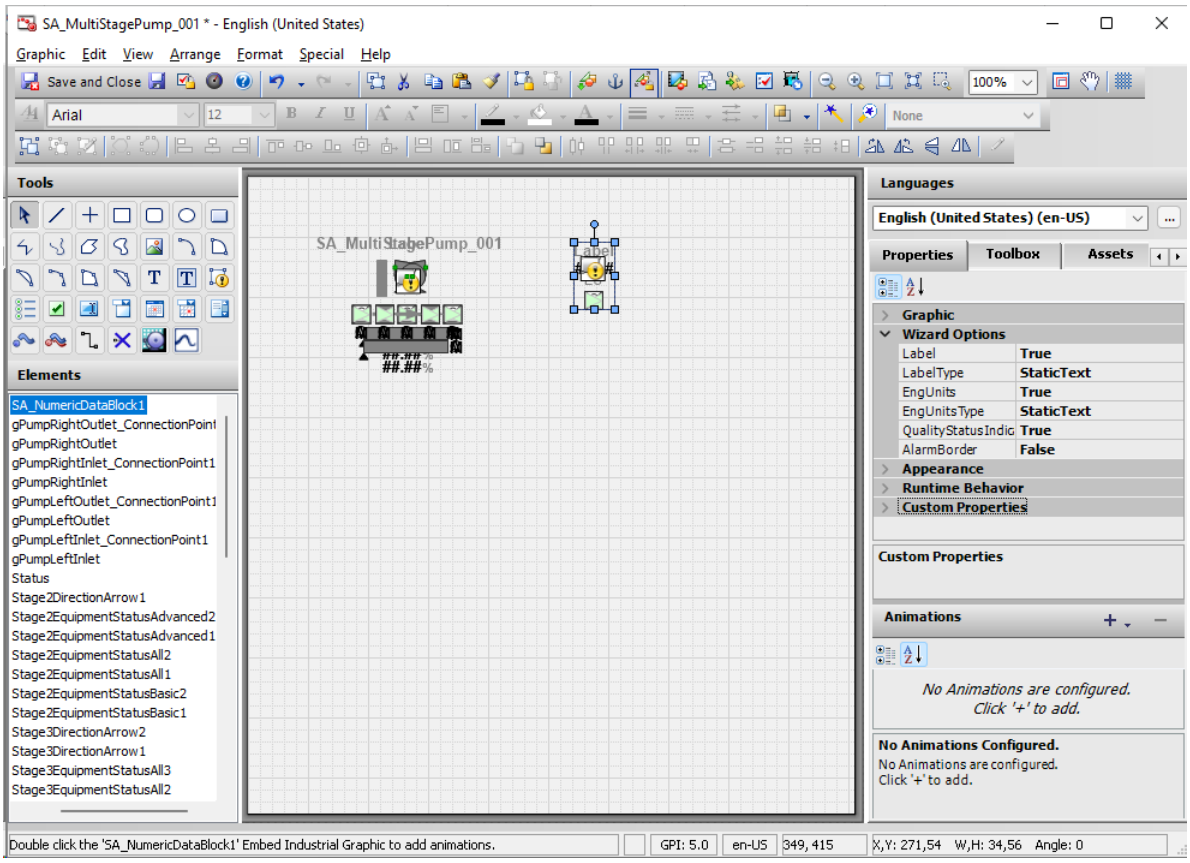
When a Symbol Wizard is finished, the Designer saves it to the Galaxy library so that it is available for use in managed InTouch applications.

For more information about creating Symbol Wizards, see Designing a Symbol Wizard in the Industrial Graphics Editor help.

Embedding Symbol Wizards into an Application

Symbol Wizards are stored in a Galaxy library just like standard Industrial Graphics. When a Consumer selects a Symbol Wizard and embeds it into a managed InTouch application, the Symbol Wizard's default configuration is selected. The Consumer can change the graphic's configuration by changing the options from the Symbol Wizard's **Wizard Options** section of the **Properties** view. Depending on the selected configuration, there can be additional configuration-related properties that can be selected by the consumer.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



After selecting a graphic configuration and changing any properties, the Consumer saves the Symbol Wizard so that it can be embedded into a window from WindowMaker.

While the InTouch application is running, the Symbol Wizard appears as the configuration selected by the Consumer. A Symbol Wizard configuration cannot be changed during run-time.

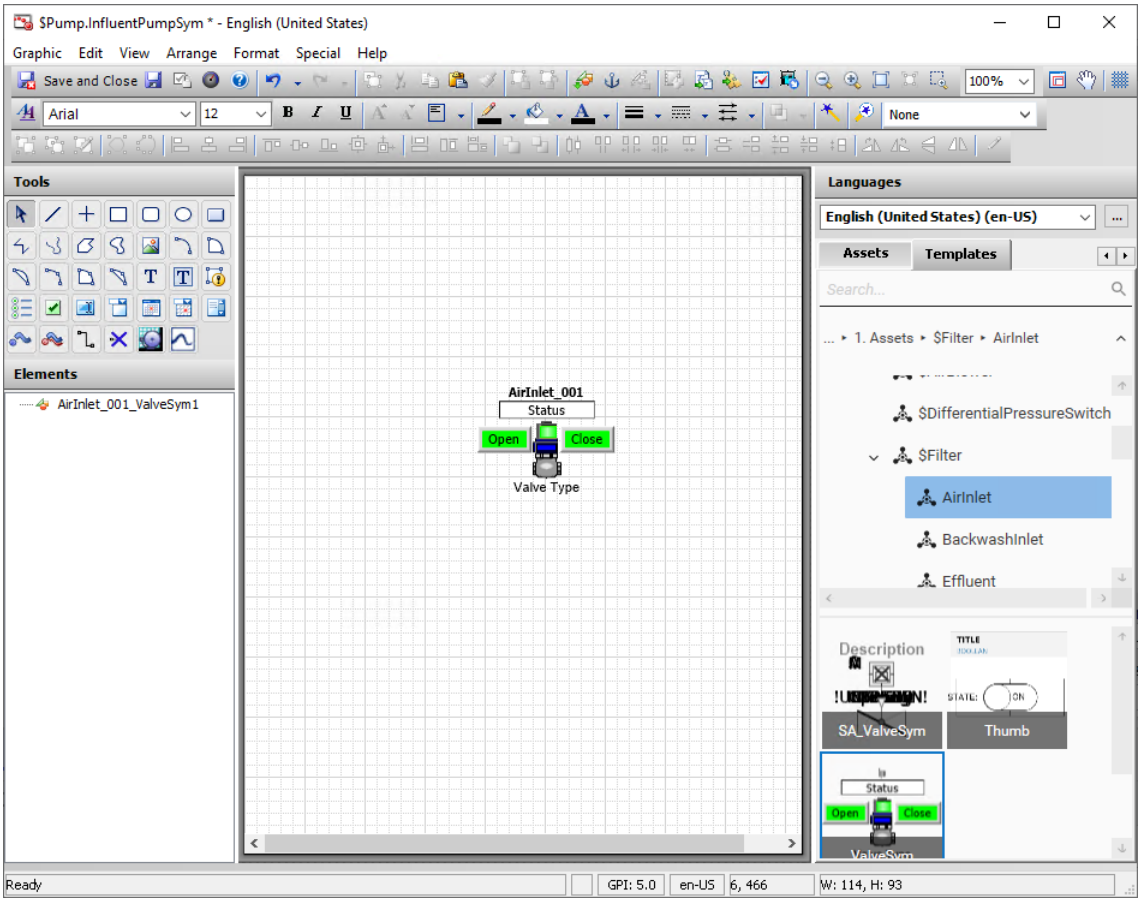
For more information about how to embed Symbol Wizards into managed InTouch applications, see Using Symbol Wizards in an Application in the Industrial Graphics Editor help.

Integrate Symbol Wizards with Object Wizards

Symbol Wizards can be combined with object wizards to give users a simple configuration workflow when adding new assets to an overview graphic. The *Application Server User Guide* describes how to build an object wizard that can be added to any object template. Then, you can associate object wizard choices with different configuration values to be applied to a linked Symbol Wizard.

Users access Object Wizards through the **Templates** tab of the **Graphic Editor** to derive instances. The configured instance is placed on the overview graphic. For detailed instructions about using Symbol Wizards with object wizards, see the topic "Create a New Instance with the Configure New Asset Editor" in the *Application Server User Guide.*

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



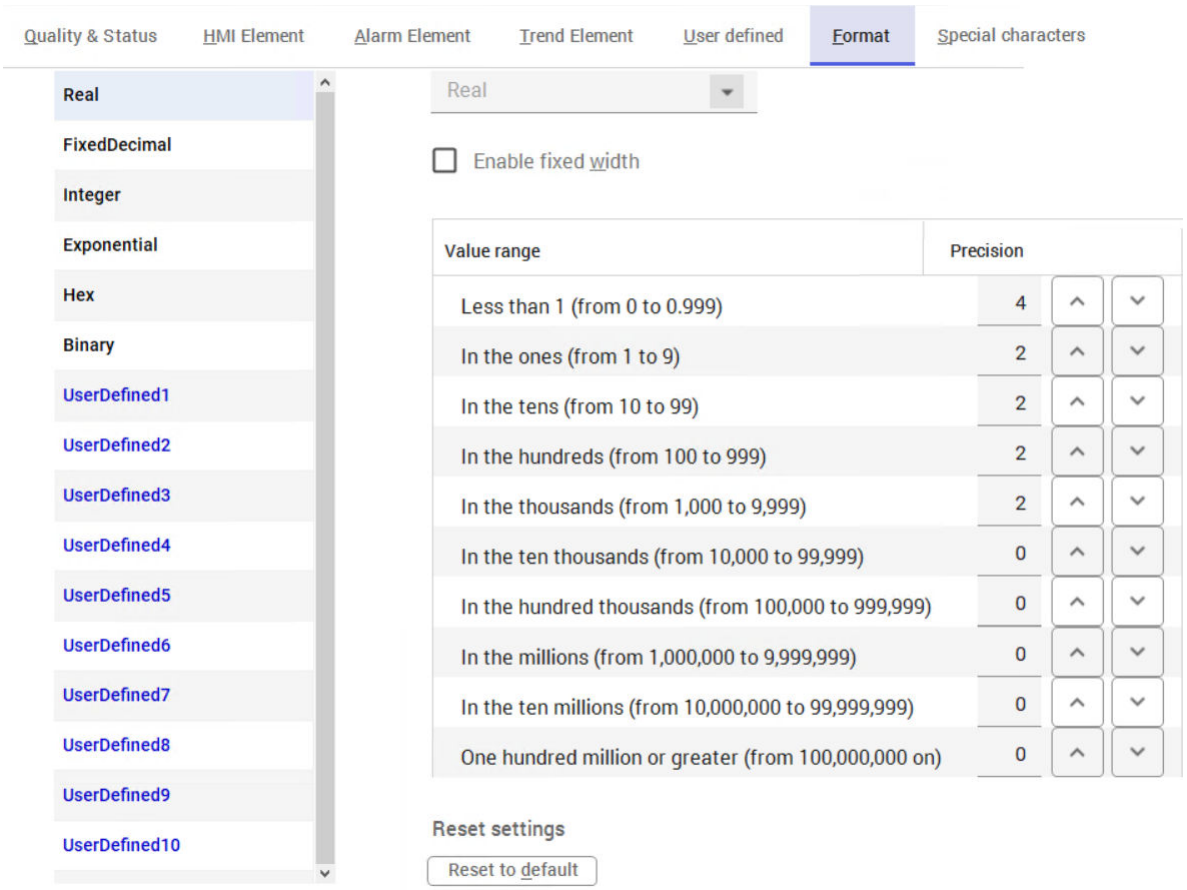
Note: AVEVA OMI lets you add a configured asset directly to a pane through the Layout Editor. However, InTouch HMI does not support adding a graphic derived from a Symbol Wizard linked to an object wizard directly onto an InTouch window. The graphic must be embedded in another graphic (for example, contained in an overview graphic) before you add it to an InTouch window.

Galaxy Style Library

A Galaxy Style Library applies a set of visual and numeric styles to ViewApps in the current Galaxy. Styles can be set for text, lines, fill colors, outlines, etc. The Galaxy Style Library contains the following style elements:

- The **Quality and Status** tab sets the physical appearance of quality and status icons. Also, the **Quality and Status** tab includes a Status group to assign the visual characteristics to error condition status icons.
- The **HMI Element** tab sets the appearance of various HMI aspects, such as background color colors, bad values, default fonts, etc.
- The **Alarm Element** tab sets the appearance of alarms.
- The **Trend Element** tab sets the appearance of the pens and background grid used to show historical trends.
- The **User defined** tab can be used to create custom style elements.
- The **Format** tab specifies the format of numbers that appear in a ViewApp.
- The **Special characters** tab sets characters to be shown at run time for bad quality/no value and values too large for a fixed field.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Open Galaxy Style Libraries

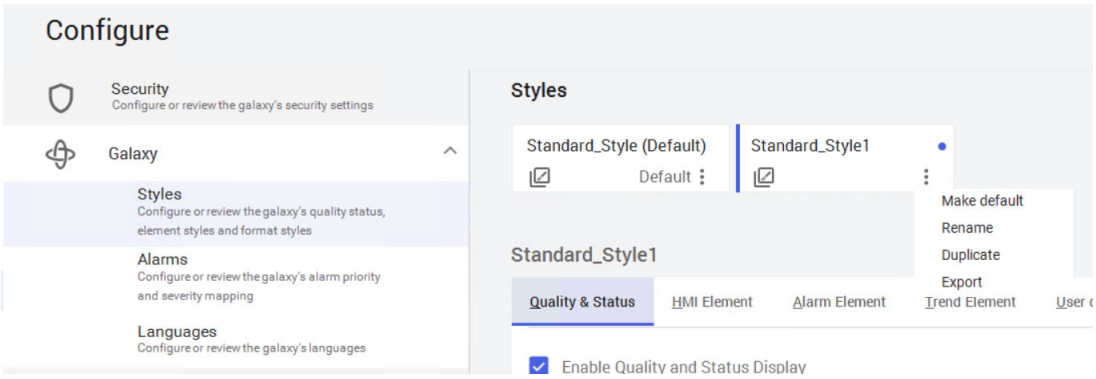
1. Select the **Galaxy** tab.
2. Select the **Configure** option, then select **Galaxy**.
3. Select the **Styles** option.

Manage Galaxy Style Libraries

To manage your collection of Galaxy Style Libraries, select the vertical ellipse above the **Galaxy Style Libraries** field to show a list of management commands.

- Make default (enabled only if more than one Galaxy Style Library is available)
- Rename
- Duplicate
- Export

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Duplicate a Galaxy Style Library

You can duplicate an existing Galaxy Style Library to add it to the list of other Galaxy Style Libraries defined for the Galaxy.

To duplicate a Galaxy Style Library

1. Select the Galaxy Style Library to be duplicated.

   A blue background indicates the Galaxy Style Library is selected.

2. Select the vertical ellipse below the Galaxy Style name.
3. Select the **Duplicate** command from the context menu.
4. Click **Save** to save the duplicated library.

The duplicated Galaxy Style Library appears with a default name of StandardStyle*n*, where *n* is a single digit. A blue dot next to the Galaxy Style is shown until the new style is saved.

Set the Default Galaxy Style Library

The default Galaxy Style Library shown from the list is active when a ViewApp starts in a Galaxy.

To set a default Galaxy Style Library

1. Select the Galaxy Style Library to be set as the default.

   A blue background indicates the Galaxy Style Library is selected.

2. Select the vertical ellipse below the Galaxy Style name.
3. Select the **Make default** command from the context menu.
4. Default is shown in parentheses next to the name of the Galaxy Style Library, and no longer appears next to the previous default library name.

Rename a Galaxy Style Library

Galaxy Style Libraries should be named using a convention that suggests the theme of the element styles or their purpose. Also, a new Galaxy Style Library should be renamed from its default name after it has been added.

To rename a Galaxy Style Library

1. Select the Galaxy Style Library to rename.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

A blue background indicates the Galaxy Style Library is selected.

2. Select the vertical ellipse below the Galaxy Style name.
3. Select the **Rename** command from the context menu.
4. Enter the new name for the library.

   A warning message is shown if you type an invalid name.

5. Click in the Galaxy Style Libraries area to deselect the Galaxy Style Library and verify if you typed a valid name. You can also press **Enter** to validate the new name.

Export a Galaxy Style Library

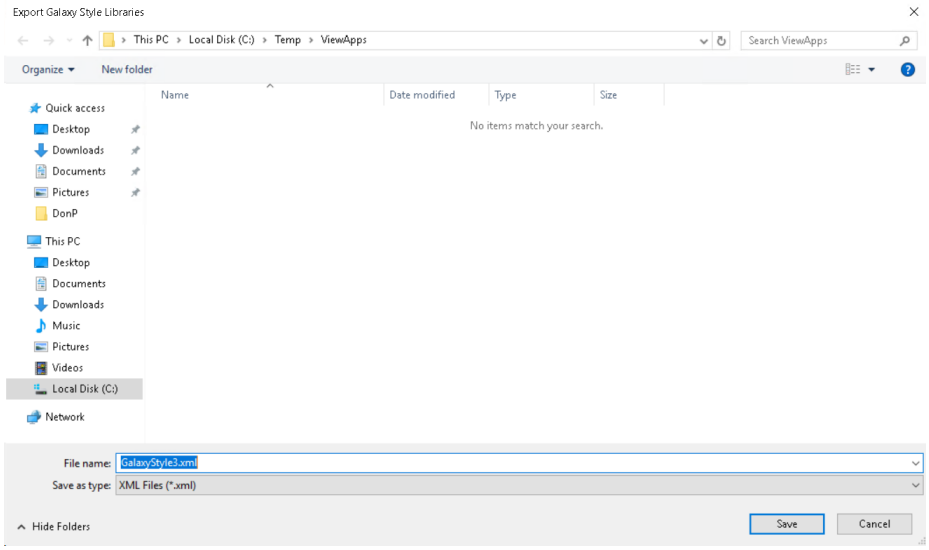
A single Galaxy Style Library can be exported as an xml file to a designated folder location.

To export a Galaxy Style Library

1. Select the Galaxy Style Library you want to export.

   A blue background indicates the Galaxy Style Library is selected.

2. Select the vertical ellipse below the Galaxy Style name.
3. Select the **Export** command from the context menu.
4. In the **Export Galaxy Style Libraries** dialog, select the folder where you want to save the xml file.



5. The default name of the exported xml file is the name of the exported Galaxy Style Library. You can edit this in the dialog.
6. Click **Save**.

Export All Galaxy Style LIbraries

All Galaxies shown in the **Galaxy Style Libraries** list can be exported together to a common folder location.

To export all Galaxy Style Libraries

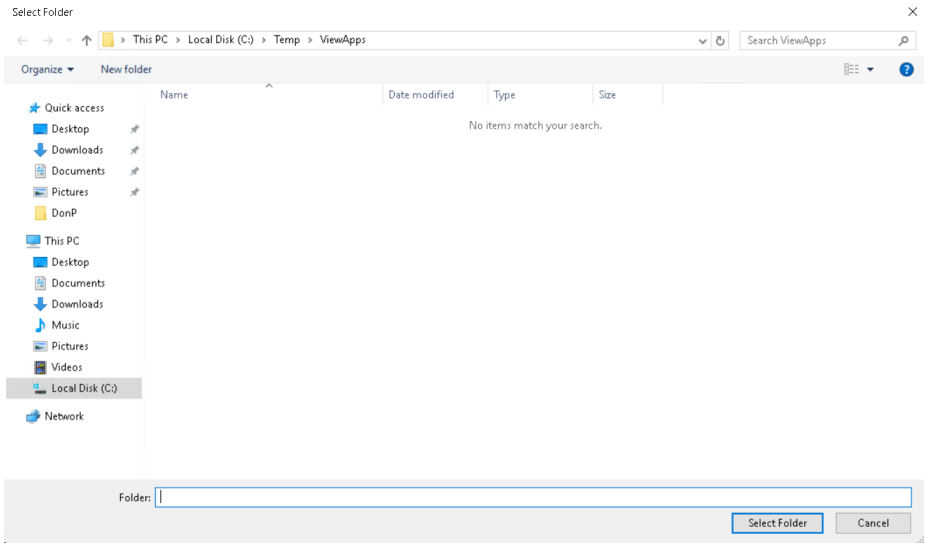
1. Select the **Galaxy** tab in the System Platform IDE.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. Select the **Export** option, then select **All Style libraries**.

   The **Export all Styles** dialog opens.

3. Select the folder where you want to save the exported xml files and click **Select Folder**.



An xml file for each Galaxy Style Library in the Galaxy is created in the folder you selected. The name of each exported xml file is the name of the exported Galaxy Style Library.

Import a Galaxy Style Library

You can import a Galaxy Style Library from an XML file.

Optional Galaxy Style Library XML files are installed with System Platform. These optional Galaxy Style Libraries are located in the ...\Program Files (x86)\ArchestrA\FrameWork\Bin\AdditionalElementStyles folder.

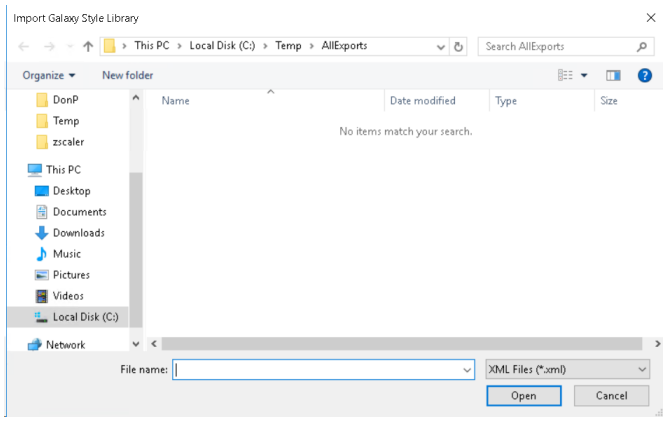
The names of the optional Galaxy Style Library XML files suggest the primary color schemes of the Element Styles within each optional Galaxy Style Library.

To import a Galaxy Style Library

1. Select the **Galaxy** tab in the System Platform IDE.
2. Select the **Import** option, then select **Style Libraries**.

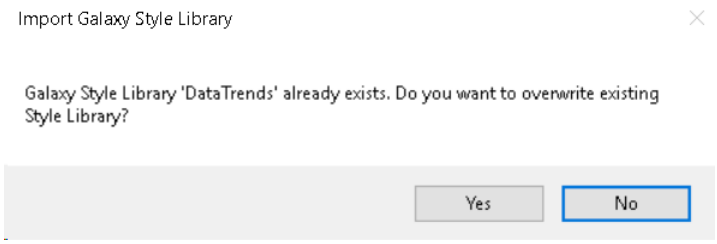
   The **Import Style Libraries** dialog opens.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. Select the folder containing the Galaxy Style Library to be imported.
4. Select the Galaxy Style Library xml file you want to import, then click **Open**. You can select multiple files for import.

   If the Galaxy Style LIbrary to be imported is a duplicate of a Galaxy Style Library that is already listed, a warning message appears with options to overwrite the existing Galaxy Style Library or not.



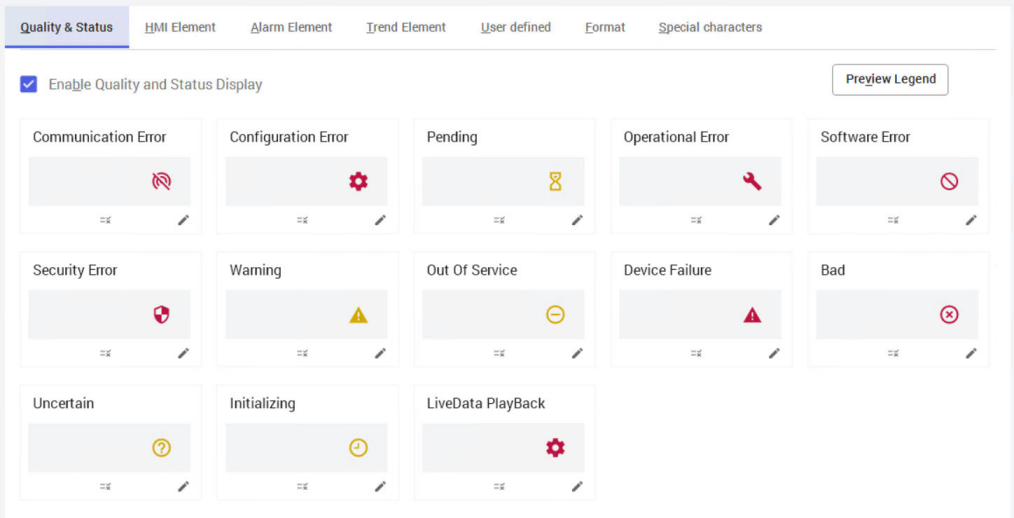
After successfully importing a Galaxy Style LIbrary, its name appears in the **Galaxy Style Libraries** list.

Quality and Status Styles

Data quality and status can be represented by a set of icons that appear in a running ViewApp. The physical appearance of icons indicates current quality and status of the data associated with an icon. The **Quality and Status** tab shows a list of status style overrides that can be modified to set the visual characteristics of quality and status states at run time.

Quality and Status (QS) indicators are listed by order of priority (left to right, with the first row listing the highest priority indicators). At run time, if more than one QS indicator is active, only the highest priority indicator is shown.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The appearance of text, fill, line, outline, blink can be set as overrides of the following **Quality and Status** icons:

- Communication Error (highest priority)
- Configuration Error
- Pending
- Operational Error
- Software Error
- Security Error
- Warning
- Out of Service
- Device Failure
- Bad
- Uncertain
- Initializing
- LiveData Playback (lowest priority)

For more information about the LiveData Playback QS indicator, see Historical Playback in the *AVEVA OMI Help*.

Understand Quality and Data Status

To show a specified status or quality at run time, you can:

- Use a Status element that shows an icon. It indicates the status or quality of specified attributes directly or those used indirectly in elements.
- Change the appearance of animated elements based on the status and quality of AVEVA attributes they use.

Showing Quality and Status with the Status Element

The Status element cannot monitor attributes of:

- Elements that are not in the same hierarchy level in the Elements List.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Elements that use the attributes in scripts.
- Elements that are invisible at run time.

For more information on how to configure status on an element, see Configuring Animation for a Status Element.

For more information on how to configure the appearance of a status element, see Setting the Appearance of a Status Element.

Showing Quality and Status by Overriding

You can override the appearance of an animation depending on its configured attributes by:

- Overriding the animation or changing the appearance of the element.
- Drawing an outline around the element.

Overriding the appearance of animations also applies to:

- Elements contained in groups.
- Elements in graphics embedded in other graphics.

Overriding the appearance of animations does not apply to:

- Elements that use the monitored attribute in scripts.
- Elements that are invisible at run time.

For more information, see Overriding Element Appearance Depending on Quality and Status of its Attributes.

Set Quality and Status Styles

You can configure your graphic to show non-good status and quality of attributes in different ways:

- A status element shows a specific icon depending on the quality and status of configured attributes or elements. See Using Status Elements, Using Status Elements.
- The text, fill, or line appearance of elements can be overridden depending on the quality and status of the attributes they reference. See Overriding Element Appearance Depending on Quality and Status of its Attributes.
- Elements are drawn with an outline depending on the quality and status of the attributes they reference. See Overriding Element Appearance Depending on Quality and Status of its Attributes.

Using Status Elements

Status elements show a specified graphic depending on the quality and status of:

- Attributes and tags configured for specific animated elements.
- One or more specified attributes or tags.

You can assign status elements to an animation in three steps:

1. Draw the status element on the canvas.
2. Associate the status element with animated elements on the canvas and/or attributes that provide the quality and status data to be monitored.
3. If needed, configure the appearance of the status element.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Drawing the Status Element on the Canvas

You can easily place a status element on the canvas to show an icon that indicates quality and status of attributes or tags contained in selected animated elements and/or specified attributes and tags.

You do this as you would with any other element. For more information, see Drawing and Dragging Elements.

Configuring the Status Element

You can associate the status element with:

- Animated elements that use attributes and tags that provide the quality and status that is to be monitored.
- Attributes and tags that provide the quality and status to be monitored.

In both cases, the appearance is set by the settings in your application. For example, in AVEVA ViewApps it is set in the **Quality and Status** tab of the **Configure Galaxy Style Library** dialog box.

For more information on how to configure this animation, see Configuring Animation for a Status Element.

Setting the Appearance of a Status Element

You can set the appearance of a status element depending on the quality and status of its referenced attributes and tags and/or attributes and tags used in its referenced elements.

You can also preview the appearance of a status element. For more information, see Previewing all Status Appearances, Previewing all Status Appearances.

You can reset the appearance of a status element to its default. For more information, see Resetting an Override Appearance to its Default, Resetting an Override Appearance to its Default.

Overriding Element Appearance Depending on Quality and Status of its Attributes

You can configure any animated element to appear differently depending on the quality and status of its associated attributes and tags if your HMI application supports it.

For animated elements, you can:

- Override the appearance of the text font, style, and blinking.
- Override the appearance of the fill style and blinking.
- Override the appearance of the line style, weight, pattern, and blinking.
- Preview all status appearances in one dialog box.
- Reset the status appearances to their defaults.
- Use an outline to indicate a specified status or quality.

Note: Instead of overriding the appearance of elements on the canvas, you can use a status element. The status element shows an icon representing quality and status of monitored attributes and tags.

Overriding the Text Appearance of Elements to Indicate Non-Good Status or Quality

If your application supports it, you can configure the text appearance of animated elements with attributes that have non-good status or quality are overridden with a specific text appearance.

Overriding the Fill Appearance of Elements to Indicate Non-Good Status or Quality

If your HMI application supports it, you can configure the fill appearance of animated elements with attributes that have non-good status or quality are overridden with a specific fill appearance.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Overriding the Line Appearance of Elements to Indicate Non-Good Status or Quality

You can configure a Galaxy so that the line appearance of animated elements with attributes that have non-good status or quality are overridden with a specific line appearance.

To override the line appearance of elements specified by a Status element

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select **Enable Quality and Status Display**.
5. Select the edit icon for a status or quality.
6. Click the **Line** tab.
7. To override the line pattern, select the **Line Pattern** checkbox and select a line pattern from the adjacent list.
8. To override the line weight, select the **Line Weight** checkbox and type a new line weight in the adjacent box. Alternatively, you can increment or decrement the line weight by clicking the adjacent boxes.
9. To override the line color:
    a. Select **Line Color** checkbox.
    b. Click the color box.
    c. Select a line color from the **Select Line Color** dialog box. For more information, see Setting Style.
10. To override the line blink behavior:
    a. Select **Blink**.
    b. Select a blinking speed from the **Speed** list (Slow, Medium, or Fast).
    c. Click the color box.
    d. Select a line blink color from the **Select Blink Color** dialog box. For more information, see Setting Style.
11. Click **Apply**.

Adding Outlines to Elements to Indicate Non-Good Status or Quality

You can configure the Galaxy so that animated elements with attributes that have non-good status or quality are shown with an outline.

To add outlines to elements to indicate non-good status or quality

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select **Enable Quality and Status Display**.
5. Click the **Outline (OI)** tab.
6. Select the edit icon for a status or quality.
7. Select **Show Outline**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

8. To set the line pattern, select **Line Pattern** and select a line pattern from the dropdown list.
9. To set the line weight, select **Line Weight** and enter a line weight. Alternatively, you can increment or decrement the line weight by clicking the adjacent boxes.
10. To set the line style:
    a. Click the color box next to **Line Color**.
    b. Select a line color from the **Select Line Color** dialog box. For more information, see Setting Style.
11. To set the line blink behavior:
    a. Select **Blink**.
    b. Select a blinking speed from the **Speed** list (Slow, Medium, or Fast).
    c. Click the color box.
    d. Select a line blink color from the **Select Blink Color** dialog box. For more information, see Setting Style.
12. Click **Apply**.

Previewing all Status Appearances

You can preview the appearance of all status overrides by showing the **Status Legend** dialog box.

To preview all override appearances

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Click **Preview Legend**. The **Status Legend** dialog box appears.
5. Click **Close**.

Resetting an Override Appearance to its Default

For any status, you can reset the default appearance:

- Text, fill, and line overrides.
- Outline settings.
- Status element settings.

To reset a status or quality to its default appearance

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select the edit icon for a status or quality.
5. Click **Reset to default**. All text, fill, and line overrides, status element icons, and outline settings are reset to their defaults.

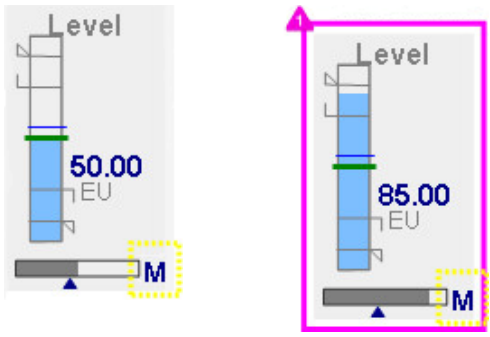
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Element Styles

An Element Style defines a set of visual properties that determine the appearance of text, lines, graphic outlines, and interior fill shown in Industrial Graphics or graphic elements. An Element Style can be applied to a graphic that sets pre-configured visual property values that take precedence over a graphic's native visual properties.

Element Styles provide the means for developers to establish consistent visual standards in their AVEVA OMI ViewApps. An Element Style can define the same visual properties of text, lines, fill, and outlines for all graphics included in a ViewApp.

Likewise, Element Styles can be applied to a graphic during runtime to show the changing status of a piece of equipment in response to events or alarms. The following example shows an alarm border animation Element Style around a meter when the process value shown by the meter transitions into an alarm state.



For information about managing and using element styles, see Element Styles in the Industrial Graphic Editor help.

Details of Element Styles

A Galaxy Style Library includes pre-defined and user-defined Element Styles, plus numerical format styles and special characters used to show exceptions at run time.

- Pre-defined element styles

  Pre-defined element styles have been configured with default values. You can modify each element style's font, fill, line, and outline properties.

  Pre-defined element styles are organized in groups belonging to the **Element Style Overrides** list by their functional purpose:

  - **Quality and Status**
  - **HMI Elements**
  - **Alarm Elements**
  - **Trend Elements**

  Many of the element styles apply to the graphic elements that comprise Situational Awareness Library graphics. An element style can define the visual properties of a graphic like its background color. An element style can also indicate a graphic's current operating state. For example, there is a set of pre-defined element styles that set the visual characteristics of machinery alarm states represented by a graphic.

  For more information about element styles that apply to Situational Awareness Library graphics, see . Then, search for the element style by name in the Situational Awareness Library Graphic help.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- User-defined element styles

  User-defined element styles do not contain preset font, fill, line, and outline properties. You must specify each pre-defined element style's properties to meet the needs of your applications.

- **Format** styles are applied to numerical values, such as set points and process variables. There are pre-defined styles for real, FixedDecimal, Integer, Exponential, Hex and Binary numbers. You can also configure user-defined formats. Format styles define range and precision for values.
- **Special characters** define two different special characters that are shown in alarm mode at run time for the following conditions:
  - Bad quality with no value: default character is "!" (exclamation point)
  - Value too large for fixed field: default character is "*" (asterisk)

You can create your own Galaxy Style Library by overriding the default style configurations and exporting the library. For information about setting Element Style overrides, see Changing Visual Properties of an Element Style.

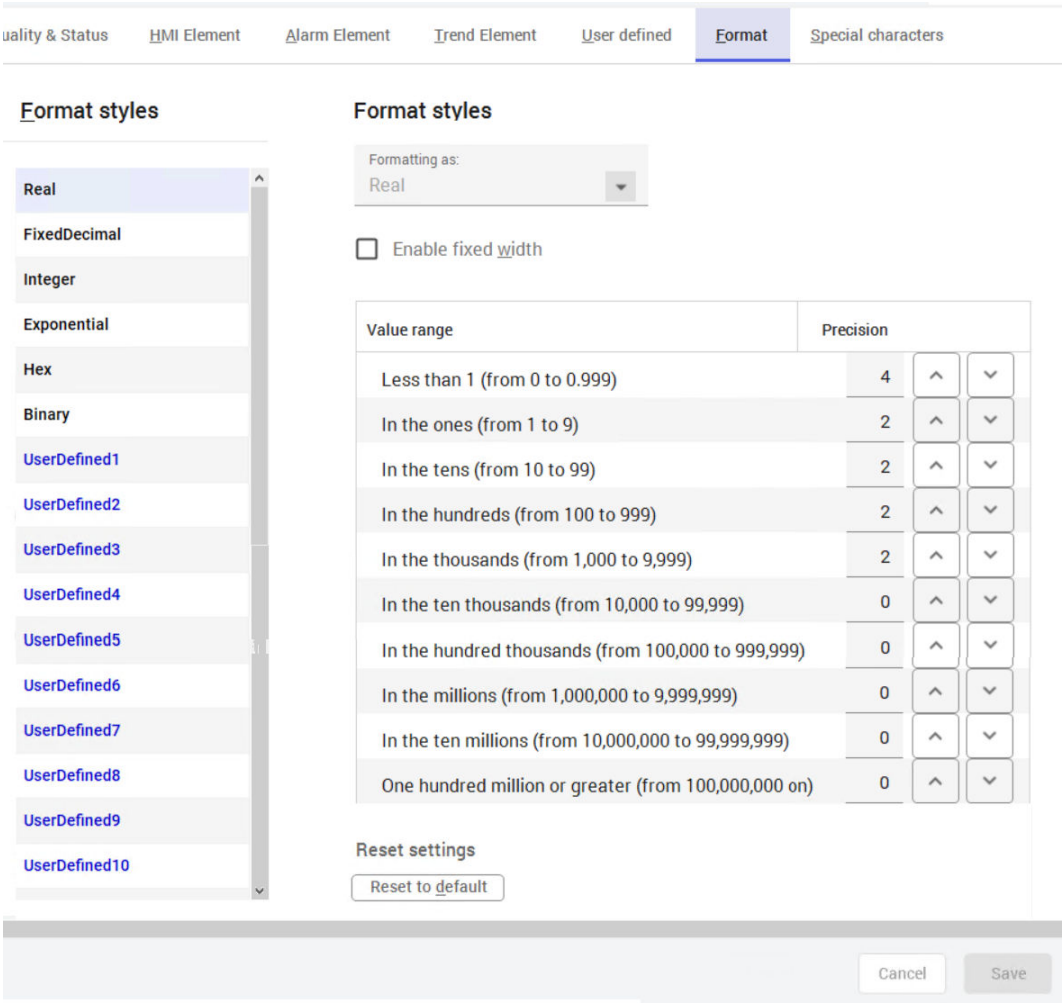
If you want to restore the original default visual properties for your graphic elements, each of the styles pages includes a **Reset to Default** button. The UI Theme page does not have a button to restore the default setting.

Numeric Formats

The **Style Library** page of the backstage includes the **Format Styles** tab. **Format Styles** provides options to individually configure Galaxy-wide styles for common types of numbers used in industrial applications.

Each global number style is assigned a unique name, which cannot be changed. A number style can be applied by name in design time and run time for an analog data type in User Input and Value Display animations. Also, grouped elements support global number styles.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: For information about setting numeric formatting by locale, see Setting Number Format by Regional Locale in the Industrial Graphic Editor help.

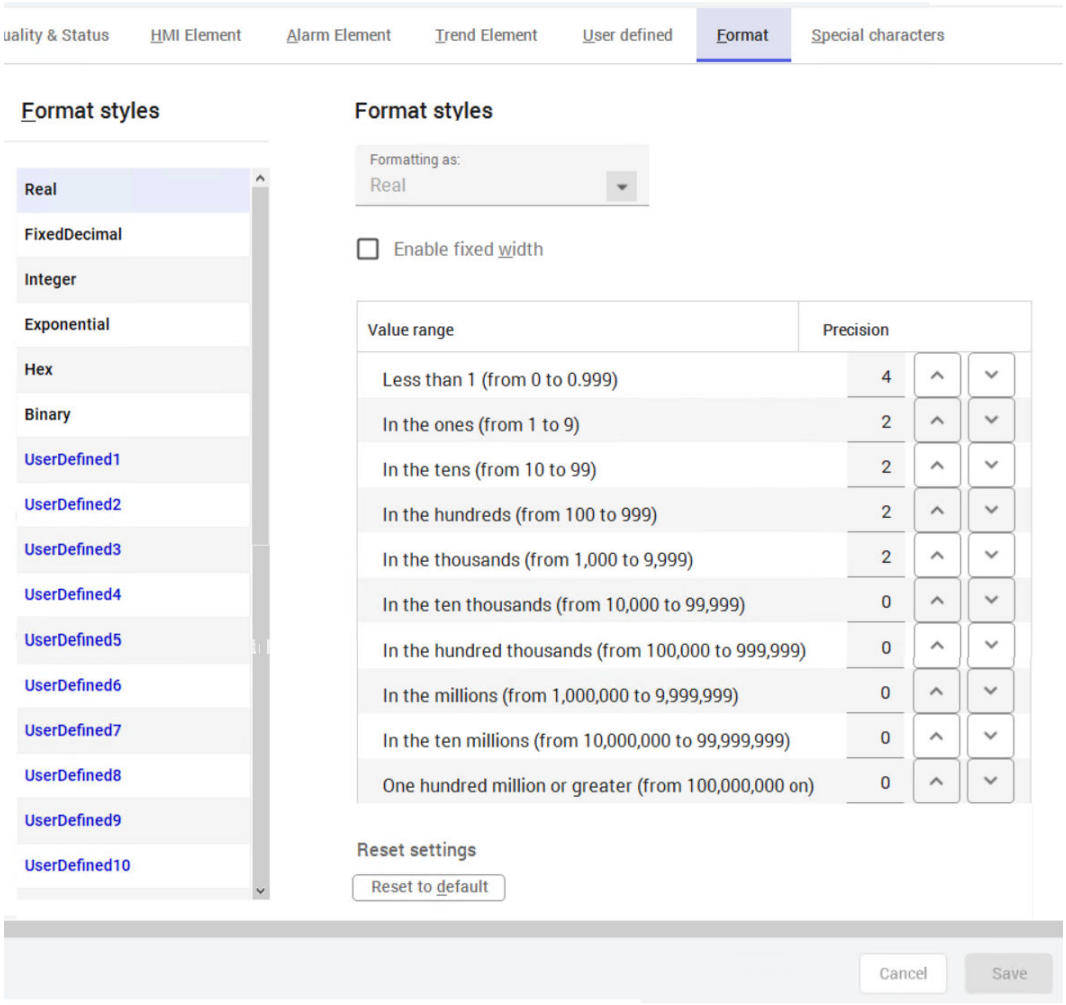
Configure Global Number Styles

You can configure a global number style by changing the values assigned to the properties for each type of number.

To configure a global numeric format style

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Format** tab to show numeric format options.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



4. Select a numeric format style from the **Format styles** list.

   The **Format** tab updates to show options for **Fixed Width**, **Precision**, and/or **Bits From**, or **To** based on the selected format style.

5. Update the fields shown for the selected format style.
6. Click **Save** to save your changes.

More Information About Numeric Formatting

The **Format** list includes a set of styles that can be individually configured to create customized number styles. The Real number format is the default for the user defined styles.

Based on the selected number style, the following fields appear on the **Format** dialog box to change the default properties.

- **Fixed Width**

  Appears for every number format style. When selected, the length of a number cannot exceed the text length of the text element (Text, TextBox, or Button) in design time. Numbers that exceed design time text length will show the special character specified in the **Value too large for Fixed Field**. The character is defined in the **Special character** tab of the **Styles** dialog.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For the Real number style, the length of the fractional part of the number is truncated to fit the design time length. If the length of the number is still too large after removing the entire fractional part, then the number will show the special character specified in the **Value too large for Fixed Field**.

- **Precision**

  Appears for the **Real**, **FixedDecimal, and Exponential** number styles. **Precision** sets the possible number of digits in the fractional part of a number to the right of the decimal point and can be set from 0 to 8.

- **Bits From** and **To**

  Appear for the **Hex** and **Binary** number formats. **Bits From** sets the starting bit position (0-31) of a hex or binary number shown during run time. **To** sets the ending bit position of a hex or binary number shown during run time.

Working with User-defined Global Number Styles

The Galaxy Style Library includes a set of 25 user-defined number styles. User-defined styles appear towards the bottom of the list of the **Format Styles** list and are named UserDefined1 to UserDefined25.

Renaming User-defined Global Number Styles

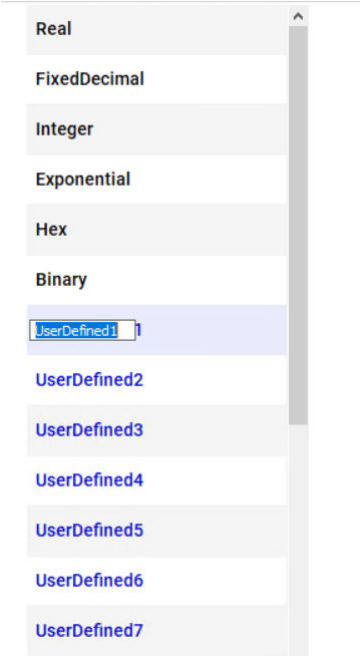
Rename a style to provide a descriptive name, to provide a group of related styles names, or to suit any other specific application needs. The renamed style will appear during configuration. For example, the renamed style will appear in the Industrial Graphics Editor when configuring an animation using Number Styles.

The renamed style will function the same during run time as it did prior to the name change.

To rename a user-defined element style

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. On the **Format** tab, navigate to the user-defined style you want to rename.
4. Right-click the UserDefined format style to select the style name and enable the text box for editing.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



5. Enter a new style name.

Importing and Exporting User-Defined Global Number Styles

You can import and export your renamed user-defined Number Style as a normal part of the Galaxy Style Library. For more information, see Importing and Exporting Galaxy Style Libraries.

Switch to Another Galaxy Style Library During Runtime

AVEVA OMI includes the Style namespace with two attributes.

- CurrentStyle is a string that is the name of a Galaxy Style Library. CurrentStyle can be included in a script to change to another Galaxy Style Library during runtime.

  CurrentStyle is retentive. A ViewApp starts with the Galaxy Style Library that was active when the ViewApp was previously stopped.

  The CurrentStyle attribute operates in the MyViewApp.Style namespace. CurrentStyle is specified in a script with the MyViewApp.Style prefix in the form MyViewApp.Style.CurrentStyle.

- Styles.CSV is a comma-separated string that is a list of the defined galaxy styles. The value is read-only and is non-retentive. It is defined by the list of galaxy styles defined in the Galaxy Style Library. This list is configured in the IDE.

Note: The CurrentStyle and Styles.CSV attributes are available for use only in AVEVA OMI ViewApps. The CurrentStyle attribute cannot be used in an InTouch HMI application because the MyViewApp. prefix is regarded as a configuration error.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Attribute	Data Type	Read/Write	Initial Value	Description
CurrentStyle	String	Yes	Retentive	Loads the name of a Galaxy Style. During runtime, the value of CurrentStyle can be changed to the name of another Galaxy Style. The appearance of graphics in the running ViewApp are updated with the element styles of the new, active Galaxy Style.
Styles.CSV	String	Read Only	No initial value	Gets the list of available Galaxy Styles in CSV format. During run time, shows the styles as a comma separated string. The value is automatically updated when new styles are added to the Galaxy Style Library in the IDE.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Chapter 3

Build applications

You can build applications with the objects and graphic standards that you built during the design phase of your project. These standards represent reusable components that can be inserted into a ViewApp in an almost assembly line process.

Note: Some of these links go to the relevant sections in other AVEVA help systems.

Create a ViewApp Framework	About screen profiles	AppFramework layouts	About ViewApp security	About alarms
Work with Graphic Elements	About graphic elements	About the graphic element property commonalities	Specific Graphic Element Properties (see About graphic-specific and element-specific properties on page 403)	Use animation
Work with Graphics and Other Media	Create and manage graphics	Manage industrial graphics	About custom properties for graphics	About using an External Content item
Write Scripts	Script editor		Add a script to a graphic	

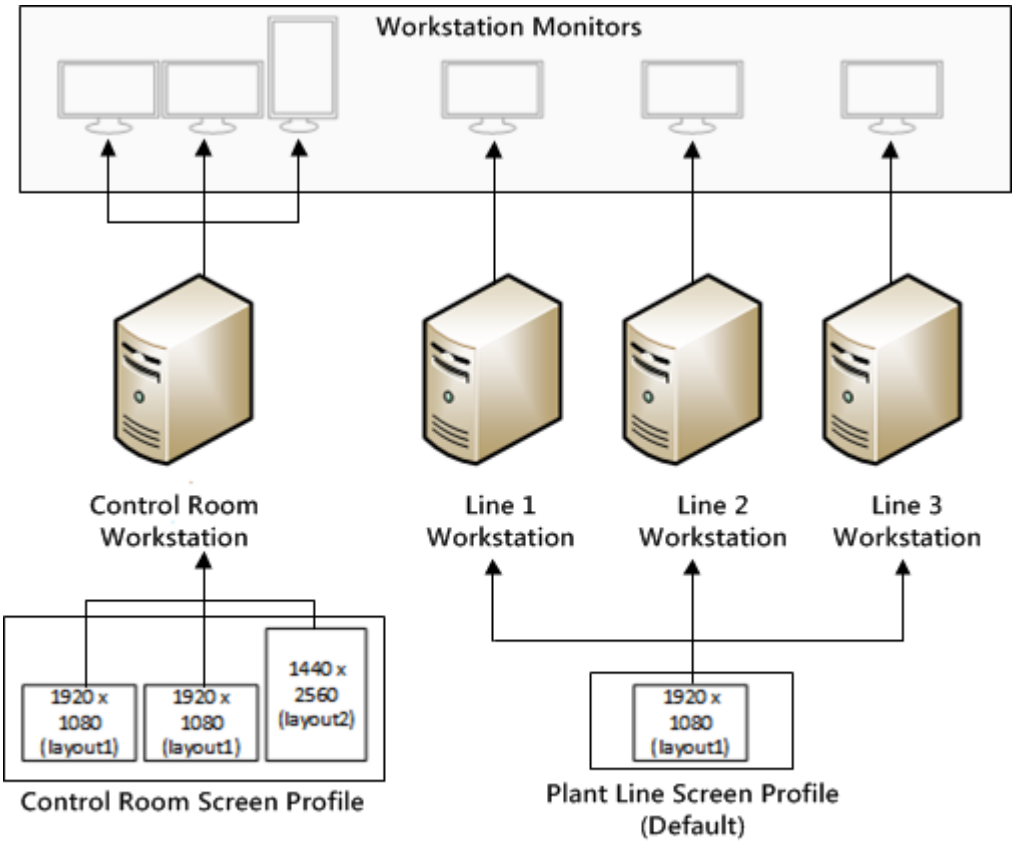
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Assemble a View Application	ViewApp editor	Switch ViewApp language at runtime	ViewApp Navigation (see About ViewApp navigation on page 514)	About client controls embedded into a graphic
Show ViewApps with InTouch Access Anywhere	About InTouch Access Anywhere	About InTouch Access Anywhere architecture	Build an InTouch Access Anywhere Secure Gateway	About using InTouch Access Anywhere

About screen profiles

A screen profile defines the physical characteristics of one or more client workstation screens that will show a running ViewApp and how these screens are arranged with respect to each other. Each screen icon that appears in a screen profile represents a physical screen.

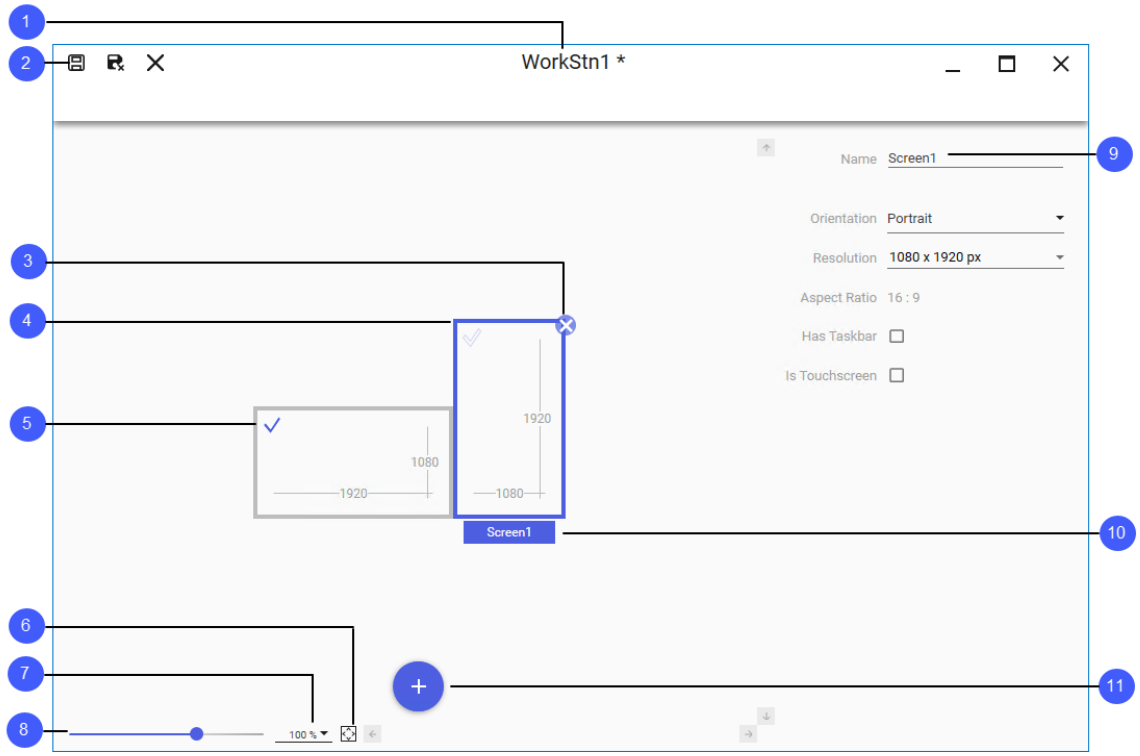
You create, manage, and configure screen profiles with the Screen Profile Editor.You must assign values to the physical properties of each screen in a profile. Also, you must arrange the screen icons to represent the actual configuration of screens attached to a client workstation with the Screen Profile Editor.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About the Screen Profile Editor

The figure below shows the various components of the Screen Profile Editor to create and configure a profile by adding screens, arranging the screens to each other, and assigning property values to each screen.



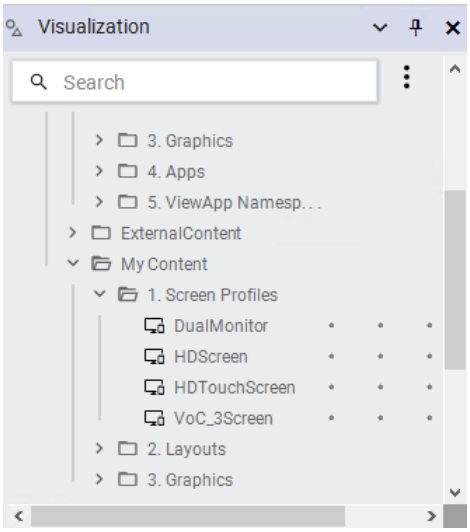
1	Screen Profile Name	Name assigned to the screen profile.
2	Command Bar	Commands to save, save and close, and close the screen profile in the Screen Profile Editor.
3	Delete Screen	Selecting the **Delete Screen** icon removes the screen from the profile. The primary screen cannot be deleted from a profile .
4	Selected Screen	A blue border indicates the current selected screen in the profile. Tapping or clicking on a screen icon within a profile sets the focus of the Screen Profile Editor to the selected screen.
5	Primary Screen Indicator	A solid blue check mark icon identifies the primary screen in the profile.
6	Zoom to Fit	Selecting the **Zoom to Fit** icon changes the size of all screens in a profile to fit entirely within the edit area of the Screen Profile Editor.
7	Zoom Percent Picker	Drop-down list of zoom percentages to resize all screens shown in a profile.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

8	Zoom Percent Slider	Moving the **Zoom Percent Slider** left or right changes the size of screens shown in a profile
9	Selected Screen Properties	Properties of the selected screen in a profile. See About screen properties for a description of each screen property.
10	Screen Name	Name assigned to a screen as the value of the screen's **Name** property.
11	Add Screen	Selecting the **Add Screen** icon adds a screen to a profile immediately to the right of the selected screen. The added screen is a duplicate of the selected screen.

Create a screen profile

You create a screen profile from the **Visualization folder**. After creating a screen profile, you continue editing and configuring the screen profile by selecting it from the list of objects shown in the **Visualization folder**.



A screen profile should replicate the display properties configured in Windows for the computer .

To match the computer's display properties, a screen profile must include values assigned to the following properties:

- Resolution of each screen in the profile: a drop down menu provides preset display resolutions, ranging from 1024 x 768, to 7680 x 4320. Or, you can enter a custom resolution. The default is 1920 x 1080.
- Orientation of each screen: landscape or portrait.
- Physical arrangement of screens: side by side, stacked vertically, or a combination.
- Primary screen: the primary screen should correspond to display 1 configured in Windows.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Touch screens: enable this property if the screen is a touch-sensitive.

Create a screen profile from the visualization folder

You create a screen profile from the **Visualization folder** of the System Platform IDE. After a screen profile is created, it appears in the **Visualization folder** with an identifying icon.

To create a screen profile

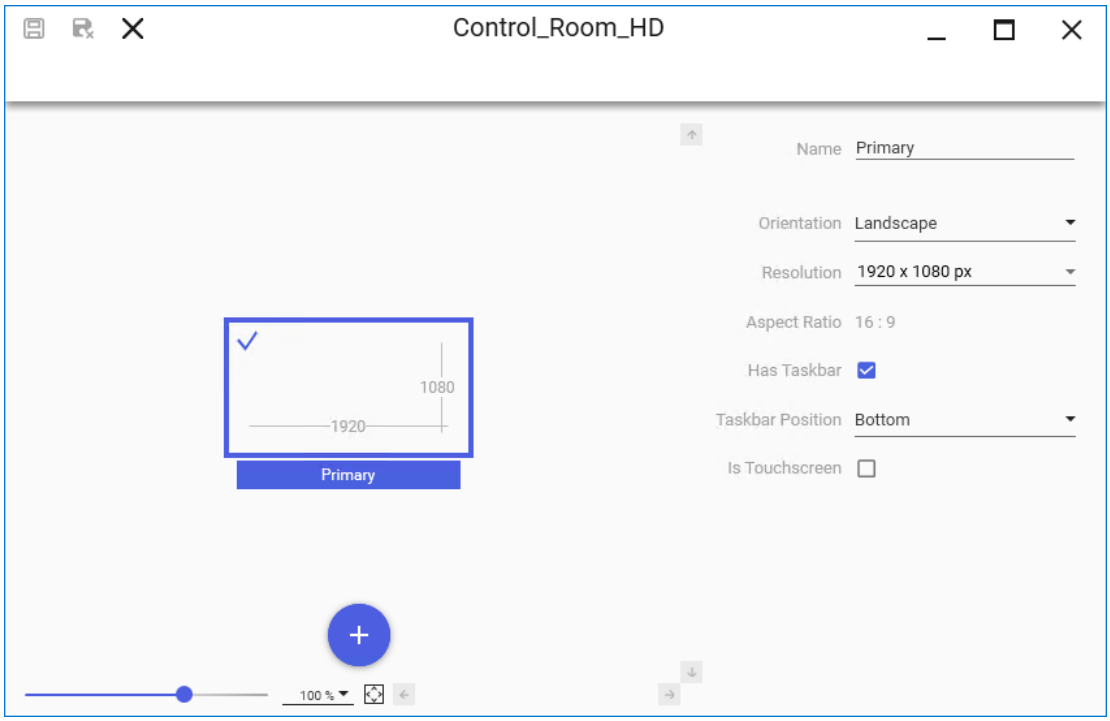
1. Open the **Visualization folder**.
2. Select **Galaxy** from the menu bar, select the **New** option, and then select **Screen Profile**.

   You can also create a screen profile by other methods:

   - **Keyboard shortcut**: Ctrl + Shift + E
   - **Shortcut menu**: Right-click within the **Visualization folder** and select **New**, and then **Screen Profile**.

   A screen profile appears in the **Visualization folder** with a default name of ScreenProfile_001 for the first screen profile that has been created. A blue background around the screen profile name indicates the name can be changed.
3. Select the screen profile from the list and rename it.
4. Select the screen profile and press Enter to open the screen profile in the Screen Profile Editor.

   After opening a screen profile, the Screen Profile Editor shows a single screen in landscape orientation with a 1920 x 1080 resolution.

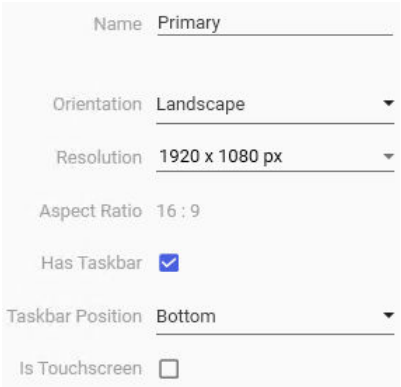


You can begin configuring the screen profile for your View Applications. For more information about configuring a screen profile, see Configure a screen profile.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure a screen profile

A screen profile specifies the physical characteristics of one or more workstation screens. For each screen defined in a profile, you must specify a screen name, the screen's orientation, resolution, and whether the screen includes a task bar and supports touch gestures. Also, you must arrange the screen icons in a screen profile to represent the configuration of actual client workstation screens, as configured within the Windows display settings (you can configure the Windows settings either before or after you configure the screen profile). For the details of each screen configuration property, see About screen properties.



Configuration Note:

A computer's Windows display properties do not necessarily need to be defined before you create a screen profile. You can create a screen profile first, and then configure the hardware screens for the computer afterwards. You can start testing with a single 1080P screen and add screens to the profile later in the testing cycle.

Add a screen to a screen profile

You can add up to a maximum of 20 screens to a screen profile. When a new screen is added, a screen icon appears to the right of the currently selected screen. The physical properties of the selected screen are assigned to the added screen.

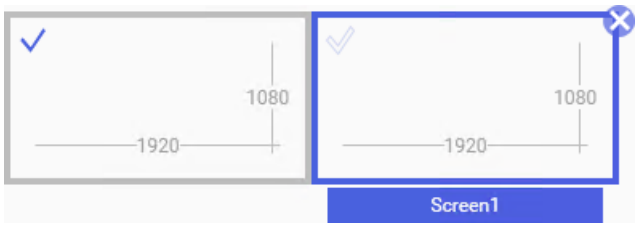
If another screen is located immediately to the right of the selected screen, the added screen is inserted and the existing screen is moved to the right.

To add a screen to a screen profile

1. Open the **Visualization folder**.
2. Open the screen profile for editing.
3. Select the screen in which you want to add a new screen to the right of it.
4. Click the **Add Screen** icon that appears beneath the screen icons.

   A screen icon appears immediately to the right of the selected screen and is named Screen*x*, where *x* is the number of the screen added to the profile.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



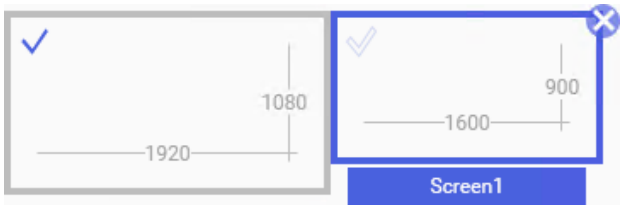
5. Rename the added screen.
6. Modify the physical properties of the new screen in the right column of the editor window.
7. Move the screen (in relation to the primary screen) so it mimics the computer display configured in Windows.
8. Save your changes to the screen profile.

Delete a screen from a screen profile

You can delete all screens within a profile except the primary screen designated with a checkmark in the top left corner of the screen.

To delete a screen from a screen profile

1. Open the **Visualization folder**.
2. Open the screen profile that contains a screen you want to delete.
3. Select the icon of the screen you want to delete.



   The primary screen shows a check mark in the screen's upper left corner. The primary screen cannot be deleted from a screen profile. If the screen you selected can be deleted, a circle with an X appears at the screen's upper right corner.

4. Click the circle on the screen to be deleted or press the Delete key.

   The screen is deleted from the screen profile and its icon no longer appears in the edit area.

5. Click **Save**.

Rearrange screens in a screen profile

The screen configuration area shows the placement of screens defined in a screen profile in relation to each other. The placement of the screen icons should represents how actual screens are arranged on the workstation that will run a View Application, as configured within the Windows display settings. The primary screen should correspond to display 1 in the Windows settings.

You use a drag and drop operation with a mouse or touch gesture to arrange the screen icons to match the position of the screens of a workstation. Typically, you arrange screens by aligning the corner of one screen to the corner of another screen.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

A valid screen arrangement must adhere to the following rules:

- At least one corner of a screen must be aligned with the corner of another screen.
- All screens in a profile must be arranged to touch at least one other screen.
- Screens must be arranged without overlapping any other screens.
- A screen arrangement supports up to a maximum of 20 screens.

To rearrange screens in a screen profile

1. Open the **Visualization folder**.
2. Open the screen profile.
3. Select the screen icon that you want to move.

   The border of the icon turns blue to show that it is selected.
4. Drag the icon to its new location.

   While dragging the icon, the border turns red. It turns blue again when the icon is aligned with the corner of another screen icon.

   If you drop the icon at an invalid position, it returns to its original position.
5. Click **Save** to save the changes.

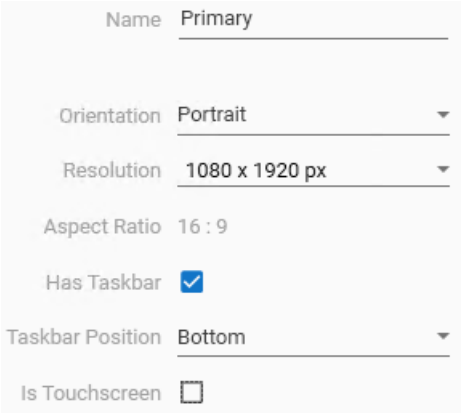
Configure a screen

The Screen Profile Editor shows the properties of the selected screen in the right column. You assign a name to a screen and configure its properties by updating the values shown in the property fields.

To configure a screen

1. Open the **Visualization folder**.
2. Open the screen profile for editing.
3. Select the screen that you want to configure properties for.

   The default Primary screen includes the following properties.



   Additional touch property fields appear when the **Is Touchscreen** property is selected.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For a complete description of all properties, see About screen properties.

4. Type a name for the screen and assign values to the screen's properties.
5. Click **Save**.

About screen properties

A set of screen properties appears in the Screen Profile Editor. These properties define the physical and functional characteristics of each screen in a profile. Assigning a value to each property is required to configure a screen.

Name Property

Each screen defined in a screen profile must be uniquely named. The border of the **Name** field turns red if you enter an invalid screen name.

Orientation Property

The orientation of a screen can be either landscape or portrait. A screen's width is greater than its height when oriented in landscape mode. In portrait mode, screen height is greater than its width.

Changing a screen's orientation changes the width and height values shown in the **Resolution** property field. For example, when the resolution is 1920 w x 1080 h, changing the orientation from Landscape to Portrait changes the resolution to 1080 w x 1920 h.

Resolution Property

The **Resolution** property field shows common screen resolutions in a drop-down list. The ratio of the width of a screen by its height in pixels is the aspect ratio. Changing orientation reverses width and height values shown in a screen resolution.

Aspect Ratio

The **Aspect Ratio** property is an automatically calculated ratio of a screen's width to height by pixel counts. The Screen Profile Editor shows the aspect ratio of a screen in the form of a well known value. If an aspect ratio does not fall in a range of a well known value, then the aspect ratio is represented by a double data type rounded to two decimal points

Has Task Bar

The **Has Taskbar** property indicates whether the selected screen shows a task bar or not. Task bar information is considered when adjusting the size of a layout to fit a maximized screen. As task bar can differ in size based on themes or physical size and is only an approximation.

If you select the **Taskbar** property then the **Taskbar Position** field appears with options to position the taskbar at the Top, Bottom, Left or Right of a screen border.

Is Touchscreen

A screen touch lock determines how screen touch gestures are supported while a ViewApp is running.

A touch lock can be configured to:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Support all touch interactions of a ViewApp.
- Prevent users from interacting with machinery while interacting with related controls on touch screens. The risk of accidentally entering a machinery command with a touch gesture is reduced by forcing the user to keep the touch lock active by pressing the lock icon continuously while operating screen controls.
- Restrict access to a subset of supported touch gestures to prevent accidental interactions with a ViewApp.

Touch Lock Configuration

You can set three Touch Lock properties when configuring a screen profile.

- **Is Touchscreen**
  - **Disabled**

    Disabled is the default. No touch lock appears on the screen while a view application is running. You can interact with a ViewApp or its contents using all supported touch gestures.
  - **Enabled**

    A touch lock appears on the screen and restricts the usage of touch gestures to protect a ViewApp.
- **Touch Lock Level**

  The **Touch Lock Level** property becomes accessible only when the **Is Touchscreen** property is enabled.
  - **None**

    The touch lock is disabled and users can interact with a view application or its contents to the full extent to which touch is supported. **None** is the default and no touch lock icon appears on the screen of a running ViewApp.
  - **Commands Locked**

    In **Commands Locked** mode, all touch gestures are blocked when the touch lock is active. A touch lock icon is green to indicate the touch lock is active or in a locked state. Mouse or keyboard interactions are still possible.

    Pressing the touch lock icon during runtime deactivates or *unlocks* the touch lock to enable you to interact with a view application using all supported touch gestures. A touch lock icon turns red to indicate the touch lock is in an unlocked state.

    Commands Locked mode does not block writes to I/O in any way, shape, or form.
  - **Two Handed Operation**

    In **Two Handed Operation** mode, single finger content selection gestures are blocked by the touch lock, but two and three finger pane level gestures are permissible. Pressing on the touch lock icon deactivates the touch lock enabling you to interact with a view application using all touch gestures including single finger content gestures.
- **Touch Lock Position**:

  Touch Lock Position specifies the placement of the touch lock icon on the screen. This property is enabled only if the **Touch Lock Level** property is set to **Commands Locked** or **Two Handed Operation**.

  Touch Lock Position provides options to place the Touch Lock icon at the bottom left or bottom right of a screen.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Scope of a Touch Lock During Runtime

- One Touch Lock can be configured for each screen assigned to a ViewApp.
- When Touch Lock is present during runtime, the possible touch gestures are specified by the configuration of that screen.
- When Touch Lock is absent or deactivated, a user can perform all supported touch actions on the corresponding touch screen.
- If there are multiple layouts per screen, then touch interactions on all layouts depend on the state of Touch Lock.
- If there are layouts that span multiple touch screens, the configuration of the Touch Lock on the primary screen of the layout determines touch functionality for other screens.

Resize a screen

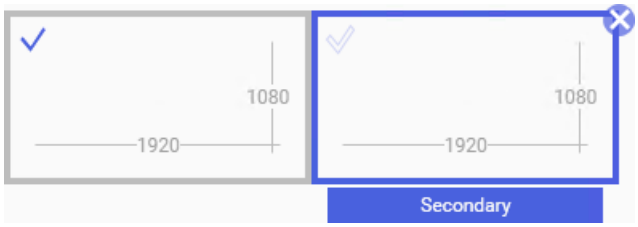
You can change the size of a screen by selecting new horizontal and vertical dimensions in the **Resolution** field. You can also change screen dimensions by changing the screen's orientation between Landscape and Portrait.

Changing the size of a screen affects the positioning of surrounding screens. A resize may result in screens that are not aligned to the edge or ones that are completely detached. These screens will then snap to the closest point that does not overlap any other screens.

The scrollable extents of the viewable screen area are updated if a screen is resized, which may cause other screens to extend beyond the boundaries of the Screen Profile Editor's viewable area.

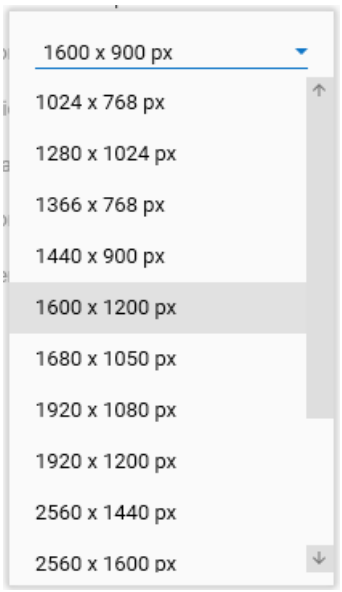
To change the size of a screen in a screen profile

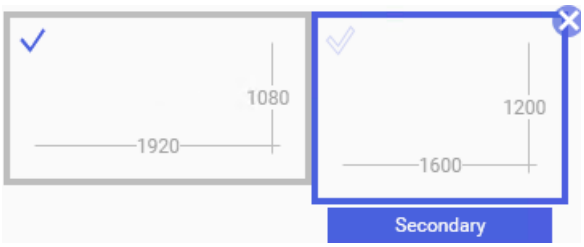
1. Open the screen profile for editing.
2. Select the screen to resize.



3. Select new horizontal and vertical dimensions from the **Resolution** field or change the screen's orientation.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





Note: You can also type a custom width and height in the **Resolution** field.

4. Click **Save**.

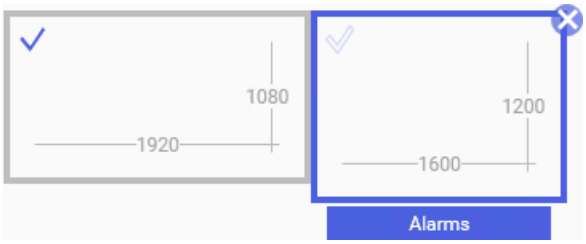
Set a primary screen

Every screen profile includes a primary screen. A primary screen represents the main screen of a workstation that will show a running View Application. The primary screen is used as the reference point when applying a screen profile to the workstation, and corresponds to display 1 in the Windows settings. The primary screen cannot be deleted from a screen profile.

The primary screen of a screen profile is identified by a solid check mark in the screen icon's upper left corner.When a non-primary screen is selected, an empty check mark is shown instead. Clicking on an empty check mark selects the screen as the primary screen of the profile.

To set the primary screen of a screen profile

1. Open the screen profile for editing.
2. Select the screen icon to designate as the primary screen of the profile.

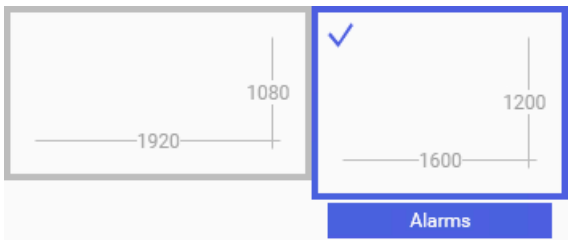


3. Move your mouse over the empty check mark at the top left of the screen icon.

   A tooltip message indicates the screen can be set as the primary screen.
4. Click the check mark.

   The check mark turns blue to indicate that this screen is now the primary screen.

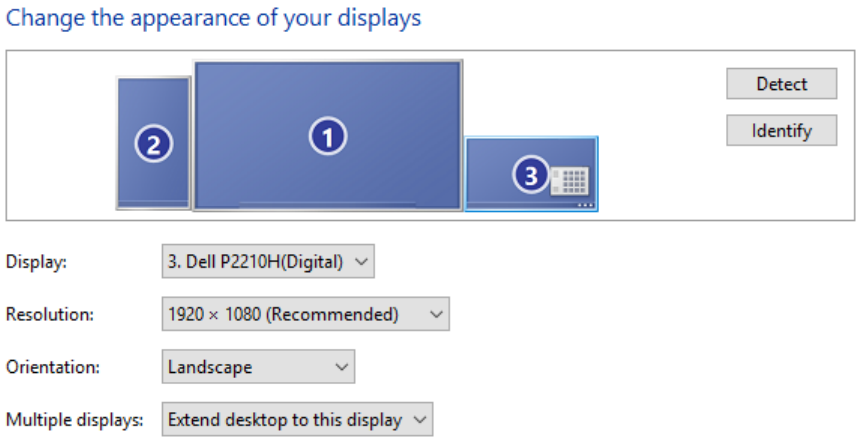
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



5. Save your changes.

Set additional screens

The following figures show examples of a computer's Window display properties and a corresponding screen profile that replicates the display properties.



In this example, the Windows display properties of the three screens attached to a workstation are:

Screen 1	3840 by 2160 resolution, Landscape orientation
Screen 2	1080 by 1920 resolution, Portrait orientation
Screen 3	1920 by 1080 resolution, Landscape orientation, Primary screen

Typically, there are three major tasks to configure a screen profile:

- Add the screens

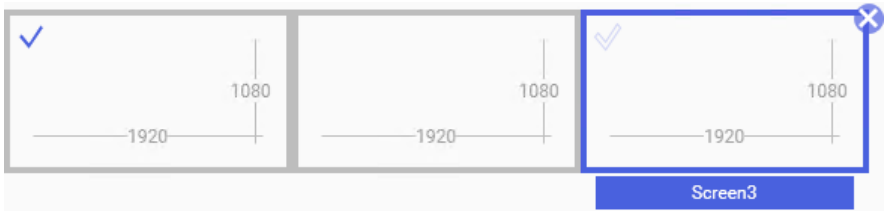
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Position the screens
- Configure the properties of the screens

The following steps demonstrate these general tasks by building an example screen profile that replicates a computer's Windows display properties shown above.

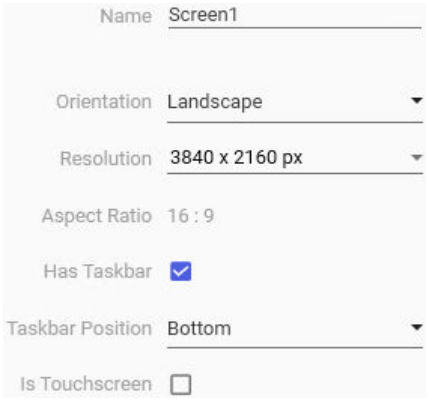
1. Open a screen profile that you created, which shows the default **Primary** screen.
2. Rename the **Primary** screen to **Screen1**
3. Add a screen, which appears to the right of the **Primary** screen and is named **Screen2**.

   Screen2 has same properties as **Screen1** except its name and it does not contain a task bar. Also, **Screen2** is the selected screen.
4. Add another screen, which appears to the right of **Screen2** and is named **Screen3**.



5. Set **Screen3** as the primary screen in the screen profile.
6. Drag **Screen2** to the left of **Screen1**.

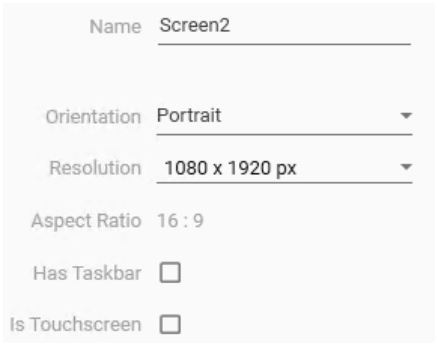
   The three screens of the profile have been created and are in the approximate configuration that matches the Windows display properties.
7. Set the properties of **Screen1** to match Screen1 of the computer's Window display properties.

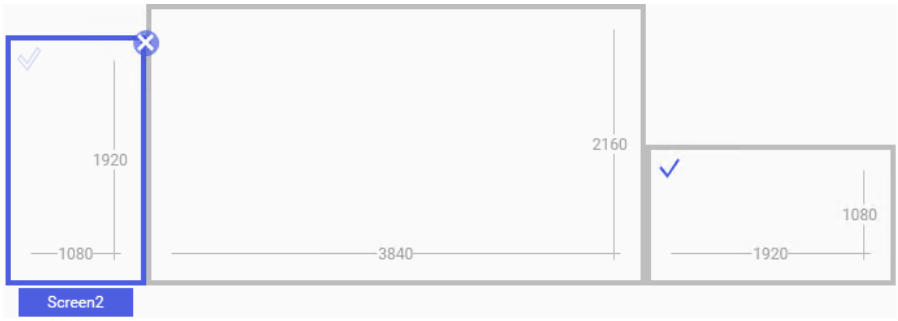


© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

If necessary, drag **Screen2** and **Screen3** to align both screens to the bottom of **Screen1**.

8. Set the properties of **Screen2** to match the properties of screen 2 of the Windows display properties.



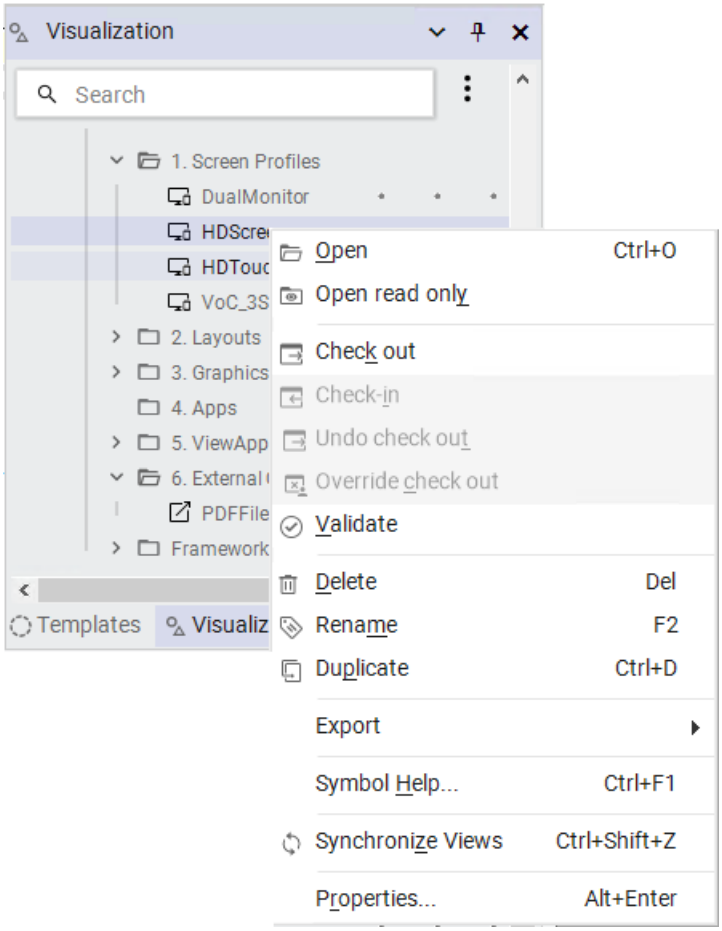


Your screen profile should match the Windows display properties.

Manage a screen profile

You can perform routine tasks to manage a screen profile from the **Visualization folder** by selecting shortcut menu options.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Delete a screen profile

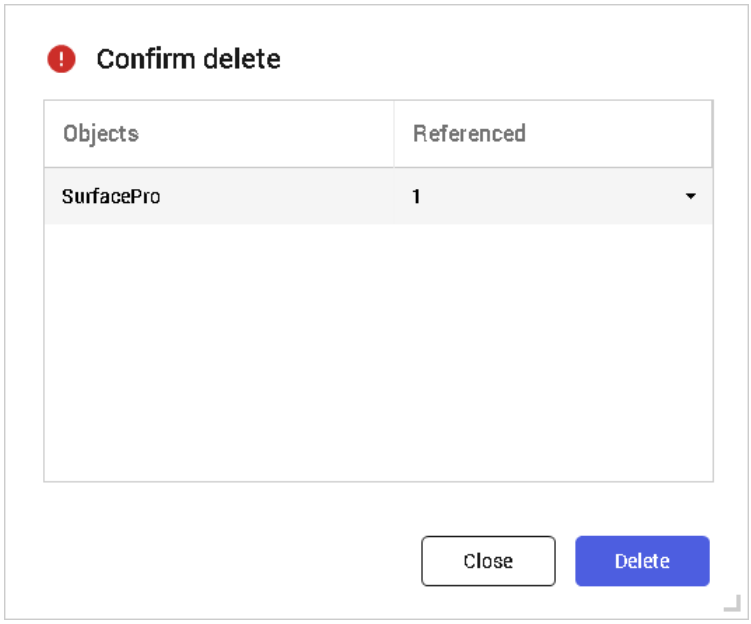
You can delete a screen profile from the Visualization folder.

To delete a screen profile

1. Open the **Visualization folder**.
2. If the screen profile you want to delete is open for editing, close it.
3. Select the screen profile(s) that you want to delete from the **Visualization folder** list.
4. Press the **Delete** key or right-click and select **Delete** from the shortcut menu.

   The **Confirm Delete** dialog box appears. The **Referenced** column shows the number of objects that the screen profile is referenced by. To see what those objects are, click the down-arrow at the right side of the **Referenced** list. This shows any layouts, ViewApp templates, or ViewApp instances that refer to the profile.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: If you delete the screen profile associated with a layout, it sets the layout's **Target** field to a red background to indicate an error condition. The resolution of the target screen is cached with the layout.

5. Click **Delete** to delete the screen profile(s).

Rename a screen profile

You can rename a screen profile by selecting a profile from the Visual Graphics list and pressing F2 or by clicking on the name to enter edit mode.

A new screen profile name is validated to ensure it meets naming requirements. If a new name is a duplicate of an existing name or breaks any name rules, a message appears indicating there is a problem with the name.

Pressing the Esc key cancels a rename request and the screen profile reverts to the original name. Committing an empty name cancels the rename request.

To rename a screen profile

1. Select the screen profile that you want to rename from the **Visualization folder** list.
2. Click the screen profile name or press **F2** to enter edit mode. A blue background appears around the screen profile name.
3. Type the new name of the screen profile.
4. Commit the new name by clicking outside of the screen profile name.

   If the new name is valid, it appears in the **Visualization folder** list. If the name is invalid, you see a dialog box that describes the problem.
5. If you typed an invalid screen profile name, click **OK** and re-enter another name.

Duplicate a screen profile

You can create a duplicate of a screen profile and then rename it. Duplicating a screen profile reduces the effort to create an entirely new screen profile by using an existing screen profile as a template.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

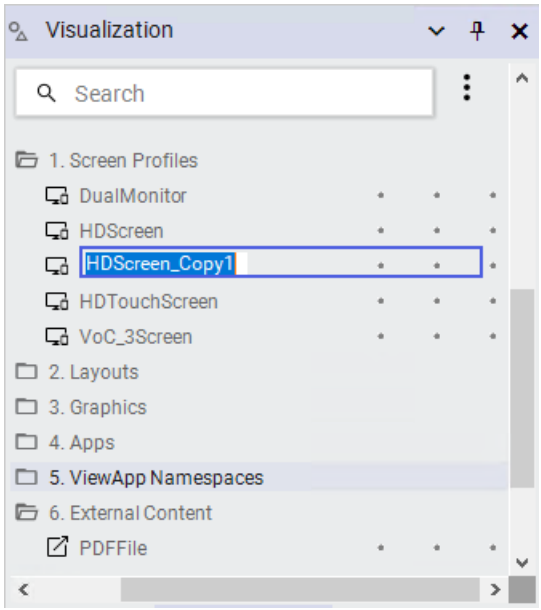
To duplicate a screen profile

1. Open the **Visualization folder** and select the screen profile that you want to duplicate.
2. In the **Home** ribbon **Edit** area, select **Duplicate**.

   You can also duplicate a screen profile by other methods:

   - **Keyboard shortcut**: **Ctrl + D**
   - **Shortcut menu**: Right-click the name of the screen profile to duplicate and click **Duplicate.**

   A new screen profile appears in the **Visualization folder** list with Copy1 appended to its name.



3. Type the name to give the duplicate screen profile.

Export a screen profile

You can export one or more screen profiles from the Visualization folder to an aaPKG file. You can then import the screen profiles from the export file to another project.

To export a screen profile

1. Open the **Visualization folder**.
2. Select the screen profiles to export.
3. In the **Home** ribbon **Export** area, select **Selected**, then select **As package**.

   You can also right-click the screen profile name and select **Export**, then **Selected object(s)**.

   The **Export Selected objects** dialog box appears with a folder tree to specify where the exported file should be saved and a field to name the export file.
4. Select the folder to save the export file.
5. Assign a name to the export file.

   The default export file name is the name of the first selected screen profile from the **Visualization folder**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

6. Click **Save**.

   A horizontal bar shows the progress of the screen profiles being loaded into the export file.

7. Verify the export aaPKG file has been placed in the folder you specified.

Import a screen profile

You can import screen profile created in another project to the Visualization folder of the System Platform IDE. Exported screen profiles are saved in an aaPKG file.

During the import process, you have the choice to keep or overwrite any screen profiles currently in the Visualization folder with the imported screen profiles.

To import one or more screen profiles

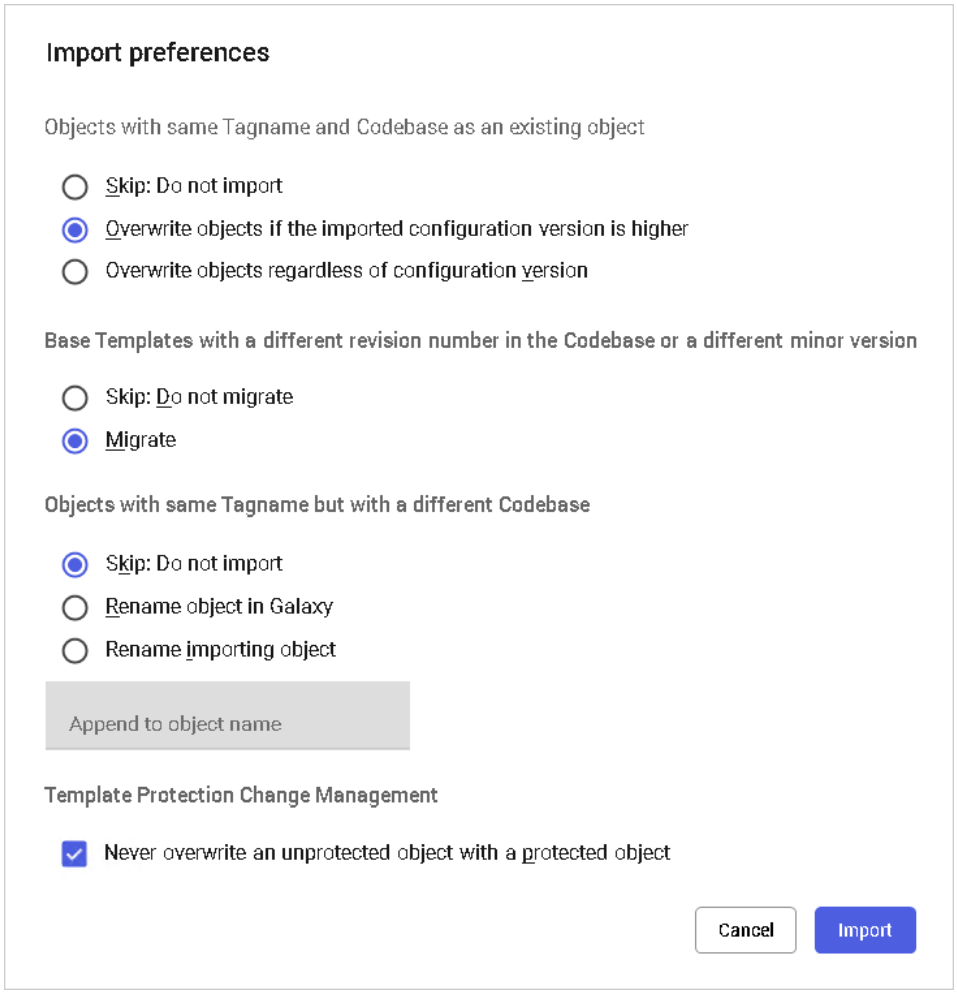
1. Open the **Visualization folder**.
2. Select the **Galaxy** ribbon, then select **Import**.
3. Select **Objects**, then **From package**.

   The I**mport Objects from package** dialog box opens.

4. Go to the folder where the aaPKG file containing screen profiles is located.
5. Select the aaPKG file and click **Open**.

   The **Import Preferences** dialog box opens with options to import screen profiles. You cannot have two screen profiles with the same name or more than one copy of the same version of a profile in the same Galaxy. When you import a screen profile, you can choose options from the **Import Preferences** dialog box how you want naming and version conflicts handled.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



6. Click **OK** to import the screen profiles.

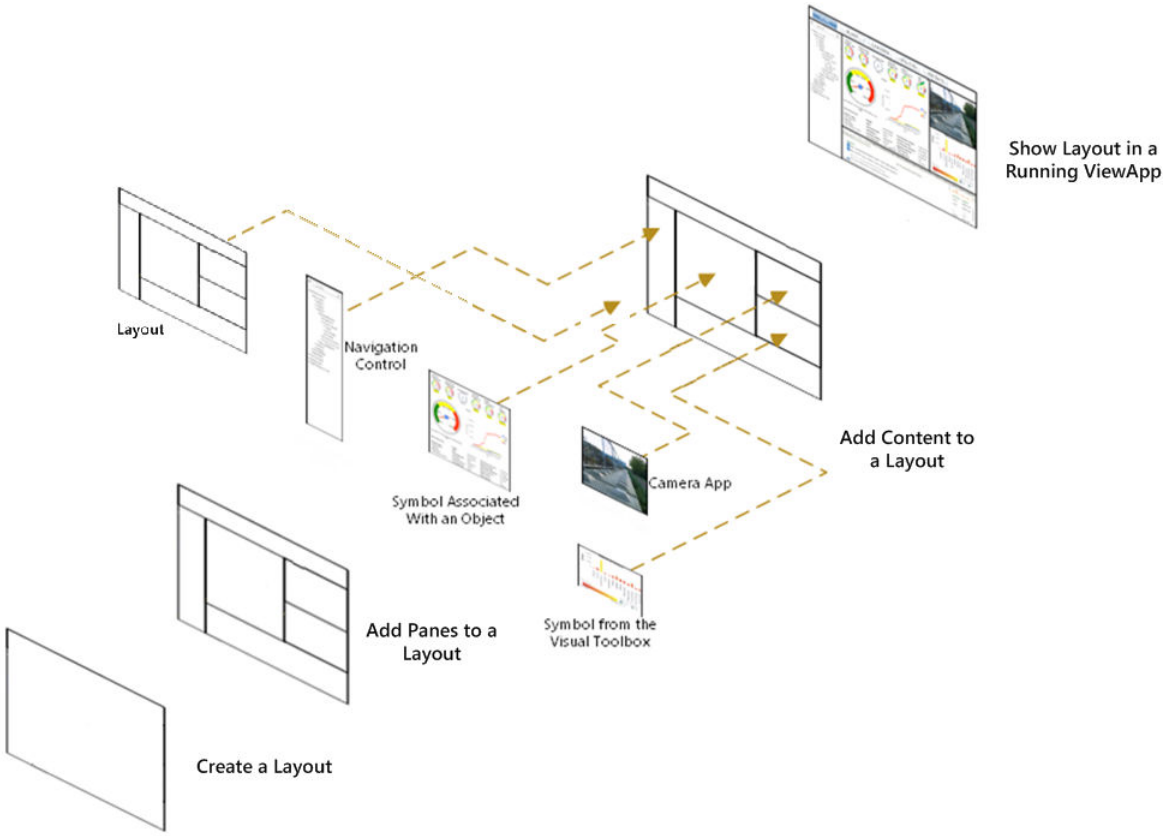
   A progress bar shows the screen profiles being imported into the Visualization folder. After finishing the import, you should see the screen profiles in the list of Visualization folder objects.

AppFramework layouts

A layout consists of one or more rectangular areas called panes that contain content shown in a ViewApp. Horizontal and vertical lines called dividers set the boundaries between adjacent panes in a layout. As part of building a ViewApp, you associate a layout to a screen in a selected screen profile. The Layout system encompasses an editor and an engine that displays a layout and its content while a ViewApp is running.

During design time, you use the Layout Editor to configure one or more layouts for a ViewApp. The Layout Editor includes controls to add panes and adjust their size and arrangement to other panes within the layout's window. The Layout Editor provides properties to configure the layout itself and each individual pane. You can also drag and drop content into panes as part of configuring a layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



In addition to containing content such as the apps and graphics that you add to layout panes, a layout can itself function as content that you add to another layout. See Add content to panes for more information.

You can also associate layouts with an Application Server object, as you would any other content item. For more information about using layouts as content, see Use a layout as content.

About the Layout Editor

The Layout Editor consists of **Build** and **Script** pages, and includes controls to create and configure panes that appear in a layout.

The Layout Editor provides Layout, Content, and Script editing modes. The Layout editing mode provides a set of components and properties to define the visual and behavioral aspects of each pane.

Content editing mode includes two different modes:

- Fixed Layout (default): This is the best choice for adding content if the ViewApp that uses the layout will be viewed exclusively on desktop-type monitors or similar displays.
- Responsive Layout: This is best choice if the layout will be used in one or more ViewApps that will be viewed by a mix of devices, such as smart phones, tablets, and monitors.

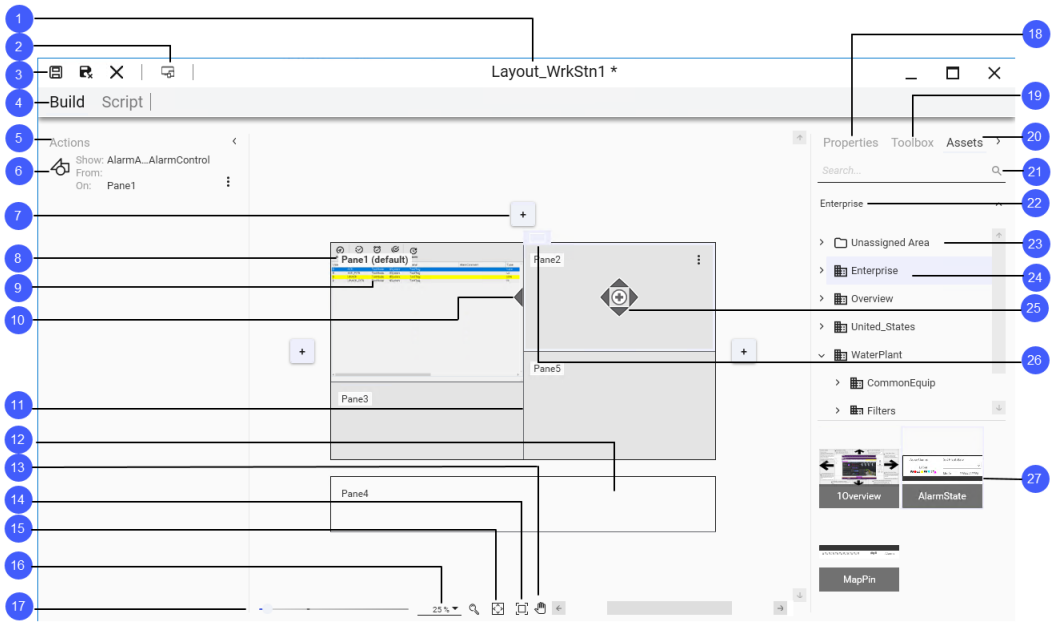
For more information about whether to use the fixed and responsive mode, see About fixed vs. responsive layouts.

Once you have begun adding and configuring content, switching from Fixed to Responsive or visa versa will clear all panes and any configuration changes that you have made will be lost. Unsaved layout scripts are not affected.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

In either content editing mode, you can select the content to appear in a pane by selecting the **Toolbox** and **Assets** tabs and dragging and dropping objects, controls, apps, or graphics into panes.

The figure below shows the various components parts of the Layout Editor when Fixed mode is selected. For more information about using the Layout Editor to configure Responsive Layouts, see About the responsive layout mode.



1	Layout Name	Name assigned to the displayed layout.
2	Fixed Layout/ Responsive Layout Button	Sets/resets the IsResponsive layout property and changes the build view from the default **fixed** layout view to the **responsive** layout view (and back again) in the Layout Editor. Note that any configuration changes made in one view are not preserved when you switch between fixed and responsive views, since the two views cannot be reconciled. If you switch views after configuring the layout, the configuration is lost.
3	Menu Bar	Commands to save, save and close, and close the layout.
4	Build/Script Selection	Switch between the **Build** and **Script** pages of the Layout Editor. • In the **Build** page, you can select between fixed and responsive layout modes, and configure the layout by adding, removing, and resizing panes, including slide-in panes, and add content such as graphics and apps to individual panes. You can also set properties for the layout, panes, and content. • In the **Script** page, you can add layout scripts that control how panes and their content execute in runtime.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

5	Action List	List of navigation actions associated with the panes of a layout. Actions are added to pane when content is dragged from **Toolbox** or **Assets** and dropped on a pane or by using the pane-drop-down menu to move content between panes.
6	Pane Action	Selected pane of a layout listed in the Pane Action list, which contains options to edit or remove the current content of the pane.
7	Add Slide-in Pane Button	Adds a slide-in pane on the side of the layout where the **Add Slide-in Pane** button is located. A slide-in pane can contain content. During runtime, users can select a tab on the interior border of a layout window to show or hide a slide-in pane.
8	Pane Name	Name assigned to a layout pane.
9	Pane Content	Thumbnail graphic representing the content added to a layout pane.
10	Extender Button	Selecting the triangular extender button extends the source pane's size in the direction pointed by the triangle to fill the area previously occupied by the target pane.
11	Pane Divider	Line representing the boundary of a pane. During configuration, a layout's pane size can be changed by moving its divider. Also, a divider's visual properties can be configured and whether users can move a divider within a pane during runtime.
12	Slide-in Pane	Optional panes that can be configured on each side of a layout window. During runtime, users select a tab to show or hide a slide-in pane, which appears within the boundaries of the layout window.
13	Pan Lock	Places a pane into pan mode to enable the user to move the view box and show content outside the boundaries of the pane.
14	Zoom to 100%	Sets the size of content shown in a layout to its actual size in the Layout Editor.
15	Zoom to Fit	Adjusts the size of content to fit entirely within a pane in the Layout Editor.
16	Zoom Box	The **Zoom Box** enables a selected area to be zoomed to the full width or height of the current editing area of the Layout Editor. The cursor shows a magnifying glass when the Zoom Box is selected. Drawing a rectangle by mouse or by touch within the editing area shows the content within the rectangle at the full width or height of the current editing area.
17	Zoom Percent Slider	Moving the **Zoom Percent Slider** left or right zooms the size of the layout shown in the Layout Editor.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

18	Properties grid	Selecting the **Properties** grid shows a list or properties of the selected item within the Layout Editor. • Layout selected: Tap or click outside the layout window to show layout properties (fixed layout only) • Pane selected: Tap or click within a layout pane to show pane properties • Divider selected: Tap or click on a divider within the layout to show divider properties (fixed layout only) • Content in a pane selected: Tap or click content within a pane to show the content's properties
19	Toolbox Tab	Selecting the **Toolbox** tab shows the hierarchical folder organization of the Visualization folder. Selecting a folder shows the content within it.
20	Assets Tab	Selecting the **Assets** tab shows the hierarchical organization of objects shown in the Model view of the Template folder.. Selecting an object shows any content associated with it.
21	Search Field	Data entry field to enter a search string to locate content with a matching name. • Search from **Properties** grid: Locates selected property type by name. • Search from **Toolbox** tab: Locates content within the Visual Toolbox by name. • Search from **Assets** tab: Locates objects in the Model view by name.
22	Breadcrumb	When the **Toolbox** tab is selected, a breadcrumb shows the serial list of graphic folder names to the content selected from the list. If the **Assets** tab is selected, a breadcrumb shows the serial list of Areas or other objects to the selected item from the list.
23	Asset List	The Asset List shows a hierarchy of Area and other types of assets shown by the Template folder Model view. An arrowhead to the left of an asset within the list indicates an asset that can be expanded to show the contained objects.
24	Selected Asset	Touching or clicking an asset shown in the list selects it. A shaded background identifies the selected asset from the list.
25	Add Pane Control	Selecting an arrowhead of the **Add Pane Control** splits the currently selected pane in half and the new pane is inserted in the direction of the arrowhead. Panes can be added up, down, left, or right of the existing pane with the **Add Pane Control**.
26	Pane Presentation Style Selector	Selecting the **Pane Presentation Style Selector** shows a drop-down list of options to select how content is hosted and presented in a pane. In **Single** mode, a pane can host one piece of content. In **Multiple** or **Tabbed** mode, a pane can host multiple items.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

27	Asset Content	The content associated with the asset selected from the list.

About fixed vs. responsive layouts

The Layout Editor lets you choose between fixed layout mode and responsive layout mode.

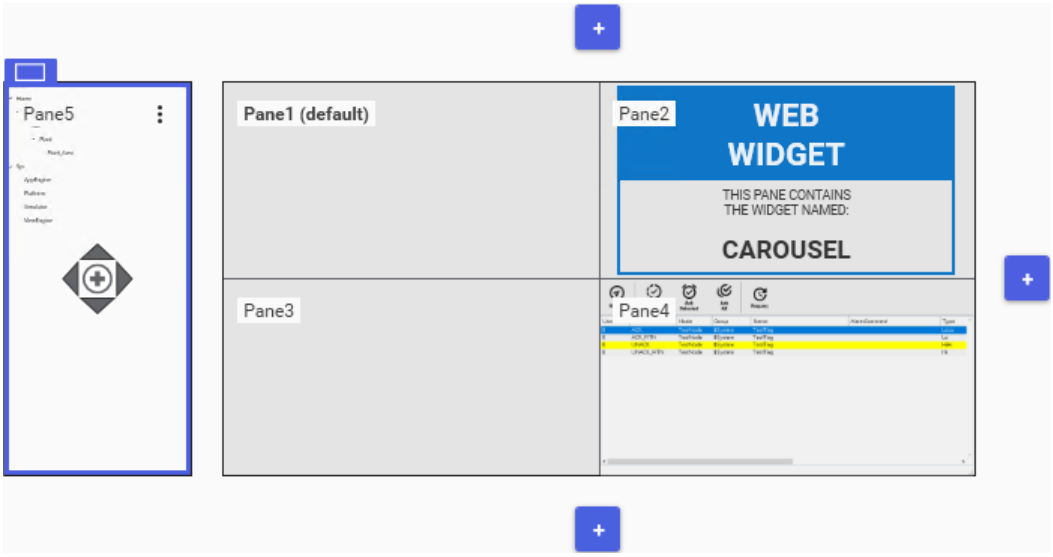
- Use the fixed layout mode if the layout you are designing is intended for use with only a single screen layout or screen form factor. For example, if the layout is only for use in a single ViewApp that will be viewed on a monitor, then using the fixed layout mode is appropriate.
- Use the responsive layout mode if the layout is intended for use with multiple screen form factors. For example, if the layout will be used in multiple ViewApps designed for viewing on different screen form factors, or if the ViewApp will be viewed on hand-held devices through the OMI web client.

The two modes are mutually exclusive, and any unsaved changes mode in one mode will be undone if you switch to the other mode. You can toggle between the responsive and fixed layout modes, by the clicking or touching the **Fixed Layout/Responsive Layout** button.

You can also switch between modes by enabling or disabling the I**sResponsive** behavior layout property. To view responsive layout properties, tap or click above the layout window, in the area that shows screen resolutions. To view layout properties, tap or click anywhere outside the layout window.

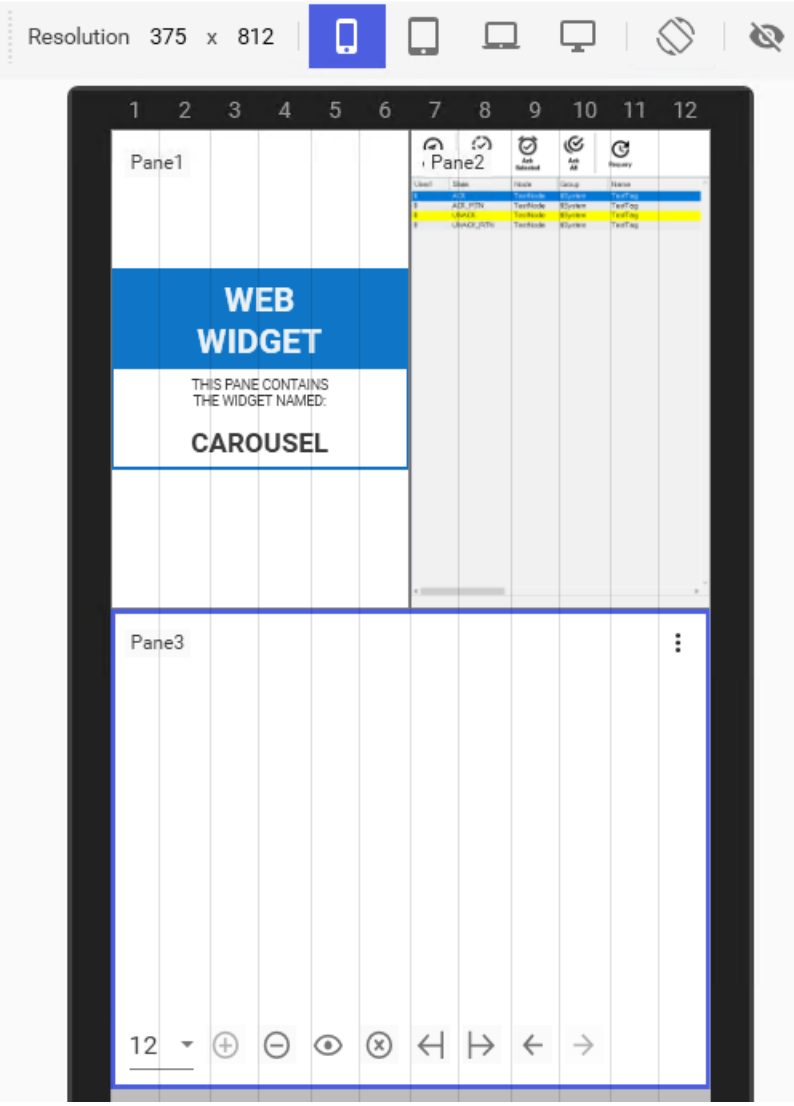
Important: All changes to the layout are discarded when you switch between modes. Note that if you switch from responsive to fixed or vice versa and then save the layout. you will lose any changes from the previous layout type, even changes that were saved. There is no undo.

You will see something similar to following image when you add content to panes in Fixed Layout mode.



You add content the same way to panes in the responsive layout mode as in fixed layout mode. In the responsive layout mode, panes are now divided into 12 columns of equal width (there is a single pane initially, as is the case with fixed layouts). Panes can be added, and width of each pane can adjusted to use anywhere from one to 12 columns. Additionally, panes in different sized screens can be configured to use a different number of columns.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



For more information about the responsive layout mode, see About the responsive layout mode.

About responsive layouts

When configuring a responsive layout, some of the important differences between it and a fixed layout are:

- In the responsive layout mode, as in the fixed layout mode, you begin with a single pane. However, panes are added to a responsive layout by selecting an Add Pane button. In the fixed layout mode, you add panes by dividing the original pane
- Different screen sizes are defined in a responsive layout. Panes can be configured to use different numbers of columns for each screen size. For example, a user on a mobile device may be shown content in a pane that takes up the entire width of the screen, while a user on a laptop is shown the same content in a pane that takes only a fraction of the screen width.
- In the responsive layout mode, you define pane widths by number of columns, not by percentage (or number of pixels) of the the total layout width. A pane can be as wide as 12 columns, or as narrow as one column. A pane that is 12 columns wide takes the full width of the layout. You can set a different number of columns for each of five different screen sizes, from extra small (mobile) to large (desktop).

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- You can configure multiple panes in a side by side arrangement if the total number of columns the panes use is 12 or less. For example, you can have three panes side by side if each pane four columns wide.

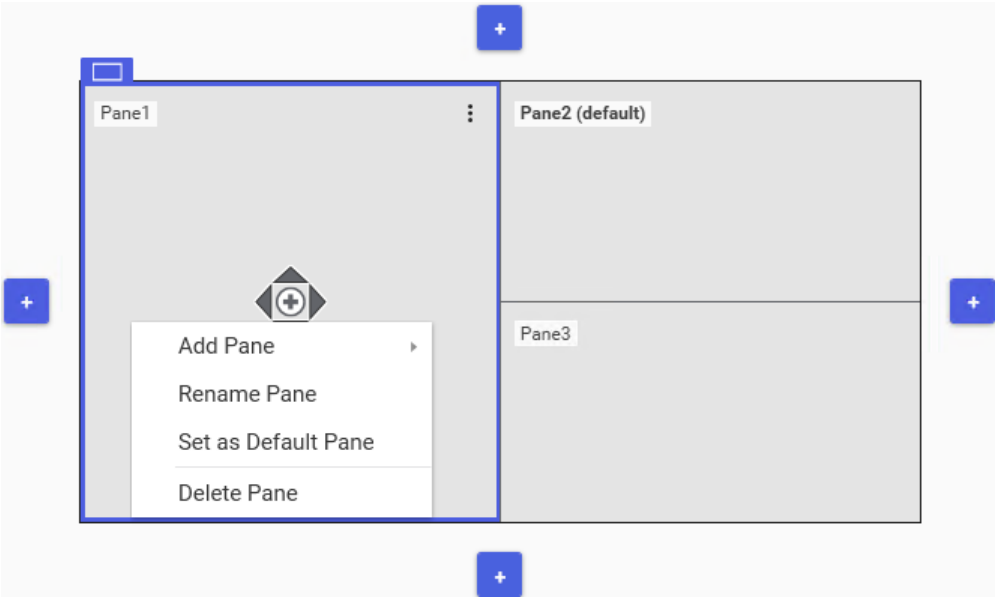
See About the responsive layout mode for more information.

Design-time editor actions: fixed layout

During design time when you are editing a layout, the Layout Editor provides shortcut menus, a Pane Options button, and shortcut keys to modify the structure of a layout and the properties of panes.

Shortcut Menu

When a layout is opened in the Layout Editor, you can show a shortcut menu by right clicking on the selected pane of a layout.



In Edit Layout mode the shortcut menu includes the following actions:

- **Add Pane**

  A sub-menu provides options to add the new pane to the Left, Right, Above, Below the selected pane.

- **Rename Pane**

  A blue background appears behind the pane name to indicate the pane can be renamed. The new name you enter must follow System Platform naming conventions.

- **Set As Default Pane**

  This action appear only if the pane you select is not the default pane of the layout. The word default placed within parentheses appears next to the pane name after it has been selected as the default pane.

- **Delete Pane**

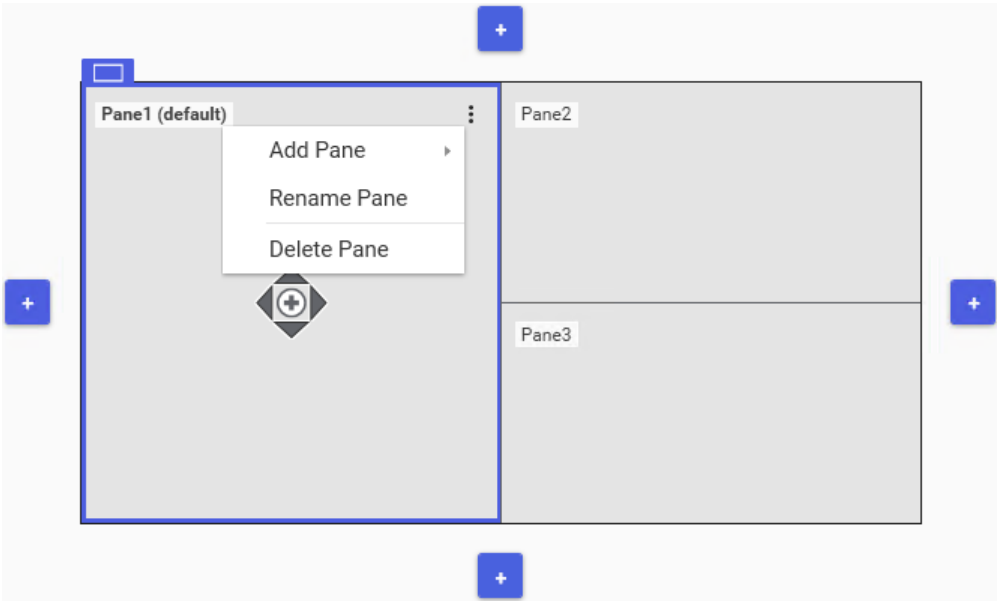
  Selecting **Delete Pane** immediately removes the pane from the layout and the space within the layout window is reassigned to the remaining panes.

In Edit Content mode the shortcut menu provides a single action to clear the content from the selected pane.

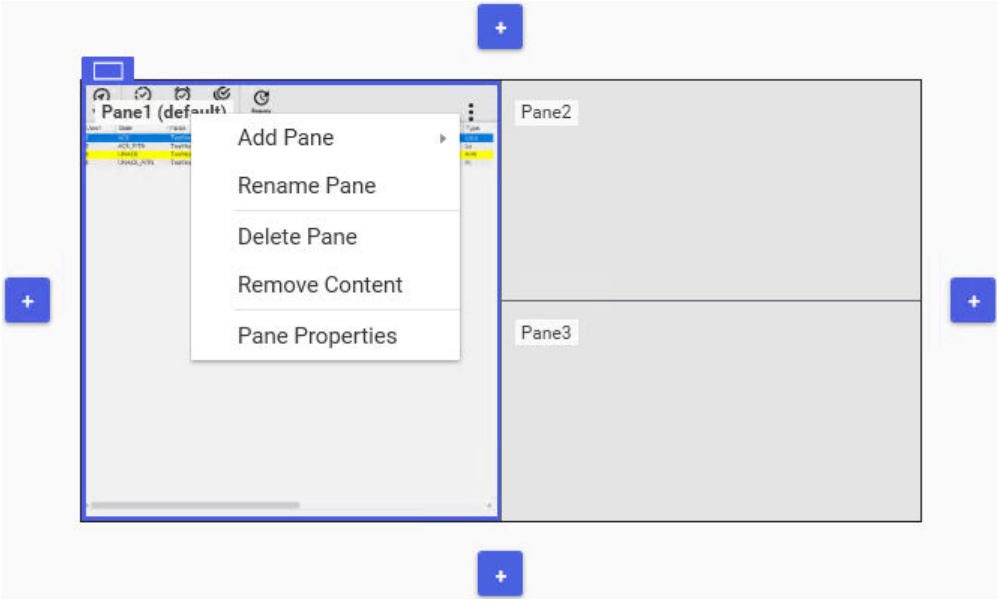
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Pane Menu Button

The Pane Options button appears at the top right corner of a pane when a pane is selected within a layout,. When a pane is not selected, placing the mouse over it shows its **Pane Options** button. Clicking or tapping the button shows a list of the same actions as a shortcut menu.



In Edit Content mode, the Pane Options button includes additional commands to remove content from the pane and edit the pane's properties.



Keyboard Shortcuts

The following keyboard shortcuts are available when a pane is selected in Layout Edit mode:

- Delete key: Remove the selected pane and all of its content.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- F2 key: Rename the selected pane
- Alt + Up, Down, Left, or Right: Split the pane in that direction if applicable.

Configure the Layout Editor

You can use a set of controls to adjust the editing areas of the Layout Editor while you are working on a layout.

Adjust the Size of Layout Editor Areas

The Layout Editor consists of three columns that show an Action List, a graphic of the layout being edited, and a tabbed area at the right to show properties.

Dividers are vertical or horizontal lines that separate the different areas of the Layout Editor. You can resize a Layout Editor area by moving its divider with a drag and drop mouse action or by touch.

Place the cursor directly over a divider to select it.The shape of the cursor changes to a double arrowhead when a divider is selected. Using a mouse, left click and hold to drag the divider to another position. Using touch, select the divider and move the divider with your finger to another position.

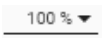
Show or Hide Layout Editor Fields

A triangle icon to the left of the name of a Layout Editor field indicates the field can be expanded or hidden by selecting the triangle.

The triangle points down when the Layout Editor field is expanded to show its content.

Change the Size of a Layout Shown in the Layout Editor

The middle area of the Layout Editor show a graphic representation of the layout currently being edited. A set of pan and zoom controls appear at the bottom of the field to change the size of the Layout graphic. Panning or zooming a layout is useful when the layout is designed for a much larger or smaller screen than the editing area of the Layout Editor.

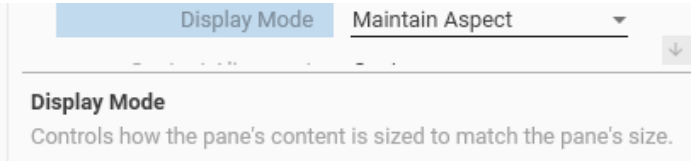
	Moving the slider left or right changes the zoom level of the layout graphic shown in the Layout Editor. You can also move the slider by rotating the wheel of your mouse.
	The Zoom Percentage control shows a drop-down list of zoom percentages. You select an option from the list to resize the layout graphic to the selected percentage.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	The Zoom Box enables a selected area to be zoomed to the full width or height of the current editing area of the Layout Editor. The cursor shows a magnifying glass when the Zoom Box is selected. Drawing a rectangle by mouse or by touch within the editing area shows the content within the rectangle at the full width or height of the current editing area.
	The Zoom to Fit control changes the size of the layout graphic to fit entirely within the current editing area of the Layout Editor.
	The Zoom to 100% control changes the size of the layout graphic to the actual size of the layout.
	You can pan to different parts of the editing area of the Layout Editor with the Pan tool. The cursor changes to a hand when pan mode is selected.

View Descriptions of Layout Properties

You can see a brief description of a selected property by selecting the name of a property shown in the **Properties** grid view. The description appears beneath the list of properties.



Create a layout 1

You access the Layout Editor from the **Visualization folder** to create a screen layout. After creating a layout, you continue editing and configuring the layout by selecting it from the content list shown in the **Visualization folder**. For more information, see Create a layout.

Create a layout 2

You create a layout from the **Visualization folder** of the IDE. After a layout is created, it appears in the list of visual objects in the **Visualization folder** with an identifying icon.

To create a layout

1. Open the **Visualization folder**.
2. On the **Home** ribbon, in the **Create** area, select **Layout**.

   You can also create a layout by other methods:

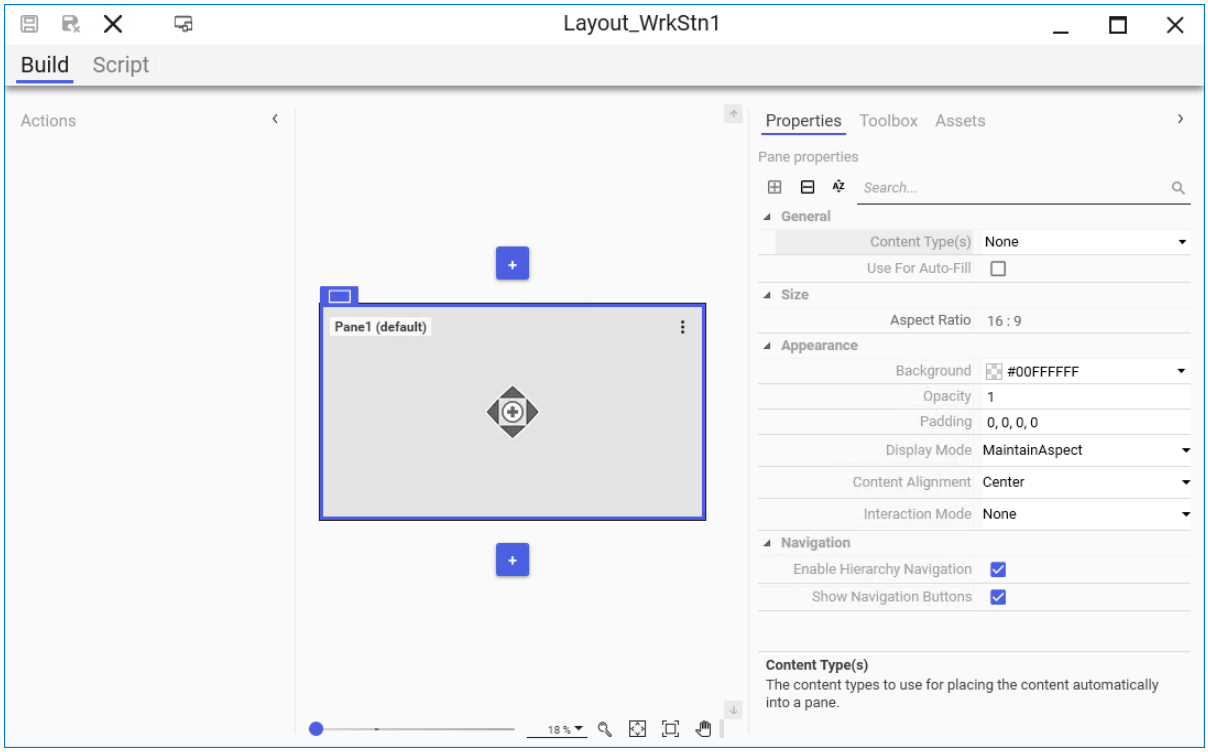
   - **Keyboard shortcut**: Ctrl + Shift + L
   - **Shortcut menu**: Right-click in the **Visualization folder** and select **New**, then **Layout**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

A layout appears in the **Visualization folder** list with a default name of Layout_001 for the first layout that has been created. A blue background around the layout name indicates the name can be changed.

3. Select the layout from the list and rename it.
4. Double-click the layout or right-click and select **Open** to show the layout in the Layout Editor. You can also press Enter to open a layout in the Layout Editor.

   After opening a layout for the first time, the Layout Editor shows a single pane with its properties listed in the right column of the editor.



   You can begin configuring the layout for your View Applications. For more information about configuring a layout profile, see Configure a layout .

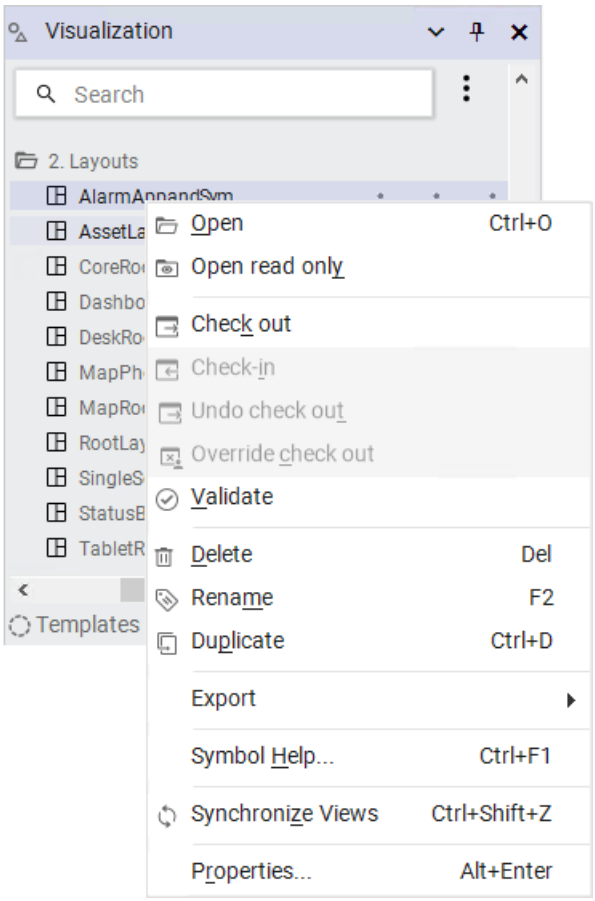
5. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

Manage a layout

You can perform routine tasks to manage a layout from the **Visualization folder**. Management tasks include:

- Delete a layout
- Rename a layout
- Duplicate a layout
- Export a layout
- Import a layout

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Delete a layout

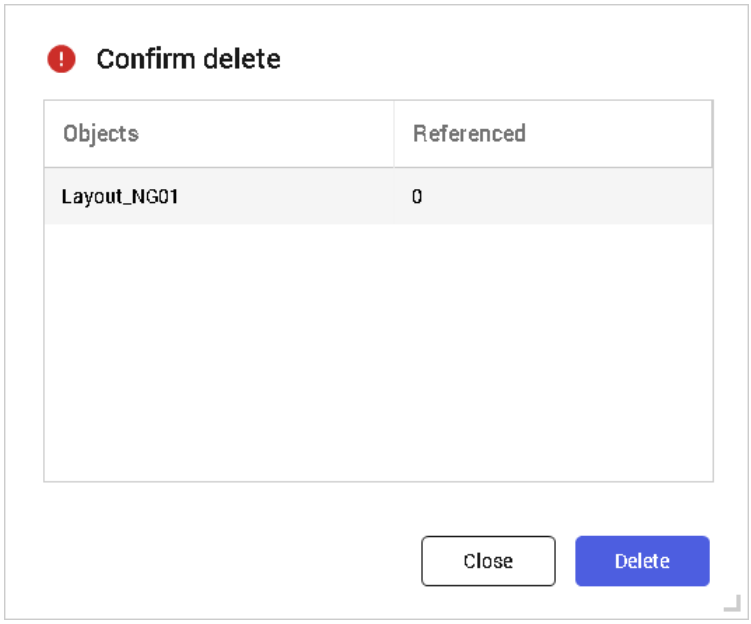
You can delete a layout from the Visualization folder.

To delete a layout

1. Open the **Visualization folder**.
2. If the layout you want to delete is open for editing, close it.
3. Select the layout you want to delete.
4. Press the Delete key or right-click the layout and select **Delete**.

   The **Confirm Delete** dialog box appears. The **Referenced** column shows the number of objects that the layout is referenced by. To see what those objects are, click the down-arrow at the right side of the **Referenced** list. This shows any screen profiles, ViewApp templates, or ViewApp instances that refer to the layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



5. Click **Delete** to delete the layout.

Duplicate a layout

You can duplicate to reduce the effort to create an entirely new layout by using an existing layout as a template.

To duplicate a layout

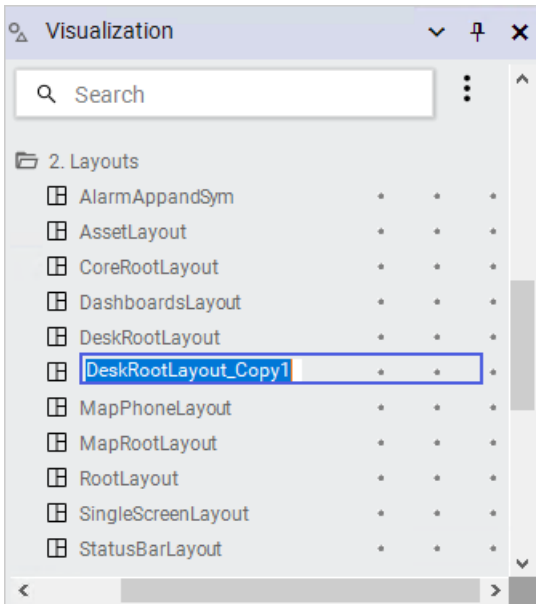
1. Open the Visualization folder and select the layout that you want to duplicate.
2. On the **Home** ribbon, in the **Edit** area, select **Duplicate**.

   You can also duplicate a layout by other methods:

   - **Keyboard shortcut**: **Ctrl + D**
   - **Shortcut menu**: Right-click the name of the layout and click **Duplicate.**

   A new layout appears in the Visualization folder list with Copy1 appended to its name.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. Select the duplicate layout and rename it.

Rename a layout

You can select a layout from the Visualization folder list and rename it by pressing F2 or by clicking on the name to enter edit mode.

A new layout name is validated to ensure it meets naming requirements. If a new name is a duplicate of an existing name or breaks any name rules, a message appears indicating there is a problem with the name.

Pressing the Esc key cancels a rename request and the layout reverts to the original name. Committing an empty name cancels the rename request.

To rename a layout

1. Select the layout that you want to rename from the **Visualization folder** list.
2. Click the layout name or press **F2** to enter edit mode. A blue background appears around the layout name.
3. Type the new name of the layout.
4. Commit the new name by clicking outside of the layout name.

   If the new name is valid, it appears in the **Visualization folder** list. If the name is invalid, you see a dialog box that describes the problem.
5. If you typed an invalid layout name, click **OK** and re-enter another name.

Export a layout

You can export one or more layouts from the Visualization folder to an aaPKG file. You can then import the layouts from the export file to another Galaxy.

To export a layout

1. Open the **Visualization folder**.
2. Select the layouts to be exported.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3. In the **Home** ribbon **Export** area, select **Selected**, then **As package.**

   You can also export a layout from the shortcut menu by right-clicking the layout name, then selecting **Export**, and then **Selected objects**.

   The **Export Selected objects** dialog box appears.
4. Select the folder where you want to save the export file.
5. Assign a name to the export file.

   The default export file name is the name of the first selected layout from the **Visualization folder**.
6. Click **Save**.

   A horizontal bar shows the progress of the layouts being loaded into the export file.

Import a layout

You can import a layout created in another Galaxy to the Visualization folder of the System Platform IDE. Exported layouts are saved in an aaPKG file.

During the import process, you have the choice to keep or overwrite any layouts currently in the Visualization folder with the imported layouts.

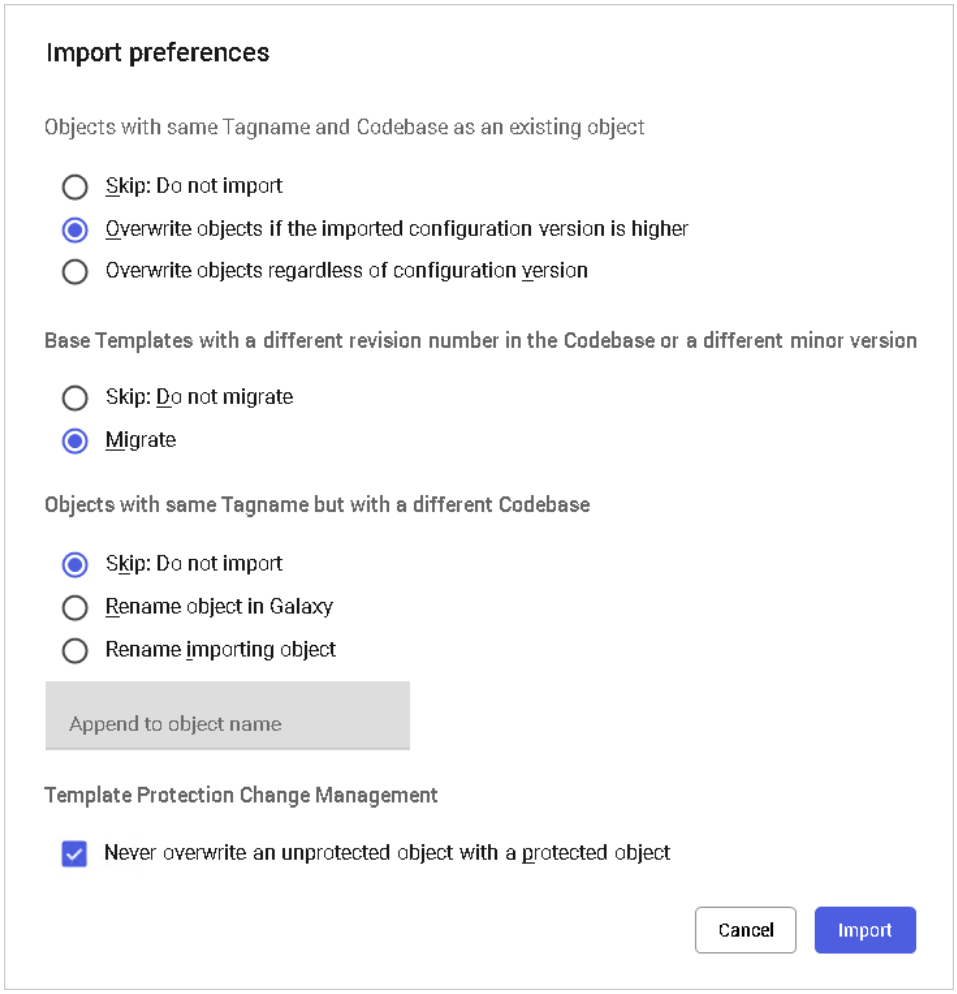
To import one or more layouts

1. Open the **Visualization folder**.
2. Select the **Galaxy** ribbon, then select **Import**.
3. Select **Objects**, then, **From package**.

   The **Import Objects from package** dialog box appears.
4. Go to the folder where the aaPKG file containing layouts is located.
5. Select the aaPKG file and click **Open**.

   The **Import preferences** dialog box opens with options to import layouts. You cannot have two layouts with the same name or more than one copy of the same version of a profile in the same Galaxy. When you import a layout, you can choose options from the **Import preferences** dialog box how you want naming and version conflicts handled.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



6. Click **Import** to import the layouts.

   A progress bar shows the layouts being imported into the Visualization folder. After finishing the import, you should see the layouts in the list of Visualization folder objects.

Configure a layout

The Layout Editor includes several editing modes to change the characteristics of the entire layout, a specific pane in a layout, a pane divider, and pane content. The following table lists the different methods of selecting an edit mode within the Layout Editor.

Edit Modes	Edit Mode Selection Methods
Layout	• Click or tap outside the boundary of a layout shown in the Layout Editor.
Pane	• Click or tap an empty pane to show the selected pane's properties. • Select **Pane Properties** from the **Pane Options** menu of a pane containing content
Divider	• Click or tap on a pane divider within a layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Content	• Click or tap within a pane containing content to edit content properties. • Select **Edit** *content_name* from the **Pane Options** menu to edit graphic content within the Industrial Graphic Editor.

When switching edit modes, the **Property** field automatically adjusts to show the properties appropriate for the selected object in the Layout Editor.

This topic describes the different tasks to configure the physical appearance of a layout. For more information about editing the properties of a layout or a pane, see Set layout properties or Set pane properties 1. See Add content to panes for the steps to add content to a pane.

Add panes to layout

Fixed Layout Mode

When in **fixed layout mode**, you can add panes by a variety of methods:

Method	**Add Pane Above**	**Add Pane Below**	**Add Pane Right**	**Add Pane Left**
Add Pane Control				
Keyboard Shortcuts	Alt + Up	Alt + Down	Alt + Right	Alt + Left
Shortcut Menu	Right-click **Add Pane**, then **Above**	Right-click **Add Pane**, then **Below**	Right-click **Add Pane**, then **Right**	Right-click **Add Pane**, then **Left**
Pane Options Button	Click **Pane Options** button, select **Add Pane**, then select **Above**	Click **Pane Options** button, select **Add Pane**, then select **Below**	Click **Pane Options** button, select **Add Pane**, then select **Right**	Click **Pane Options** button, select **Add Pane**, then select **Left**

Initially the Layout Editor shows a single pane. As panes are added to a layout, dividers are automatically created between adjacent panes. Dividers are one pixel thick, but can be changed by setting the divider's **Thickness** property . By default, the newly added pane will be one pixel less in size.

When a pane is added, the Layout Editor divides the selected pane in half regardless of whether it is variable or fixed. If the selected pane has a minimum size that is more than half its current size, then the selected pane will be set to its minimum size and the new pane will take the remaining available size.

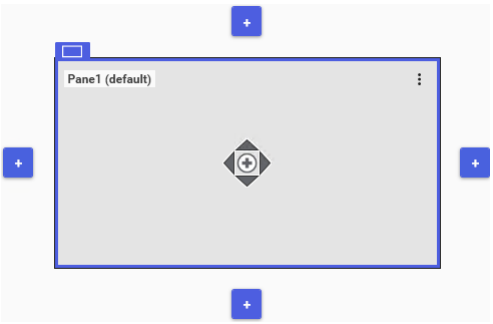
To add a pane to a fixed layout

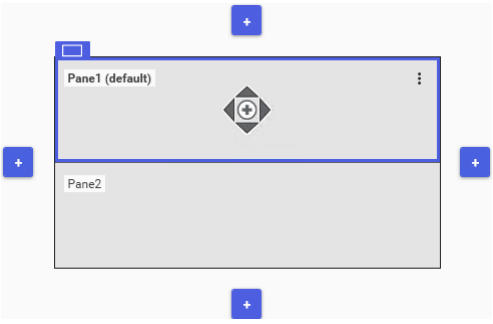
1. Open a layout from the **Visualization folder**.

   The **Add Pane** control appears in the center of the default pane with arrows to create a new pane above, below, right, or left of the current pane.

2. Select an arrow of the **Add Pane** control to add a pane to the layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





The new pane that is added is named Pane<*n*>, where *n* is a number from 1 to 50.

Note: A fixed layout supports up to a maximum of 50 panes. You will not be able to add panes beyond the 50 pane limit.

3. Select **Save** to save your changes to the layout.

Responsive Layout Mode

When in **Responsive layout mode**, just click the **Add Pane** button to add a new pane. Each new pane is added below the last pane added.

Add Pane Button	

To add a pane to a responsive layout

1. Open a layout from the **Visualization folder**.
2. Switch to responsive mode by enabling the **IsResponsive** layout property, or by selecting the **Fixed/ Responsive Layout** button. The **IsResponsive** property is visible when layout properties are selected (click above the layout to switch to the layout properties view).
3. Press the **Add Pane** button.

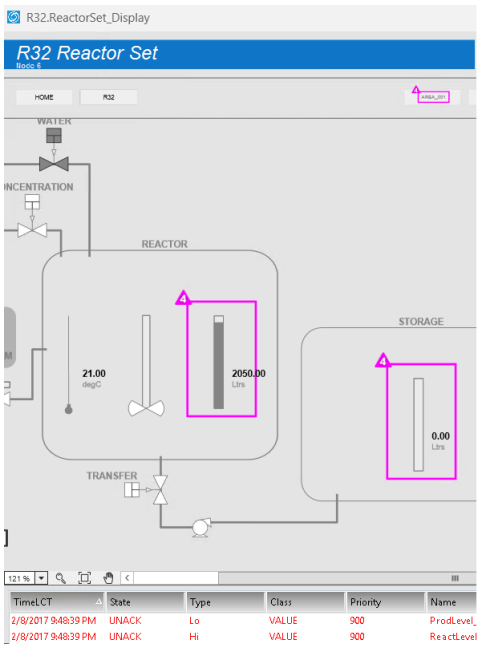
   The new pane that is added is named Pane<*n*>, where *n* is an integer.

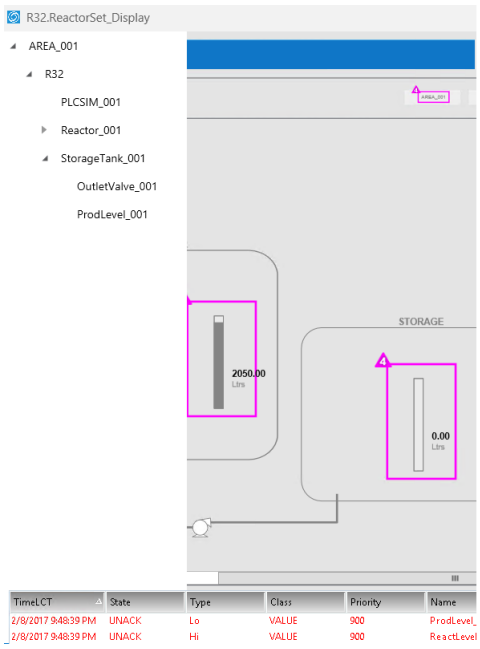
   Note: A responsive layout does not automatically limit the number of panes you can add. However, as you add panes and content, more computer resources are consumed which can impact performance, both during configuration and at runtime.

Configure a layout: slide-in panes

A layout can have a slide-in pane configured at each side of a layout window. During runtime, slide-in panes are hidden until the user selects the tab at the side of a layout to extend the slide-in pane and show its contents. In the example below, selecting the tab at the left border of the layout shows a NavTree control in a slide-in pane. Clicking or tapping outside of the slide-in pane hides it

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





Configure a layout: add slide-in panes

You can add up to a maximum of four slide-in panes to a layout.

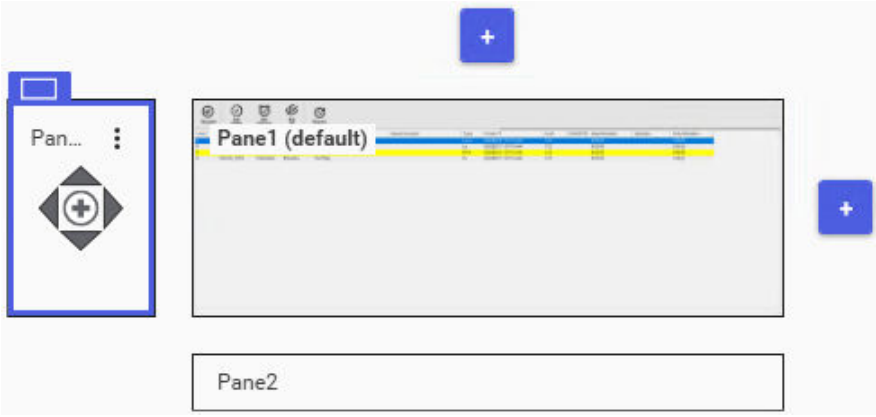
To add slide-in panes to a layout

1. Open a layout in the Layout Editor.

   The Layout Editor shows four Add buttons with a plus sign at each side of a layout.

2. Select the square on the side of the layout in which you want to add a slide-in pane.

   The added slide-in pane appears next to the layout.



All standard layout configuration operations can be done to a slide-in pane. The one exception is that a slide-in pane cannot be set as the layout's default pane.

Slide-in panes are included in a layout's maximum pane count. If the maximum pane count has been reached, the pane Add buttons are disabled within the Layout Editor.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

A slide-in pane always spans the full height or width of a layout window. A slide-in pane can be divided into smaller panes, but the frame of the slide-in pane is always the full width or height of the layout window.

The initial width of a slide-in pane on the left or right is 25% of the layout's width, and the initial height of a slide-in pane on the top or bottom is 25% of the layout's height.

When the dimensions of a layout are resized, slide-in panes are resized accordingly. When the width is changed, the top or bottom slide-in panes are resized to match the new width. When a layout's height is changed, the left or right slide-in panes are resized to match the new height.

Right clicking a pane within the slide-in pane or selecting the **Pane Options** menu button and selecting **Delete Slide-In Pane** removes the slide-in pane from the layout. Any navigation actions associated with the slide-in pane are removed from the Action List. The default pane in the layout is selected and the layout is zoomed to fit.

Resize slide-In panes

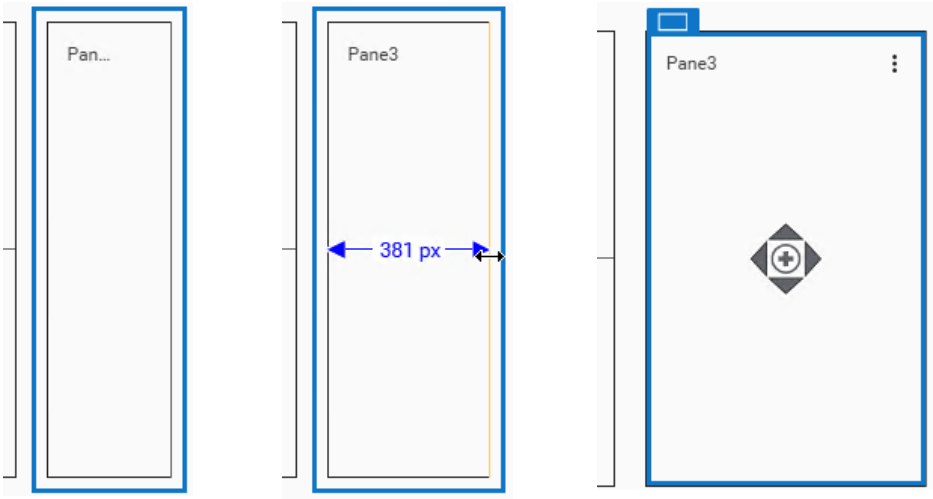
By default, a slide-in pane extends to 25 percent of a layout's width or height during runtime. You can resize slide-in panes with the Layout Editor. The upper and lower slide-in panes can be resized vertically to change their height. The left and right slide-in panes can be resized horizontally to change their width.

The size of a slide-in pane can be increased up to the maximum layout width or height minus 16 pixels. This small 16 pixel band provides sufficient space within the layout window for the user to click outside of the slide-in pane to hide it during runtime.

To resize a slide-in pane

1. With the Layout Editor, open the layout containing the slide-in pane to resize.
2. Tap or click the outside border of the slide-in pane.

   The cursor changes to a double arrowhead and the outside border of the selected slide-in pane changes color to indicate the pane can be resized. The pixel width of the slide-in pane also appears while it is being resized.



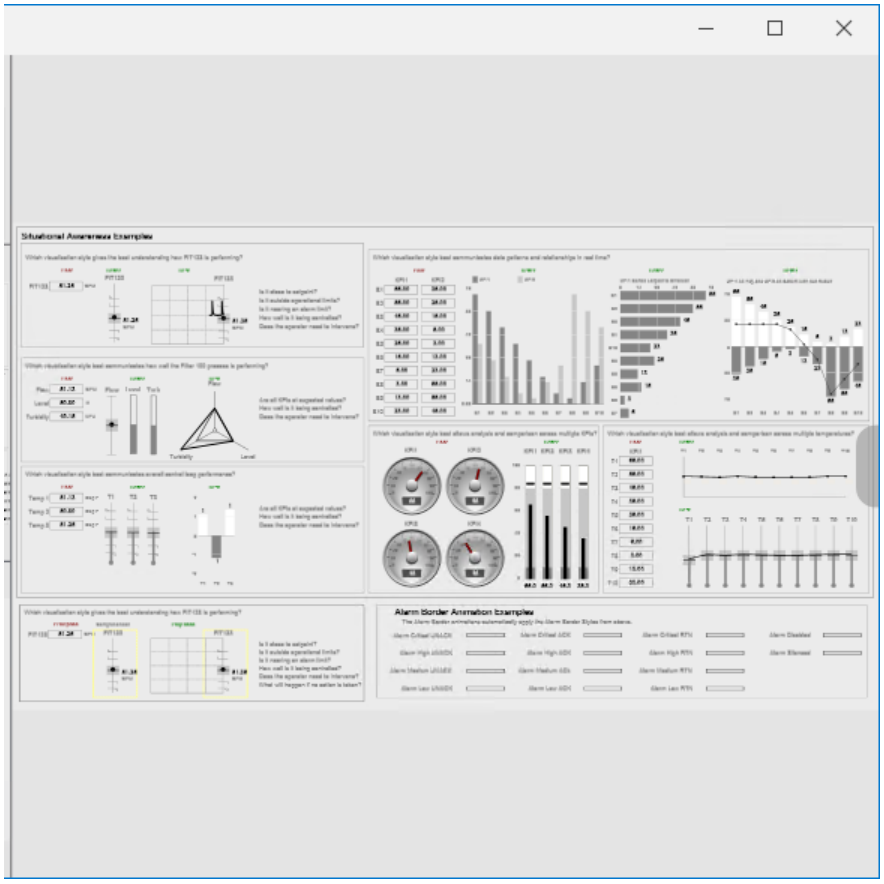
3. Drag the pane border to the desired position to change the size of the pane.
4. Save your changes to the layout.

Show slide-in panes at runtime

During runtime, users can show a slide-in pane by one of the following methods:

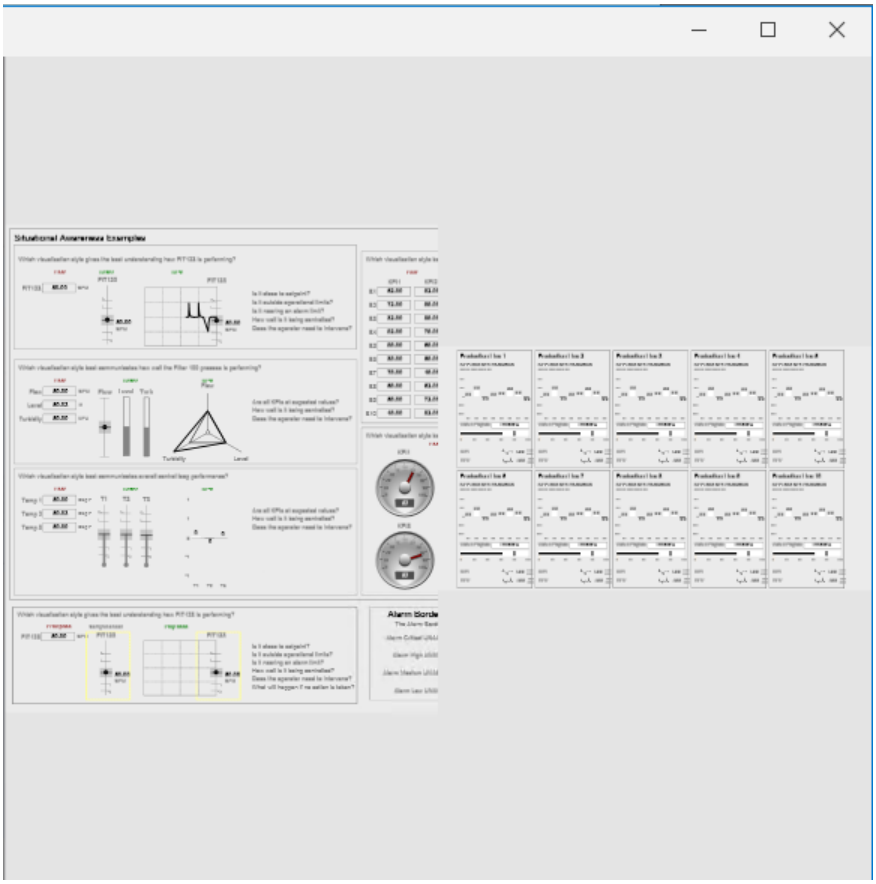
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Click or tap on the slide-in pane tab.
- Press the keyboard shortcut Alt+Shift+<Arrow>, where <Arrow> is the arrow key in the direction away from the side with the slide-in pane. For example, if you have a slide-in pane at the left side of a layout, press the right arrow key.
- Perform a three-finger swipe gesture in the direction away from the side of the layout containing a slide-in pane.



The slide-in pane and its contents appear over the existing pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The background of the whole slide-in pane is the same as the layout's configured background color. If the layout's background color is transparent, the slide-in pane has an opaque white background behind it

When the layout window is not the same size as the layout's configured size, the slide-in pane size changes to remain at the same proportion as the window size.

Users can hide a slide-in pane while the ViewApp is running by one of the following methods:

- Click or tap outside of the slide-in pane, including changing focus to another window
- Press the Esc key.
- Perform a three-finger swipe gesture towards the side of the ViewApp window containing a slide-in pane.

Rename a pane

You can rename a pane within the Layout Editor by three methods.

Current Pane Name	Click or tap on the current pane name to type a new name.
Pane Options Menu	Click or tap the Pane Options help and select **Rename Pane**.
Shortcut Menu	Right-click within the pane and select **Rename Pane**.

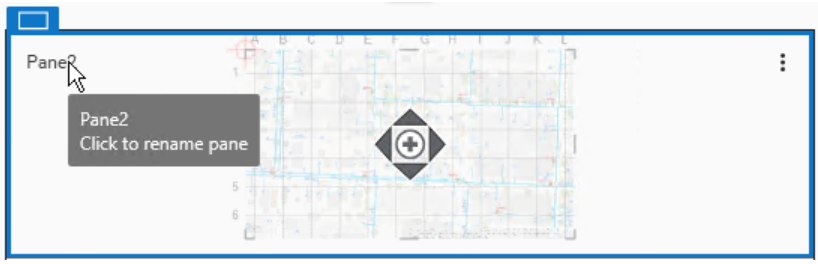
To rename a pane

1. Open a layout from the **Visualization folder**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. Select the pane to rename.

   When you place the cursor directly over the current pane name, a tooltip indicates the pane can be renamed.



3. Click the name of the pane.

   A blue background behind the name of the pane indicates you are in edit mode.

4. Type a new name for the pane.

   You can press Esc to cancel the rename request and keep the pane's current name.

5. After typing the new name, click outside of the name box or press Enter to commit the new name.

   If you typed an invalid pane name, a red border appears around the edit box and a tooltip describes the problem.

6. Click **Save** to save the new pane name.

Remove a pane from a layout

You can remove a pane from a layout. If the removed pane is the default pane, the remaining pane with the largest space within the layout is assigned as the default pane. The freed space of the removed pane is assigned to the layout's remaining panes.

You can remove a pane from a layout by three methods.

Delete Key	Select the pane to delete and press the Delete key.
Pane Options Menu	Click or tap the **Pane Options** button and select **Delete Pane**.
Shortcut Menu	Right-click the pane and select **Delete Pane**.

To remove a pane from a layout

1. Open a layout from the **Visualization folder**.
2. Select the pane you want to remove from the layout.
3. Select the **Pane Option** button to show a list of pane commands.
4. Select **Delete Pane** from the list.

   The pane is removed immediately.

5. Click **Save** to save your changes to the layout.

Resize a pane

During design time, you can change the size of a pane by moving its divider or changing its size properties.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

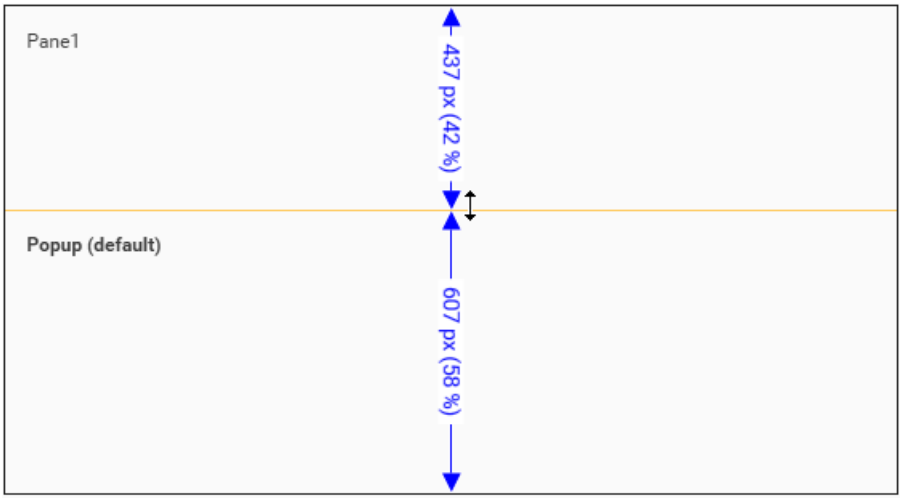
To resize a pane using a divider

1. Open the layout from the **Visualization folder**.
2. Select the pane to resize.
3. Place the cursor over the divider between two panes to select the divider.

   The mouse cursor changes shape and the divider changes color when the divider is selected.

   Note: If you are using a touch device, you can also select and move a divider using standard touch gestures.
4. Drag the divider to change the size of the panes.

   As you move the divider, you see the percentage and pixel counts of the two panes sharing the divider.



5. Save your changes.

To resize a pane using Layout properties

You can also adjust pane size by changing the value of a pane's **Width** or **Height** Layout property.

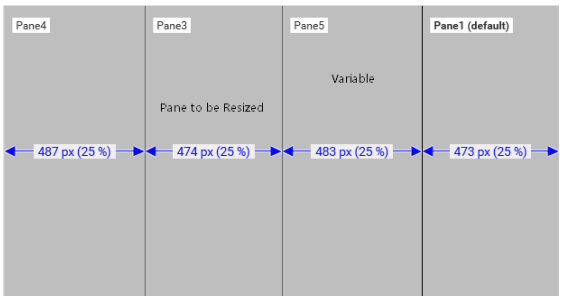
The **Width Sizing** and **Height Sizing** properties include **Fixed** or **Variable** values that determine whether a pane's width or height will change or remain fixed when the parent pane or layout is resized during runtime. A **Fixed** sizing property maintains the pane's configured size while a **Variable** sizing property enables the pane to change size in response to changes in the layout's size during runtime.

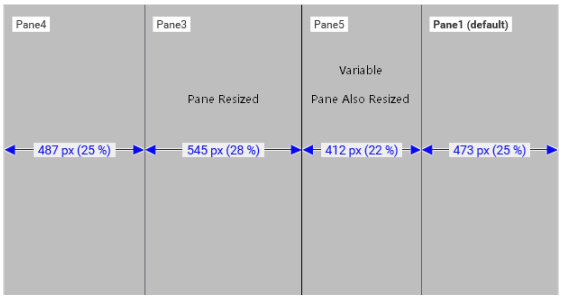
When a pane is resized by changing its **Width** or **Height** property, another pane in the same row or column is resized according to the following precedence rules.

Note: After means to the right or below the pane being resized. Before means to the left or above the pane being resized.

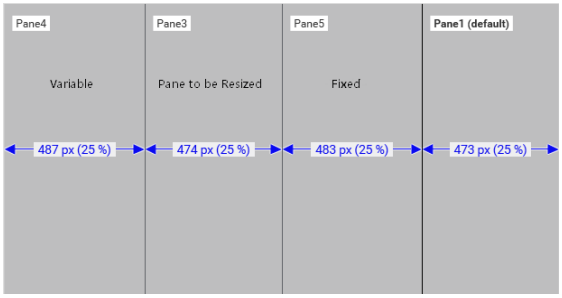
- The pane immediately after the resized pane if its **Width Sizing** and **Height Sizing** properties are set to **Variable**

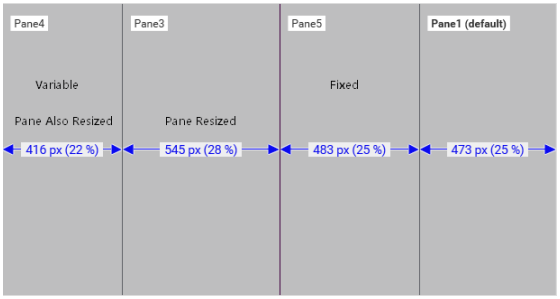
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



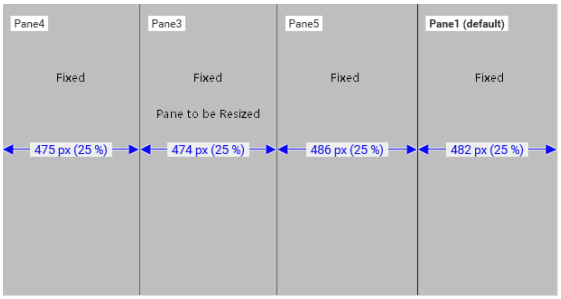


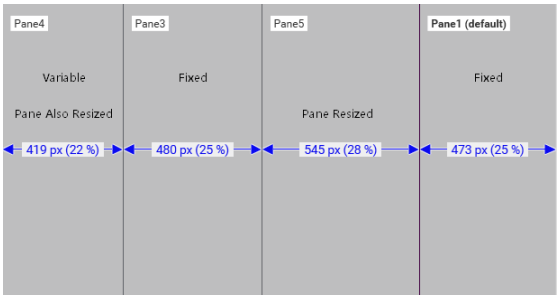
- The pane immediately before the resized pane, if its **Width Sizing** and **Height Sizing** properties are set to **Variable**



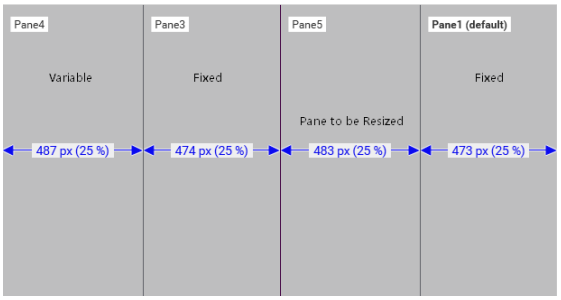


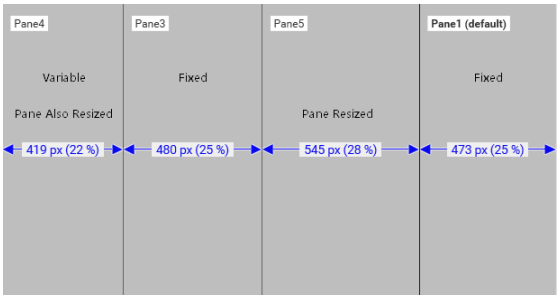
- The first pane after the resized pane whose **Width Sizing** and **Height Sizing** properties are set to **Variable**





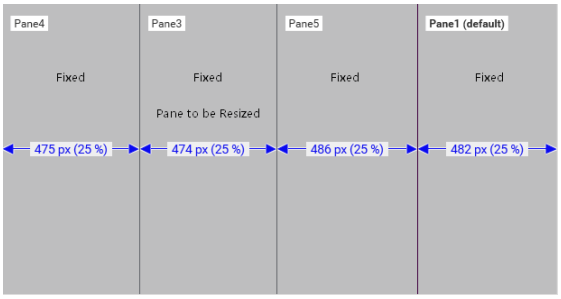
- The first Variable pane before the resized pane

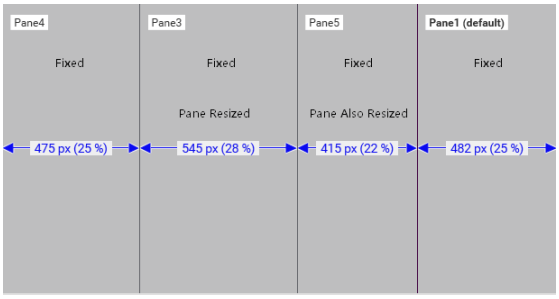




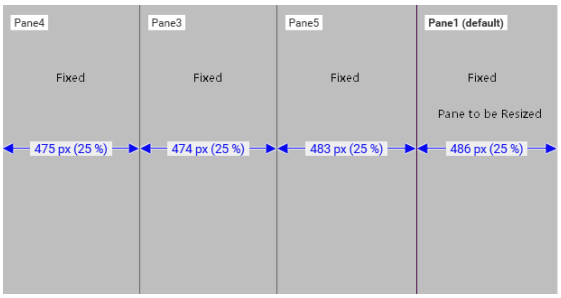
- The pane immediately after the resized pane if no pane's **Width Sizing** and **Height Sizing** properties are set to **Variable**

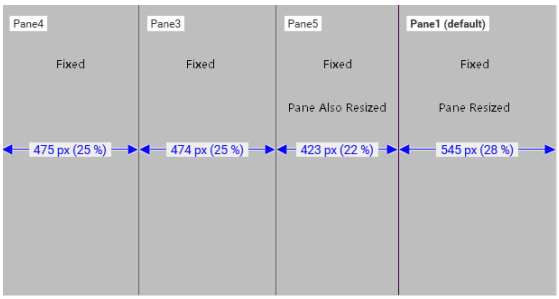
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





- The pane immediately before the resized pane if all other panes in the row or column are to the left or beneath and whose **Width Sizing** and **Height Sizing** properties are not set to **Variable**





If an appropriate pane is not found that matches a precedence rule, the pane sizing algorithm continues to the next precedence rule until a pane is found that matches the requirements of the precedence rule.

Resize a pane during runtime

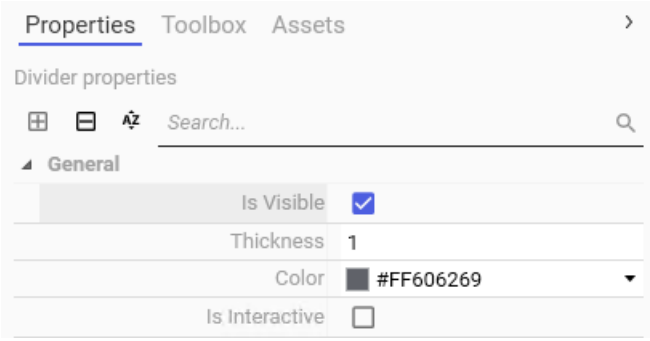
Layouts can be designed to enable users to change the size of a pane during runtime by selecting and moving a pane divider. When selected, the shape of a mouse cursor changes to a double arrowhead to indicate to the user the pane can be resized.

To enable runtime pane resizing

1. From the Layout Editor, select a divider that you want to be interactive during runtime.

   Divider properties appear in the **Properties** area of the Layout Editor.

2. Select the **Is Interactive** property.

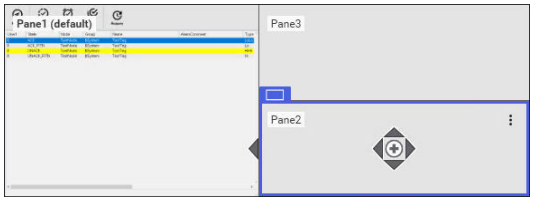


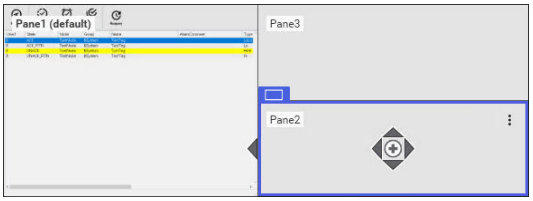
Extend panes in a layout

Extenders are triangular arrowheads that appear on vertical and horizontal dividers between panes of a layout.

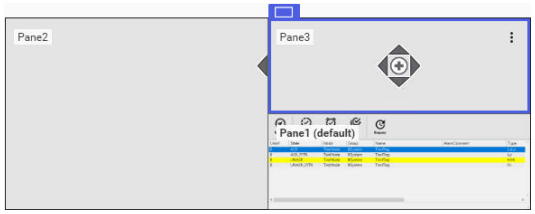
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

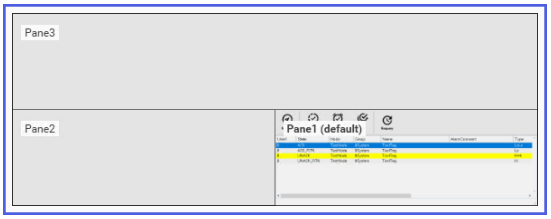
An extender enables the source pane to extend into the area of a target pane. In the example below, Pane 2 is the source pane that will be extended into Pane 1, which is the target pane.





Extending a pane effectively rotates the orientation of the first parent common to both the source and target panes. In the following example, Pane 1 is the parent of Pane 2 and 3. When Pane 3 is extended into Pane 2, Pane 1's orientation changes from a column to a row.





For more information about the rules of extending panes, see Extend panes in a layout.

To extend a pane in a layout

1. Open the layout from the **Visualization folder**.

   The default pane of the layout is selected.

2. Select the pane to be extended.

   An extender appears on a pane border that can be extended.

3. Tap or click the extender to extend the pane
4. Save your change.

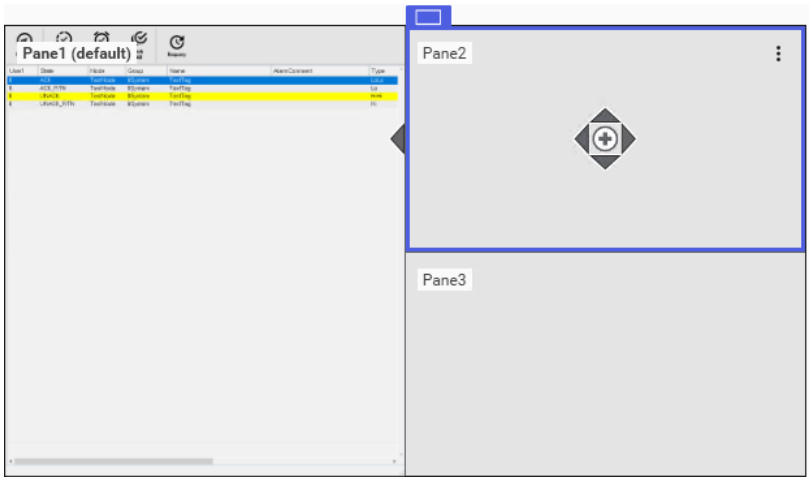
Extend panes in a layout

A layout shows panes arranged in vertical columns or horizontal rows. The Extend Pane command enables a source pane fitting the correct criteria to be extended Up/Down/Left/Right into a sibling target pane. In the following examples, Pane 1 is the parent pane.

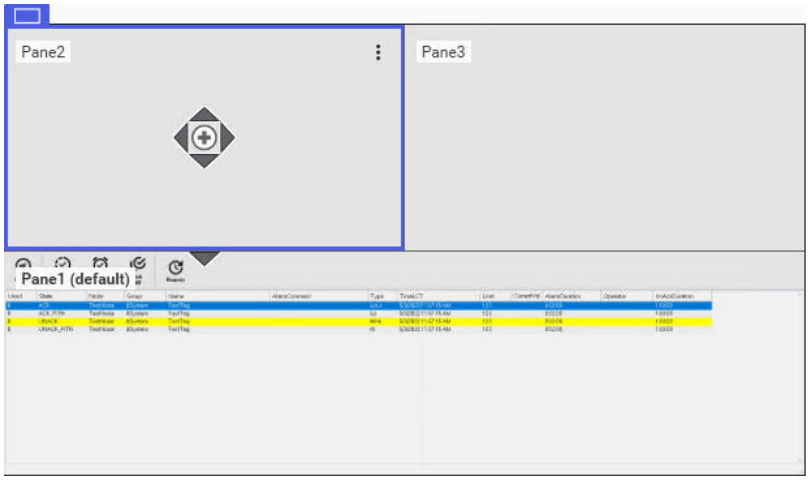
A pane can be extended based on the following rules:

- A pane that is a row of its parent pane can be extended left or right

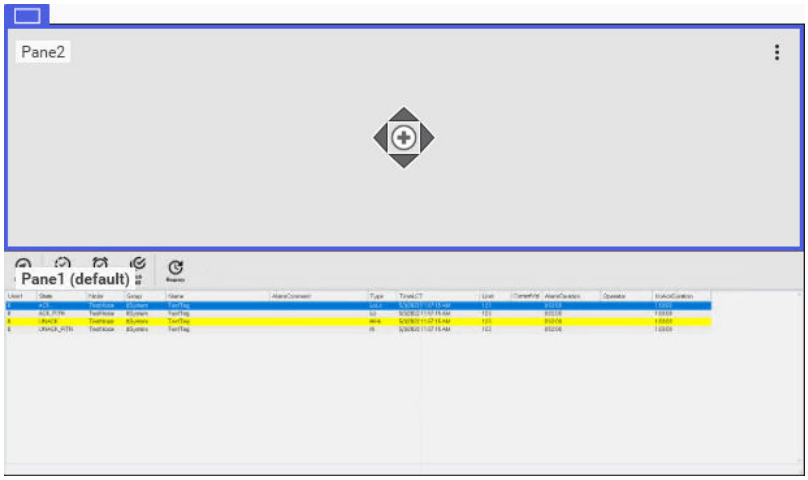
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



- A pane that is a column of its parent pane can be extended up or down

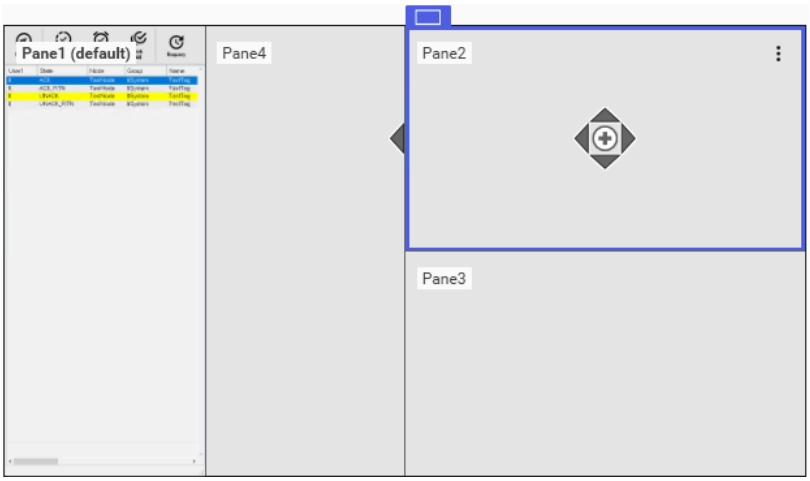


- A pane adjacent to the border of a layout window cannot be extended outwards beyond the layout's border.
- A pane that is an immediate child of its parent pane cannot be extended



- A pane can be extended into its parent pane’s immediate siblings

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



- A pane cannot be extended if the number of panes will exceed the maximum of 50

Set the default pane of a layout

A layout includes a default pane, which should be the pane that is most likely to be updated with changed content during runtime. A default pane is commonly used with navigation actions when no pane is specified.

A layout containing only one pane is the default. If you add one or more panes, the original Pane 1 is designated as the default pane. You can assign a new default pane if you do not want to use Pane 1 as the default pane.

To select the default pane of a layout

1. Open the layout from the **Visualization folder**.
2. Select the pane that you want to assign as the default pane.
3. Right-click the selected pane to show the shortcut menu.
4. Select **Set as Default Pane**.

   The name of the pane is shown in bold and the word default appears in parentheses to the right of the pane name.
5. Select **Save** to save your changes.

Set the pane presentation style

Content in a layout can be shown by three pane presentation styles

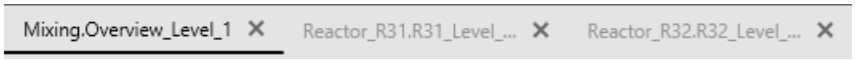
- **Single**

  In Single mode, the pane hosts a single content item (Layout, Display, or Graphic) shown during runtime. Single is the default pane presentation style.
- **Multiple**

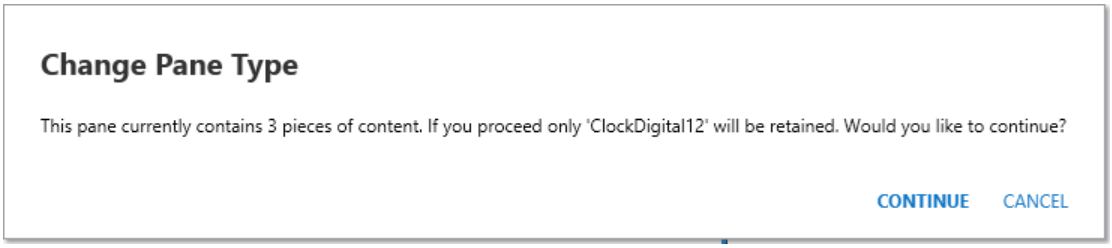
  In Multiple mode, two or more content items are added to a multi-content pane by a script or a navigation API. Each content item is placed in a separate pane. During runtime, the user cannot select an item from Multiple panes. A script or navigation API controls the ordering of the panes within the stack. The content in the top most pane is visible during runtime.
- **Tabbed**

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

In Tabbed mode, two or more content items are added to a multi-content pane. Each content item is placed in a separate tabbed pane. During runtime, the user selects a tab to show its content in the pane.

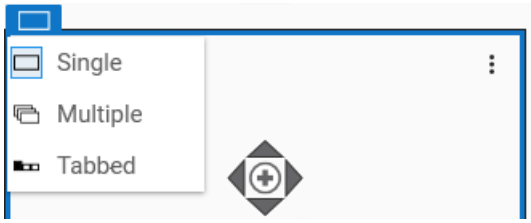


When you change a pane's presentation style from Single to either Multiple or Tabbed, any existing content in the pane is maintained. However, when you change the presentation style from Multiple or Tabbed to Single, you see a confirmation message that all content except the current selection in the action list will be removed from the pane.



To set the presentation style of a pane

1. Open the layout from the **Visualization folder**.
2. Select the pane to set a presentation style for.
3. Select the **Presentation Style** selector at the top left corner of the selected pane.



4. Select a presentation style from the list.

   The icon shown in the **Presentation Style** selector changes to match the presentation style you selected.
5. Select **Save** to save your changes.

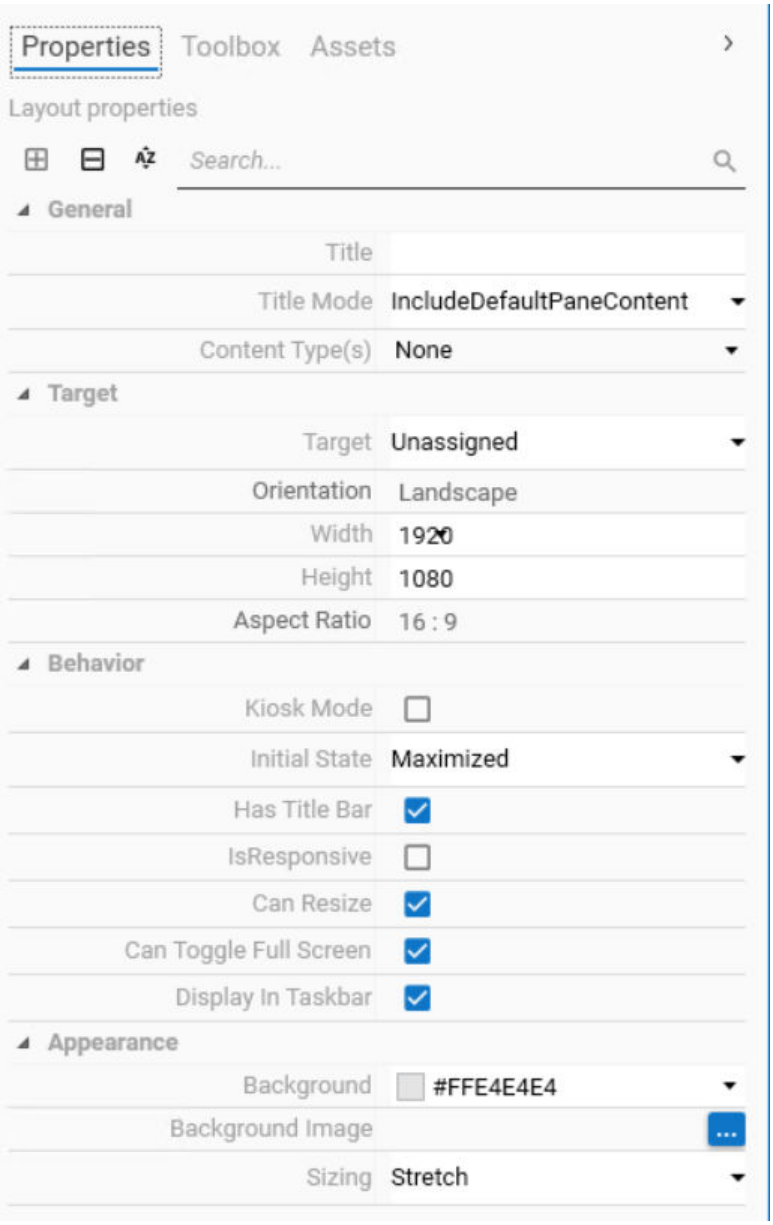
Set layout properties

Layout properties define the overall visual characteristics and runtime behavior of a layout. For information about pane properties, see Set pane properties 1.

You can show layout properties from the Layout Editor by clicking or tapping outside of the displayed layout.

The following table shows the default values of layout properties. When no pane or divider is selected, layout properties show the properties of the overall layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Other property fields appear when some layout properties are assigned specific values. The following table lists any prerequisite property values in the **Prerequisite Property Values** column.

The following table lists all layout properties. The properties are organized into the categories shown in the **Properties** window of the Layout Editor. You can also view layout properties in alphabetical order by selecting the **Show Alphabetical** icon next to the **Search** field.

You type values or select values from drop-down lists to assign values to layout properties. A red background appears when you enter an invalid property value.

Short descriptions of the properties appear at the bottom of the **Properties** window for each selected property.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Property Category	Property	Default Property Value	Prerequisite Property Values	Description
General				
	Title	<Empty>	None	Text shown in the title bar of the layout's window during runtime.
	Title Mode	IncludeDefault PaneContent	None	Options specify whether the name of the default content shown in the default pane is appended to the title of the layout or not. **TitleOnly**: Shows only the title assigned to the layout. **IncludeDefaultPaneContent**: The name of the content placed in the default pane is appended to the title of the layout window.
	Content Type	<Empty>	None	Specifies the type of content represented by a layout. Select the arrow at the right of the **Content Type** field to show a drop-down list of content types. One or more content types can be selected from the list.
Target				
	Target	Unassigned	None	Specifies if the layout is intended for a screen, a pane of another layout, a pop-up, or is unassigned.
	Screen Profile	N/A	Target=Screen	Name of the screen profile containing a screen that the layout is intended to be displayed on. For more information about associating a screen profile to a layout, see Select a screen profile layout property.
	Screen	N/A	Target=Screen	Name of the screen within a Screen Profile that the layout is designed to be displayed on.
	Layout	N/A	Target=Pane	Name of the layout containing the target pane that this layout is designed to be hosted in.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	Pane	N/A	Target=Pane	Name of the pane within the selected layout that this layout is designed to be hosted in.
	Orientation	N/A	Target=Screen or Target=Pane or Target=Popup	Orientation of the layout(Portrait or Landscape) determined by the selected screen profile screen or pane. Read only.
	Width	N/A	None	Width of the layout in pixels.
	Height	N/A	None	Height of the layout in pixels.
	Aspect Ratio	1.91:1	None	Calculated aspect ratio of the width to height size of a layout.
Behavior				
	Kiosk Mode	False	None	When set to True, the System Menu, and Window Controls (Maximize/ Restore, Minimize and Close) buttons and their underlying functions are inaccessible during runtime.
	Initial State	Maximized	Kiosk Mode=False	Initial state of a layout's window during runtime. The initial state of the layout window can be one of the following: Maximized Full Screen Normal Minimized
	Has Title Bar	True	Kiosk Mode=False	When set to True, a title bar appears at the top of the layout's window, which is 33 pixels in height. As a result, the height of a layout designed for a screen resolution of 1920 x 1080 is adjusted to 1047 pixels to accommodate the height of the title bar.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	IsResponsive	False	Kiosk Mode=False	When set to True, the Layout editor enters responsive layout mode. For more information about Responsive Layouts, see About the responsive layout mode. See the following table for a set of layout properties that are unique to responsive layouts and only appear when the **IsResponsive** property is set to True.
	Can Resize	True	Kiosk Mode=False	When set to True, the user can resize the running View Application window by dragging the horizontal or vertical border. Also, selecting either the Minimize or Maximize Windows can be selected to resize the layout.
	Can Toggle Full Screen	True	Kiosk Mode=False	When set to True, the user can press the F11 key to toggle the running View Application window in and out of full screen mode.
	Display In Taskbar	True	None	When set to True, the layout's window will appear in the Windows taskbar.
Appearance				
	Background	#FFE4E4E4	None	Background color of a layout or a pane. Select the arrow at the right of the property field to show a color picker to set a background color.
	Background Image	<Empty>	Valid image file types / extensions: bmp, jpeg, jpg, gif, png,tif, tiff, ico, wmf, emf	Image file used as the background of a layout. The image spans all panes of a layout. Select the file picker button to open a file browser window, then navigate to an image file and select **Open** to set the background image. To remove a background image, select the **X** that appears next to the image name.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	Sizing	Stretch		Sets the method for resizing the Background Image file to fit the layout. **Stretch:** matches the height and width of the image file with the layout size. This can skew the image aspect ratio. **Horizontal:** matches the width of the image file to the width of the layout. This preserves the image aspect ratio, but can result in empty space above and below the image (letter-boxing), or clipping the top and bottom of the image. **Vertical:** matches the height of the specified image file to the height of the layout. This preserves the image aspect ratio, but can result in empty space to the right and left of the image (letter-boxing), or clipping the sides of the image. **Tile:** preserves the original height and width of the specified image file, but tiles (repeats) it to fill the entire layout. If any blank space appears around the image due to letter-boxing, the background color of the layout is shown.

The following table shows layout properties that apply only to responsive layouts. The **IsResponsive** property must be set to True to show these properties.

Property Category	Property	Default Property Value	Prerequisite Property Values	Description
Extra Small				
	Column Span Extra Small	12	IsResponsive=True	Number of columns a pane can span when a responsive layout is displayed on an extra small screen.
	Column Push Extra Small	0	IsResponsive=True	Number of columns to push a responsive layout pane when displayed on an extra small screen.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	Column Pull Extra Small	0	IsResponsive=True	Number of columns to pull a responsive layout pane when displayed on an extra small screen.
	Pane Offset Small	0	IsResponsive=True	Number of columns to offset a responsive layout pane when displayed on a small screen.
	Pane Offset Extra Small	0	IsResponsive=True	Number of columns to offset a responsive layout pane when displayed on an extra small screen.
Small				
	Column Span Small	0	IsResponsive= True	Number of columns a pane can span when a responsive layout is displayed on a small screen.
	Column Push Small	0	IsResponsive= True	Number of columns to push a responsive layout pane when displayed on a small screen.
	Column Pull Small	0	IsResponsive= True	Number of columns to pull a responsive layout pane when displayed on a small screen.
	Show Pane Small	True	IsResponsive= True	When set to True, a pane can be shown on a small screen.
	Pane Offset Small	0	IsResponsive= True	Number of columns to offset a responsive layout pane when displayed on a small screen.
Medium				
	Column Span Medium	0	IsResponsive= True	Number of columns a pane can span when a responsive layout is displayed on a medium screen.
	Column Push Medium	0	IsResponsive= True	Number of columns to push a responsive layout pane when displayed on a medium screen.
	Column Pull Medium	0	IsResponsive= True	Number of columns to pull a responsive layout pane when displayed on a medium screen.
	Show Pane Medium	True	IsResponsive= True	When set to True, a pane can be shown on a medium screen.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	Pane Offset Medium	0	IsResponsive= True	Number of columns to offset a responsive layout pane when displayed on a medium screen.
Large				
	Column Span Large	0	IsResponsive= True	Number of columns a pane can span when a responsive layout is displayed on a large screen.
	Column Push Large	0	IsResponsive= True	Number of columns to push a responsive layout pane when displayed on a large screen.
	Column Pull Large	0	IsResponsive= True	Number of columns to pull a responsive layout pane when displayed on a large screen.
	Show Pane Large	True	IsResponsive= True	When set to True, a pane can be shown on a large screen.
	Pane Offset Large	0	IsResponsive=True	Number of columns to offset a responsive layout pane when displayed on a large screen.
Navigation				
	Enable Hierarchy Navigation	True	IsResponsive= True	When set to True, a responsive layout pane supports Natural Navigation by touch gestures or graphic buttons.
	Show Navigation Buttons	False	IsResponsive= True	When set to True, a responsive layout pane shows graphic buttons for Natural Navigation.

Set layout properties values

You set the properties of a layout by entering values in the data entry fields of the right window of the Layout Editor.

To set values of layout properties

1. Open a layout from the **Visualization folder**.
2. Select outside of the layout in the edit area to ensure no pane or divider is selected.
3. Select the **Properties** grid of the Layout Editor to show the layout's current property values.

   Not all layout properties may appear after selecting the **Properties** grid. Some properties only appear after other layout properties are set to specific values. For more information about each layout property, see Set layout properties.
4. Assign values to layout properties.

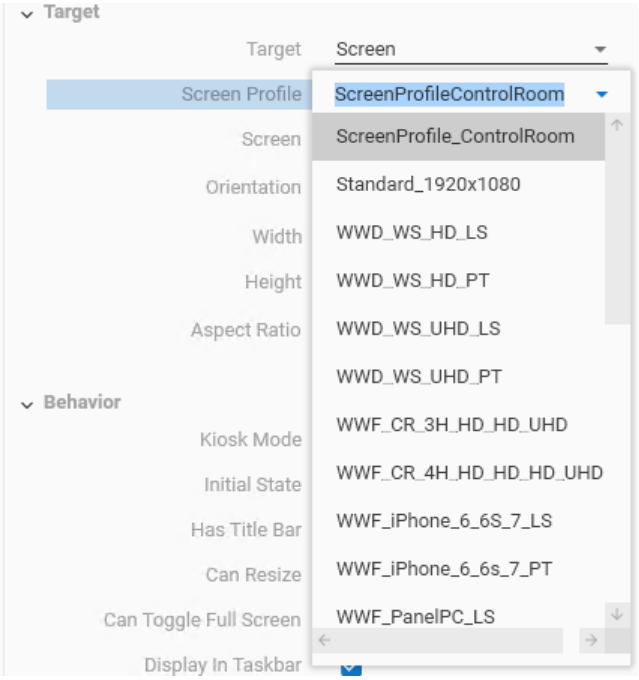
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

If you enter an invalid value in the property grid, the background color of property field changes to a a light red. You can hover over the field with your mouse to show a tooltip error message describing why the entered value is invalid.

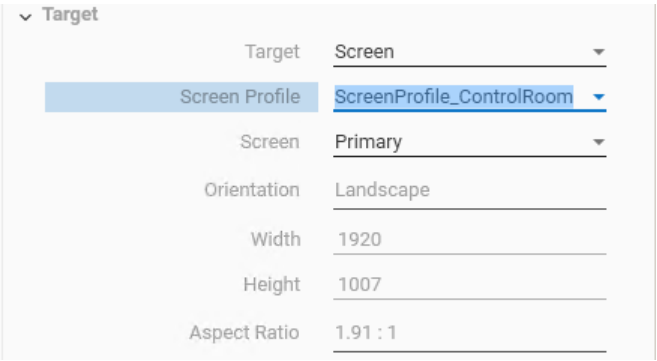
5. Select **Save** to save your changes to the layout properties.

Select a screen profile layout property

If the **Target** property is set to **Screen**, the **Screen Profile** property appears in the **Properties** window containing a drop-down list of screen profiles defined in the Galaxy.



When a screen profile is selected from the list, the **Screen** property appears containing a list of all screens defined in the selected screen profile.



Based on the screen assigned as the value of the **Screen** property, the layout is updated to match the size of the target screen and sizing rules are enforced. The adjusted size of the target screen is based on the screen's resolution and the following layout properties:

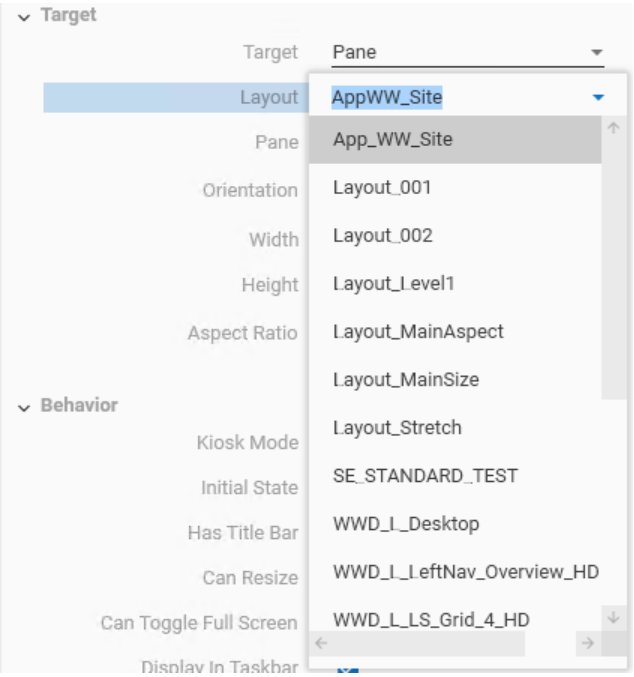
- If the layout's **Initial State** property is FullScreen, the size is not changed
- If the layout's **Has Title Bar** property is True, the height is reduced by 33 pixels

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

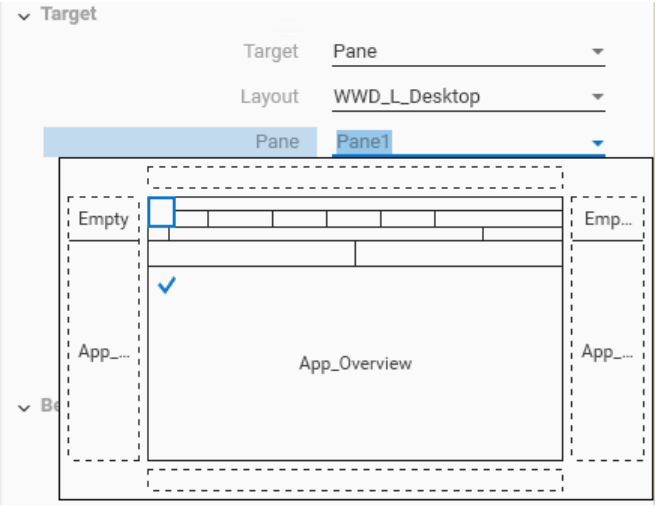
If the screen profile assigned to the **Target** property is deleted, then the **Screen Profile** property shows a red background in the entry field and an an error message. If the screen profile assigned to the **Target** property is renamed, the updated screen profile name appears the next time the layout is opened in Layout Editor.

Select a pane layout property

If the **Target** property is set to **Pane**, the **Layout** property appears in the **Properties** window containing a drop-down list of layouts defined in the Galaxy.



When a layout is selected from the list, the **Pane** property appears. Selecting the arrow at the right of the **Pane** field shows a representation of the selected layout with its panes identified by name.



When a pane is selected the **Pane** property is updated with the selected pane's name. If the pane selected is the default pane then the **Pane** property is assigned the value of <Default>.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When the target pane is changed, the size of the layout being edited will be updated to match the size of the new target pane, and the sizing rules are enforced.

If the target pane is either deleted or renamed, then **Pane** property shows an error indication with red background and corresponding tooltip message – “Could not find pane ‘PaneName’ in target layout ‘LayoutName’.

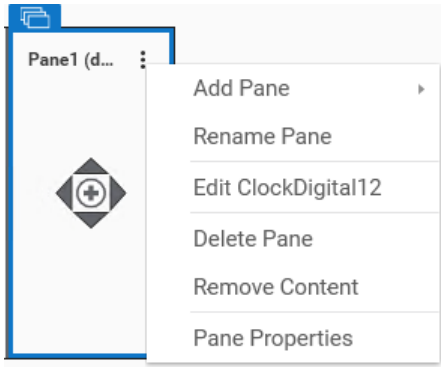
If the selected layout is deleted or renamed, then the **Layout** property shows an error indication with red background and corresponding tooltip message – “Could not find target layout ‘LayoutName’.”

Set pane properties 1

Pane properties define the overall visual characteristics and runtime behavior of the selected pane within a layout. For information about layout properties, see Set layout properties.

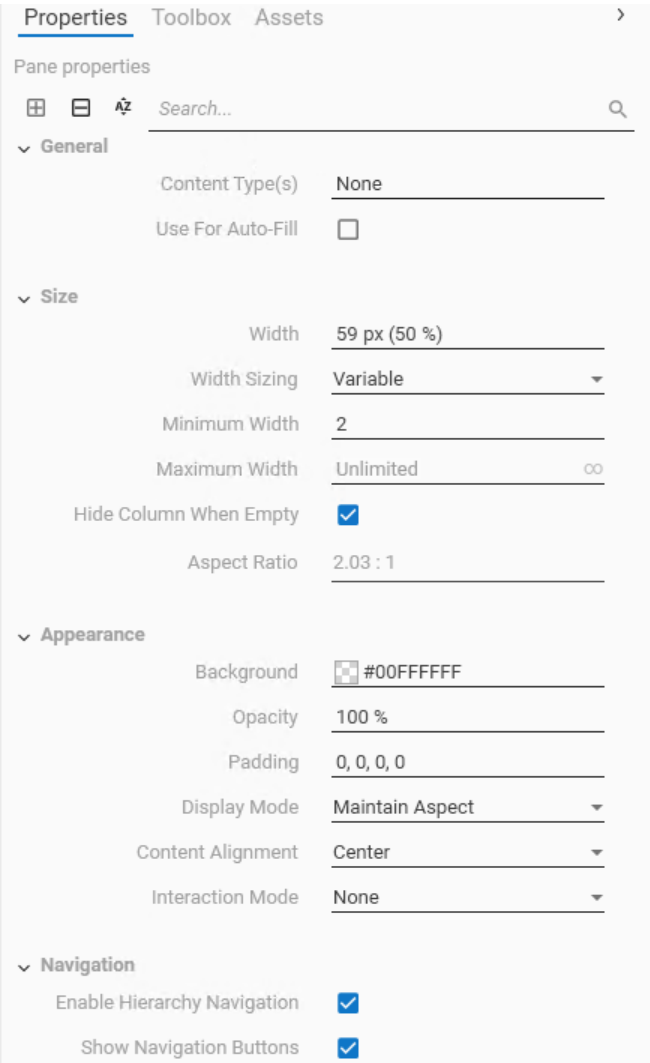
You show the properties of a pane within the Layout Editor by the following methods based on whether the pane contains content or not.

- When the pane is empty, click within the pane to show its properties,
- When the pane contains content, select the pane's **Pane Options** menu and select **Pane Properties**.

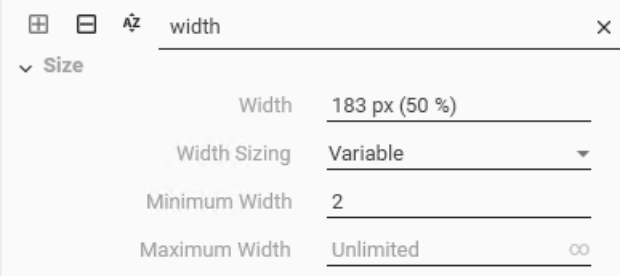


The following image shows the default property values of a pane that extends the full width of a layout . Other property fields appear when properties are assigned specific values or the state of the selected pane. The **Prerequisite Pane Properties** column of the table shown below list any other pane requirements to show a pane property.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



You can search for a specific property by entering its name in the **Search** field. Any properties that contain your search string appear beneath the **Search** field.



You type or select values from drop-down lists to assign values to pane properties. A red background appears when you enter an invalid property value.

The following table lists all pane properties. The properties are organized into the categories shown in the **Properties** window of the Layout Editor. You can view pane properties in alphabetical order by selecting the **Show Alphabetical** icon next to the **Search** field. Short descriptions of the properties appear at the bottom of the **Properties** window for each selected property.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Category	Property	Default Value	Prerequisite Pane Properties	Description
General				
	Content Types	Any		Types of content intended to be displayed in a pane. Select the arrowhead at the right of the field to show a list of supported content types that can be selected. Any is the default content type. See Set pane properties: content types.
	Use For Auto-Fill	False		When True, pane content is filled automatically by Auto-Fill mode. For more information about Auto-Fill, see About auto-fill navigation.
Size				
	Width	N/A	Pane Width < Layout Width	Width of the pane in pixels and percentage of total layout width.
	Width Sizing	Variable	Pane Width < Layout Width	Specifies if pane width can be resized during runtime or remains at a fixed width.
	Minimum Width	2	Pane Width < Layout Width	Minimum width the pane can be resized to at runtime.
	Maximum Width	Unlimited	Pane Width < Layout Width	Maximum width the pane can be resized to at runtime.
	Hide Column When Empty	True	Pane Height < Layout Height	When True, a layout column is hidden if all panes within the column do not show content during runtime. See Set pane properties: hide when empty.
	Height	N/A	Pane Height < Layout.Height	The Height of the selected pane in pixels.
	Height Sizing	Variable	Pane Height < Layout.Height	Specifies if pane height can be resized during runtime or remains at a fixed height.
	Minimum Height	2	Pane Height < Layout.Height	Minimum height the pane can be resized at runtime.
	Maximum Height	Unlimited	Pane Height < Layout.Height	The maximum height of a pane when resized at runtime.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	Hide Row When Empty	True	Pane Width < Layout.Width	When True, a layout row is hidden if all panes within the row do not show content during runtime. See Set pane properties: hide when empty.
	Aspect Ratio	None		Calculated aspect ratio of the width to height dimensions of a pane.
Behavior				
	Can User Close	True	Presentation Mode = Tabbed	When True, users can close pane tabs at runtime.
	Maximum Items	20		The maximum number of items that can be open in the selected pane while the ViewApp is running.
Appearance				
	Background	#00FFFFFF		Background color of a pane. Select the arrow at the right of the field to show a color picker control to select a background color.
	Opacity	100%		Opacity percentage applied to a pane background and all content.
	Padding	0, 0, 0, 0		Number of pixels of blank space between a pane border and content within a pane.
	Display Mode	Maintain Aspect		Method used to change the size of content when a pane is resized during runtime. See Set pane properties: display mode.
	Show Zoom Control	True	InteractionMode = Pan Only or Pan and Zoom	When True, controls are shown during runtime for the user to pan or zoom a pane.
	Horizontal Scroll Bar Visibility	Auto	InteractionMode = Pan Only or Pan and Zoom	Specifies how the horizontal scrollbar is displayed when pane content exceeds the width of the pane.
	Vertical Scroll Bar Visibility	Auto	InteractionMode = Pan Only or Pan and Zoom	Specifies how the vertical scrollbar is displayed when pane content exceeds the height of the pane.
	Content Alignment	Center	DisplayMode = MaintainSize or MaintainAspect	Horizontal and vertical alignment of content within a pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	Interaction Mode	None		Specifies if the user can pan or zoom a pane during runtime to show content that exceeds the size of a pane.
Navigation				
	Enable Hierarchy Navigation	False		When set to True, gestures or buttons can be used to navigate with natural Navigation.
	Show Navigation Buttons	True	Enable Hierarchy Navigation = True	When set to True, the pane shows buttons for Natural Navigation.
	Enable Content Navigation	True	Pane Presentation Style = Multiple	When set to True, the pane supports content navigation using gestures or buttons.

Set pane properties 2

You set the properties of a pane by entering values in the data entry fields of the right window of the Layout Editor. If you enter an invalid property value, the background of the entry field turns red.

To set pane properties

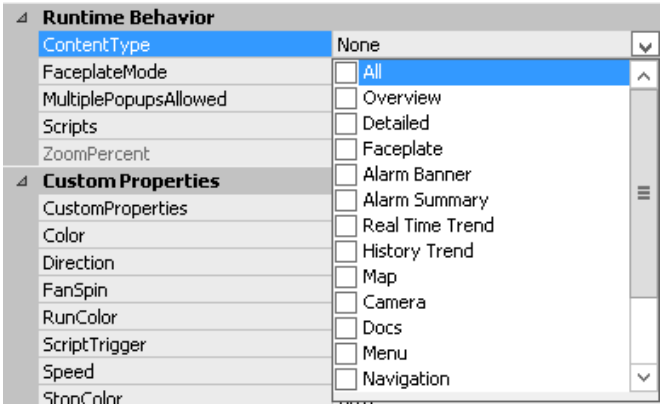
1. Open a layout from the **Visualization folder**.
2. Select the pane whose properties you want to change.
3. Select the **Properties** grid in the right column of the Layout Editor to show the pane's current property values.

   Not all pane properties may appear after selecting the **Properties** grid. Some properties only appear after other properties are set to specific values.
4. Accept the default value or enter new values in the data entry fields.
5. Select **Save** to save your changes.

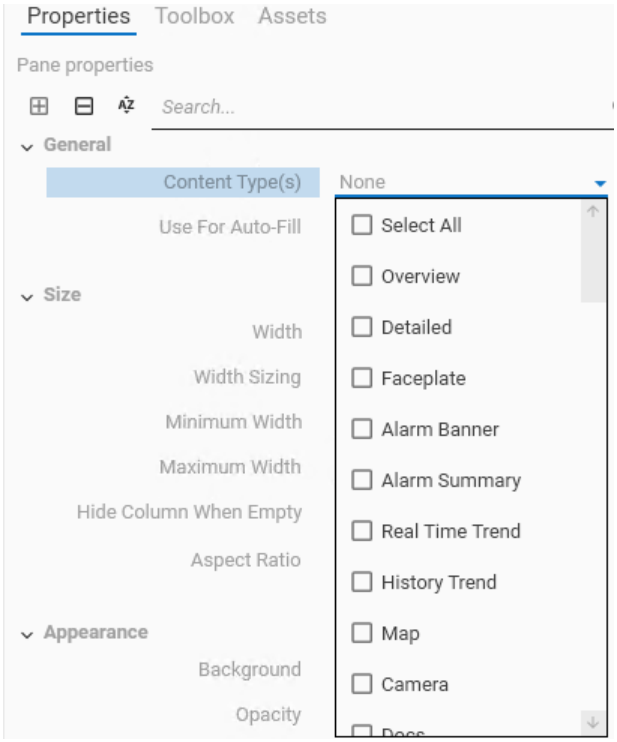
Set pane properties: content types

A content type is a classification method to determine the types of content that can be shown in a layout pane during runtime. The Industrial Graphic Editor provides the **ContentType** property to define the content type of a graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



A corresponding layout pane property defines the type of content that can be added to a pane during runtime. When there is a match between graphic and pane content types, the content can be added to the pane during runtime and focus set to it.



You can select multiple content types as values for the **Content Types** pane property.

Set pane properties: display mode

The **Display Mode** property can be assigned three values that determine how content appears during runtime when the size of the containing pane changes size.

Display Mode Values	Description

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

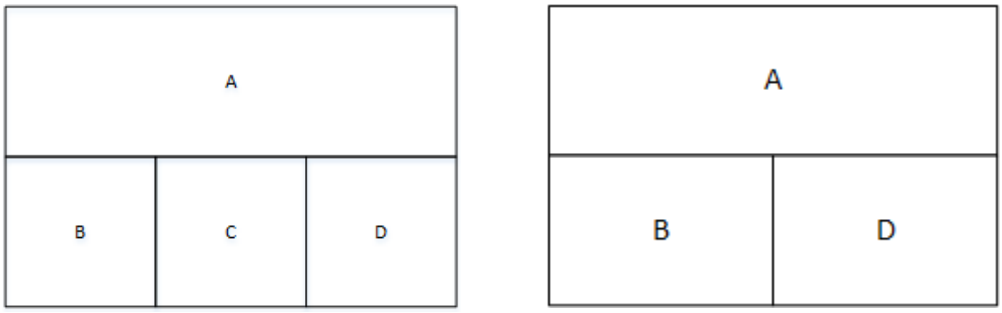
Maintain Aspect	The width and height ratio of content is maintained when a pane is resized larger or smaller. In the case when content exceeds the pane's width or height, the content is aligned within the pane based on on the assigned **Alignment** and **Padding** property values. The content is truncated in the dimension that is larger than the pane's width or height. In the case when the entire content fits within a pane's dimensions, the content is positioned within the pane based on the assigned **Alignment** and **Padding** property values.
Maintain Size	The dimensions of a piece of content remain constant when a pane is resized. When the content is smaller than the pane, the content is positioned within the pane based on the assigned **Alignment** and **Padding** property values. In the case when the size of a piece of content exceeds the dimensions of a pane, the content is centered in the pane and truncated in any dimension that exceeds the pane's width or height.
Stretch	The width and height of a piece of content adjust to fit the dimensions of a pane. In the case of a pane becoming smaller, the **Alignment** property value is ignored.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

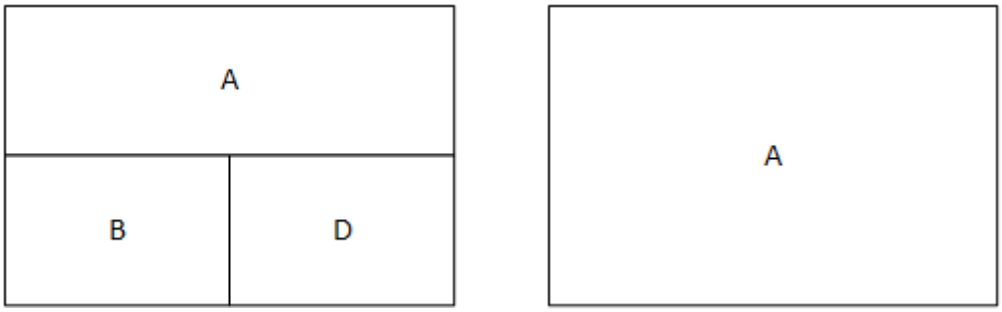
Set pane properties: hide when empty

If the **Hide Row When Empty** and **Hide Column When Empty** properties are set to True, a pane can be hidden during runtime when it is without content or the content is invalid.

In the following example, pane C has been configured with the **Hide column When Empty** property set to True, so it is not shown when it does not contain any content.



An entire row or column of panes can be hidden by configuring the appropriate property. In this example, if pane C is also configured with the **Hide Row When Empty** property set to True, this property propagates to panes B and D. When all three panes (B, C,and D) are empty, the entire row of panes is hidden in the layout.



Set pane properties: auto-fill

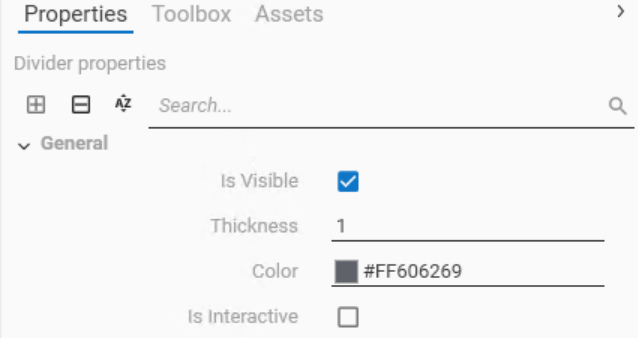
The **Use for Auto-Fill** pane property determines if each navigation item can control whether the selected pane of a layout can be auto-filled or not. In Auto-Fill mode, the content shown in a layout's panes can come from different navigation items in the ViewApp's navigation model. Using hierarchy navigation in any pane will use the specific navigation item for the swiping behavior which placed that original content in the pane.

Auto-Fill has a set of options that control how content is automatically placed in panes without the user having to individually configure the content shown in each pane. For more information about Auto-Fill options, see About auto-fill navigation

Set layout divider properties

If you select a pane divider in a layout, divider properties that can be assigned values appear in the **Properties** area.

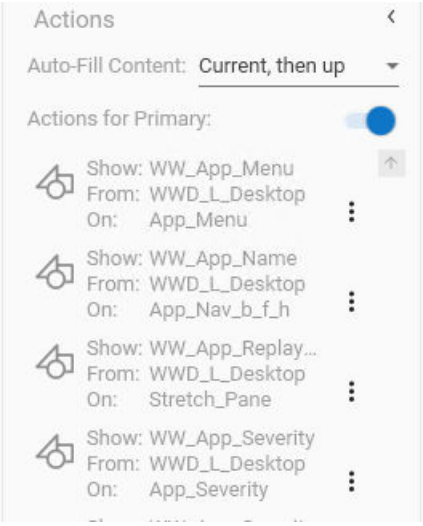
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Property	Default Value	Prerequisite Pane Properties	Description
Is Visible	True	Divider selected	When True, the pane divider is visible while the ViewApp is running
Thickness	1	Divider selected and Is Visible = True	Thickness of a pane divider in pixels.
Color	#FF606269	Divider selected and Is Visible = True	Color of a pane divider. Select the arrow at the right of the field to show a divider color picker.
Is Interactive	False	Divider selected and Is Visible = True	When True, the user can move the divider bar to resize panes during runtime.

Add content to panes

You can drag and drop content to panes. Assigning content to a pane adds a Show Content action to the layout's navigation Action List, which coordinates the automatic filling of panes during runtime.



Content from the Toolbox or Assets tabs can be dragged into a pane of the layout. Content can include:

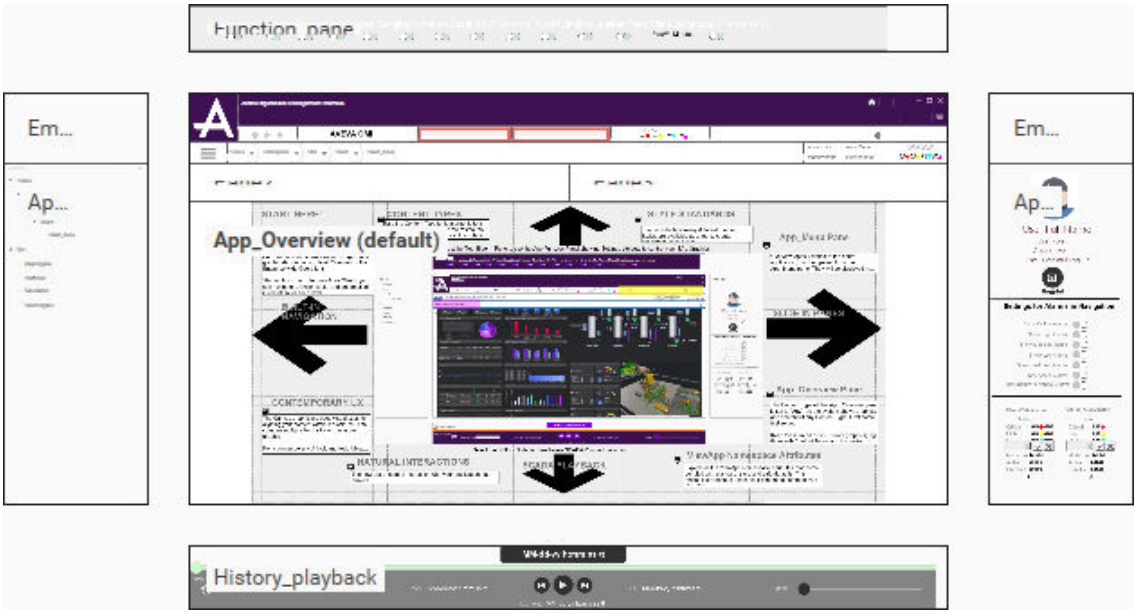
- Graphic

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Layout
- Control
- App Instance

While content is dragged over a layout, the background color of the target pane changes to indicate the pane will receive the content when it is dropped. The shape of a mouse cursor also changes to indicate the content can be dropped on the pane. After content has been dropped, a thumbnail appears within the pane representing its content.

Note: Using a touch device, you can add content to a pane by selecting and holding the content to move it over the target pane.

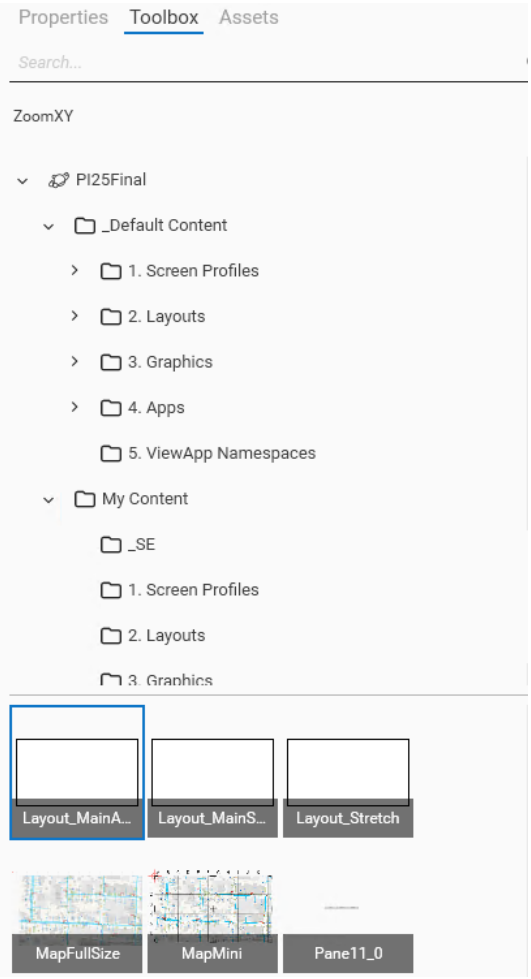


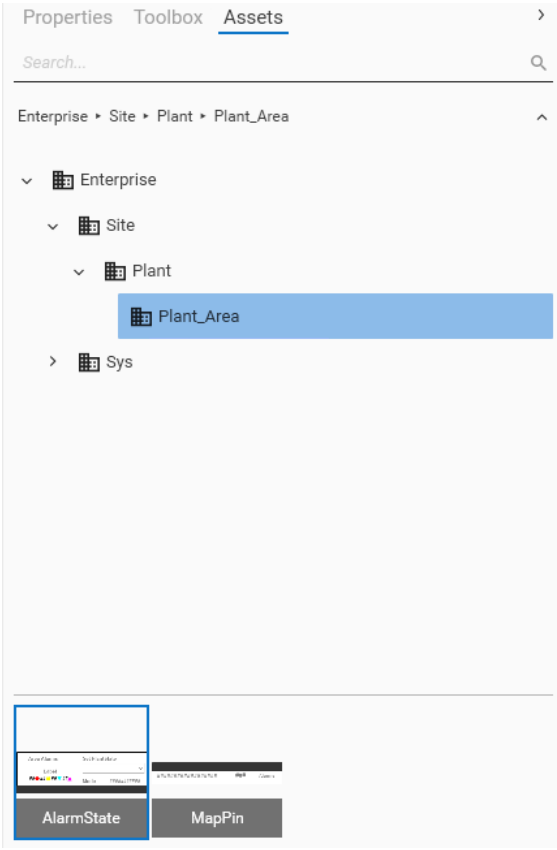
Browse for content to add to layout

You can browse the **Assets** and **Toolbox** panes to find content that can be added to a layout. Each item shown in the **Assets** view represents an Area or an Asset as configured in the System Platform IDE. Each item shown in the **Toolbox** view represents either a folder, a graphic, or a control library. Selecting an item from a browse tree selects the item and displays any content belonging to that item in the content area beneath the browse list.

Placing your mouse cursor directly over a Toolbox or Assets content thumbnail shows a horizontal bar with the folder path to the selected content.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





If a listed item contains sub-items, an expand arrow appears to the left of the item. Clicking or touching the expand arrow retrieves and displays the sub-items. The current selection does not change until an item is selected. If the number of items exceeds the amount of vertical space in the content area of the Layout Editor, a scrollbar appears to scroll the list

If there is insufficient horizontal space to display an item name, a horizontal scrollbar appears at the bottom of the tree to enable you to scroll the content into view.

Search for content to add to layout

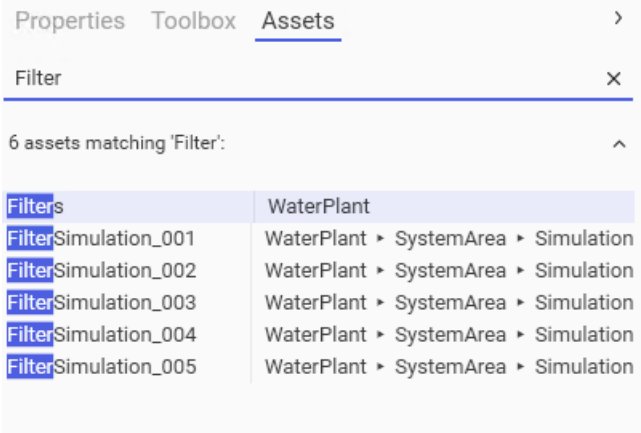
You can search within the currently displayed toolbox folders or assets by entering a term in the **Search** field. Search terms are case-insensitive and locate content or assets regardless of capitalization.

While you are entering a search string, an X appears in the **Search** field to indicate the string can be deleted. If you click the X or press the Esc key, the search string is cleared and no search is conducted.

Asset Searches

An Assets search is applied across the entire Galaxy. The search results show the first batch of Areas and Assets containing the search term in a multi-column list displayed in ascending alphabetical order. A message beneath the **Search** field indicates the number of items found that match the search string.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



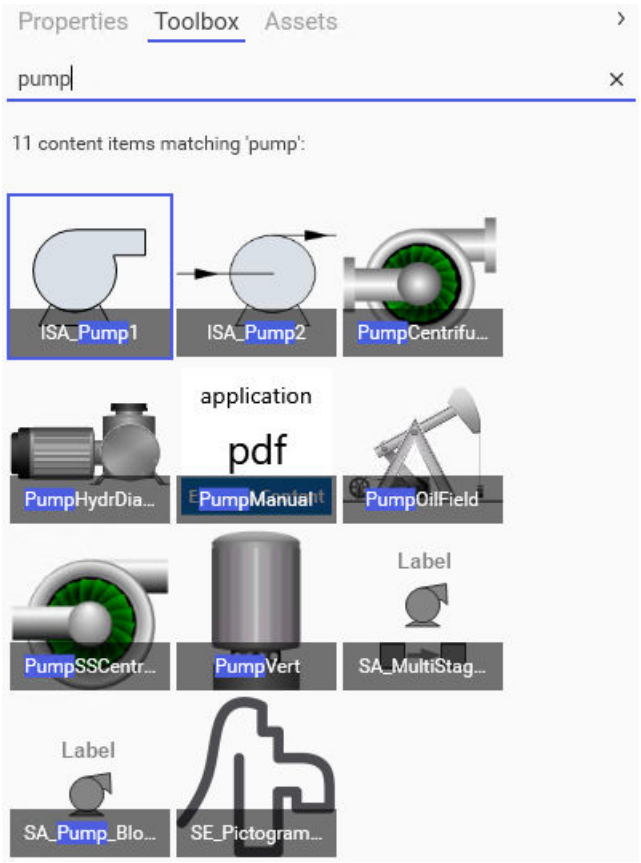
The search results show the search term highlighted within the names of the matching items. You can select an item from the search results list. Selecting an item updates the content view with all content that is owned or linked to the asset you selected.

Placing the cursor over a content thumbnail or tapping it shows the folder in the Visualization folder where the item is located.

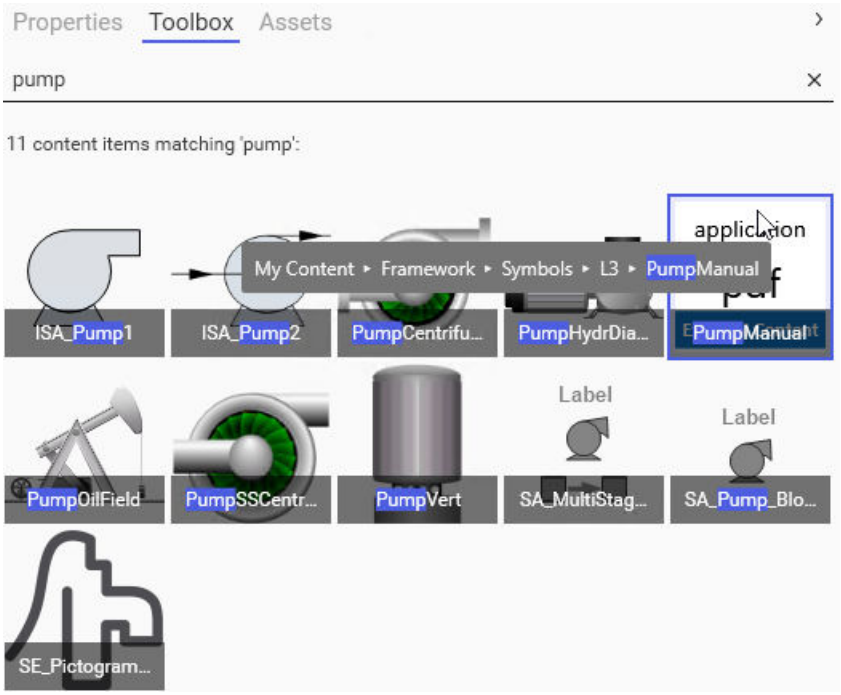
Toolbox Searches

A Toolbox search is applied across the entire Visualization folder. The search results show thumbnails of content that match the search term in the content area by ascending, alphabetical order. A message beneath the **Search** field indicates the number of items found in the search that match the search string.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Placing the cursor over a thumbnail in the search results list or tapping it shows the folder in the Visualization folder where the item is located.



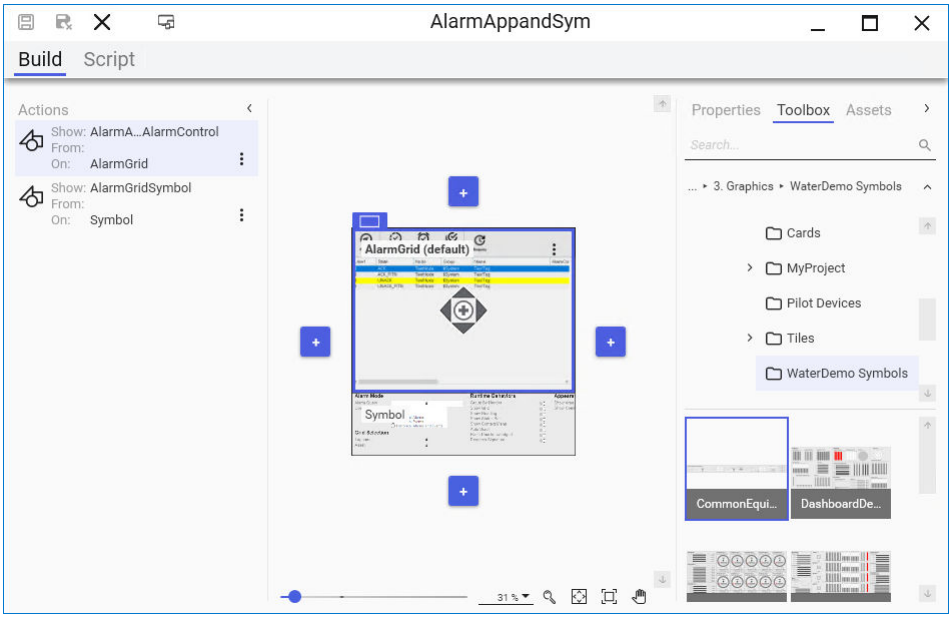
Search results items in the content area can be selected and dropped in a layout pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About adding content to an empty pane

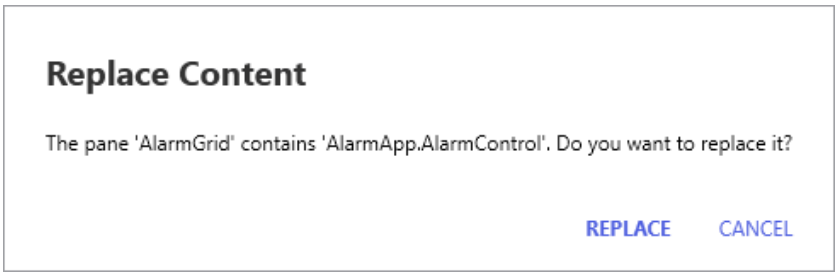
If you add content into an empty pane, then,

- The **Content** property is set to the name of the content added to the pane
- The **Pane** property is set to the name of the pane hosting the content.
- The **Actions** list updates to show an entry containing the name of the content and the name of the pane hosting the content
- A preview of the content appears in the pane



Note: If a layout is added to a pane within the same layout being edited, the cursor changes appearance to indicate the layout cannot be added.

If the target pane being receiving content is a single content pane and already contains content, your see a dialog box asking whether the current content should be replaced or not.



Select **Replace** to update the **Content** property. The thumbnail of the new content appears in the pane.

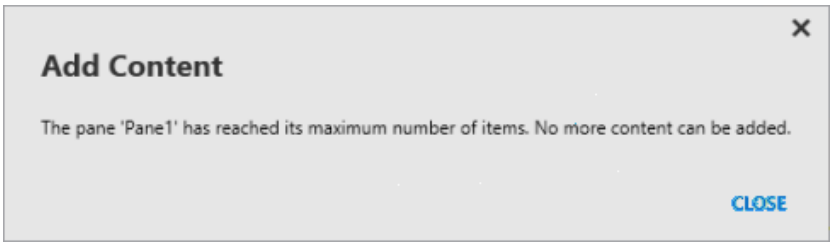
Select **Cancel** to cancel the add content process. The existing content remains in the pane.

Add content to multi-content panes

If a pane is defined as a multi-content pane, dropping a second content item adds the content and the pane is selected. The **Pane** property updates to show the name of the most recent content added to the pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

If you attempt to drop a content item that will exceed the maximum limit set for the pane, the add content action will fail. A dialog box appears notifying you that the maximum number of content items has been reached and no more content can be added to the pane.



Move content to another pane

You can move content from one pane to another within the layout. The **Pane** property includes a thumbnail representation of the layout where you can select the target pane to move the content into.

The behavior of the Layout Editor is based on the presentation styles of the source and target panes and the number of content items currently in these panes.

- If content is moved from a pane specified with a Single presentation style to another Single presentation style pane, the source pane will become empty and the target pane will contain the moved content. New content can be added to the original source pane after its original source content has been moved.
- If content is moved from a pane specified with a Single presentation style to a multi-content presentation style pane, the source pane will become empty and the target pane will contain the moved content unless the maximum content limit has been exceeded.
- If content will be moved from a multi-content presentation style pane to another pane, the **Actions** list shows the default content to be moved with a highlighted background. If you want to move other content from the multi-content pane, select the item from the **Actions** list before selecting the target pane.

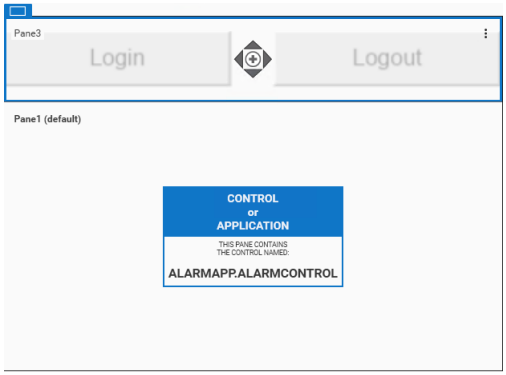
To move content to another pane within a layout

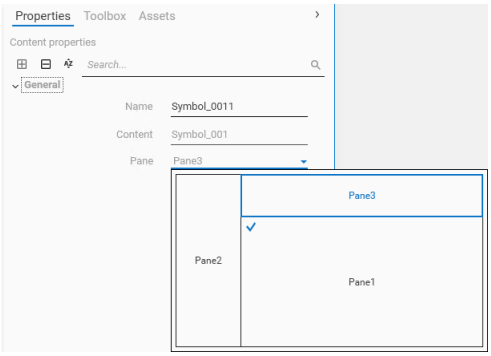
1. Select the pane containing content that you want to move to another pane.

   If you select a multi-content pane, the top most content item shown in the **Actions** list is the selected default content to be moved.

2. Select the **Properties** grid to show the selected pane's properties.
3. Select the arrow at the right of the **Pane** property field to show a thumbnail of the layout.

   The layout thumbnail identifies the selected source pane with a blue border and text.





© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

If the source pane is a multi-content pane, the **Actions** list shows the pane's content as separate items.

4. If you are moving content from a multi-content pane, select a content item from the **Actions** list to be moved.
5. Select another pane in the thumbnail to move the content from the selected pane.

   The **Actions** list updates to show the contents of each pane within the layout.

About the responsive layout mode

Use the responsive layout mode if the layout you are designing is intended to be used in one or more ViewApps that will be viewed by a mix of devices, such as smart phones, tablets, and monitors. If the layout is only for use in a single ViewApp that will be viewed on a monitor, then using the fixed layout mode is appropriate.

Four different screen resolutions (break-points) are defined for a responsive layout, as listed below. Each screen resolution represents a break-point, which represents the screen width at which a layout designed for that particular screen width takes effect. The layout design presented to the user is determined by the width, in pixels, of the viewing device. Screen height does not play any role in determining screen size for the responsive layout. Default layout settings are configured at the smallest screen size (mobile), and propagate to the larger screen sizes, unless overridden. See About responsive layout propagation for more information.

Note: The dimensions (width x height) shown for each screen size within the layout editor are nominal dimensions.

- **Mobile (extra small layout):** This layout is used when the viewing screen is 767 pixels or less. See Display a web ViewApp on a mobile device for information about ViewApp appearance at runtime.

  The parameters set for the Mobile layout are used for each of the larger screen sizes, unless an override is set for larger screen size. An override set for Tablet, for example, would also apply to the Laptop and Desktop screen sizes, unless also overridden at one of those larger screen sizes.
- **Tablet (small layout):** This layout is used if the width of the viewing screen is between 768 and 991 pixels.
- **Laptop (medium layout):** This layout is used if the width of the viewing screen is between 992 and 1199 pixels.
- **Desktop (large layout):** This layout is used if the width of the viewing screen is 1200 pixels or more.

Note: If you change the orientation of a device, the height of the screen becomes its width. For example, if you are viewing a layout on a mobile device and turn it sideways, the screen resolution becomes 812 x 375 (using the nominal dimensions in the Layout Editor). Therefore, the **Laptop** layout would be used instead, since it is used for screen widths between 768 and 991 pixels. You can demonstrate this by configuring a pane in a Mobile layout, and then select the **Rotate Screen** icon. The layout automatically switches from the Mobile to the Laptop layout. You can demonstrate the same effect by dragging the right side of the layout to change its width.

You can change the number of columns used for a pane at each of the defined screen resolutions. By default, new panes span the full width of the layout (12 columns). To create side-by-side panes, set the number of columns that each pane uses so the total number of columns for all the panes is 12 or less. If you have two 12-column panes configured for mobile devices, you may want to have the panes side-by-side when viewed in larger devices. In this case, you could set the width of each pane to 6 columns.

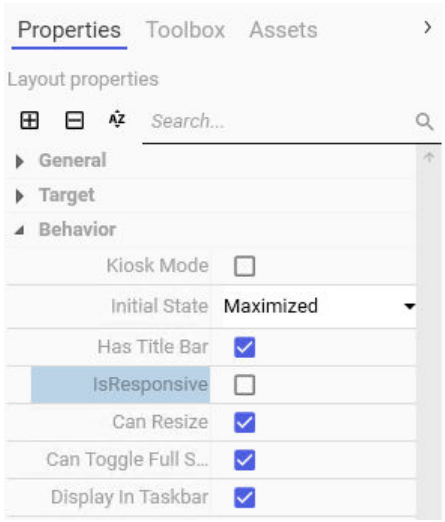
Build a responsive layout

To build responsive layouts, you first toggle the Layout Editor to responsive mode by doing one of the following:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

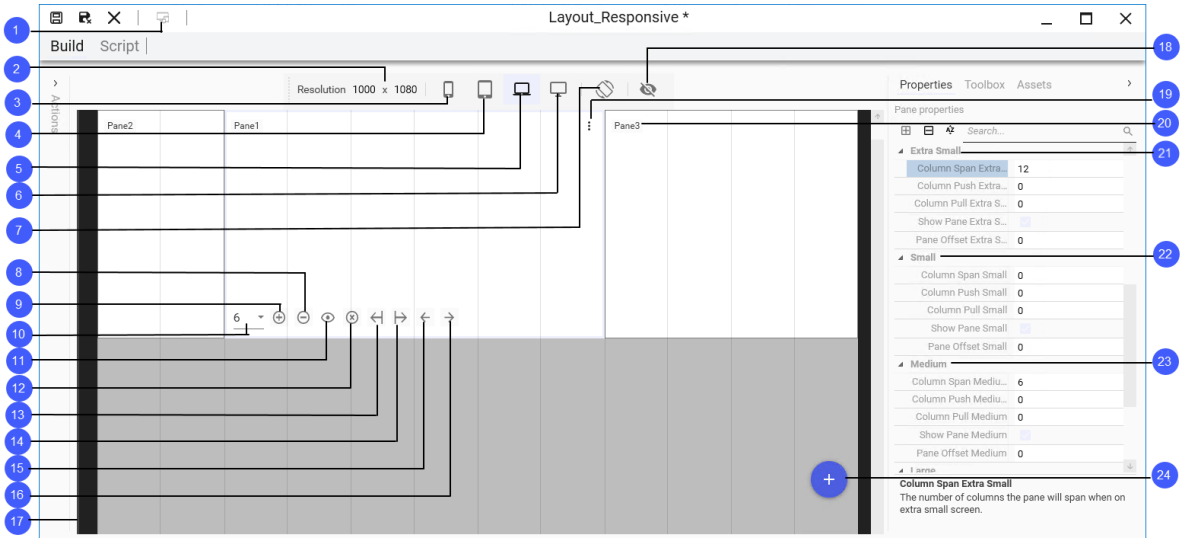
- Select the Fixed Layout/Responsive Layout button, located at the top left of the Layout Editor.

- Enable the **IsResponsive** property, located under Layout Properties. To see layout properties, the layout itself must be selected (not a pane).



Note: Anytime you switch between layout modes, all changes to the layout are discarded. There is no undo.

The figure below shows the various parts of the Layout Editor when Responsive mode is selected. For general information about using the Layout Editor, see About the Layout Editor.



1	Fixed Layout/ Responsive Layout Button	Sets or clears the **IsResponsive** layout property. When IsResponsive is set, the build view changes from the default **fixed** layout view to the **responsive** layout view. Note that any configuration changes made in one view are not preserved when you switch between fixed and responsive views, since the two views cannot be reconciled. If you switch views after configuring the layout, all unsaved changes are lost.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2	Current screen resolution	Shows the layout height and width, in pixels, of the layout as it is currently displayed in the editor. Entering a height or width value changes the displayed layout to match the new dimension. If a width value is entered, the applicable screen size icon (mobile, tablet, etc.) that corresponds to the value is automatically highlighted, and the displayed layout changes responsively to match the configuration for that screen size.
3	Mobile icon	Selecting the **Mobile** icon changes the layout to match the settings entered for Extra Small layouts. Mobile settings apply to screens 767 pixels in width or less.
4	Tablet icon	Selecting the **Tablet** icon changes the layout to match the settings entered for Small layouts. Tablet settings apply to screens widths between 768 pixels and 991 pixels.
5	Laptop icon	Selecting the **Laptop** icon changes the layout to match the settings entered for Medium layouts. Laptop settings apply to screens widths between 992 pixels and 1199 pixels.
6	Desktop icon	Selecting the **Desktop** icon changes the layout to match the settings entered for Large layouts. Desktop settings apply to screens that are 1200 pixels in width or more.
7	Rotate layout	Changes the height of the currently-displayed layout to its width (and width to height). If this change in width exceeds the threshold value of the current layout size, the layout automatically changes to match the parameters of the new layout size. For example, rotating the layout when Mobile is selected changes its width to 812 pixels (the height setting), and thus the layout settings for Tablet devices take effect.
8	Increase Column Span	Each click increments the column span of the pane by one, up to the maximum of 12 columns.
9	Decrease Column Span	Each click decrements the column span of the pane by one, down to the minimum of 1 column.
10	Column Span	.Sets the number of columns for the selected pane. The default is 12 (100% of layout width). You can type a number from 1 to 12, or click the arrow to select from a list.
11	Hide Pane	Sets or clears the **Show Pane** property for the currently-selected screen size only. Unlike other settings, the **Hide Pane** setting does not propagate to larger screen sizes. Hidden panes are not shown runtime, and are hidden at design-time unless **Show Hidden panes (18)** is selected. The default setting of Show Pane is true for each screen size. The ShowContent() method does not work with panes that are hidden.
12	Delete Pane	Deletes the pane from the layout. The selected pane is deleted from the entire layout, not just the currently-selected screen size. If there is only one pane in the layout, the pane cannot be deleted.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

13	Insert Pane before selected	Adds a new pane to the layout. The new pane is inserted before the currently-selected pane.
14	Insert Pane after selected	Adds a new pane to the layout. The new pane is inserted after the currently-selected pane.
15	Move Pane Before	Moves the the pane in the layout to the position immediately before its current position. If the pane is the first in the layout, the **Move Pane Before** button is disabled. The new position for the pane applies to all screen sizes.
16	Move Pane After	Moves the the pane in the layout to the position immediately after its current position. If the pane is the last in the layout, the **Move Pane After** button is disabled. The new position for the pane applies to all screen sizes.
17	Resize bar (layout border)	Select and drag the **resize bar** (thick black border) from either side of the layout to increase or decrease currently-displayed layout width. To change the height of the layout, select and drag the resize bar from the bottom. To change width and height simultaneously, select and drag the **resize handle** (not shown) at the bottom-right corner of the resize bar.
18	Show/Hide hidden panes	Shows or hides any hidden panes for the currently-selected screen size only. Hidden panes for other screen sizes are not affected. If hidden panes are shown, this applies to design-time only, not runtime.
19	Pane Option menu	Opens the context pane option menu. The options are: • Rename Pane • Set as Default Pane • Delete Pane If the pane contains content, additional options appear in the menu: • Edit <Content Name> • Remove Content • Content Properties (toggles to **Pane Properties** when Content Properties are shown)
20	Pane name	This is an editable text field. Rename a pane by clicking on the name and entering a new one. You can also use the **Rename Pane** option in the **Pane Option menu**.
21	Extra small screen size properties	The default pane properties that are set for the screen size. These propagate to each of the larger screen sizes, unless overridden. Similarly, overrides serve as defaults for larger screen sizes, unless subsequently overridden at a larger screen sizes.
22	Small screen size properties	The set of pane property overrides that are set for the small (Tablet) screen size.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

23	Medium screen size properties	The set of pane property overrides that are set for the medium (Laptop) screen size.
24	Add Pane Icon	Adds a new pane to the layout. The new pane is inserted after the last pane in the layout.

About responsive layout methods

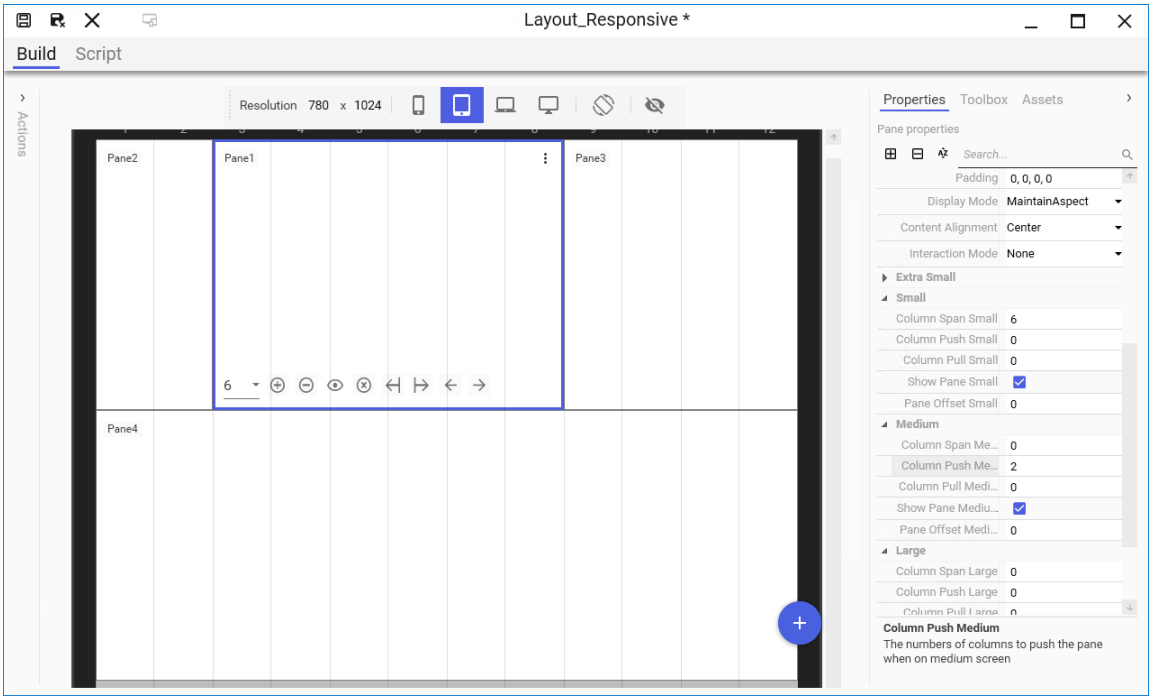
There are several ways to configure a responsive layout. These are:

- **Mostly Fluid:** this is the most basic and the simplest type of responsive layout to configure. The only layout property that requires configuration in a mostly fluid layout is the **Column Span** property, which can be modified to define a different width for a pane at each screen resolution. Even though responsive layouts are easy to configure using only the column span parameters, the Mostly Fluid method provides sufficient capability for most applications.
- **Column Drop:** in addition to setting different widths with the Column Span parameter, the Column Drop method makes use of the **Column Push** and **Column Pull** parameters. This allows content to flow differently at different screen sizes. One scenario for using column drop would be an application that has a navigation screen followed by an overview display. When viewed on a large screen, it makes sense for navigation to be shown above the overview display. However, when viewed on a mobile phone, a user would prefer to have the overview shown at the top of the screen, instead of having to scroll down to get to the main content.
- **Layout Shifter:** this is the most complex method for creating responsive layouts. The Layout Shifter method lets you show different content when the ViewApp is opened on different size screens. As with the Column Drop method, Layout Shifter uses the Column Push and Column Pull properties, and also uses the **IsVisible** property to selectively allow a different pane that contains different content to be shown at specific screen resolutions. For our example scenario, we will again consider a ViewApp with a navigation pane. In this case, we not only want to move the navigation pane up for viewing on larger screens, we also want to change its orientation from horizontal to vertical. To effect this, we create two panes: one with a vertical navigation app and the other with a horizontal navigation app. In addition to using Column Push and Column Pull, we will also use the **IsVisible** property so that the pane with applicable content is shown only when it should be.

Column Offset is another pane property that can be used, but this property is primarily used to align a pane as needed at different screen resolutions, rather than for enabling column shifts. For example, you can use Column Offset to center an "orphaned" pane at certain resolutions.

The figure below shows Pane 1 is selected and overrides have been entered for Small (Tablet) and Medium (Laptop) screen sizes. For Small screen sizes, **Column Span** is set to 6. This also becomes the setting for larger screen sizes, unless a new override is set at a larger size. For medium layouts, **Column Push** has been set to 2. This setting will also be in effect for larger screen sizes, unless another override is set. Pane 4 is offset by 2 columns. However, the setting is not shown because the pane is not selected.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

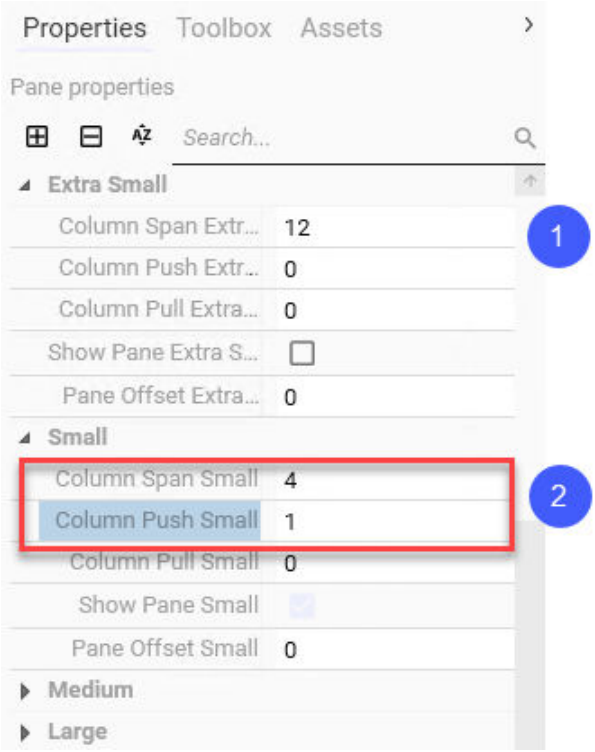


About responsive layout propagation

Responsive layout properties, with the exception the Show Pane property, always propagate from smallest to largest (see below for more information about Show Pane). Therefore, you initially set the properties at the smallest screen size. These become the default parameters for the larger screen sizes. Each layout size has its own configuration parameters, and you can set the number of columns for a pane at each screen size. You can also modify the display behavior of a pane at a particular screen size by using column push/pull and offset parameters. You can also set whether or not a pane will be visible at a particular screen size. Use the following parameters to set pane size and display behavior. These are configurable for each screen size. Always configure the smallest screen first, since the parameters propagate from smallest to largest.

- **Column Span:** Sets the width of the selected pane by number of columns. The minimum is one, and full layout width is 12 (maximum). The default is 12.
- **Column Push:** Moves the selected pane to the right by the number of columns specified. Only the selected pane moves. If there is a pane to the right, it covers the pushed pane. The default is 0.
- **Column Pull:** Moves the selected pane to the left by the number of columns specified. Only the selected pane moves. If there is a pane to the left, it is covered by the pulled pane. The default is 0.
- **Pane Offset:** Similar to column push, but offset is also pushes any panes to the right by the same number of columns. Unlike Column Push, an offset pane is not covered by the adjacent pane to the right. The default is 0.
- **Show Pane:** Determines if the pane is visible or hidden for a particular screen size. The default is visible (Show Pane is enabled). The Show Pane property only applies to the screen size for which it is set; it does not propagate. The ShowContent() method does not work with hidden panes.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



1 Default pane settings, configured at smallest screen size (Extra Small).

2 Override settings for Column Span and Column Push, configurable for each larger screen size (in this case, Small). The overrides for Small apply to Small screens and larger. Settings with 0 (Column Pull and Pane Offset) indicate no override.

- You can enter overrides for one or all of the larger screen sizes. Any overrides set at one screen size become the default settings at larger screen sizes. For example, if you override settings for Laptop (medium) size, those overrides become the default settings for the Desktop (large) size as well.
- When a parameter is set to 0, it indicates that there is no override at that particular level of screen size, and that the default parameters set for a smaller screen have not been overridden.

Responsive Layout: Mostly Fluid

Column Span Property

The Mostly Fluid method for configuring a responsive layout uses the **Column Span** property to set how much of the layout width a pane use at a given screen size. Recall that Column Span can be any integer from 1 to 12, where 12 indicates that the pane will be 100% (12/12 columns) of the layout width. If Column Span is set to 6, the pane will be 50% (6/12 columns) of the layout width.

- The Column Span default for each pane is set by the Extra Small (Mobile) setting. The default is 12 and propagates to all screen sizes, if no overrides have been entered for a larger screen size.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- When the Column Span property is set to 0 for Small (Tablet) or larger screen sizes, it indicates that the parameter is using a setting that was configured for a smaller screen. If an override is set, the override propagates to larger screen sizes, unless a different number is set at a larger size.
- A number other than 0 indicates an override. It is not necessary to set overrides for every screen size. You only need to override a setting if you want to change the **Column Span** that was configured for a smaller screen setting.

To Configure a Mostly Fluid Responsive Layout

1. Open the layout in the **Layout Editor**.
2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/ Responsive** button in the title bar.
3. Add panes for each content item by clicking the **New Pane** button .

   Each new pane is added with default settings for width and height. The default width setting is 12 columns and the default height setting is 300 pixels; the height setting is constant for all screen sizes.
4. To resize a pane, select both the pane and the applicable screen size to which you want to apply overrides.
   - Pane width is set by the **Column Span** property. This is located under the applicable screen size area of the Pane Properties grid (Extra Small, Small, Medium, Large, Extra Large). The default width, which is displayed under the Extra Small (Mobile) screen size category, is 12 columns. This indicates that the pane will take up the full layout width.

     For each of the larger screen size, Column Span is initially 0. This indicates that a previous setting for a smaller screen size will be used. Change the pane width as needed by setting a new Column Span. Screen sizes set to 0 will use the setting from a smaller size.
   - Pane height is set by the **Height** property (located in the **Size** properties area of the Layout Editor), and applies to all screen sizes. The default height is 300 pixels. Minimum height is 1 pixel. Change pane height as needed to accommodate the content that will be added.

   When the combined column span of two or more adjacent panes is 12 or less, the panes will be displayed side-by-side for the screen size(s) that match that configuration. For example, if **Column Span** for three adjacent panes is set to 4 in each pane, the panes will fit in a single row (3 panes x 4 columns = 12).
5. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, panes that have Column Span overrides will move. You can observe the same behavior in the ViewApp editor, once you have added the layout to a ViewApp. To check runtime behavior, use the **Preview** button in the ViewApp Editor, or deploy the ViewApp. As you resize the ViewApp, panes with Column Span overrides will move as the width of the ViewApp changes.

1. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

Mostly fluid responsive layout example

1. Open the layout in the **Layout Editor**.
2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/ Responsive** button in the menu bar.
3. Add panes for each content item by clicking the **New Pane** button . Each new pane is added with default settings for width and height. You can modify these settings later.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

4. Select the **Mobile** (Extra Small) screen size and check the pane properties. Each pane should show its default property settings.
   - Use the default setting as is (12 columns).
5. Select the **Tablet** (Small) screen size.
   - For the **Tablet** screen size, change the **Column Span** property from 0 to 6 for both Pane 1 and Pane 2. These two adjacent panes will now be shown side-by-side. To display additional adjacent panes side-by-side (but in a different row), change their **Column Span** properties as well to numbers that add up to 12 or less.
   - You can leave the **Height** property at the default, or modify it as needed. Panes that are shown side-by-side do not need to have the same height setting, and the height setting remains the same for all screen sizes.

   Note: Property settings propagate from smallest screen size to largest. When a parameters is set to 0, as in this case, it indicates that the parameter set for a smaller screen (in this case, Mobile), will be use the **Mobile** settings. It is not necessary to set overrides for every screen size. You only need to override a setting if you want to change the **Column Span** from the prior (smaller) screen setting.
6. Select the **Laptop** (Medium) screen size. For this size device
   - For the **Laptop** screen size, change the **Column Span** property from 0 to 4 for Pane 1, Pane 2, and Pane 3. These three adjacent panes will now be shown in one row. To display additional adjacent panes side-by-side (but in a different row), change their **Column Span** properties as well to numbers that add up to 12 or less.
7. Select the **Desktop** (Large) screen size.
   - For the **Desktop** screen size, change the **Column Span** property from 0 to 3 for Panes 1, 2, 3 and 4. These four adjacent panes will now be shown in one row. To display additional adjacent panes side-by-side (but in a different row), change their **Column Span** properties as well to numbers that add up to 12 or less.
8. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, panes that have Column Span overrides will move to allow panes that total 12 columns or less to next to each other.
   - Once you have added the layout to a ViewApp, you can observe the same behavior in the ViewApp editor. To check runtime behavior, use the **Preview** button in the ViewApp Editor, or deploy the ViewApp.
   - As you resize the ViewApp, panes with Column Span overrides will move as the width of the ViewApp changes.
9. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

Configure a column drop responsive layout

Column Push/Column Pull Properties

The Column Drop method for configuring a responsive layout uses the **Column Push** and **Column Pull** properties in addition to the **Column Span** property. See Responsive Layout: Mostly Fluid for information about the Column Span property.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- The **Column Push** property moves the selected pane to the right by the number of columns specified.
- The **Column Pull** property moves the selected pane to the left by the number of columns specified.

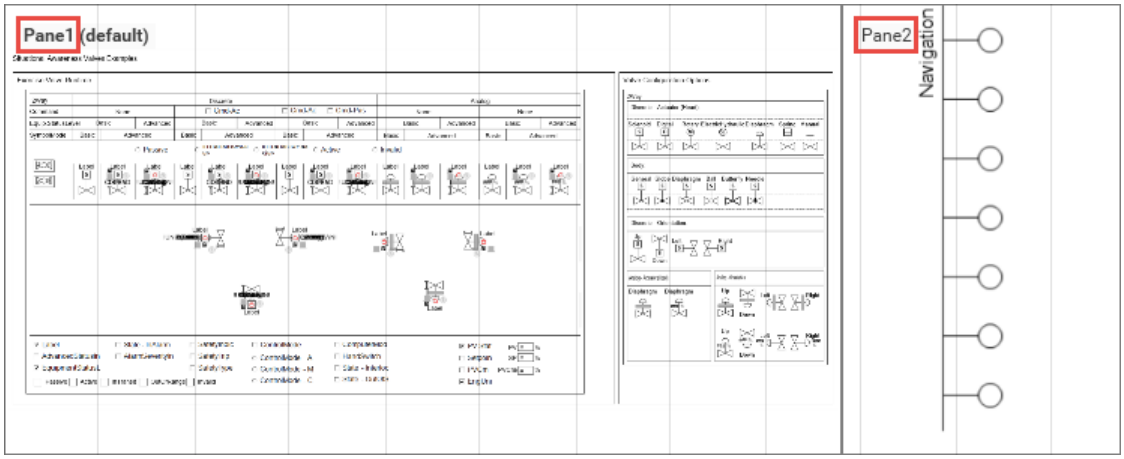
Column Push and Column Pull let you rearrange the order in which panes are presented at different resolutions. For example, when viewed on a mobile device, you probably want to have the main content, such as an overview graphic, shown first and followed less critical content, such as a navigation pane that the user scrolls down to view. However, when these panes are resized for viewing on the desktop, the navigation pane ends up on the right side of the screen instead of to the left.

Use the Column Push and Column Pull properties to transpose the positions of two adjacent panes when using larger screen sizes.

To Configure a Column Drop Responsive Layout

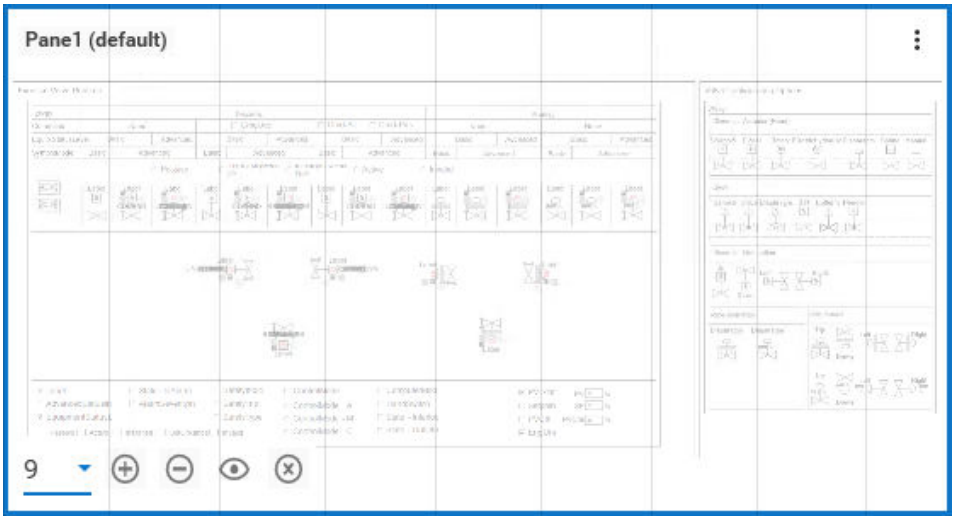
1. Open the layout in the **Layout Editor**.
2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/ Responsive** button in the menu bar.
3. Add panes for each content item, as described in Responsive Layout: Mostly Fluid .
4. Resize panes using the Column Span property, as described in Responsive Layout: Mostly Fluid .
5. To transpose two adjacent panes:

   The follow screenshot shows two adjacent panes that have been resized to fit a single row when viewed on a Desktop (large) screen. The main content of the layout is in Pane1 and navigation is in Pane2. For this procedure, Column Span for Pane1 is set to 9, and Column Span for Pane2 is set to 3.

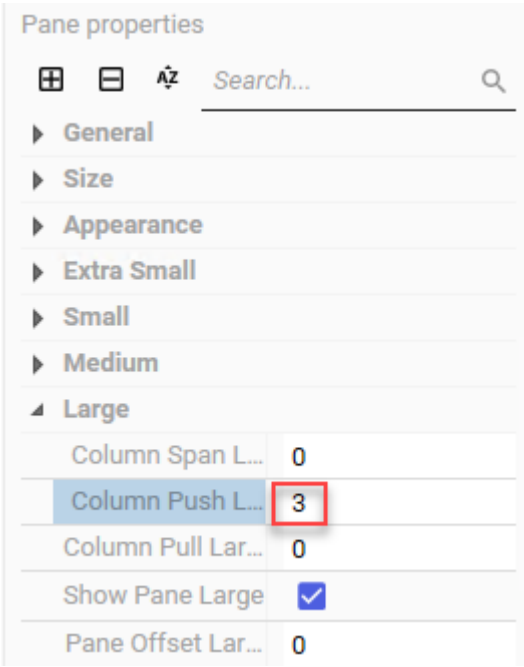


   a. Select the pane you want to move right (Pane1).

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

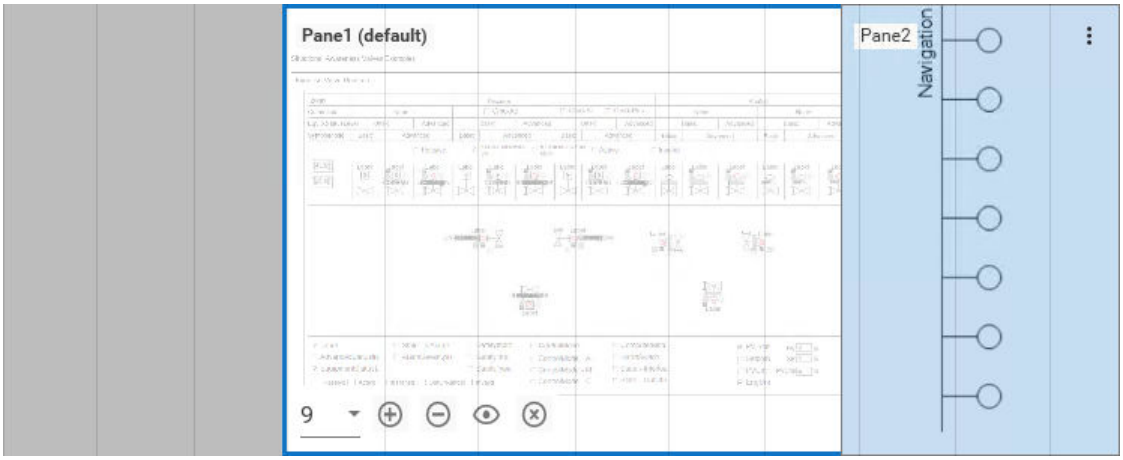


b. Expand the screen size setting (Small, Medium, Large or Extra Large) that you are configuring, and set **Column Push** equal to the **Column Span** (3) of the pane to the right (Pane2).

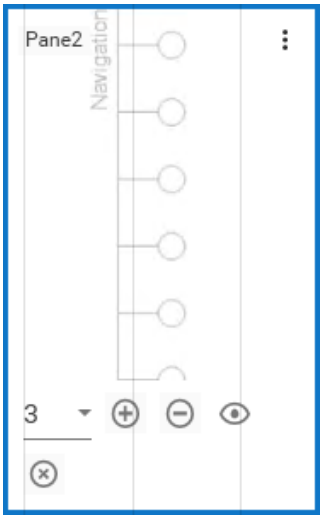


Note that Pane1 moves three columns to the right and slides under Pane2.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



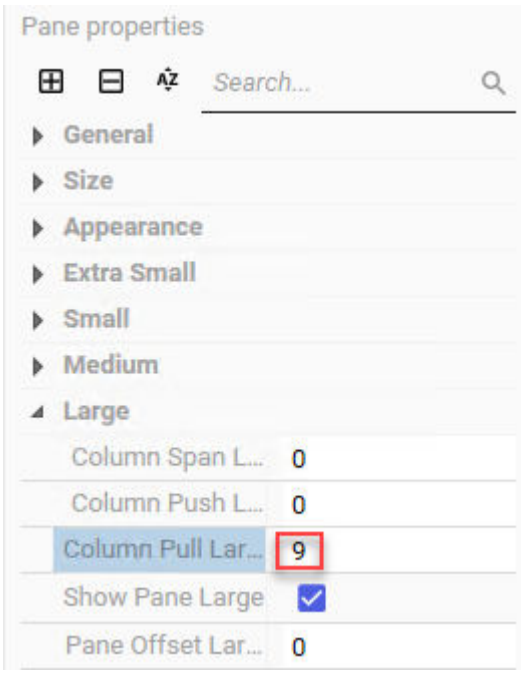
c. Select Pane2.

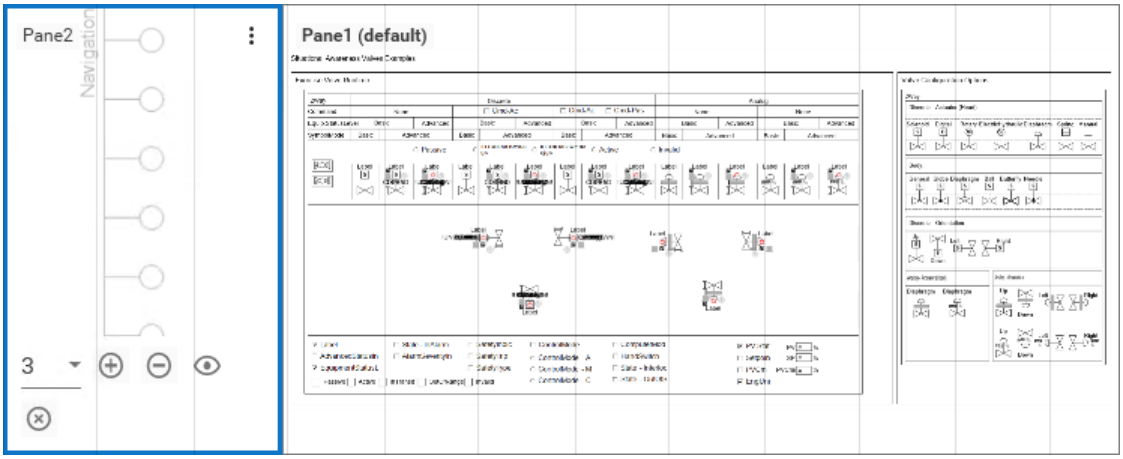


d. Set **Column Pull** equal to the **Column Span** of the pane to the left (Pane1).

Pane 2 now slides to the left side of the layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





6. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, Pane1 and Pane2 will switch positions.
7. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

To check runtime behavior, add the layout to a ViewApp. You can use the **Preview** button in the ViewApp Editor, or deploy the ViewApp. As you resize the ViewApp, the two panes will switch positions as the width of the ViewApp changes.

Configure a layout shifter responsive layout

Show Pane Property

The Layout Shifter method for configuring a responsive layout uses the **Show Pane** property to allow different content to be displayed as a function of screen size. Layout Shifter also uses **Column Push** and **Column Pull** properties, introduce in the Column Drop method. **Column Span**, as the basic property that sets the width of a

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

pane, is used in all responsive layout methods.. See Responsive Layout: Mostly Fluid for information about the Column Span property, and Configure a column drop responsive layout for more information about the Column Push and Pull properties.

The Show Pane property lets you show different content, contained in different panes, at different resolutions. Returning the navigation example presented in Configure a column drop responsive layout , we may want to use vertically-oriented content at some screen resolutions, and horizontally-oriented content at other resolutions. With the Show Pane property, we can selectively show and hide content, depending on the width of the screen the user is using. The Show Pane property lets us conditionalize panes and their content, so they are shown only at specific screen sizes.

Note: The ShowContent() method does not work with hidden panes.

Use the Show Pane property along with other responsive layout properties to show/hide and change positions of selected panes.

For additional information about the responsive web design, including the Layout Shifter method, see "Learn Responsive Design" at web.dev.

To Configure a Layout Shifter Responsive Layout

1. Open the layout in the **Layout Editor**.
2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/ Responsive** button in the menu bar.
3. Add panes for each content item, as described in Responsive Layout: Mostly Fluid . As you add panes, think of the conditional
4. Resize panes using the Column Span property, as described in Responsive Layout: Mostly Fluid .
5. Configure Column Push/Pull if needed for larger screen sizes, as described in Configure a column drop responsive layout .
6. Determine which content is needed for each screen size, and the position in which the content should be displayed. For example:
   - For mobile devices, you want to show several several panes that contain important graphics. You do not need to show a navigation pane, or you want the navigation pane to have its content oriented horizontally (such as in the Breadcrumb navigation app) and below the other content.
   - For desktop displays, you want the navigation pane to be the first pane and to have its content oriented vertically, such as in the NavTree navigation app.
7. For each applicable screen size for the conditional panes, select the pane and then enable or disable the **Show Pane** property.
8. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, panes will be shown or hidden.
9. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

To check runtime behavior, add the layout to a ViewApp. You can use the **Preview** button in the ViewApp Editor, or deploy the ViewApp. As you resize the ViewApp, the conditional panes will be shown or hidden as the width of the ViewApp changes.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure a layout shifter responsive layout example

The layout shifter pattern is the most complex of the responsive layout designs. Unlike a mostly fluid or column drop design, layout shifter actually presents users with different pieces of content when viewing the application at different screen sizes.

Show Content Property

In addition to the Column Span, and Push/Pull properties used for mostly fluid and column drop designs, respectively, the **Show Content** property allows different content to be shown or hidden at specific screen sizes.

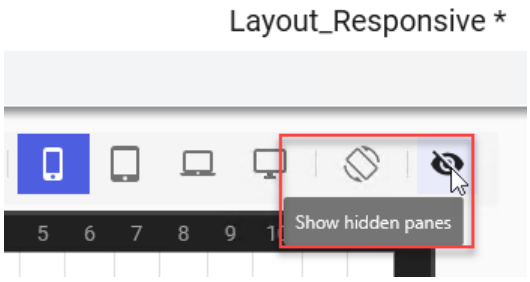
The following example leverages Show Content to show and hide different navigation panes. For smaller screen sizes, we will show the navigation pane arranged horizontally across the screen, while for larger screen sizes, we will show a vertical navigation tree at the left of the screen. In the example, we will:

- Show the navigation pane with content arranged horizontally below the main content for mobile devices.
- Show the navigation pane with content arranged horizontally above the main content for tablets and laptops.
- Show the navigation pane with content arranged vertically to the left of the main content for large and extra large screens.

Note that this is a simplified example that is intended only to illustrate how you can use the Show Content property. Depending on your requirements, you can combine all the elements of mostly fluid, column drop, and layout shifter within a single layout.

1. Open the layout in the **Layout Editor**.
2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/Responsive** button in the menu bar.

   Pane 1 configuration (small device navigation)
3. Select **Pane 1** and set its height to 100 (the pane height property is under the **Size** group of pane properties and has a default value of 300). Retain the default settings for all other pane properties.
4. Click the **Show Hidden Panes** button at the top of the layout to show panes regardless of how the **Show Pane** property is set. Showing hidden panes can help as you configure panes and their content.



5. With **Pane 1** selected, make sure that the **Show Pane Extra Small** and **Show Pane Small** properties are enabled (checked). Disable (uncheck) the following Show Pane properties:
   - **Show Pane Medium:** disabled
   - **Show Pane Large:** disabled
   - **Show Pane Extra Large:** disabled

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: The **Show Pane** property, unlike the rest of pane properties, does not propagate and must be explicitly enabled or disabled for each screen size. The default state of the Show Pane property is enabled.

6. From the **Toolbox** tab, add the **NavBreadcrumbControl** navigation app to Pane 1. The NavBreadcrumbControl is a horizontally-oriented navigation control.

   Pane 2 configuration (large device navigation)

7. Add a pane (Pane 2) by clicking the **New Pane** button .
8. If the **Toolbox** tab is still selected, reselect the **Properties** grid.
9. With **Pane 2** selected, make sure that the **Show Pane Medium** and **Show Pane Large** properties are enabled (checked). Disable (uncheck) the following Show Pane properties:
   - **Show Pane Extra Small:** disabled
   - **Show Pane Small:** disabled
10. Set the **Column Span Medium** property to 2 columns for **Pane 2**. This setting propagates to the larger screen sizes. Since Pane 2 is not shown for mobile (extra small) and tablet (small) devices, **Column Span** for these smaller screen sizes is irrelevant.
11. From the **Toolbox** tab, add the **NavTreeControl** navigation app to Pane 2. The NavTreeControl is a vertically-oriented navigation control.

    Pane 3 configuration (main content)

12. Add a pane (Pane 3) by clicking the **New Pane** button .
13. With **Pane 3** selected, configure the **Column Span Medium** property to 10 columns. This setting propagates to the larger device sizes, but the settings for **Column Span Extra Small** and **Column Span Small** do not change from 12 columns.
14. Add an overview graphic or other graphic to Pane 3. This will function as the main content for the layout.
15. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, the panes containing navigation content (vertical or horizontal) will be hidden or shown as applicable to the screen size.
    - Once you have added the layout to a ViewApp, you can observe the same behavior in the ViewApp editor. To check runtime behavior, use the **Preview** button in the ViewApp Editor, or deploy and launch the ViewApp.
    - As you resize the ViewApp, the Show Pane property determines which pane is shown as the width of the ViewApp changes.
16. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

Responsive layout limitations

Limitations Inherent in a Responsive Layout

The following parameters and properties cannot be changed in a responsive layout:

- The values defined for responsive layout breakpoints cannot be changed.That is, the transition screen widths between layout sizes (extra small, small, medium, etc.) are fixed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- The **AspectRatio** property is not supported for panes within a responsive layout.
- The DisplayMode property will not have any effect if a responsive layout is placed as the content of a pane in another layout.
  - Width: The responsive layout will occupy the full width of the pane in which it is placed.
  - Height: If the height of the responsive layout exceeds the height of the pane in which it is placed, a vertical scroll bar will appear.
- When a responsive layout is placed as the content of a pane in a fixed layout, the width of the fixed pane (NOT the width of the fixed layout) determines which breakpoint (screen size) the responsive layout will use. For example, consider a fixed layout that is 1280 pixels wide. It contains a pane that is 640 pixels wide that holds a responsive layout. In this case, the responsive layout uses the extra small (mobile) breakpoint, since the extra small size allows a layout up to 767 pixels wide. The overall size of the fixed layout is ignored.

InTouch Web Client Limitations

The following limitations apply when you view a ViewApp that uses a responsive layout in the InTouch Web Client:

- The InTouch Web Client does not support placing a responsive layout as the content of a fixed layout pane.
- The InTouch Web Client does not support placing a fixed layout as the content of a responsive layout pane.
- These limitations do NOT apply to the AVEVA OMI ViewApp at runtime.

Build layout scripts

The **Layout Editor** includes a tab for adding and editing scripts for execution at runtime. You can use different triggers, such as opening or closing the layout, to start the execution of a layout script, or setting an interval at which the script executes while the layout is open.

On Show layout scripts are executed before any graphic scripts contained within graphics in the layout, while On Hide layout scripts are executed after the graphic scripts. Other script types do not have a specific order of execution and may not execute in the same sequence each time. You can also add scripts triggered by events in an app or control, such as PropertyChange, MouseDown, etc.

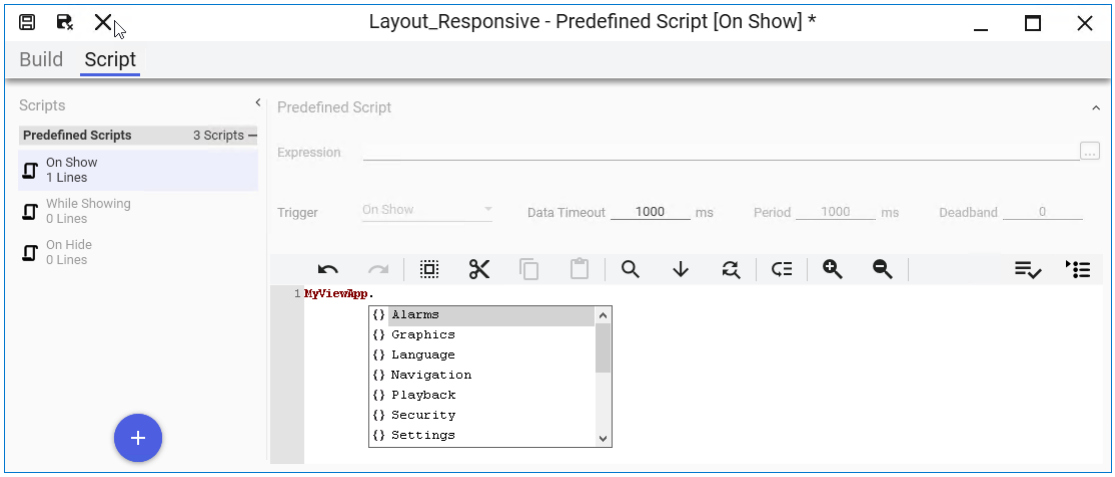
Layout scripts work much like the graphic scripts that can be added to graphics to enable animation of the graphic or its elements. For more information about graphic scripts, see About Symbol Scripts.

The layout script editor includes the following features:

- Autocomplete: as you type, the editor presents a dropdown list of namespaces, attributes, custom properties, exposed app properties, scripting methods, and other items that match the characters that you have entered. The items in dropdown list are filtered by context, so only valid items are listed in the dropdown.

  Note: All public app properties will appear in the autocomplete list, even though not every public app property that appears in the list can be scripted. Properties that are not data-bound cannot be scripted. You can script public properties that contain their backing logic within the app itself. However, some apps which wrap around an underlying control, such as the Alarm App, contain pass-through properties which may require additional qualification. In a pass-through property, the backing logic is contained within the wrapped control. Therefore, to script a pass-through property, you must invoke the control namespace, even though autocomplete, by virtue of the property being public, will allow you to enter the property name with or without the control namespace. See Alarm client control pass-through properties for details.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The Layout Script Editor autocomplete feature works the same way as the Application Server QuickScript Editor and the Graphic Script Editor. For more information about Application Server scripting, see the Application Server Scripting Guide.

- Event handler: add scripts that are triggered by user interaction with an app, such as a mouse click or a menu opening.
- Search function: includes options to find, find next, or replace a specified string.
- Undo/redo.
- Script validation: flags invalid entries in the script.

The script editor contains three separate areas:

- A pane at the left side of the editor lists all scripts within the layout. This list always includes the three built-in scripts (**On Show**, **While Showing**, and **On Hide**), and any named scripts and event-handler scripts that you add. Built-in scripts cannot be deleted or renamed, and are grayed-out until a script is added (regardless of the validity of the script). The number of lines in the script appears under the listed script name. If the script is not configured, the script shows that contains 0 lines. You can collapse and expand the script-list pane by selecting the chevron at the top right of the pane.
- The top pane of the script editor contains an area to enter an expression and script trigger type. Other fields in the top pane are for timeout, period, deadband, and whether to include quality changes. These fields are enabled only if they are applicable to the selected trigger.
- The main portion of the script editor has buttons for undo/redo, cut, duplicate, zoom in/out, and other editor functions. A button at the right of the pane lets you fully expand the editing pane. The cursor position is tracked by row and column numbers displayed at the bottom of the pane.

Types of layout scripts

The **Script** tab shows the scripts that have been configured for the layout:

- **Built-in scripts.** There are three built-in scripts that are automatically added to all layouts. These are listed in the **Script** tab and are always shown, whether or not they have been configured. The three built-in scripts are:
  - On Show
  - While Showing

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

  - On Hide
- **Named scripts**. You can add and configure an unlimited number of named scripts. These are expression-based scripts that can use any of the following triggers:
  - On True
  - On False
  - While True
  - While False
  - Data Change (this can also include a change in data quality)
- **Event-handler scripts.** These can be added and configured for apps and controls in the layout that contain public events. There are numerous types of events that can be scripted, and the events will be different between apps. Some examples of the types of events that can be scripted include:
  - Got Focus
  - Mouse Move
  - Mouse Double Click
  - Drag Over
  - Drop
  - Key Up
  - Context Menu Opening

About built-In layout scripts

Built-in scripts for layouts are similar in both concept and function to the Predefined Scripts for graphics that can be invoked through the **Industrial Graphic Editor**. For more information about the graphic built-in scripts, see Predefined graphic script configuration. Built-in scripts are triggered by the state of the layout itself, and do not use an expression or reference.

- **On Show:** The script runs one time when the layout opens. You can specify a timeout period for the script in milliseconds to stop the script if a referenced attribute cannot be accessed.
- **While Showing:** The script runs at a specified interval (in milliseconds) while the layout is open.
- **On Hide:** The script runs one time when the layout closes. There are no configurable trigger parameters.

You can have only one built-in script of each type, for a maximum of three built-in scripts, per layout. See About Built-in scripts for more information.

About named scripts

You can add named layout scripts that are triggered by an expression or reference. The expression or reference is used to define the specified condition. There is no limit on how many expression-based scripts that you can include in a layout.

- **While True:** The script runs periodically, at a specified interval (in milliseconds), for as long as the expression remains true.
- **While False:** The script runs periodically, at a specified interval (in milliseconds), for as long as the expression remains false.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

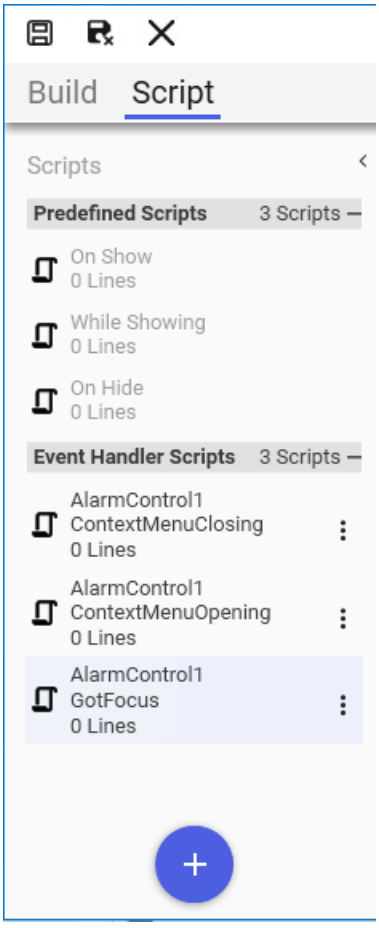
- **On True:** The script runs one time when the expression transitions to true.
- **On False:** The script runs one time when the expression transitions to false.
- **Data Changed:** The script runs whenever data referenced by the expression changes. You can specify a deadband to filter out inconsequential data changes. In addition, you can specify if quality changes will also be used to trigger the script to run.

See Add a named script for more information.

About event-handler scripts

You can script against public events contained in apps and controls. When you add an app or control to the layout that contains public events, you can add event handler scripts from either the Build or Script tab that are triggered by selected events.The script is then triggered when, for example, the app detects a MouseDown or DataChange event.

Event-handler scripts are named for the app or control and the selected event, and cannot be renamed. They appear in the Scripts pane at the left of the **Script** page, and are listed in alphabetical order, after build-in scripts and named scripts. Like the other two categories of scripts, event handler scripts are listed in their own folder. See Layout event handler scripts for more information.



Namespaces for use with layout scripts

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Layout scripts can leverage both local and global namespaces, including the About the MyContent namespace, which is local to each individual layout, and About ViewApp namespaces, which are global across the Galaxy. The MyContent namespace can be referenced by layout scripts and expressions used within that layout. This provides the ability to isolate attributes within an individual layout, as opposed to the ViewApp namespace, which can be referenced by all layouts and ViewApps in the Galaxy.

Event-handler scripts, unlike named and built-in scripts, do not require the use of the MyContent namespace when scripting content properties and methods. To script a property or method, start by entering the name of the control instead of "MyContent." The autocomplete function will allow you to see the properties and methods contained in the control as you type.

About Built-in scripts

You can configure one of each type of built-in scripts for a layout. There is no requirement to configure built-in scripts. Unconfigured scripts are not used at runtime and do not generate errors or warnings.

Built-in scripts are named for their trigger type, and cannot be renamed or deleted. A built-in script is not triggered by an expression, but instead uses the state of the layout to define when the script executes. The built-in scripts are::

- **On Show:** The script runs one time when the layout opens.
- **On Hide:** The script runs one time when the layout closes.
- **While Showing:** The script runs periodically for as long as the layout remains open.

For details about the Script Editor, see Build layout scripts. See also About the MyContent namespace for information about referencing different content contained in the layout.

Note: The three **Built-in Scripts**, each named for a trigger type (**On Show**, **While Showing** and **On Hide**), cannot be deleted from the script list. Only one built-in script per trigger type can be configured for a single layout.

Configure built-in script

To configure a built-in script

1. Open the layout in the Layout Editor and select the **Scripts** tab.
2. Select one of the three built-in scripts from the script list in the left pane of the **Scripts** tab:
   - **On Show:** Use this to configure a script that runs one time when the layout opens.
   - **While Showing:** the script runs periodically while the layout is open.
   - **On Hide:** the script runs one time, when the layout closes.
3. Set any applicable timing parameters for the script.
   - **On Show:** the default **Data Timeout** is 1000 ms. See Tips to ensure proper On Show script execution for more information about configuring the timeout period.
   - **While Showing:** the default **Period** is 1000 ms. This value sets the interval at which the script will run while the layout is showing.
   - **On Hide:** There are no configurable timing parameters for an On Hide script. The script runs only once when the layout closes.
4. Enter your script in the main edit box. Autocomplete helps by displaying valid items, based on the initial characters you enter. The script syntax is the same as the syntax of AutomationObject scripting.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

5. Enter your script in the main edit box. enter.Autocomplete helps by displaying valid items, based on the initial characters you enter. The script syntax is the same as the syntax of AutomationObject scripting.
   - You can validate your script for syntax errors prior to saving by pressing the **Validate Script** button.
   - To clear all data entered for the the script, including the expression, press the **Clear Script** button. There is no undo for this action.
6. To save the script, select the **Save** or **Save and Close** button.
7. The script is validated automatically. If there are warnings or errors generated, a dialog listing the errors is displayed. You have the following options:
   - Select **Yes** to save the script and preserve the errors.
   - Select **No** to discard the changes you made to the script. If you selected **Save and Close**, the Layout editor closes and you will lose any changes you made to the script. If you selected **Save**, the Layout Editor remains open and you can resume editing the script.
   - Select **Cancel** to resume editing the script. The Layout Editor remains open and you can resume editing the script.

Tips to ensure proper On Show script execution

When an **On Show** script includes external references to ApplicationObject attributes, data from these attributes may not be immediately available when the **On Show** script runs. As a result, the script might not work properly.

To mitigate this possibility, you can enter a value in the **Data Timeout** field. The **Data Timeout** period sets a time limit to allow reference data to become available. At the end of the **Data Timeout** interval, or before the timeout expires if the reference data is available sooner, the **On Show** script is executed. Note that data quality is not checked, only that the data is available.

If the **Data Timeout** expires before all reference data is available, the **On Show** script is still executed. However, the script might not work properly.

You can reduce the possibility of linking to invalid data by having the script first test the quality and status of the attribute before using it in the rest of the script.

The default value in the **Data Timeout** field is 1000 ms (one second). The maximum data time-out value is 30,000 ms (30 seconds).

Note the following issues regarding **On Show** scripts and the **Data Timeout** function:

- The **Data Timeout** function is not available for the other trigger script types. It would be rare for external reference data to not be available in time for those scripts.
- The execution of the **On Show** script is not delayed if there is an invalid reference (that is, the reference's quality is Bad).
- Named scripts are blocked until the **On Show** script has completed, so some could be missed. For example, the named script OnDataChange might not run for the first few updates.
- Delayed **On Show** scripts within nested embedded graphics might run out of order for the different nested levels. If the outer-most level is delayed but the inner levels are not delayed and are executed immediately, the order of execution will be changed.

Note: You can increase the effective data time out period for an **On Show** script by using the **Additional settings** feature when editing an OMI ViewApp. See Set additional properties for a ViewApp for more information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Add a named script

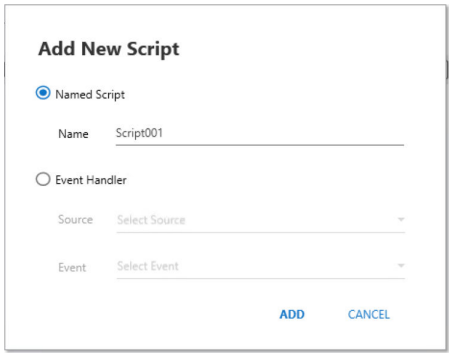
In addition to the built-in scripts that are triggered by the status of the layout, you can add scripts that are triggered by expressions, or by changes in values and/or data quality. These expression-based scripts are called "named scripts." You can configure a named script to run:

- One time when an expression changes from false to true or true to false, or when a value or its quality changes. Scripts that use the On True, On False, and Data Change triggers will run once when the specified event occurs.
- Periodically, at the specified interval, when the trigger is set to While True or While False.

Each added script can contain only one trigger type.

To add a named script to a layout

1. Open the layout in the Layout Editor and select the **Script** tab.
2. Click the **Add Script** button.
3. Select the **Named Script** choice from the pop-up window.



4. To rename the script from the default "Script00*n*," click or touch the script name in the **Name** field and edit the name.
5. Click **Add**. The new script is created in the Scripts list.

   To rename the script after it has been added to the Scripts list, click or touch the script name and edit the name.

To configure a named script

1. In the **Expression** field, enter an expression, value or reference. As you type, an autocomplete menu will present any valid items that start with the characters you are typing.

   The expression acts as data source for the script trigger.
2. In the **Trigger** field, select the script trigger:
   - **While True** triggers the script to run periodically when the expression is true. The default **period** for **While True** scripts is 1000 ms. To change the period, enter a new value in the field.
   - **While False** triggers the script to run periodically when the expression is false. The default **period** for **While False** scripts is 1000 ms. To change the period, enter a new value in the field.
   - **On True** triggers the script to run one time when the expression transitions to true from false.
   - **On False** triggers the script to run one time when the expression transitions to false from true.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **Data Change** triggers the script one time when the value of the expression or its quality changes. You can specify a **deadband** value to filter insignificant changes. To trigger the script for changes to both value and quality, select the **Quality Changes** check box.

3. If you are writing/editing a **While True** or **While False** script, specify the frequency that the script will be executed by entering an interval (in milliseconds) in the **Period** field.

   The **Period** field specifies the length of the interval from the start of one script execution cycle to the next, while the layout remains open. If the **Period** field is set to a very short interval, system performance may decrease. Enter an interval value in milliseconds.

   - Default: 1000 ms (one second)
   - Maximum: 360,000 ms (six minutes)

   Note: Data Timeout, Deadband, and **Quality Changes** fields are not applicable to, and are disabled for **While Showing** scripts.

4. Enter your script in the main edit box. Autocomplete helps by displaying valid items, based on the initial characters you enter.
   - You can validate your script for syntax errors prior to saving by pressing the **Validate Script** button.
   - To clear all data entered for the the script, including the expression, press the **Clear Script** button. There is no undo for this action.
5. To save the script, select the **Save** or **Save and Close** button.
6. The script is validated automatically. If there are warnings or errors generated, a dialog listing the errors is displayed. You have the following options:
   - Select **Yes** to save the script and preserve the errors.
   - Select **No** to discard the changes you made to the script. If you selected **Save and Close**, the Layout editor closes and you will lose any changes you made to the script. If you selected **Save**, the Layout Editor remains open and you can resume editing the script.
   - Select **Cancel** to resume editing the script. The Layout Editor remains open and you can resume editing the script.

About the ShowContent() function

Note: The Show/Hide Content() script functions are for use with AVEVA OMI ViewApps only. These functions cannot be used with InTouch HMI applications.

The Show/Hide Content() script functions let you write scripts for AVEVA OMI layouts, Industrial Graphics, Application Server objects to display a graphic, layout, or external content item (for example, a web page, video, or text file) inside a specific pane within an AVEVA OMI ViewApp. The Show/Hide Content() script functions are complementary to the Show/Hide Graphic script functions and the Show/Hide Graphic animation feature.

- Use the Show/Hide Graphic script functions to load or hide a graphic in a modal or modeless popup, in an InTouch HMI application or an AVEVA OMI ViewApp.
- Use the Show/Hide Content script functions to load or hide a graphic, layout, or an external content item into a pane in an AVEVA OMI ViewApp. When a Layout linked to an asset is added to a pane in another layout, any empty panes in the linked Layout that have the "Use for Autofill" flag enabled will attempt to fill the empty panes with content from the linked asset, using the “Current Only” autofill setting.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The Show/Hide Content functions let you specify:

- The specific content to show or hide in a pane.
- The pane in which to show or hide the content.
- The screen that contains a pane in which to show or hide the content.
- The type of content you want to show or hide.
- Any property overrides to apply when showing the content.
- Whether to confine the search for a pane in which to show or hide the content to the source screen, the primary screen, or to search all screens, and whether to search in the top level layout or a nested (embedded) layout.

Configure the Show/Hide Content script functions

Use the Show/Hide Content functions in layout scripts, Industrial Graphic scripts, or Application Server object scripts to populate a pane with a specific graphic, layout, or external content object in an AVEVA OMI ViewApp. You can call the same script multiple times to refresh content, or to show the content in a different pane. If you choose to display the same content in different panes, you can alter its settings through the ParameterOverride parameter.

If ShowContent calls the same content item while it is open, ShowContent searches for open content items that match the specified parameters in the call. Any matching items are closed and then reopened. If the content has changed, the latest changes are shown when the content is redisplayed.

Important: The Show/HideContent functions can be used in a graphic or layout action script, named script and pre-defined script. Although the system allows you to include it in a server script, such as Start Up, On Scan, Off Scan, Shut Down and Execute, you will not be able to execute the function at runtime.

To include a script that contains the Show/Hide Content functions within a script

1. Open the System Platform IDE.
2. Open a new or existing graphic or layout.
   - To add a layout script, click the **Script** tab.
   - To add a graphic script, create a graphic and open it fro editing. Double-click it to open the **Edit Animations** page, then open the script editor and click the **Display Script Function Browser** icon. The **Script Function Browser** appears. Click the ShowContent() script function in the **Graphic Client** list, and then click **OK.** The following code snippet is added:

     ```
     Dim contentInfo as aaContent.ContentInfo;
     contentInfo.Content = "<ContentName>";
     ShowContent( contentInfo );
     ```

   Note: You can click **Help** to view the Help after you have selected any Graphic Client script function.
3. Modify the script as needed. ContentInfo is a predefined structure that contains data members listed below. Content is the only required parameter. For more information about the Show/Hide Content, see the AVEVA Scripting Guide.
   - **Content:** name of the content (graphic, layout, or external content object) to be loaded into the pane. Content is a string and is the only required parameter.
   - **Name:** name property of the content. This is automatically created when the content is added to the layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **ScreenName:** name of the screen that contains the pane in which to load the content. ScreenName is a string.
- **SearchScope:** when ScreenName is not specified, **SearchScope** determines which screens within the Screen Profile are searched for a pane in which to place the content. SearchScope is an enum that can be set to one of the following values:
  - **Self:** This is the default. Searches for matching content within the panes of the layout from which the ShowContent call was made. Self will search from a nested (embedded) layout, if the call was made from that layout. The remaining SearchScopes search ONLY the top level if there are nested layouts.
  - **AllScreens:** Searches the top level layout on all screens, starting with the source screen, then the primary screen, and then any remaining screens in alphabetical order. The search stops as soon as a matching pane is found. If a matching pane is not found, Content is placed in the default pane of the source screen.
  - **SourceScreen:** Searches the top level layout only in the source screen (the screen that made the ShowContent call).
  - **PrimaryScreen:** Searches the top level layout only in the primary screen, as designated in the Screen Profile.
- **PaneName:** name of the pane in which to load the content. PaneName is a string.
- **ContentType:** specifies the type of content to be loaded. The ContentType can be matched against the Content Type designation that was set for a pane. ContentType is a string.
- **PropertyOverrides:** specifies overrides for custom properties. This parameter is only valid if a graphic is the designated content. It is not valid if a layout has been designated. PropertyOverrides is a ValuePair structure.
- **OwningObject:** specifies an automation object as an owning object.

Example

```
Dim contentInfo as aaContent.ContentInfo;
Dim cpValues [2] as aaContent.PropertyOverrideValue;
cpValues[1] = new aaContent.PropertyOverrideValue("CP1", "20", true);
cpValues[2] = new aaContent.PropertyOverrideValue("CP2", "Pump.PV.TagName", false);
contentInfo.Content = "S1";
contentInfo.ContentType = "Overview";
contentInfo.OwningObject = "Enterprise";
contentInfo.PaneName = "Pane1";
contentInfo.ScreenName = "Wall";
contentInfo.PropertyOverrideValues = cpValues;
contentInfo.SearchScope = aaContent.SearchScope.PrimaryScreen;
ShowContent ( contentInfo );
```

About the best pane match algorithm

When some or all of the optional parameters are not provided or are invalid, ShowContent follows a set of rules to determine in which pane the specified content should placed. In addition, the SearchScope parameter can be used to limit which screens will be searched for a pane.

- If SearchScope is set to SourceScreen, only the screen that made the ShowContent call is searched. If no matching pane is found, Content is loaded into the default pane of the SourceScreen.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- If SearchScope is set to PrimaryScreen, only the primary screen in the Screen Profile is searched. If no matching pane is found, Content is loaded into the default pane of the PrimaryScreen.
- If SearchScope is set to AllScreens, every screen contained in the Screen Profile can be searched. This is the default behavior. Searches progress as follows and stop at the first match:
  - SourceScreen
  - PrimaryScreen
  - Any additional screens are searched in alphabetical order.
  - If no matching pane is found, Content is loaded into the default pane of the SourceScreen.
- If SearchScope and ScreenName are not specified and ShowContent is called from an embedded layout:
  - ShowContent searches locally for a matching pane within the embedded layout.
- If SearchScope or ScreenName is specified and ShowContent is called from an embedded layout:
  - ShowContent searches for a pane globally and uses the SearchScope and/or ScreenName settings to determine which screens within the Screen Profile associated with the ViewApp will be searched.

Order of precedence for determining content placement

Content is the only required parameter. Three values are also checked according to the following order of precedence to determine placement of the content:

1. PaneName: optional parameter
2. ContentType: optional parameter
3. Type of content: this is part of the Content definition and is not specified in the ShowContent call.

Search steps (SearchScope not specified or set to AllScreens)

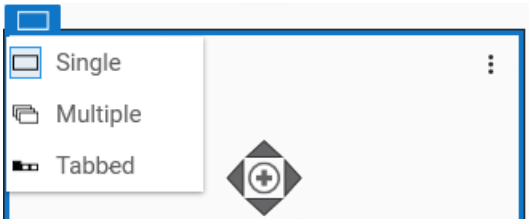
Scenario 1: PaneName PaneName is specified	Assumes SearchScope not set or set to AllScreens. Otherwise, search is confined to the named screen. 1. Search SourceScreen for PaneName. 2. If not found, search PrimaryScreen for PaneName. 3. If not found, search any additional screens in alphabetical order for PaneName. The search stops as soon as a matching PaneName is found.
Scenario 2: ContentType PaneName is not specified OR matching PaneName not found AND ContentType is specified	Assumes SearchScope not set or set to AllScreens. Otherwise, search is confined to the named screen. 1. Search SourceScreen for a pane that supports the ContentType. 2. If not found, search PrimaryScreen for pane that supports the ContentType. 3. If not found, search any additional screens in alphabetical order for a pane that supports the ContentType. The search stops as soon as a pane that supports the ContentType is found.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Scenario 3: Type of content PaneName is not specified OR matching PaneName not found AND ContentType not specified or not found	Assumes SearchScope not set or set to AllScreens. Otherwise, search is confined to the named screen. 1. Evaluate type of content for the specified Content. 2. Search SourceScreen for a pane that supports the type of content. 3. If not found, search PrimaryScreen for pane that supports the type of content. 4. If not found, search any additional screens in alphabetical order for a pane that supports the type of content. The search stops as soon as a pane that supports the type of content is found.
Scenario 4: No matching pane No matches found from previous searches (scenarios 1 through 3)	When SearchScope is not set or set to AllScreens: If no match is found using PaneName, ContentType, or type of content, Content is loaded into the default pane of the SourceScreen. When SearchScope is set to SourceScreen: Only the SourceScreen is searched and if no match is found, Content is loaded into the default pane of the SourceScreen. When SearchScope is set to PrimaryScreen: Only the PrimaryScreen is searched, and if no match is found, Content is loaded into the default pane of the PrimaryScreen.

Content display rules

The way content is displayed differs slightly between panes that are configured as single content panes versus those configured as multi-content panes. The differences are described below. Pane configuration (single or multi-content) is set by opening the **Presentation Style** selector at the top left of the pane in the **Layout Editor**. See Set the Presentation Style of a Pane (see Set the pane presentation style on page 283) in the *System Platform Help* for additional information about single and multiple content modes.



If there are multiple ShowContent() calls, but with different parameters, each ShowContent() call opens a different pane or tab. The content in each pane or tab is configured with the parameters that are specified in the call.

Content display rules – single content pane

If the pane already contains content, the existing content is closed and the content specified by ShowContent() then is loaded into the pane. This occurs even if the existing content is the same as the specified content.

Content display rules – multi-content or tabbed pane

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For panes that are configured as multiple-content or tabbed panes,the exact behavior will depend on the structure of the pane.

- If the specified Content is already open in the pane or a tab, the pane or tab remains open and the existing Content is replaced with a new instance of the same Content.
- If the pane or tab has been split (that is, contains child panes), or contains a nested layout, existing panes and/or tabs remain open, and the specified content is loaded into a new tab or pane and the focus is set to the new tab or pane.
  - This is true as long as the number of tabs or child panes is less than the maximum available. The default setting allows up to 20 panes or tabs, but you can increase this to 50.
  - If the maximum number of tabs or child panes has been reached, the pane that has been updated last (oldest content) is closed and a new tab or pane is opened in its place.

Layout event handler scripts

The Layout Editor lets you add event-handler scripts to a layout if you have already added an app or control to a pane within the layout that contains public events. There are two ways to add an event-handler script after dropping an app or control onto a pane:

- From the **Build** tab
- From the **Script** tab

When you add an event-handler script, triggered by a specific event that occurs in the selected app, the script is added to the scripts that are listed under "Event Handler Scripts "on the **Script** tab of the Layout Editor. You can then select and configure the script in the script editor.

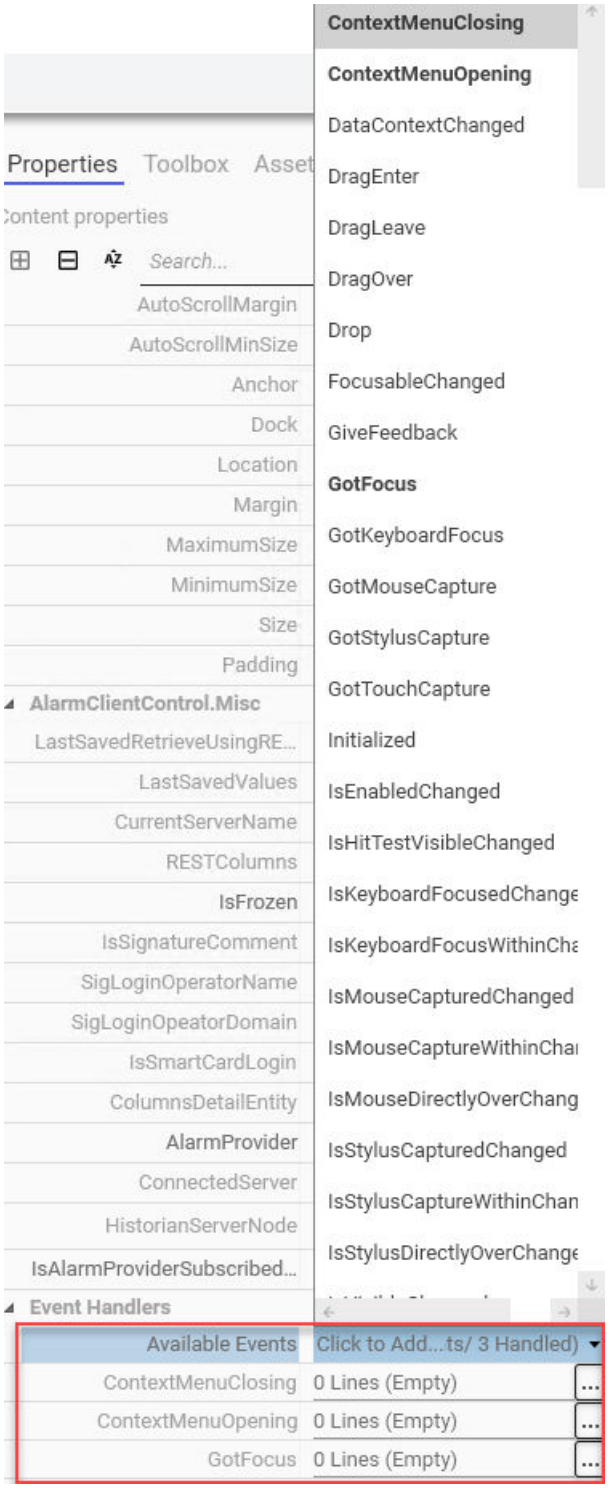
Note: Event-handler scripts, unlike named and built-in scripts, do not require the use of the MyContent namespace when scripting content properties and methods. To script a property or method, start by entering the name of the control instead of "MyContent." The autocomplete function will allow you to see the properties and methods of the control as you type. However, apps based on Windows Presentation Foundation (WPF) or .NET Controls may not provide the related event parameters in autocomplete. See Layout event handler parameters for more information.

Add an event-handler script to a layout from the Build tab

To add an event-handler script to a layout from the Build tab

1. Select the pane that contains the app or control that you are scripting.
2. Select **Properties** grid and scroll to the bottom of the properties list.
3. If necessary, expand the **Event Handlers** section to show **Available Events**.
4. To add a script to an event, click on **Click to Add Event...**
5. The list of public public events and methods contained in the app or control is shown.
6. Select the event name for which you want to add a script. After you select the event name, it is shown in **bold** type in the list, and listed below the **Click to Add Event...** option.

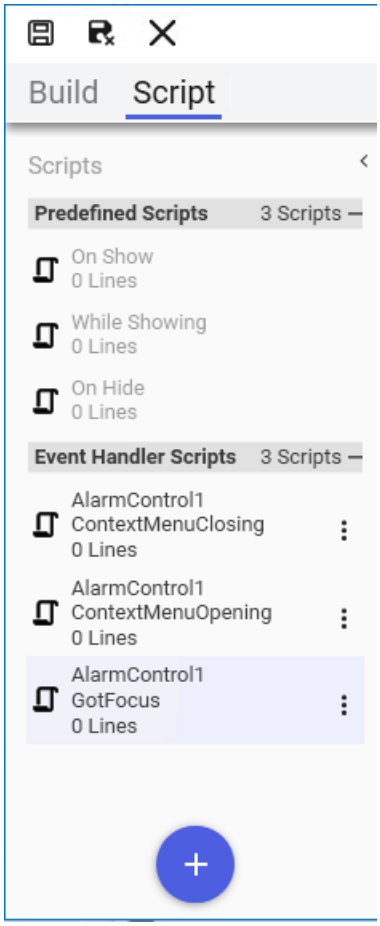
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



7. Switch to the **Script** tab. The new Event Handler script that you just added is listed alphabetically, by name of app and event, under **Event Handler Scripts** in the left column of the editor.

   The first line shows the name of the app, the second line shows the event, and the third line shows the number of lines in the script.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

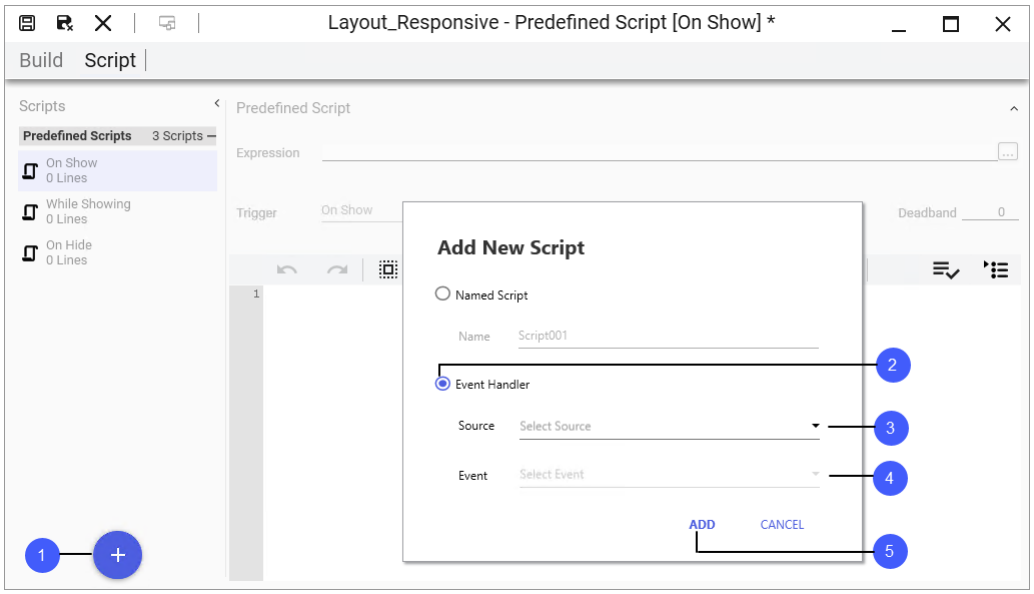


Add an event-handler script to a layout from the Script tab

To add an event-handler script to a layout from the Script tab

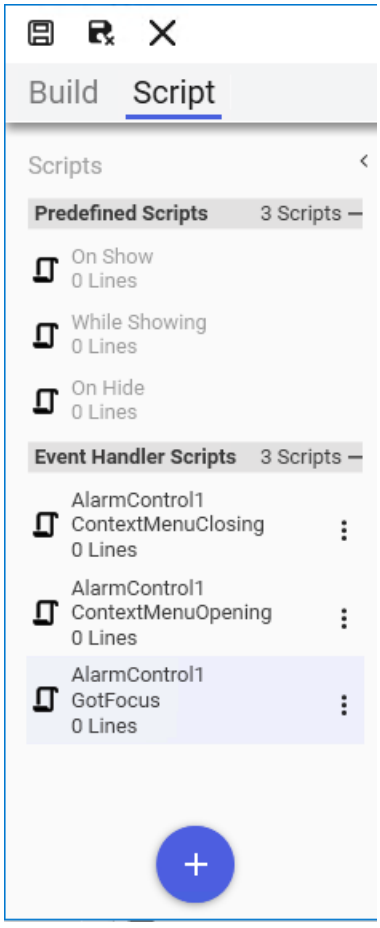
1. Click the **Add Script** button.
2. Select the **Event Handler** choice from the pop-up window.
3. Select the app/control name from the **Source** pulldown list.
4. Select the event name from the **Event** pulldown list.
5. Click **Add**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



6. The new Event Handler script that you just added is listed alphabetically, by name of app and event, under **Event Handler Scripts** in the left column of the editor.

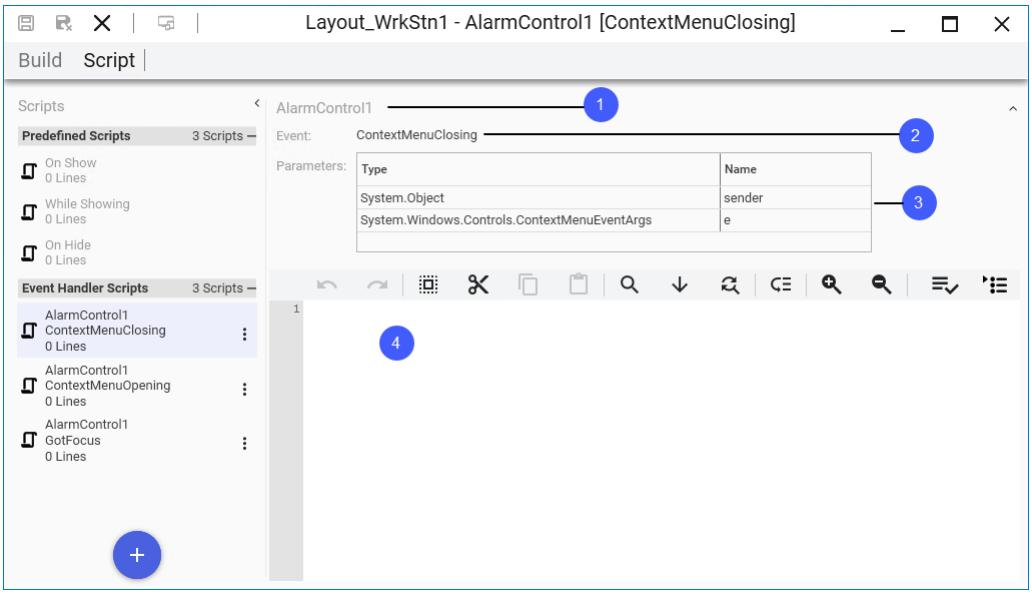
   The first line shows the name of the app, the second line shows the event, and the third line shows the number of lines in the script.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure an event handler script

When you add an Event Handler script and select the script name from the list of scripts on the **Script** tab, the app name, event, and parameters associated with the event are shown in a grid at the top of the page, above the script editing area.



1	App/Control Name
2	Event Name
3	Parameter Grid
4	Script Editing Area

In the parameter grid, you can change the order in which the parameters are shown by clicking on either of the column headers. Parameters can be shown in ascending or descending order by name or type.

All public events, methods, and .NET dependency properties are exposed for apps that are compiled against the current version of the SDK. If an app was compiled against a previously released version of the SDK, only .NET dependency properties are exposed.

When you create an Event Handler script, the event name appears in bold in the list of available events.

To configure an event handler script

1. To change a parameter name, select its name in the parameter grid and enter a new name.
2. Enter your script in the main edit box. You can directly access properties and methods that belong to the app/control, without using the MyContent namespace .Autocomplete helps by displaying valid items, based on the initial characters you enter.
    - You can validate your script for syntax errors prior to saving by pressing the **Validate Script** ☑ button.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- To clear all data entered for the the script, including the expression, press the **Clear Script** button. There is no undo for this action.

3. To save the script, select the **Save** or **Save and Close** button.
4. The script is validated automatically. If there are warnings or errors generated, a dialog listing the errors is displayed. You have the following options:
   - Select **Yes** to save the script and preserve the errors.
   - Select **No** to discard the changes you made to the script. If you selected **Save and Close**, the Layout editor closes and you will lose any changes you made to the script. If you selected **Save**, the Layout Editor remains open and you can resume editing the script.
   - Select **Cancel** to resume editing the script. The Layout Editor remains open and you can resume editing the script.

Event handler script limitations

The events for existing custom apps that were compiled against an older version of the AVEVA OMI SDK are filtered, and only .NET dependency properties will be exposed for event scripting. This ensures that public properties that may not have been intended to be used for scripting cannot be used this way.

Do not create event handlers for events that occur during, or shortly after, the creation of the app instance or short. For example, do not create an event handler for the "Initialized" event.

Do not create event handlers for touch events. These will not be executed at runtime.

Layout event handler parameters

Apps based on Windows Presentation Foundation (WPF) or .NET Controls may not provide the related event parameters in autocomplete as you are entering an event handler script in the Layout Script Editor. Typically, this occurs for events, such as mouseover, doubleclick, etc., and other events related to WPF types.

To enable autocomplete for event parameters within your event handler script, import the corresponding DLL where the event parameter type is defined. Use the System Platform IDE import function to import the DLL as a script library.

Layout execution order

In general, graphic scripts run before layout scripts run. This is to ensure that any references within the layout script will exist when a layout script calls the reference. The exception to this rule is a layout On Hide script, which, when triggered by the layout closing, will run before any graphic scripts that are still waiting to execute. Layout scripts thus have the following precedence with regards to graphic scripts:

- The On Hide layout script has precedence over graphic scripts.
- Graphic scripts have precedence over the On Show layout script.
- Named layout scripts can run in any order (not necessarily in the listed order), and may be interspersed with graphic scripts.

Any named script that is triggered by the Data Change trigger type runs the first time when the reference is subscribed to. This behavior is different than the Data Change trigger behavior of Application Server scripts and can take considerable time in intermittent networks.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Data binding in a layout

As you add apps and graphics to a layout, the exposed properties of the apps and the public properties of graphics are added to the MyContent Namespace. For more information, see About the MyContent namespace. You can view and edit the property values of a selected app or graphic you select the **Properties** grid of the **Layout Editor**. The properties can be used in runtime, both directly by setting references and through scripting.

A unique content name is generated and displayed in the **Layout Editor** at the time you drag and drop an app or graphic onto a layout pane. The generated name uses the original graphic or app name and appends a number. For example, if you drop a graphic named "MyGraphic" on the pane, the unique generated name will be "MyGraphic1" for the first usage of the graphic; if you drop the same graphic on another pane the generated name will be "MyGraphic2." The syntax for referencing a custom property in MyGraphic1 is:

```
MyContent.Mygraphic1.CustomPropertyName
```

The autocomplete function of the Layout Editor works with the MyContent namespace and presents a dropdown list of apps, graphics, custom properties, and exposed app properties contained within the namespace. The items in dropdown list are filtered by context, so only valid items are listed in the dropdown.

In addition to exposed app properties, any WPF dependency properties contained within apps are also exposed, and you can bind to them as well.

If you attempt to bind to a property that does not exist, an error is displayed in the **Layout Editor**. If you do not correct the error before saving and/or closing the layout, a warning is displayed that the reference could not be resolved, and if you save the layout, it will be shown to be in a warning state in the **Visualization folder**.

About the MyContent namespace

The MyContent namespace lets you create references in scripts to tie together the content added to a single layout. The layout content can consist of graphics, apps, other layouts, and controls. The MyContent namespace encompasses everything the layout contains, including:

- Public custom properties of graphics associated with the layout
- Exposed properties of apps contained in the layout
- WPF dependency properties of apps contained in the layout

The exposed properties of each app are listed in the AppManifest.xml file for the app.

Note: Event-handler scripts, unlike named and built-in scripts, do not require the use of the MyContent namespace when scripting app or control properties and methods. To script a property or method, start by entering the name of the control instead of "MyContent." The autocomplete function will allow you to see the properties and methods of the control as you type.

Properties designated as private are not accessible in the MyContent namespace. Each layout has its own MyContent namespace and you cannot create references to properties in another layouts via this namespace. If you need to reference properties in a different layout, use the ViewApp namespace and create attributes as needed for the references. See About ViewApp namespaces for more information.

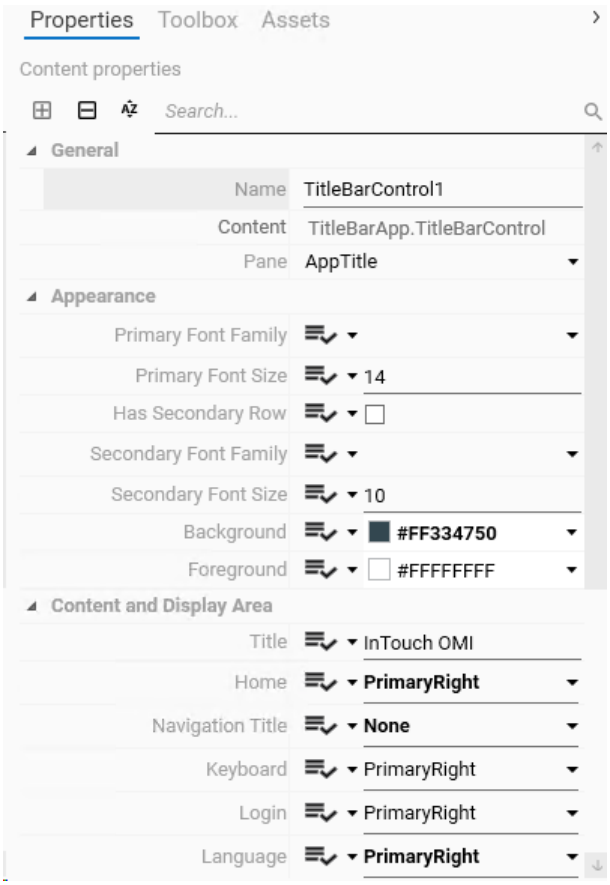
Note: MyContent Namespace properties cannot be referenced from an object script. References from object scripts will validate successfully within the script editor, but a configuration error message will be displayed when you save the object. Objects that contain a script that references a MyContent Namespace property will remain in a warning state and the reference will not resolve.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Add property binding

Public properties for both **AVEVA OMI apps** and Industrial Graphics that contain custom properties are visible and configurable in the Layout Editor. Configuration consists of overriding default settings such as enabling or disabling a property, setting the binding mode of a property, or changing a string value. You can view the properties of an app or graphic by selecting the pane that contains the app or graphic and then selecting the **Properties** grid. Private properties are not exposed and not configurable through the Layout Editor.

Note: Although the Industrial Graphic Editor enables you to view graphic custom properties that are set to private, the properties cannot be overridden, either in the Industrial Graphic Editor or Layout Editor, unless you first set the property to public in the Industrial Graphic Editor. You cannot view graphic custom properties or app properties set to private from the Layout Editor. If you want to override an app property that is set to private, you must change it programmatically. There is no graphical interface for changing app properties from private to public.



App property binding can be set to static, dynamic/read-only, dynamic/write-only, or dynamic/read-write, as follows:

Constant	Static binding to the specified property value.
In	Dynamic property value with read only binding by the control to a reference.
Out	Dynamic property value with write only binding by the control to a reference.
InOut	Dynamic property value with read/write binding by the control to a reference.

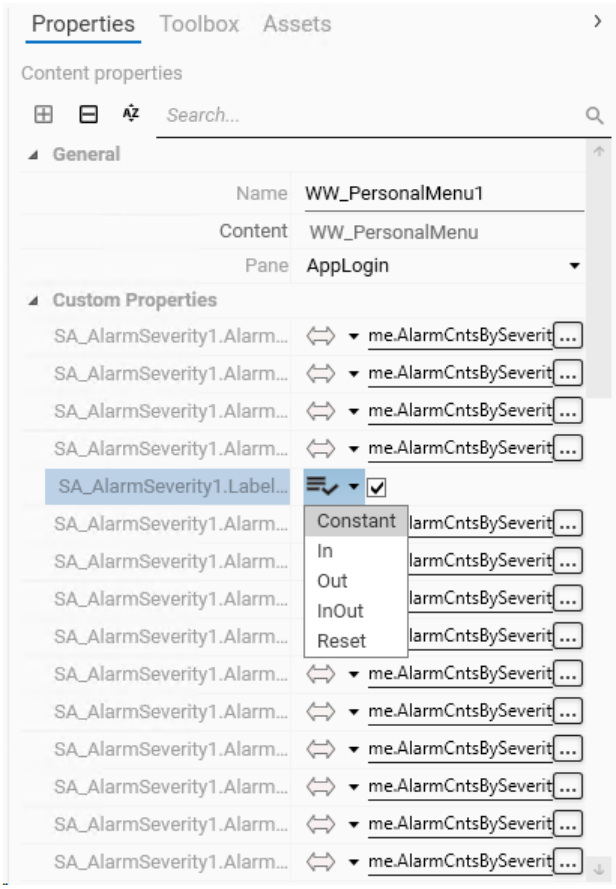
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Reset	Reset to the default property value.

When you override an app custom property, a "revert" symbol appears to the right of the property. This signifies that the default value of the property has been overridden. You can revert to the default value of the property by clicking on or touching the revert symbol. The revert symbol does not appear when you override other app properties, such as general or appearance properties.

Set custom property binding type

Public app properties are visible and configurable in the Layout Editor. Similarly, graphic custom properties that are public are also visible and configurable in the Layout Editor. Configuration consists of overriding default settings such as enabling or disabling a property, setting the binding mode of a property, or changing a string value. You can view the custom properties by selecting an embedded graphic and then selecting the **Properties** grid on the Layout editor.



Custom property binding can be set to static, dynamic/read-only, dynamic/write-only, or dynamic/read-write, as follows:

Constant	Static binding to the specified property value.
In	Dynamic property value with read only binding by the control to a reference.
Out	Dynamic property value with write only binding by the control to a reference.
InOut	Dynamic property value with read/write binding by the control to a reference.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Reset	Reset to the default property value.

Use a layout as content

You can link a layout to an asset or template, in the same way that you link a graphic. Similarly, you can link layouts to an object wizard. See Link to shared content in the visualization folder for more information.

OwningObject Property

When you embed a graphic in another graphic, or add a graphic to a layout, the OwningObject property is exposed. The OwningObject property references the object that hosts the graphic or layout, and can be modified at runtime or via a bound value to the property.

If the OwningObject property of the layout is modified, all pre-existing subscribed references are removed and the layout is reloaded in its original state. If the layout contains empty panes set to use AutoFill, the new OwningObject property is used to build the relative references to the content.

AutoFill Limitations

In a multi-galaxy environment, AutoFill is unable to fill the panes if the owning object is set to an object in the remote galaxy.

Multi-Content Pane Limitations

A layout should not be added to multi-content **stacked** panes because navigation buttons will not be visible. If a layout is added to a multi-content **tabbed** pane, navigation is possible by using the tabs. See About ViewApp navigation for more information.

Effects of autofill on a layout linked to an asset

When a Layout linked to an asset is added to a pane in another layout, any empty panes in the linked Layout that have the "Use for Autofill" flag enabled will attempt to fill the empty panes with content from the linked asset, using the “Current Only” autofill setting. For more information, see About auto-fill navigation.

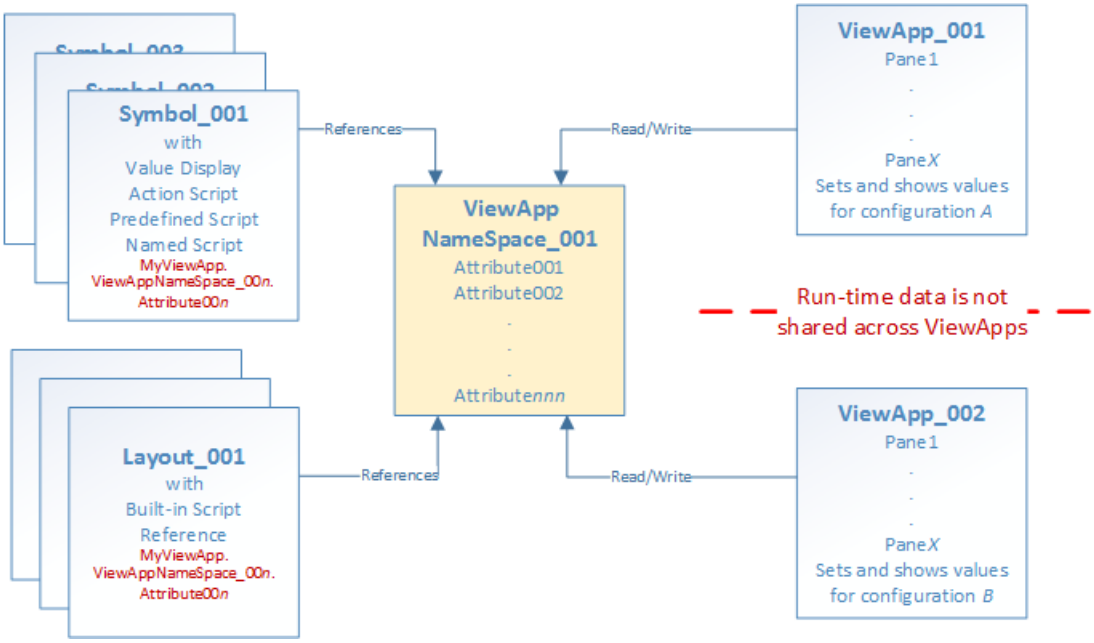
About ViewApp namespaces

ViewApp Namespaces are repositories for custom ViewApp attributes. A ViewApp Namespace can be thought of as a tag bucket, and the ViewApp Namespace attributes as individual tags. ViewApp Namespace attributes are functionally similar to graphic custom properties, but are not bound to individual graphics. To invoke autocomplete for ViewApp namespaces and attributes when creating a graphic or layout script in the Script Editor, enter the keyword "MyViewApp." For more information, see About custom properties for graphics.

ViewApp Namespaces are owned by the Galaxy in which they are created. You can can reuse ViewApp Namespaces across multiple Viewapps, but while configuration data is shared across all ViewApps, runtime values can only be shared within a single ViewApp. Each ViewApp maintains its own copy of the ViewApp Namespace data. ViewApp Namespace attributes can be used in graphic scripts and references, animation scripts, and layout scripts and references. When you create references to ViewApp attributes in graphic or layout scripts and animations, attribute values can be shared across different panes of a running AVEVA OMI ViewApp. The ViewApp Namespace attributes are referenced much like static ViewApp attributes for alarms, navigation, language, etc. For more information, see "My ViewApp Attributes" in the *AVEVA OMI SDK Help* for more information about the static ViewApp attributes.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: ViewApp Namespace attributes cannot be referenced from an object script. References from object scripts will validate successfully within the script editor, but a configuration error message will be displayed when you save the object. Objects that contain a script that references a ViewApp Namespace attribute will remain in a warning state and the reference will not resolve.



ViewApp Namespace Editor

The ViewApp Namespace Editor lets you perform the following functions:

- Add, duplicate, and remove ViewApp attributes
- Rename ViewApp attributes
- Set ViewApp attribute data types
- Configure initial ViewApp attribute values
- Set ViewApp attributes as read-only and/or retentive
- Add descriptions to ViewApp attributes

ViewApp Namespace Elements

Each ViewApp Namespace attribute consists of the following elements:

- Name: this is a string and must match galaxy naming rules
- Description: the description is shown when browsing for attributes in the Galaxy Browser
- Data type: supported data types and default values are:
  - Boolean – false (unchecked)
  - Double – 0.0
  - Elapsed time – 00:00:00.0000000
  - Float – 0.0

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

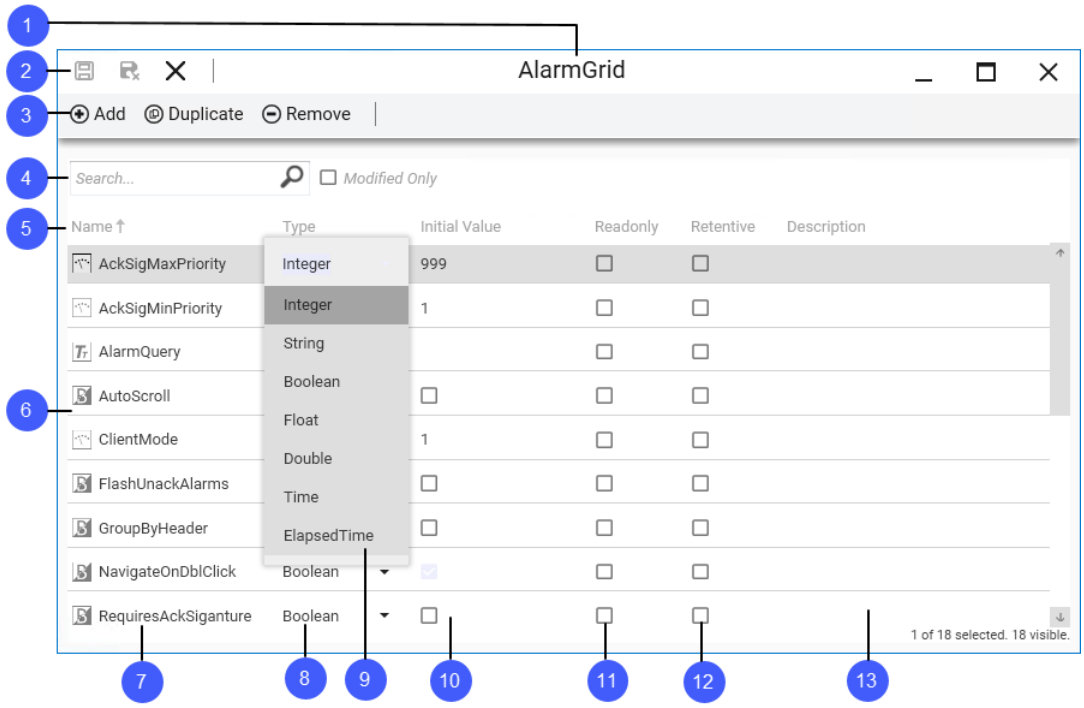
- Integer – 0
- String – empty string ("")
- Time – current time, in the format 2/18/2018 11:36:14:385 AM

Note: Wherever possible, the initial value of the attribute persists when the attribute type is changed. The most notable exception to this behavior occurs when changing an attribute type from Boolean to String. In this case, the initial value is set to an empty string.

- Read-only option: when selected, the value will not change at runtime. Default value is unchecked (false).
- Initial value or expression: expressions can be used as values
- Retentive value: default value is unchecked (false). The last runtime value of a retentive attribute is saved during a clean shut-down of the ViewApp, and will be restored the next time the ViewApp is launched. An attribute cannot be both read-only and retentive.

About the ViewApp namespace editor

The figure below shows the various components of the **ViewApp Namespace Editor**. Use the ViewApp Namespace Editor to create and configure custom attributes that can be used by all ViewApps in the Galaxy.



1	ViewApp Namespace Name	Name assigned to the ViewApp Namespace. An asterisk (*) after the name indicates that the ViewApp Namespace has unsaved changes.
2	Menu Bar	Commands to save, save and close, and close the ViewApp Namespace in the editor.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3	Attribute Command Bar	Commands to add, duplicate, or remove attributes. You can use the **Ctrl** or **Shift** key to select multiple attributes for duplication or removal.
4	Search Box	Enter text to filter the attribute list. The text can appear in any part of the attribute name. If the "Modified Only" checkbox has been selected, only attributes that have changed since the last save are shown.
5	Active Sort Column	An up or down arrow next to the column name indicates that attributes are currently sorted by that column and the sort direction.
6	Attribute List	The attributes contained in the ViewApp Namespace are displayed in the attribute list. Attributes with unsaved changes are shown in **bold** type.
7	Name Column	Names of the attributes in the ViewApp Namespace. Unsaved name changes can be reverted by pressing the **Esc** key, provided that you have not navigated away from the changed name field.
8	Data Type Column	Data types of the attributes in the ViewApp Namespace. The data type is also indicated by an icon that precedes the attribute name.
9	Data Type Drop Down	Assign a data type from the drop down to configure the selected attribute.
10	Initial Value	Initial value setting for the attribute that is shown at runtime. Unsaved value changes can be reverted by pressing the **Esc** key, provided that you have not navigated away from the changed value field.
11	Read-Only CheckBox	Select the box to designate an attribute as read-only. Default value is false (Read/Write). An attribute cannot be read-only and retentive.
12	Retentive CheckBox	Select the box to designate an attribute as retentive. Default value is false (non-retentive). An attribute cannot be retentive and read-only.
13	Attribute Description	Add a description that will appear as a tooltip.

Create a ViewApp namespace

When you first create a new Galaxy, the Visualization folder does not contain a default ViewApp Namespace. To create a ViewApp Namespace, you can:

- On the **Home** ribbon, select **Namespace** in the **Create** area.
- Use the keyboard shortcut **Ctrl + Shift + V**.

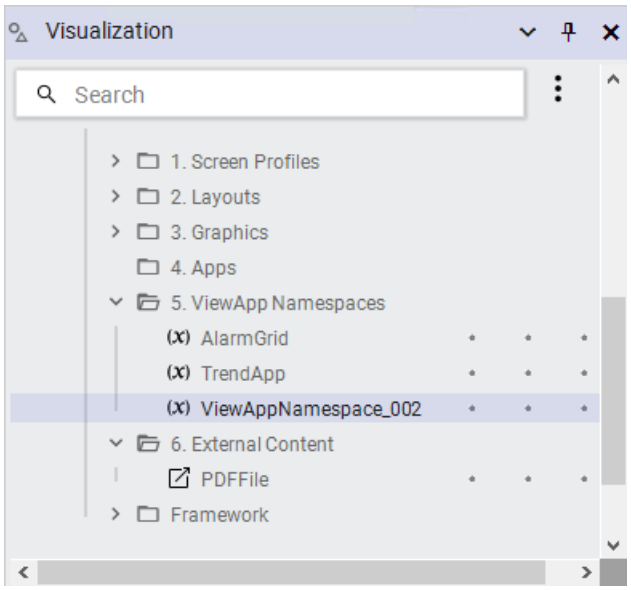
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Right-click in the **Visualization folder** to bring up the shortcut menu. Then, select **New > ViewApp Namespace**.

Any of these actions creates a new ViewApp Namespace in the Visualization folder with the default name **ViewAppNameSpace_001**. After you create the ViewApp Namespace, you can:

- Rename, duplicate, delete, check out, or export the NameSpace.
- Open the ViewApp Namespace in the editor and add or edit attributes. After creating ViewApp Namespace attributes, you can reference them from graphic and layout scripts.

Note: ViewApp Namespace attributes cannot be referenced from an object script. References from object scripts will validate successfully within the script editor, but a configuration error message will be displayed when you save the object. Objects that contain a script that references a ViewApp Namespace attribute will remain in a warning state and the reference will not resolve.



Add and configure custom ViewApp namespace attributes

This topic describes the general steps to add and configure custom ViewApp Namespace attributes into a ViewApp using a simple example.

The **ViewApp Namespace Editor** includes menu bar commands to **Add**, **Duplicate**, and **Remove** custom attributes.

Add	Adds a custom attribute to the ViewApp Namespace. The default name for the first attribute added is Attribute001. The number within the attribute name is incremented by one each time you add a new attribute. If you change the name of an attribute, the next added attribute uses the new name and increments the number contained within the name by one (the number can appear anywhere within the name). If the new name does not contain a number, "001" is appended to the name of the duplicated integer.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Duplicate	Duplicates the selected attribute and all of its properties. The only exception is that the attribute number contained in the name is incremented by one. If the selected attribute does contain a number, "001" is appended to the name of the duplicated attribute. You can select multiple attributes using the **Shift** or **Ctrl** key, and duplicate all selected attributes in a single operation.
Remove	Removes the selected attribute. You can select multiple attributes using the **Shift** or **Ctrl** key, and remove all selected attributes in a single operation.

The following procedure describes how to use these commands to create and configure ViewApp Namespace attributes.

To add custom attributes to a ViewApp Namespace

1. In the **Visualization folder**, open a ViewApp Namespace for editing.
2. Select **Add** from the menu bar of the **ViewApp Namespace Editor** to add a custom attribute.
3. Rename the attribute if necessary. To revert a pending name change, press **Esc**.
4. Configure the properties of the attribute.
   a. Select **Type** from the dropdown menu and then choose the type of attribute you are adding: Integer, String, Boolean, Float, Double, Time, or Elapsed Time.
   b. Set the initial value.
      - To revert a pending change to an initial value (other than the initial value of a Boolean), press **Esc**.
      - For Booleans, a checkbox sets the value. The default is unchecked for false.

      Note: Wherever possible, the initial value of the attribute persists when the attribute type is changed. The most notable exception to this behavior occurs when changing an attribute type from Boolean to String. In this case, the initial value is set to an empty string.
   c. Set whether the attribute is read-only.
   d. Set whether the attribute is retentive. See About the Attributes Page for more information.

      Note: An attribute cannot be set to both retentive and read-only. Since the value of a read-only attribute cannot be changed at runtime, there is no purpose in setting the value to persist across restarts.
   e. Enter a description for the attribute.

To duplicate custom attributes in a ViewApp Namespace

1. Open the ViewApp Namespace in the **ViewApp Namespace Editor**.
2. Select one or more attributes to duplicate. Use the **Shift** or **Ctrl** key to select multiple attributes.
3. Select **Duplicate** from the menu bar of the **ViewApp Namespace Editor**. This will create copies of the selected attributes. Duplicated attributes will have the same root names, types, initial values, read-only/retentive selections, and descriptions as the originals.
4. Rename the attributes as needed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

5. Configure attribute properties as needed. To revert a pending change to an attribute name or initial value (other than the initial value of a Boolean), press **Esc**.

Export a ViewApp namespace

ViewApp NameSpaces can be exported as standalone, reusable objects for use in other Galaxies. For example, you could create a ViewApp NameSpace that is broadly applicable across a variety of Galaxies for multiple locations. Once the ViewApp NameSpace has been imported into another Galaxy, it can be customized as needed.

A ViewApp NameSpace will be bundled automatically with other objects you are exporting, if one of the linked layouts or graphics you are exporting contains a reference to the ViewApp NameSpace. When you select a ViewApp for export, a dialog will ask if you want to include ViewApp NameSpaces with the export (along with navigation hierarchy assets and automation objects that host graphics). If you select the checkbox to include these items, then all ViewApp NameSpaces will be bundled with the exported ViewApp and its associated navigation hierarchy assets and automation objects.

Use the same method to export ViewApp NameSpaces that you use to export other objects. See Export objects for additional information.

Import a ViewApp namespace

To reuse one or more ViewApp NameSpaces that were previously created in anther Galaxy, you can export the ViewApp NameSpaces you want to use into an .aaPKF (or .aaPDF file, if you need to save the existing configuration data). Then, import the file into the Visualization folder of the new Galaxy. See Import objects for additional information.

Once you have imported a ViewApp NameSpace into the Visualization folder, you can open and configure it with the **ViewApp NameSpace Editor**.

Create references to ViewApp namespace attributes

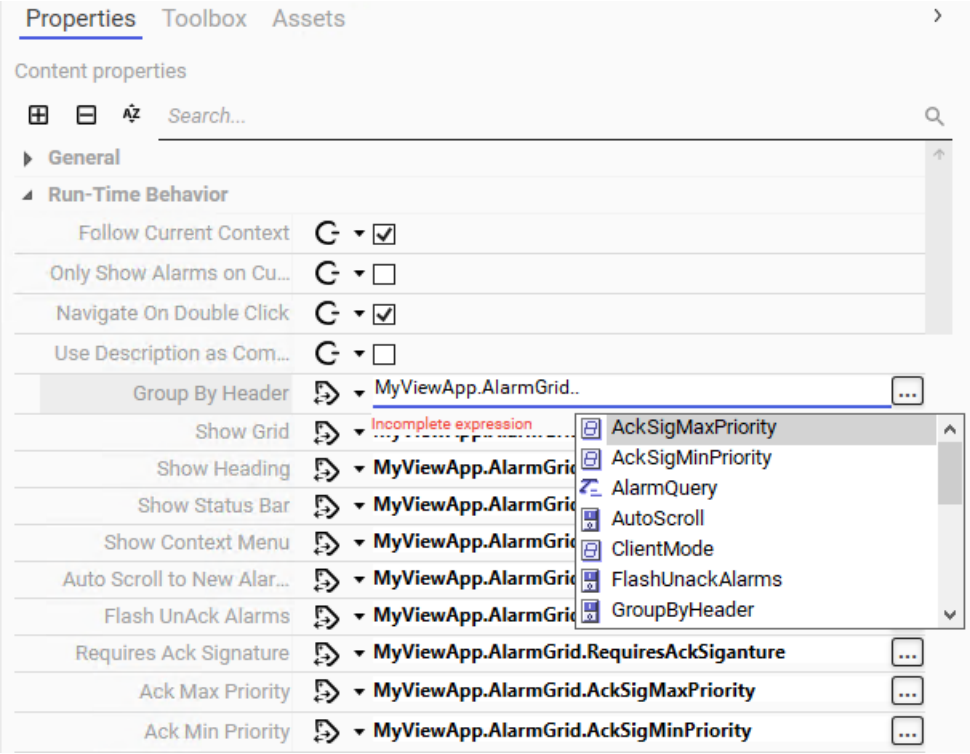
In order for a ViewApp NameSpace to have value, you must create references to its attributes. You can create references to ViewApp NameSpace attributes from:

- Graphic scripts
- Graphic animations
- Layout scripts

Note: ViewApp Namespace attributes cannot be referenced from an object script. References from object scripts will validate successfully within the script editor, but a configuration error message will be displayed when you save the object. Objects that contain a script that references a ViewApp Namespace attribute will remain in a warning state and the reference will not resolve.

References can also be configured on application properties that support data binding. This is supported in the **Properties Grid** of the **Layout** and **ViewApp** editors, as shown below. See Configurable Properties in Archestra Apps (see Configurable properties in AVEVA apps on page 109) for additional information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Each ViewApp NameSpace attribute reference must be prefixed with the reserved system name "MyViewApp." For example, to reference "Attribute001" in "ViewAppNameSpace_001," the reference would be:

MyViewApp.ViewAppNameSpace_001.Attribute001

Autocomplete will display items in the autocomplete list box to help you as you begin typing references in the script editor or animation editor. Just double-click the item you want from the list box (for example, "MyViewApp") or select the item and press the **Enter** key. Each part of the attribute name must use a dot (period) as a separator.

About retentive attributes

Retentive attributes for ViewApp NameSpaces differ in how they behave from other Application Server objects and from InTouch retentive tags. The table below compares the runtime behavior of retentive attributes/tags for:

- ViewApp NameSpace retentive attributes
- Application Server retentive attributes. This includes retentive attributes for application and system objects such as:
  - User Defined objects
  - Analog Device/Discrete Device objects
  - Area objects
  - WinPlatform objects
- InTouch retentive values

ViewApp NameSpace Retentive Attribute Values	**Application Server Objects Retentive Attribute Values**	**InTouch Tag Retentive Values**

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Retentive attribute values are saved when the ViewApp is closed. The saved value is loaded to the ViewApp when it is reopened. If more than one instance of a ViewApp is open, values for the last ViewApp that is closed will overwrite all previous values for the other ViewApps.	Values for retentive attributes are saved to disk every 10 seconds. Retentive attribute values are preserved across deployments. When an object is undeployed, the last saved value is restored at the next deployment	Retentive tag values are retained when the application stops, and are loaded when the application starts again. WindowViewer uses the retained valued as the initial tag value when it restarts, but does not write the value to the I/O device. I/O values are only updated after the I/O Server scans the device.

About alarms

You can create AVEVA OMI applications that generate alarms and events to provide the status of a running business process.

- Alarms warn about process conditions that can potentially cause problems. Typically, you set up an alarm to become active when a process value exceeds a defined limit. For example, you can set an alarm for a pump that warns when no fluid pressure is detected.

  An alarm is an abnormal condition that requires immediate attention. An operator usually acknowledges an alarm. Application Server objects handles the real-time reporting of alarms and provides special clients for viewing them.

- Events represent normal system, application, or user occurrences that produce status messages. A typical event occurs when an operator logs on to an application at the beginning of a work shift. Application Server can detect events, store them as historical data, and report them to client applications.

Application Server objects include built-in event and alarming reporting capabilities. You must configure alarms for each object in the IDE to use the event and alarm functions.

Understanding Events

Events represent normal system, application, or user occurrences that produce status messages. A typical event occurs when an operator logs on to an application at the beginning of a work shift. Application Server can detect events, store them as historical data, and report them to client applications.

An event indicates a significant occurrence that is detected, reported, and saved as historical data. Events can be detected by any Application Server component including automation objects and the OCMC.

Events provide a means for any software component to log information about a system, application, or operator action. Application Server components use an event API to send event messages to Alarm/Event distributors and the Historian. Also, events are sent to client applications like AVEVA OMI to be shown in real-time trends or reports.

Types of events

Application Server creates several types of runtime event messages, which are saved as historical data.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- System events

  Event messages are logged when a Platform, Engine, Area, DI Network or DI Device start or stop. Any system action that affects a large number of objects is logged as an event.

  A System event message contains the following information:

  - Event type, which is System.
  - Tagname, which is the name of the object generating the event.
  - Tag description, which is a short description of the object generating the event.
  - Area, which is the name of the area that contains the object generating the event.
  - Event description, which describes the system action and can be either Started or Stopped.
  - Timestamp, which is the current system time.

- Security-audit (operator change) events

  Event messages are logged to the Alarms and Events subsystem when an operator logs on to or logs off from an application, and when a logon attempt fails (for example, due to an incorrect password). Also, security-audit events are logged when an operator changes actions by means of User sets.

  A Security-audit event message contains the following information:

  - Event type, which is operator change.
  - Timestamp, which is the date and time when the operator change event occurred. The timestamp of the event is the current AppEngine scan time.
  - Tagname, which is the name of object generating the event.
  - Prim.attr, which is the reference string of the attribute being changed.
  - Tag description, which is a short description of the object. For Secured and Verified writes, this will contain the following:
    - Type of write (Secured or Verified)
    - Comment from the user, if any
    - Reason description, if any
  - FieldAttribute description, if the attribute is a FieldAttribute and has a description; otherwise, this is the Object description.
  - Area, which is the name of the area that contains the object generating the event.
  - UserEngine, which is the name of the view engine or other user engine requesting the operator change.
  - Operator1, which is the full name of the primary operator requesting a change. The full name is an attribute of the UserProfile.
  - Operator2, which is the full name of the secondary operator validating the change, if any.
  - Old value, which is the previous value of an attribute.
  - New Value, which is the new value of an attribute.

- Application (or process) related events

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Application event messages are generated by application objects in response to process actions. For example, application event messages are created when a process pump starts or stops.

An Application event contains the following information:

- Event type, which is data change.
- Timestamp, which is the date and time when the application event occurred. The timestamp assigned to the event is the timestamp of the attribute associated with the event, if available. Otherwise, the event timestamp is the current AppEngine scan time.
- Tagname, which is the name of the object generating the event.
- Prim.attr, which is the reference string of the attribute being changed.
- Tag description, which is a short description of the object.
- Area, which is the name of the area that contains the object generating the event.
- Old value, which is the previous value of an attribute.
- New Value, which is the new value of an attribute.

Understanding Alarms

Alarms warn about process conditions that can potentially cause problems. Typically, you set up an alarm to become active when a process value exceeds a defined limit. For example, you can set an alarm for a pump that warns when no fluid pressure is detected.

An alarm is an abnormal condition that requires immediate attention. An operator usually acknowledges an alarm. The System Platform framework handles the real-time reporting of alarms and provides special clients for viewing them.

Application and system objects can detect and generate an alarm. To detect an alarm, a system or application object sets a Boolean Attribute flag to indicate whether the alarm condition of the object is currently true or false.

To report an alarm, the object must contain an alarm feature. The alarm feature makes a reference to the object's Boolean flag to determine whether the alarm condition is true. It then combines this information with the current alarm mode to determine whether to report a this as an active or inactive alarm state. An alarm feature is dedicated to reporting a single alarm condition's state. The alarm feature send alarm notification messages to System Platform alarm and event distributors.

Every alarm notification includes a set of fields containing data that describes the alarm. Some alarm notification data is saved as historical data. The following list describes all fields sent with an alarm notification.

- TagName: The name of the object generating the alarm. Saved as historical data.
- Name: The name of the alarm. Saved as historical data.
- InAlarm: A Boolean value that indicates whether the object's alarm state is currently active or inactive. Saved as historical data.
- Quality: The current quality of the data upon which the alarm is based. Saved as historical data.
- OnTimeStamp: The time that the attribute value transitioned to an alarm state. The value timestamp of the attribute is used, if available. If not, the timestamp is the AppEngine scan time. Saved as historical data.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- OffTimeStamp: Time that the alarmed attribute value returns to normal. The value timestamp of the attribute is used, if available. If not, the OffTimeStamp is the AppEngine scan time. If an active alarm is disabled, it forces a return to normal and the timestamp is the current time. Saved as historical data.
- Category: An integer between 1 and 14 that identifies the type of alarm and source of the alarm. These values are associated with Internationalized category labels.

  Alarm category labels can be localized to other languages. Application Server uses the default Galaxy language to retrieve these strings and send them to InTouch. The alarm category labels appear in InTouch and InTouch history as the default Galaxy language strings. Saved as historical data.
- Priority: An integer value from 1 to 999 indicating the severity of the alarm. An alarm priority of 1 is most urgent and 999 least urgent.
- TargetValueReference: An optional field that makes a reference to the target value of the alarm. Not saved as historical data.
- ActualValueReference: An optional field that makes a reference to the actual attribute value for the alarm condition. Not saved as historical data.
- TargetValue Snapshot: An optional field containing the target value of the attribute at the time when the alarm became active. Saved as historical data.
- ActualValueSnapshot: An optional field containing the actual value of the attribute at the time when the alarm became active. Saved as historical data.
- EngUnitsReference: The reference to the engineering units string for the condition. Saved as historical data.
- AcknowledgedFlag: Indicates whether the alarm is acknowledged or not. If this flag is FALSE, the alarm is still unacknowledged. Saved as historical event data.
- AcknowledgeTime: Indicates the time when the alarm was acknowledged if the AcknowledgedFlag is TRUE. Saved as historical data at the time of acknowledgement.
- AcknowledgeUserId: The string containing the name of the user who acknowledged the alarm. Saved as historical data at the time of acknowledgement.
- AlarmMode: Indicates if the alarm mode is Enabled, Silenced or Disabled. Saved as historical data at the time when the alarm mode changes.
- Message text describing the alarm, which can be statically or dynamically constructed. The message typically contains the alarm description, the exceeded limit value, and possibly the variable value. For an alarm feature, this message is retrieved by means of a reference to a string/international string attribute in the object. The reference is setup at ObjectDesigner time for alarms. If none is specified, then the common feature short description attribute is utilized. This field is also provided in the alarm feature section and can be dynamically generated and scripted. Saved as historical data.
- Area: The name of the area that contains the object generating the alarm. Saved as historical data.

Types of alarms

Application Server supports the following types of alarms:

- State alarms, which are also known as Boolean alarms
- Limit alarms
- Target deviation alarms

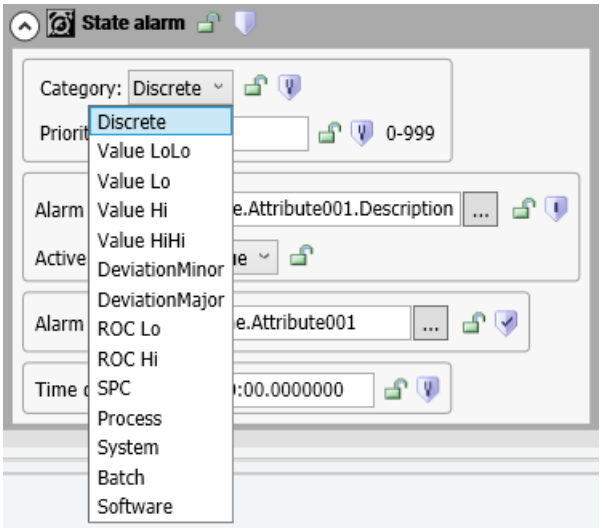
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Rate of change alarms

The types of alarm that you can configure are based on the data type of the attribute's value.

State Alarms

A state alarm set on an attribute of Boolean data type corresponds to a discrete tag with two possible states. When you create a state alarm, you configure whether the active alarm state corresponds to the TRUE or FALSE state of the attribute.



You can set an alarm category. Valid categories are:

Category	Description
Discrete	A discrete value event or alarm, such as a change of state.
Value LoLo	A continuous value is significantly below the acceptable range.
Value Lo	A continuous value is below or is approaching the low acceptable range.
Value Hi	A continuous value is above or approaching above the high acceptable range.
Value HiHi	A continuous value is significantly above the acceptable range.
DeviationMinor	A value has a minor deviation (plus or minus) from the target or setpoint.
DeviationMajor	A value has a major deviation (plus or minus) from the target or setpoint.
ROC Lo	The rate of change for a value is too slow.
ROC Hi	The rate of change for a value is too fast.
SPC	A value deviates from the SPC target/range.
Process	An alarm or event associated with the physical process/plant has occurred.
System	An alarm or event associated with the automation system has occurred.
Batch	An alarm or event associated with a batch process has occurred.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

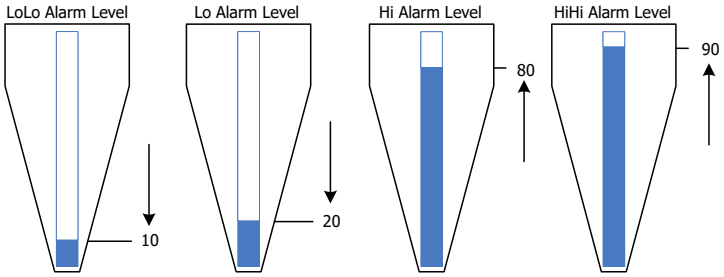
Software	An alarm or event associated with a software operation or logic (such as divide by zero in script) has occurred.

You can set an alarm message and Priority for a state alarm. The time deadband sets the length of time that an attribute value must continuously remain in an alarm or unalarmed state. The time deadband filters out rapid, transitory value spikes.

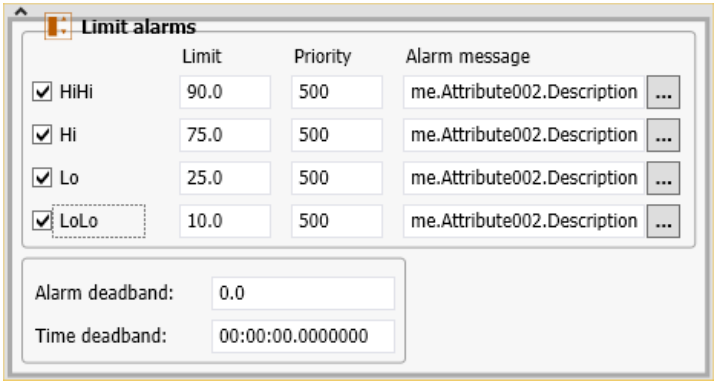
The timestamp when a state alarm becomes active or inactive is the most current timestamp of the corresponding input value. If there is no timestamp associated with the alarmed value, the AppEngine timestamp is used instead.

Limit Alarms

A limit alarm compares the current value to one or more predetermined alarm limits within the attribute's full range of values. If the value exceeds a limit, an alarm occurs.



You can individually select and configure values and priorities for the LoLo, Lo, Hi, and HiHi alarm limits. You can set individual messages for each alarm limit.



You can also configure alarm and time deadbands for limit alarms. The alarm deadband is an absolute value, in engineering units, that the attribute value must exceed the configured HiHi, Hi, Lo, or LoLo limit before the alarm is triggered, or that the attribute value must revert to before the limit alarm resets to its inactive state.

For example:

- The HiHi alarm limit is set to 90.
- The alarm deadband is set to 5.
  - The attribute value must exceed 95 (limit plus deadband value) to trigger the HiHi alarm.
  - The attribute value must fall below 85 (limit minus deadband value) to reset the HiHi alarm to an inactive state.

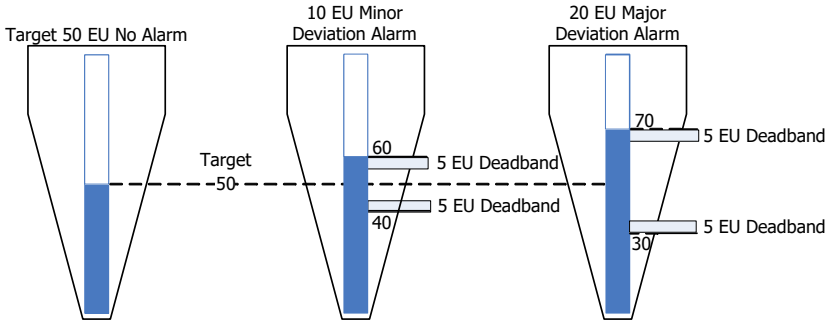
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The time deadband sets the length of time that an attribute value must continuously remain in an alarm or unalarmed condition. The process variable must remain above or below the indicated limit for at least the indicated deadband time before the application object updates the status of the alarm CONDITION Boolean. Then, standard Alarm feature logic determines whether to take that updated alarm condition and report changes to the alarm state or not.

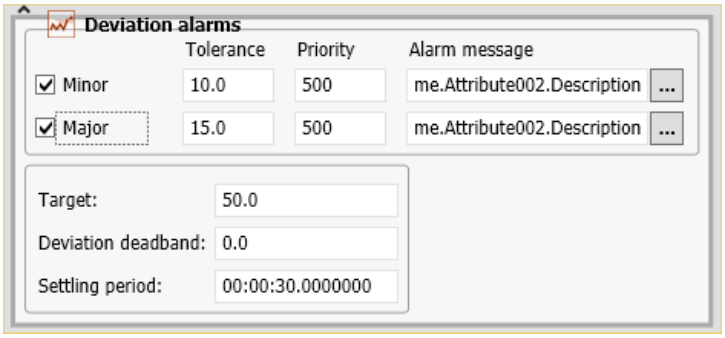
The timestamp when a limit alarm becomes active or inactive is the most current timestamp of the corresponding input value. If there is no timestamp associated with the alarmed value, the AppEngine timestamp is used instead.

Target Deviation Alarms

A target deviation alarm compares the current attribute value to a target Engineering Units value. Then, the absolute value of the difference is compared to one or more alarm deviation limits expressed in EngineeringUnits.



You can individually select and configure values and priorities for the minor deviation limit and the major deviation limit. You can set individual messages for the minor and major deviation alarm limits.



The deviation alarm's settling period is the time allowed for the attribute value to reach an expected target value after a device starts. No alarm can occur during the settling period.

You can also configure a value for a deviation deadband, which is expressed in Engineering Units. The deadband range sets a threshold that an attribute value must change from a deviation limit to reset the alarm to the inactive state.

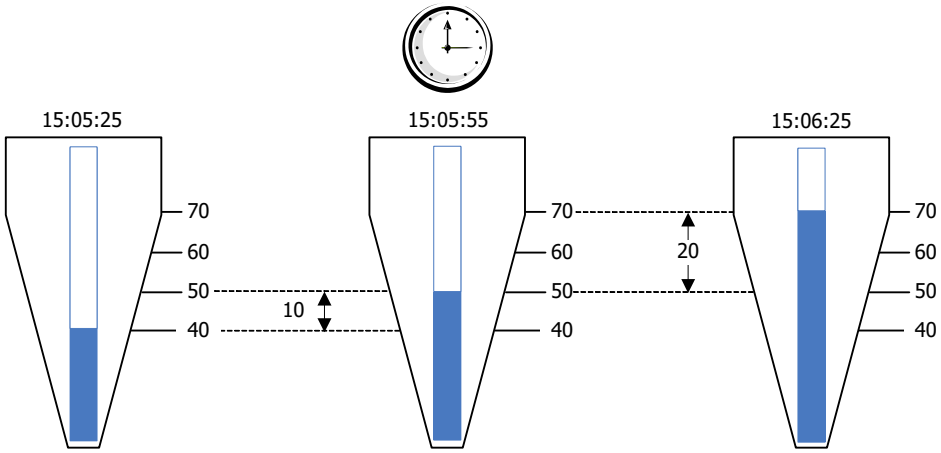
The timestamp when a deviation alarm becomes active or inactive is the most current timestamp of the corresponding input value. If there is no timestamp associated with the alarmed value, the AppEngine timestamp is used instead.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

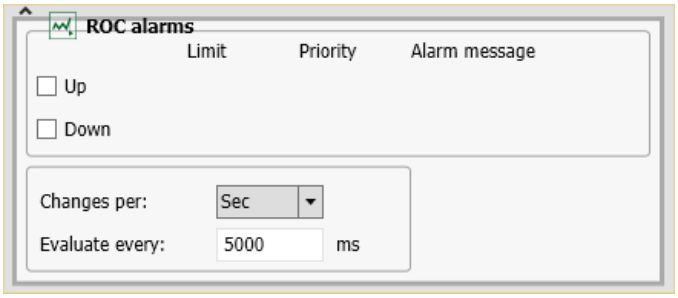
Rate of Change Alarms

A rate of change alarm identifies when an attribute value is changing too quickly over time. For example, you can set a rate of change alarm for a tank level that indicates when the pump inlet pressure is too high.

Rate of change is the calculated slope, which is the absolute difference between the current and previous attribute values divided by a specified interval. When the slope (positive or negative) exceeds a specified value, a rate of change alarm occurs. For example, if a tank volume increases from 17 to 45 liters over a 5 minute interval, the calculated slope is 5.6 liters per minute. If you set your rate of change alarm limit to 5.0 liters per minute, a rate of change alarm condition exists.



Alarm limits are expressed in the Engineering Units of the attribute's value over an interval, which can be per second, minute, hour, or day.



You can select and configure the value and priority for the upward and downward ROC limits. You can set individual messages for ROC alarms that exceed the upward or downward limits.

The timestamp when a rate of change alarm becomes active or inactive is the most current timestamp of the corresponding input value. If there is no timestamp associated with the alarmed value, the AppEngine timestamp is used instead.

Statistical Alarms

A statistical alarm is one in which a statistic is calculated, based upon an attribute. If the statistic exceeds some pre-set limit, the object flags the alarm condition as TRUE.

Setting Alarm State with Object Attributes

The Application Server alarm enable/disable mechanism includes four attributes to set an object alarm mode and report alarm status:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- AlarmModeCmd Attribute
- AlarmInhibit Attribute
- AlarmMode Attribute
- _AlarmModeEnum Attribute

AlarmModeCmd Attribute

AlarmModeCmd is a writable attribute that sets the current commanded alarm mode for the object. You set the AlarmModeCmd to enabled, silenced, or disabled with a script, user input, or from an attribute configured to read from an input source.

AlarmInhibit Attribute

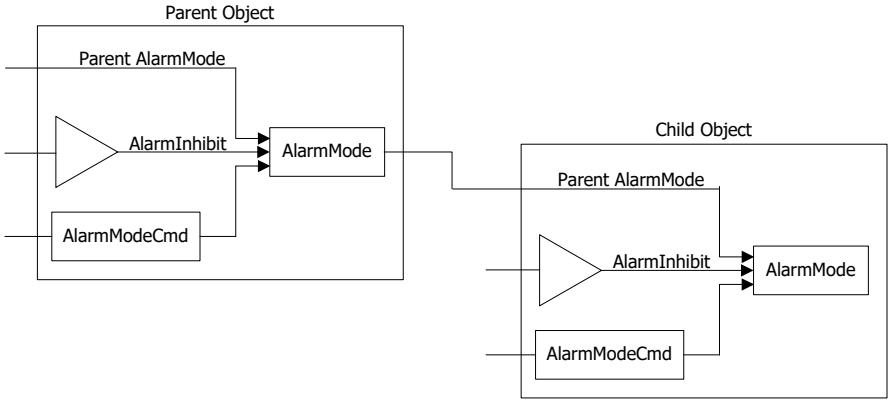
The AlarmInhibit attribute disables one or more alarms when set to TRUE. The value of the AlarmInhibit attribute is typically set by a script, manually by the user, or from an attribute configured with an input feature. If the AlarmInhibit attribute is set TRUE, all alarms of the object and of any contained objects are disabled.

When the AlarmInhibit attribute is set to FALSE, alarms are not inhibited and the object AlarmMode and parent object AlarmMode determine whether alarming is enabled, silenced, or disabled.

AlarmMode Attribute

The AlarmMode is a calculated attribute that identifies the object alarm mode and is based upon the current values of an object's:

- AlarmModeCmd attribute
- AlarmInhibit attribute
- Parent object AlarmMode attribute



Application Server checks the AlarmModeCmd and AlarmInhibit attributes of an object and the AlarmMode status of the parent object. Application Server then updates the object's AlarmMode attribute to reflect the most restrictive setting.

All individual alarms use the object's AlarmMode status to determine whether they are enabled, silenced, or disabled.

_AlarmModeEnum Attribute

The _AlarmModeEnum attribute can be used by any object to obtain an enumeration of permissible settings for the AlarmModeCmd attribute.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Alarms and Buffered Data

An object with alarms enabled detects the alarm condition and sets an associated alarm condition attribute to true. If the object detects the alarm condition based on an attribute with buffered data (HasBuffer property set to true), it enables buffer support for the condition attribute and sets the condition attribute's Buffer property to a VTQ buffer of true/false values representing alarm conditions.

Alarm Detection

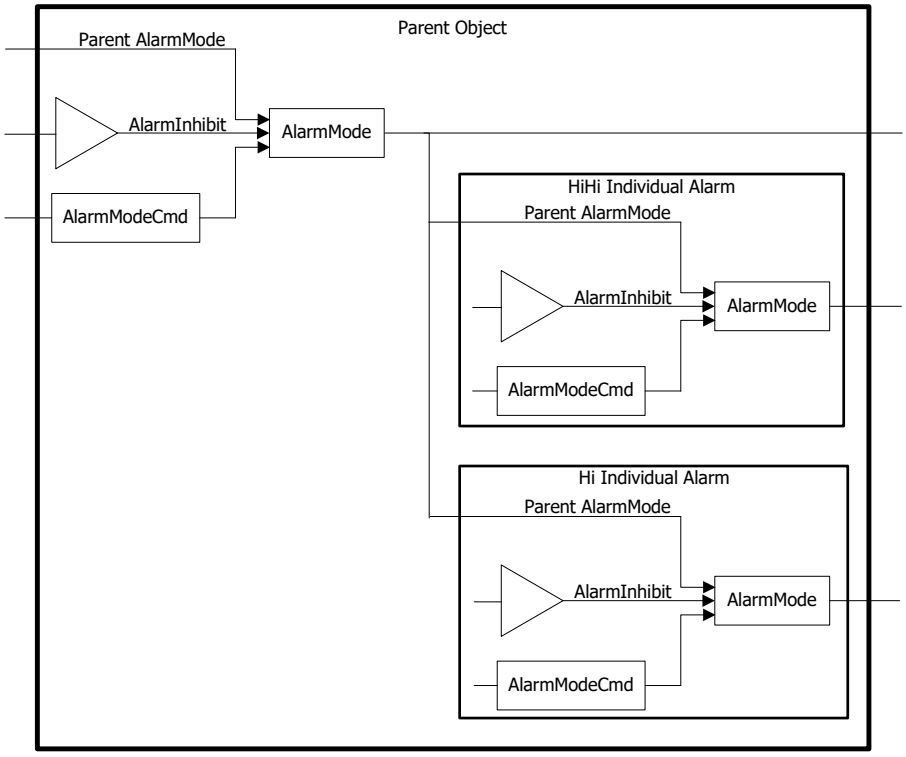
The Buffered Data feature will pass buffered alarm VTQs to the alarm subsystem in the same manner as with buffer data not enabled. When enabled, the Buffered Data feature can generate multiple alarm messages in the same scan.

For more information on alarm detection, see About Buffered Data and Alarms and Events.

Setting Alarm State for Individual Alarms

You can set individual alarms within an object for each type of alarm. For example, you can set alarms for each of the limits of a level alarm.

The following figure shows an object's individual alarms for the HiHi and Hi alarm limits.



The calculated AlarmMode attribute value of an individual alarm uses the same inputs as an object alarm. The parent AlarmMode attribute is from the object itself. As with object alarms, the individual alarm mode is set to the most restrictive input state. For example, if the object's AlarmMode state is disabled and the individual alarm's AlarmInhibit is FALSE, the individual alarm is disabled.

Each individual alarm is autonomous from other individual alarms in an object. The AlarmMode of an individual alarm is not propagated to other alarms. Unlike inhibit for the entire object, inhibit of an individual alarm does not affect the alarms of any contained objects. You can selectively enable, silence, or disable an individual alarm and not set other alarms to the same value within the object hierarchy.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Enabling, Silencing, and Disabling Alarms

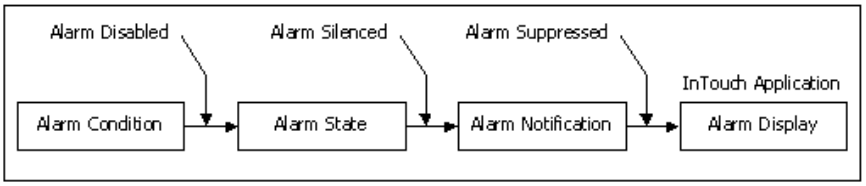
Alarms can be enabled, disabled, or silenced while an application is running. An object's alarm state can be set at the Area level, at the container object level, or at the individual object. In addition, individual alarms within a single object can be enabled, silenced, or disabled.

- Enabled: All alarms for an object are reported to client applications and saved as historical data. The enabled state is less restrictive than the silenced or disabled alarm states.
- Silenced: All alarms for an object are detected and logged in the Historian alarm and event history database. Silenced alarms are not logged to the InTouch alarm database, and are not shown in alarm clients displaying current alarms or recent history alarms. The silenced alarm state is more restrictive than the enabled state, but less restrictive than the disabled state.
- Disabled: No alarms for the object are detected. The alarm is return-to-normal until the alarm is re-enabled. The disabled state is more restrictive than the silenced and enabled alarm states. A disabled alarm does not require acknowledgement.

When an alarm state changes from silenced to enabled, the following applies:

- Only the last occurrence of the alarm appears in alarm clients.
- Only the last occurrence of the alarm is logged into the alarm database.
- Only the last occurrence of the alarm is saved to history.

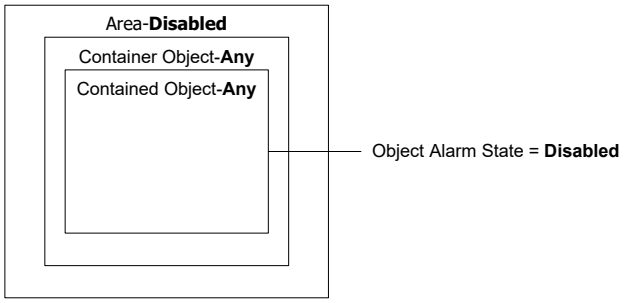
The following figure shows how the different alarm modes affect the different phases of an alarm. In the case of Alarm Suppressed, selected alarms can be filtered out of the InTouch AlarmViewer display by an operator, or programmatically by a script.



To ensure that alarmed attributes always have current data, the alarm feature always registers a reference to the alarmed attribute. This guarantees that Message Exchange never suspends updates for this attribute. Even if alarms are disabled for a particular attribute, the alarmed attribute cannot enter an Advanced Communication Management Suspended state.

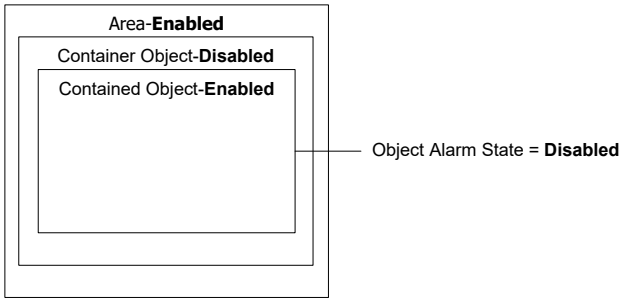
The object hierarchy and alarm states determine the final alarm condition of an object.

- An Area object's alarm state determines the alarm state for all alarms of objects that belong to the Area.

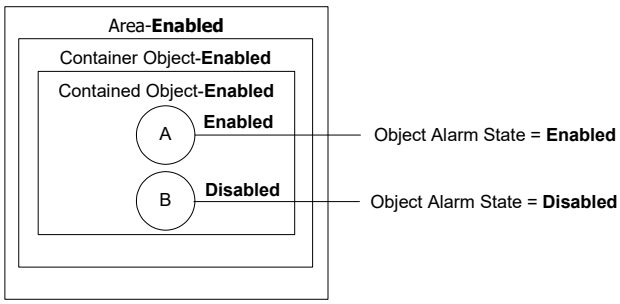


- The most restrictive setting within an object hierarchy determines the object's alarm state.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



- When an individual object alarm is silenced or disabled, it applies only to that alarm and not to other alarms belonging to the object.



  The alarms on any contained object are not affected. The disabled or silenced state of an individual alarm does not propagate downward through the object hierarchy to the alarms of any contained or assigned object.

Enabling Alarms

To enable an object's alarms, you must ensure that the AlarmModeCmd and AlarmInhibit attributes are enabled for the object, its container, and its area. An event, including the user's name, is generated indicating the object's alarms are enabled.

When object alarms are enabled, you can enable, silence, or disable an individual alarm.

Silencing Alarms

When object alarms are silenced, an individual alarm that is enabled or silenced is forced to be silenced. When object alarms are silenced, an individual alarm can be disabled.

Disabling Alarms

When object alarms are disabled, an individual alarm that is enabled or silenced is forced to be disabled.

When object alarms are enabled and an individual alarm is enabled or silenced, the individual alarm can be inhibited. This forces the individual alarm to be disabled.

When object alarms are silenced and an individual alarm is enabled or silenced, the individual alarm can be inhibited. This forces the individual alarm to be disabled.

When object alarms are inhibited, an individual alarm that is enabled or silenced is forced to be disabled.

Shelving Alarms

You can shelve alarms to temporarily hide them from displays for a fixed period. Alarms continue to be historized, even when they are shelved.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Shelving typically is used to reduce alarm "noise", or to temporarily suppress alarms that might be triggered during system modifications or repairs, allowing you to focus on correcting other more urgent alarms.

Shelving is similar to silencing an alarm, but shelved alarms differ from silenced alarms in the following ways:

- Shelved alarms have a built-in associated time-out period. Shelved alarms are automatically unshelved when the configured shelving period expires. You can also manually unshelve alarms and return them to an active, displayed state.
- Alarm shelving must be enabled at an area level, but shelving applies only to individual alarms. You cannot shelve a hierarchy of alarms for an entire area or for an entire object. You cannot propagate alarm shelving through the Model View hierarchy.
- The system enforces role-based limitations on permission to shelve alarms, alarm severity levels that can be shelved, and the total number of alarms a user can shelve.

The system tracks who shelved the alarm, from what workstation, the reason for shelving the alarm, when shelving began, and when shelving will expire. Shelved alarms aggregate in similar fashion to silenced alarms. For information about alarm aggregation, see Using Aggregated Alarm Information.

A set of seven attributes provide run-time alarm shelving information and control:

AlarmShelveCmd	User writeable. Use this attribute to shelve or unshelve an alarm. Default values: Duration = 0, Reason = ""
AlarmShelved	Read-only, Boolean value. Shows True if alarm is shelved, False if alarm is unshelved. Default value: False
AlarmShelveStartTime	A read-only date/time stamp indicating when alarm shelving began, based on the engine time when the shelving request was received, Default value: Blank
AlarmShelveStopTime	A read-only date/time stamp equal to the AlarmShelveStartTime plus the duration for which the alarm is to be shelved. Default value: Blank
AlarmShelveReason	A read-only string value providing the reason for which the alarm was shelved or un-shelved by the Alarm Shelve command. Default value: Blank (See AlarmShelveCmd attribute.)
AlarmShelveUser	Read-only, the name of the user who most recently shelved or un-shelved the alarm with the Alarm Shelve command. Default value: Blank

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

AlarmShelveNode	Read-only, the name of the computer node from which the user most recently shelved or un-shelved the alarm with the Alarm Shelve command. If the node is hosted in a Terminal Server client session, the node and the TSE ID are both identified. Default value: Blank

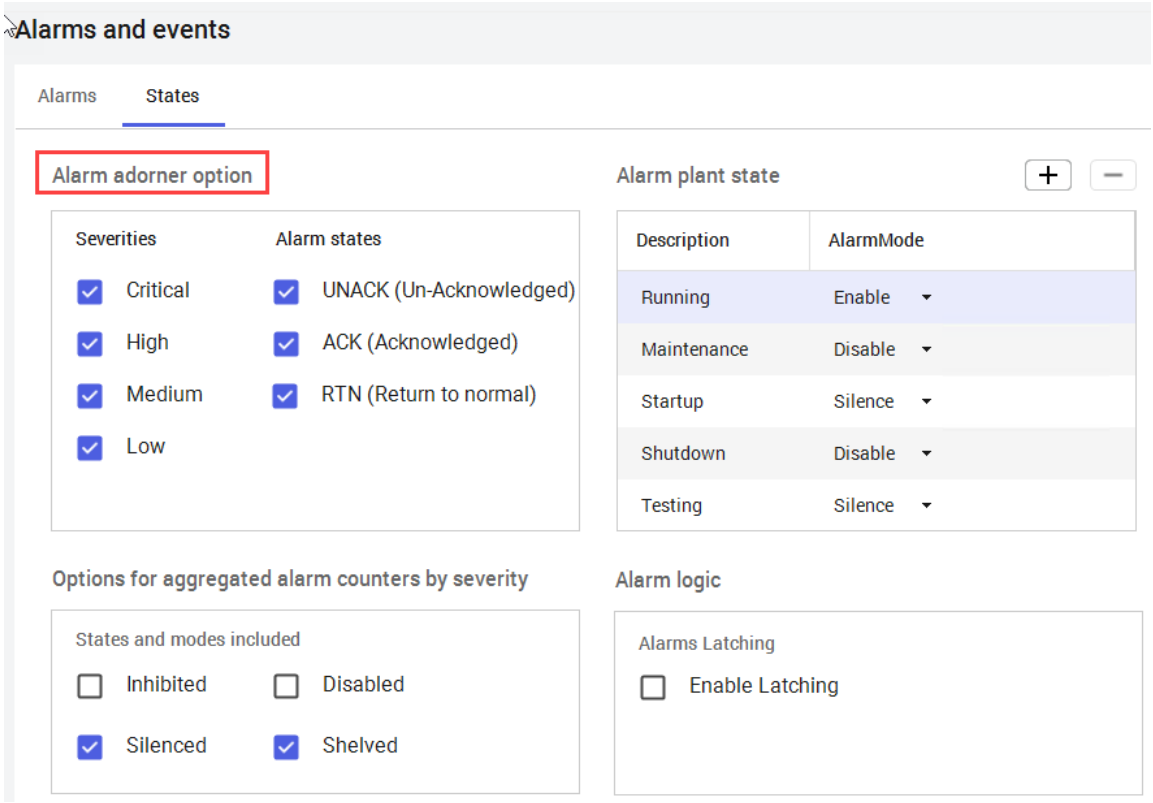
Enabling Alarm Shelving

Alarm shelving is configured from the IDE **Configuration** menu, and is enabled on the Area object.

To enable alarm shelving

1. Enable at least one security role to configure alarm shelving. Associate the relevant user with that role, if not already associated, before proceeding to step 2.
2. From the ribbon, select **Galaxy**, then select **Configure**.
3. Select **Galaxy**, then select **Alarms**.

   The **Alarms** page appears.



4. Select the severity level in the **Shelve** column. By default, severity levels 3 (medium) and 4 (low) are enabled.
5. In the IDE, open the relevant Area object editor. Click the **Shelve Alarms** checkbox to enable shelving for that area. Shelving is enabled by default.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Shelving Alarms at Run Time

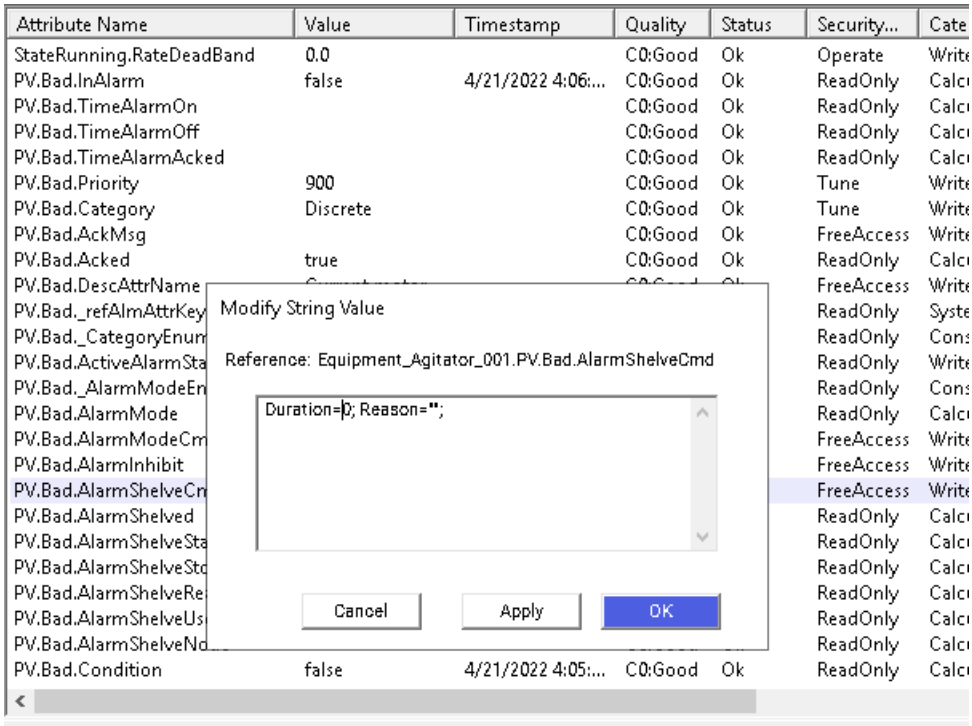
Use a run-time client to shelve and unshelve alarms. From Application Server, you can use Object Viewer to monitor and control shelved alarm attributes. Shelved alarms can also be monitored and controlled from an embedded Alarm Control and at least one animation to write to the AlarmShelveCmd attribute. For information about using the Alarm Control to shelve alarms, see Shelve and Unshelve Alarms at Run Time in the AVEVA OMI help.

To shelve or unshelve an alarm in Object Viewer

1. In Object Viewer, select the object in the **Console Tree** (left pane) which contains the alarm you want to shelve. The object attributes display in the **Details Pane** (right pane).
2. In the Attributes list, find the attribute configured with the alarm you want to shelve.

   For example, "Inlet_001.Discrete_001".
3. In the Attributes list, below the attribute in step 2, find the **AlarmShelveCmd** attribute.

   For example, "Inlet_001.Discrete.001.AlarmShelveCmd".
4. Double-click the .AlarmShelveCmd attribute. The **Modify String Value** dialog opens.



   a. To shelve an alarm, enter a duration in hours or in decimal fractions if the duration is to be less than one hour. The duration cannot be blank.

   b. Enter a reason between the quotation marks for shelving or unshelving. The reason cannot be blank. Click **OK**.

   c. To unshelve a shelved alarm, enter a value of "0".

The six read-only alarm shelving attributes will display the current shelving status information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Throttling Alarms

Alarm throttling prevents network and message queues from flooding during periods of high alarm activity. Throttling sets a maximum number of transitions into and out of an alarm state within a defined period. Alarm transitions that exceed the throttling limit are not reported.

The WinPlatform object includes tuning parameters that specify the maximum alarm rate per second on an engine. The Alarm throttle limit attribute must be used in conjunction with the scan period to determine the maximum number of alarms that can occur in a scan cycle. You can set the alarm throttle limit when you configure the WinPlatform object. For more information about setting the scan period and alarm throttle limit, see Configuring WinPlatform Object Alarms.

Alarm messages can be throttled for alarms going into alarm and out of alarm state. Alarm acknowledgement messages are not throttled. Users can still acknowledge alarms even when alarm throttling is active. Users can still disable alarms for objects or areas when the alarm rate causes throttling to occur. Also, the alarm inhibit and the disable/enable/silence messages are not throttled.

If an active alarm is disabled, the going out of alarm and disabled messages are sent to the notification distributor. If alarms are being throttled, the going out of alarm message can be throttled. The disabled message is accepted regardless of the throttling status. The going out of alarm message is not raised again. This can result in never logging a message for that alarm indicating it went out of alarm. The final alarm list in the Notification Distributor still shows the correct alarm state.

Propagating Timestamps with Alarms and Events

An alarm feature always registers a reference to the alarmed attribute. This registered reference guarantees that Message Exchange never suspends updates for the alarmed attribute. Even if alarms are disabled for a particular attribute, the Advanced Communication Management feature cannot suspend the attribute.

Be aware that the time stamp propagates to all masked bits of an integer attribute, even if only one of the bits changes.

For example, you have an Integer address in a PLC that represents 16 different alarm states. You assign ObjA.Attr_Integer to point to the PLC address. You then split the bits to different alarm attributes, adding one attribute for each alarm and naming them ObjA.Attr_Alarm00 to ObjA.Attr_Alarm15. Each attribute has an input source that refers to a different bit of ObjA.Attr_Integer. For example, ObjA.Attr_00.InputSource -> me.Attr_Int.00, and so on. At run time, when bit 00 is changing in the PLC, all of the attributes (ObjA.Attr_Alarm00 to ObjA.Attr_Alarm15) get a new time stamp, as all the bits changed. This can cause incorrect time stamps for alarms.

Alarms or Events Become Active

For alarms or events, if an attribute has no timestamp, the current AppEngine scan time is used instead. If an attribute has a timestamp, the timestamp of the value that caused the alarm to occur is used as the value for the object's AlarmOnTime attribute.

When an alarm is silenced or disabled, and an alarm condition becomes TRUE, when the alarm is later enabled, the timestamp for AlarmOnTime is the timestamp of the most recent attribute change, not the time at which the alarm was enabled.

For an event, if an attribute has a timestamp, the timestamp of the value that caused the event to be reported is used for the EventTime. A System Event timestamp is the current system time.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Alarms Become Disabled

If an active alarm becomes disabled, the alarm is forced to return to normal. The timestamp corresponds to the current time when the alarm became disabled, not the timestamp of the attribute nor of the alarm condition. Otherwise, the assigned timestamp is the AppEngine scan time.

Alarms Revert to Normal

If an attribute value has a timestamp, the timestamp of the value that caused the alarm to revert to normal is used for the AlarmOffTime.

If alarm is based upon several attributes or upon several values of a single attribute, the most recent timestamp is used when assigning values to the AlarmOnTime or AlarmOffTime.

Alarm acknowledgement

Alarms are acknowledged by users who view unacknowledged alarms from their client applications like InTouch HMI or AVEVA OMI. Only alarms of certain priority levels need to be acknowledged.

The basic workflow to acknowledge an alarm consists of the following general steps:

- The alarm is detected and reported to any subscribed alarm clients. The alarm is unacknowledged unless it is of a priority level that does not require acknowledgement.
- An authorized user attempts to acknowledge the alarm from the client. The user can type an optional comment when acknowledging the alarm.
- The user's acknowledge request is sent to the detecting object's alarm feature. The acknowledgment must pass through standard security checks first. The acknowledge request contains the user's name and any alarm comment.

For the alarm utility, the alarm is acknowledged within the alarm feature immediately. The user name, comment, and acknowledged time are also saved. Alarm comments can be localized into any supported language.

The acknowledge is considered an alarm state change, which is sent to all subscribed clients. When an alarm is acknowledged, the current AppEngine timestamp is used as the acknowledgement time.

After an alarm is acknowledged for the first time, any additional (extra) acknowledgement attempts for the same alarm are rejected and an error is returned.

Acknowledging Alarms with Signature Required

You can use the SignedAlarmAck( ) script function in Industrial Graphics to require a signature to acknowledge alarms. See Signature Security for Acknowledging Alarms in the *Creating and Managing Industrial Graphics User Guide* for information about adding the script function to a graphic. Then, embed the graphic in the WindowMaker window or an AVEVA OMI layout.

When you acknowledge the alarms at run time, the script function checks for a required signature. If a signature is required, it checks Galaxy security settings and alarm priority. If any alarm in the list falls within the configured conditions, you must provide your signature to acknowledge the alarms. You must sign in to acknowledge the alarms if no user is logged on to the InTouch or OMI application.

If any alarm in the list is requires a signature, then a signature is required for the entire list of alarms.

Note: The SignedAlarmAck( ) script function supports only System Platform alarms.

By inspecting the Alarm Viewer or the Alarm Control client, you can determine if the alarm acknowledgements were successful.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configuring Alarms

Alarming capabilities are a part of object templates, but they are not implemented until you configure the object in the IDE. After alarms are configured, you can view Application Server alarms in a client visualization application like InTouch HMI or the AVEVA OMI Alarm App.

You can add latched alarms as an additional alarm state for the Galaxy to provide an extra level of visibility and oversight to alarm conditions. See Alarm Latching for more information.

Configuring an object to be an alarm provider includes the following general sequence of steps:

1. Decide whether alarm notification is needed for each possible alarm condition of an object. For example, a command time-out alarm for a valve if the output command fails to move the valve.
2. Edit the object and set an attribute that specifies alarming.
3. Edit the object from the IDE and assign values to alarm attributes.
4. Configure the alarm properties. Typically, the fields that require configuration are Category, Priority and Description.
5. Configuring any limit fields to set an alarm. For example, the feedback time-out time limit.

You can add alarm detection and reporting capabilities to objects that were not originally developed to detect alarms. You do this by configuring alarm features for the object's attributes.

Configuring Alarming for System Objects

To enable alarming for your application, you need to configure your Galaxy's WinPlatform and AppEngine objects as alarm providers. Both system objects report their own alarms.

Client applications subscribe to application object alarms by the area containing the objects. Client applications can also subscribe to WinPlatforms and AppEngines directly. These are called "pseudo-areas." They do not need to be assigned to an area for a client to see the alarms, although the user may want to assign them to an area, such as for simplifying the alarm query in the InTouch Alarm Viewer or Alarm DB Logger.

The following list shows the alarms for each system object.

- WinPlatform
  - Excess CPU load alarm
  - Low disk space alarm
  - Excessive page faults alarm
  - Low memory alarm
  - Engine failure alarm
  - Engine checkpoint failure alarm
  - Object quarantined condition
  - Subscription folding condition
  - Scheduler scan overrun condition

  A WinPlatform object includes a general communication alarm when it loses contact with the areas to which it is subscribed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- AppEngine
  - Checkpoint failure alarm
  - Object quarantined condition
  - Subscription folding condition
  - Scheduler scan overrun condition
  - Redundancy failover alarm
  - Redundancy Standby unavailable
  - Redundancy Standby not ready
- ViewEngine
  - Checkpoint failure alarm
  - Object quarantined condition
  - Subscription folding condition
  - Scheduler scan overrun condition

WebViewEngine, Area and InTouchViewApp system objects do not include any alarms.

Configuring WinPlatform Object Alarms

You select the areas of your Galaxy to monitor for alarms from the WinPlatform object. Also, you can select specific alarms for the status of the WinPlatform itself.

You must specify that the WinPlatform object is an InTouch alarm provider to subscribe to alarms from the various areas (and pseudo-areas) of the Galaxy and report them to the InTouch Alarm Manager.

Configuring Communication Failure Alarm Priority

InTouch Alarm Provider reports any communication outage to the InTouch Alarm Manager. Communication alarms are displayed and must be acknowledged. By default, the priority of a communication failure is "1", the highest priority.

There can be instances when a communication failure is not a high-priority alarm, and should not require the attention of a high-priority alarm. Shutting down or undeploying a platform as a normal operation can result in communication failure alarms, for example.

To indicate whether the alarm is of high or low importance, the communication failure alarm priority is configurable at design time as an integer of range 1–999. InTouch Alarm Provider must be enabled in the WinPlatform object editor, **General** tab.

Selecting the Register using "Galaxy_<GalaxyName>" instead of "Galaxy" option

When you select the **Register using "Galaxy_<Galaxy name>" instead of "Galaxy" option**, you enable ITAlarmProvider contained in WinPlatform to register the ITAlarmSubsystem using a provider name of Galaxy_GalaxyName.

For example, if your galaxy is named Andromeda, the ITAlarmSubsystem registers as Galaxy_Andromeda.

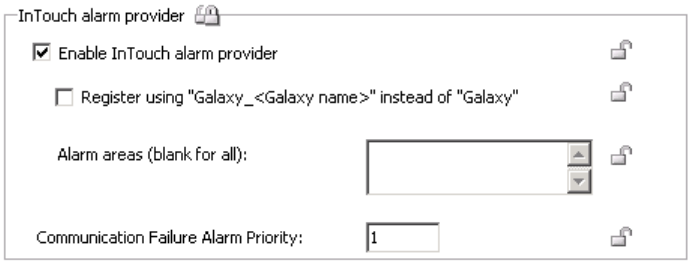
This configuration can be useful if you are using an Alarm Client to display alarms from multiple galaxies. For more information about multiple galaxies, see Working with Multiple Galaxies.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- This means that all InTouch applications which query alarms from this platform should be modified to use Galaxy_GalaxyName!AreaName.
- Select this option when the Alarm Control, the embedded alarm client (EAC), or the AlarmViewControl in InTouch queries alarms from different galaxies.

To configure a WinPlatform object to be an alarm provider

1. Open the WinPlatform object in the Object Editor.
2. Click the **General** tab.



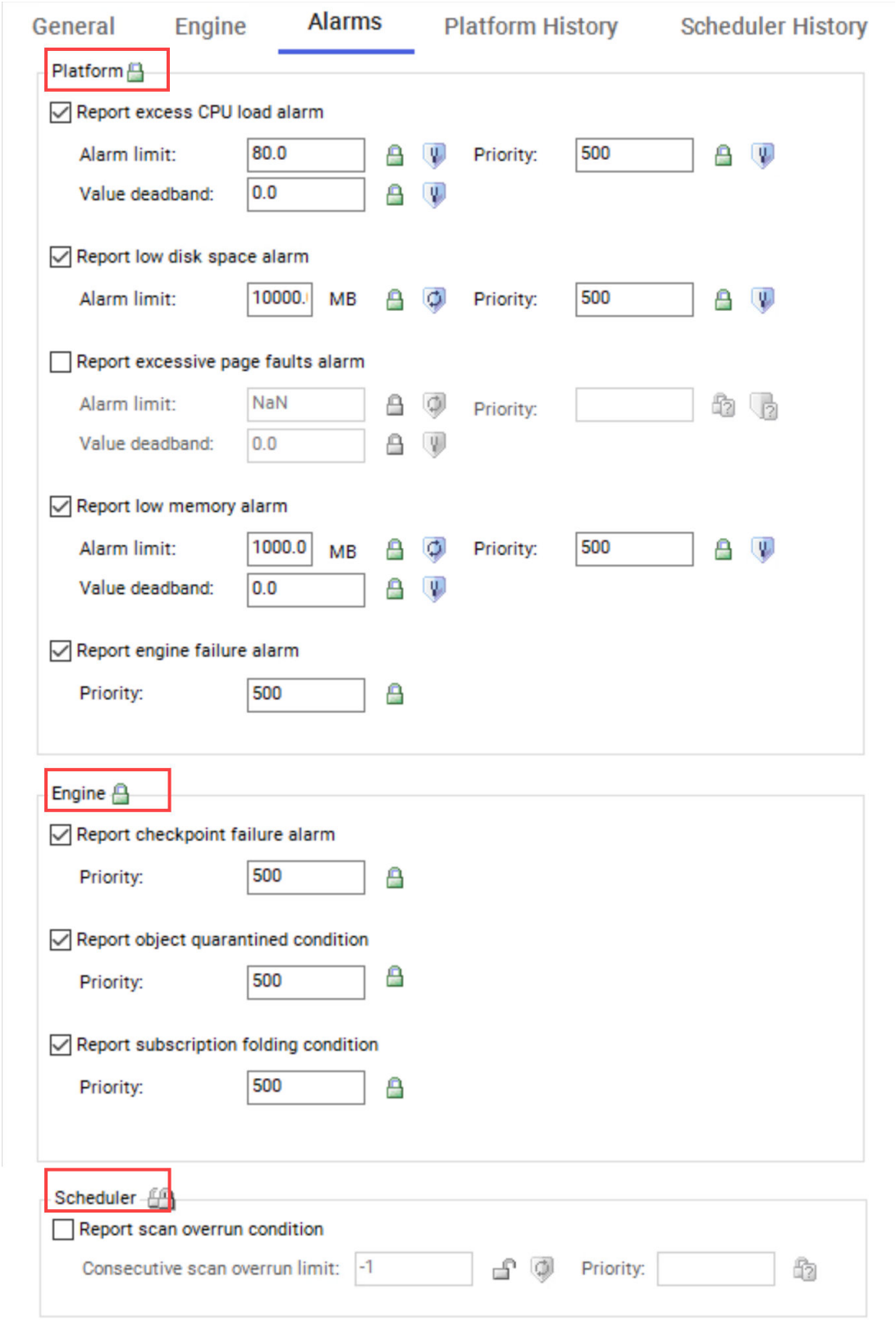
3. Select the **InTouch alarm provider** check box.
4. Select the **Register using "Galaxy_<GalaxyName>" instead of "Galaxy"** check box to enable Galaxy_<GalaxyName> registration for alarm comment language switching. An information box appears. Click **OK** on the information box to continue.
5. In the **Alarm Areas** box, type the names of areas to subscribe to for alarms.

   If you leave the **Alarm Areas** box blank, the WinPlatform subscribes to all areas in the Galaxy.

   If you want to subscribe to only selected areas within the Galaxy, insert a space between each area name. For example:

   Area1 Area2 Area3
6. Enter a number from 1 to 999 to configure the **Communication Failure Alarm Priority**.
7. Click the **Engine** tab to show the **Alarm throttle limit** box. Either accept the default throttle limit of 2000 alarms per second, or enter another value. A value of 0 disables alarm throttling. For more information about alarm throttling, see Throttling Alarms.
8. Click the **Alarms** tab to show platform, engine, and scheduler alarms that can be set for the WinPlatform object.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



9. Select the check box next to each alarm that you want to enable for the WinPlatform object.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

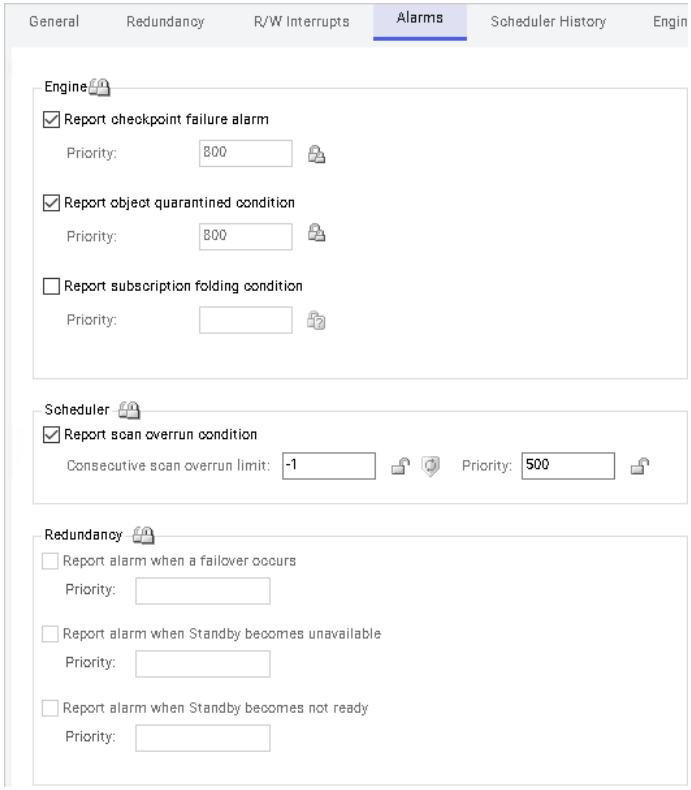
10. Set the limit, value deadband, and priority for each alarm you selected.
11. Save and close the Object Editor.
12. Check the object in to the Galaxy.
13. Deploy the object in an on scan state.

Configuring Alarms for an AppEngine Object

You can set AppEngine attributes that determine whether alarms are enabled for an engine, scheduler, and redundant failover engine.

To configure AppEngine object alarms

1. Open the AppEngine object in the Object Editor.
2. Click the **Alarms** tab to show engine, scheduler, and redundancy alarms that can be set for the AppEngine object.



3. Select the check box next to each alarm that you want to enable for the AppEngine object.
4. Set the priority for each alarm you selected.
5. Save and close the Object Editor.
6. Check the object in to the Galaxy.
7. Deploy the object in an on scan state.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configuring Alarms and Events for Application Objects

The following table shows the different types of alarms that can be specified for application objects. The table shows the application objects containing native alarm attributes.

You can also set alarms for an object's attributes. The types of alarms available depend on the attribute type. For more information about setting alarms for attributes, see Setting Alarms on the Attributes Page.

Application Object	**Alarm Types**				
	State	Limit	Target Deviation	Rate of Change	Statistics
AnalogDevice	X	X	X	X	
DiscreteDevice	X				X
FieldReference					
Sequencer	X				
SQLData					
UserDefined	X	X	X	X	

The following list shows the types of alarms for each application object in more detail.

- AnalogDevice
  - Level alarms (HiHi, Hi, Lo, LoLo) [limit alarms]
  - Rate of Change alarms (Up, Down)
  - Target Deviation alarms (Minor, Major)
  - PV Bad Value alarm [state alarm]
- DiscreteDevice
  - Uncommanded change alarm [state alarm]
  - Command time-out alarm [state alarm]
  - Active1 state alarm [state alarm]
  - Active2 state alarm [state alarm]
  - Fault state alarm [state alarm]
  - Active1 state duration alarm [statistical alarm]
  - Active2 state duration alarm [statistical alarm]
- Sequencer
  - Execution halted [state alarm]
  - Condition trigger failure [state alarm]
  - OnEntry output failure [state alarm]
  - OnExit output failure [state alarm]

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- UserDefined (Attributes can be alarmed)
  - PV State alarm [state alarm]
  - PV Bad Value alarm (that is, bad quality) [state alarm]
  - Attribute alarm features

    State alarms

    Limit alarms (HiHi, Hi, Lo, LoLo)

    Rate of Change alarms (Up, Down)

    Deviation alarms (Minor, Major)

    Bad Value alarm (that is, bad quality)

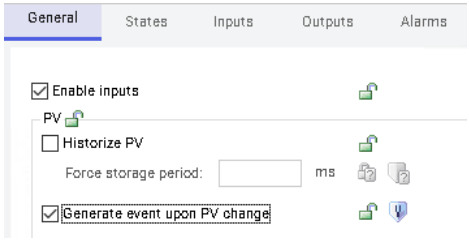
There are no built-in alarms for these application objects:

- FieldReference
- SQLData

You can also configure your application objects to generate an event each time the object's PV value changes.As mentioned above, you can also configure an alarm on any object for any attribute. See Setting Alarms on the Attributes Page for more information.

To configure alarming and events for application objects

1. Open the application object with the Object Editor.
2. For a DiscreteDevice, click the **General** tab, enable inputs, and then enable **Generate event upon PV**.



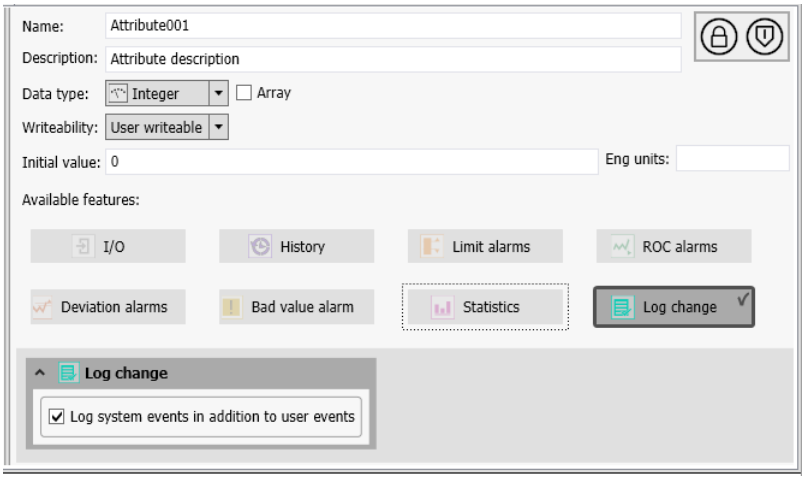
   For objects that do not include this check box, use the **Attributes** page.

   Note: If you are using field attributes instead of Attributes, use the **Field Attributes** page.

   a. Enable the **Log change** feature.

   b. Enable **Generate event upon change**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. Select or clear the check box based on whether you want to generate an event each time the object's PV value changes.
4. Click the tab that lists alarm attributes.
   - For the AnalogDevice object, click **Alarms**.
   - For the DiscreteDevice object, click **Alarms**.
   - For the Sequencer object, click **Settings**.
   - For the UserDefined object, click **Attributes** (or **Field Attributes**, if you are using field attributes). See Setting Alarms on the Attributes Page for further directions.
5. Select the check box that enables alarming for the object.
6. Assign values to the attributes for the type of alarm you selected by completing the following steps:
   a. Assign values to the alarm limits based on the type of alarm.
   b. Assign an alarm priority (1-999) for each limit you set.
   c. Accept the default alarm message or include another message for each alarm limit.
   d. Assign values to the remaining attributes based on the type of alarm you selected. For more information about other alarm attributes, see Types of alarms.
7. Save your object changes and close the Object Editor.

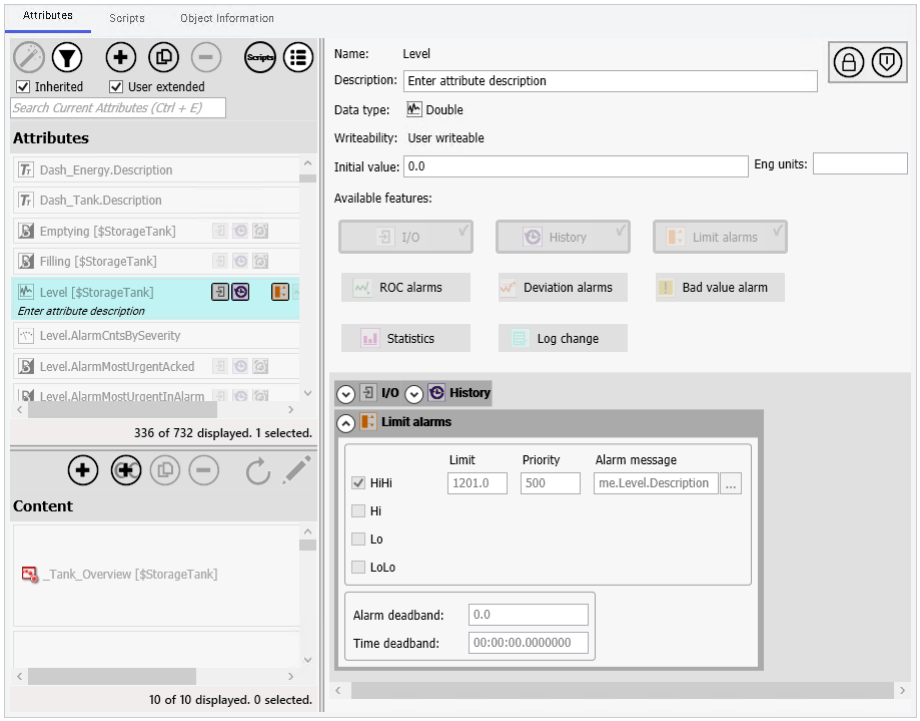
Setting Alarms on the Attributes Page

You set alarms for all Application Server objects in the Attributes page in the Object Editor.

To specify alarms for an object's attributes

1. On the **Attributes** page of the Object Editor, select an attribute from the **Attributes List**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. Select the alarm features you want to configure for the attribute. For Boolean data types, you can select:
   - **I/O Alarm**
   - **State alarm**
   - **Bad value alarm**

   For Integer, Float and Double data types, you can select:
   - **Limit alarms**
   - **ROC alarms**
   - **Deviation alarms**
   - **Bad value alarms**

   Note: Depending on the object type, some of these alarm types may not be available.

3. For each alarm feature that you activate, enter the settings for each alarm.
   a. Assign values to the alarm limits based on the type of alarm.
   b. Assign an alarm priority (1-999) for each limit you set.
   c. Accept the default alarm message or include another message for each alarm limit.
   d. Assign values to the remaining attributes based on the type of alarm you selected. For more information about other alarm attributes, see Types of alarms.
4. To configure event logging for an object, select the **Log change** feature and click the checkbox **Log system events in addition to user events**.
5. Save your object changes and close the Object Editor.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Distributing Alarms and Events

After you configure object instances for alarm detection, deploy the instances and put them On scan. The instances begin checking for alarm conditions.

When an alarm is detected, or an event occurs, a notification is reported to its alarm and event distributor, which is running on the same AppEngine.

These alarm and event distributors include:

- **Area objects**: Area objects report detected alarms through the Area, which distributes them to alarm and event clients.
- **WinPlatform objects**: Report their own alarms and events.
- **AppEngine objects**: Report their own alarms and events.
- **Device Integration objects**: Report their own alarms and events

The Area object plays a key role in alarm and event distribution. All objects belong to an Area. Areas can contain sub-Areas. Alarm and event clients are configured to subscribe to a set of Areas.

Areas provide a key organizational role in grouping alarm information and assigning it to users who use alarm and event clients to monitor their Areas of responsibility.

WinPlatforms, AppEngines and Device Integration objects do not report their alarms and events to Area objects even though they belong to Areas. This allows alarm clients to receive alarm notifications without any dependencies on Area objects. For example, a deployed and running WinPlatform can report alarms even though its Area is not deployed and running.

Alarm-event distributor objects maintain a list of all currently active alarms and inactive but unacknowledged alarms. They do not maintain a list of events, which are routed to clients that are currently subscribed at the time of the event.

You can configure a WinPlatform to act as an InTouch Alarm Provider in the run-time environment.

The WinPlatform sends an alarm through the InTouch Distributed Alarm System to InTouch clients when the WinPlatform loses communication with an Area that it subscribes to. This condition typically occurs during a network outage with computers hosting those Areas.

In a network outage, the WinPlatform InTouch Alarm Provider sends an alarm for each disconnected Area that it subscribes to, including all of its alarm distribution hierarchy. Each of these alarms is a high priority alarm that contains the name of the Area to which communication is lost. These communication problem alarms must be acknowledged.

Although they still appear in the historical record, any current alarms from the disconnected Area drop from the InTouch client's summary list. They can no longer be acknowledged.

When communication to the disconnected Areas is restored, any unacknowledged alarms generated in those Areas are sent to the alarm client.

Subscribing to Alarms and Events from a Client

Clients indicate interest in alarms and events by subscribing to an Area. When subscribing to an Area, the subscription is actually to all notification distributors within that Area.

For example, if an Area contains sub-Areas, those sub-Areas are subscribed to. If WinPlatforms, AppEngines or Device Integration objects belong to an Area, those objects are also directly subscribed to.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When a notification distributor receives an alarm and event subscription from a client, the notification distributor provides the client with the following:

- A list of all current alarm conditions, including unacknowledged return-to-normal conditions.
- An alarm condition state change. A state change includes transitions into or out of alarm (return to normal) and change in acknowledged flag.
- An event occurrence.

Alarm and event subscription requests do not include filters, for example, only show alarms greater than a specific priority value. All alarm and event messages received by the notification distributor are sent to all subscribed clients. Filtering is provided as a display option by clients.

Using InTouch HMI as the Alarm and Event Client

InTouch run-time clients subscribe to event reports from a Galaxy. Application Server reports alarms to the InTouch Distributed Alarm System, which subscribeS to alarm and event reports from a Galaxy.

An InTouch client application can visualize Application Server components. An InTouch alarm client can show alarm information for new, unacknowledged alarms, including all required fields.

The new alarm is in the unacknowledged state. An operator can view alarms, acknowledge alarms, disable alarms, and enable alarms from the client application running in InTouch WindowViewer.

Understanding the Syntax of Alarm Queries

InTouch alarm queries subscribe to alarm and event information from objects within a Galaxy. The alarm and event queries can be in the form of user input or a script.

Alarm query syntax must be in one of the following forms:

\Provider!Area

or

\\Node\Provider!Area

You can have one or more references in a query separated by spaces.

You can also optionally append a tagname filter at the end, separated by another exclamation mark:

\Provider!Area!Filter

\\Node\Provider!Area!Filter

The filter can have a wildcard * character at the beginning or at the end, but not both.

The \\Node at the beginning is only important if you want to query for alarms from a provider on another computer. Otherwise, you can leave it off and the reference is assumed to be a provider on the local computer. The provider name Galaxy refers to alarms and events that get reported by the WinPlatform configured as an InTouch alarm provider on that computer node.

Alarm Query Syntax when Register Using Galaxy_<GalaxyName> is Enabled

In the WinPlatform object, when you enable InTouch alarm provider, you can enable **Register using Galaxy_<GalaxyName> instead of Galaxy**. This option will register the platform to the alarm subsystem using the Galaxy name prefixed by "Galaxy_" instead of just the word "Galaxy". This allows an InTouch application to monitor alarms from multiple Galaxies and avoid name conflicts.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Syntax changes slightly when Galaxy_GalaxyName is enabled:

- Use \\ for computer name.
- Use \ for Galaxy or Galaxy_<GalaxyName>.
- Use ! for Area.

For example: \\Galaxy\MyGalaxy!Area001.

If Galaxy_GalaxyName is not enabled in WinPlatform, then the default behavior described in the previous section applies.

You can determine if Galaxy_<GalaxyName> has been enabled by monitoring the run-time attribute of the platform ITAlarmProvider.ProviderNameAsGalaxyNameEnabled.

Examples of Alarm Queries

- You can submit a query to get all alarms from Area1 and all other alarms within Area1, as reported by the WinPlatform object on the local computer.
  ```
  \Galaxy!Area1
  ```
  The query returns all alarms and events from all objects directly contained in Area1 and any sub-areas contained by Area1. This hierarchy is determined by what is configured in the Model View in the IDE.
- If Area1 and Area2 are two separate mutually exclusive areas, you can submit a query for alarms from both areas.
  ```
  \Galaxy!Area1 \Galaxy!Area2
  ```
- If you're on NodeA and the WinPlatform is on NodeB, you can submit a query for the alarms from the remote computer.
  ```
  \\NodeB\Galaxy!Area1
  ```
- You can submit a query for all alarms from objects whose name begins with "Tank" in the TankFarm1 area.
  ```
  \Galaxy!TankFarm1!Tank*
  ```
  The trailing wildcard character matches alarms from all objects with names that begin with "Tank" like Tank001, Tank002, TankUpper, or TankLower.
- You can submit a query for specific alarm types. For example, you can submit a query for all HiHi alarms in the TankFarm1 area.
  ```
  \Galaxy!TankFarm1!*.HiHi
  ```
- You can submit a query for all types of alarms from a specific object within an area.
  ```
  \Galaxy!TankFarm1!Tank752.*
  ```
  The trailing wildcard character matches all alarm types for Tank752.

Alarm Requirements for InTouch Client Applications

For Application Server alarming to function, the following conditions must be met:

In Application Server:

- One or more Area objects are deployed and running.
- The source object is on scan.
- The source object’s Area is on scan.
- Alarming must be enabled for the target object.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- An InTouch alarm provider on any WinPlatform in the Galaxy.

In InTouch:

- The InTouch client application is running in WindowViewer.
- An InTouch alarm ActiveX control is placed in a window and configured as an alarm consumer for the Galaxy.
- The user has logged into InTouch with security enabled and is authorized to acknowledge alarms for the object that is in the alarm state. If the user only wants to view alarms, security authorization is not required.

Application Server validates the user has sufficient security privileges to acknowledge the alarm.

If the user does not have privileges to acknowledge alarms, the user can attempt to acknowledge the alarm, but the Galaxy rejects the acknowledgment request.The alarm remains unacknowledged in the InTouch Alarm display.

The rejected alarm acknowledge event is recorded in InTouch Event History if the user attempting the acknowledgement has a valid Galaxy user account. Otherwise, the rejected acknowledgement is not recorded as an event.

You can acknowledge multiple alarms by providing a valid signature. At run time, the Industrial Graphics SignedAlarmAck() script function checks a set of alarms and conditions to find out if any alarm from the list requires a signature in order to acknowledge it. If so, you must provide your user name, password and domain or your Smart Card details and PIN to authenticate yourself and acknowledge the alarms. To be able to provide the Smart Card details, your computer must be configured for Smart Card authentication.

Note: Smart Card authentication is not supported in multi-galaxy environments for read/write operations to remote galaxies.

If even one alarm in the list requires a signature, you must provide a signature to acknowledge the alarms. The SignedAlarmAck() function will acknowledge ALL the alarms in the list.

With the SignedAlarmAck( ) function, you can provide credentials and acknowledge an alarm or group of alarms even if you are not the logged-on operator.

Alarms and Event Implemenation in Application Server and Run-Time Applications

Application Server and its run-time applications (AVEVA OMI and InTouch HMI) implement alarms and events. The following table shows the similarities and differences between the products.

Item	InTouch	Application Server/OMI
Alarm configured or detected by	Within a tag	Within an object
Alarm Classes (client column)	Only certain classes of alarms are supported or detected: DSC, VALUE, DEV, ROC, SPC.	No system-wide distinction for classes. Alarms are tied to a Boolean that can be triggered from any logic.
Alarm Type (Sub-class) (client column)	Discrete, LoLo, Lo, Hi, HiHi, MinorDev, MajorDev, ROC, SPC. Client column.	No sub-class. The Alarm feature name is the closest concept. For example, ".PVHiAlarm". Mapped from Category.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Item	InTouch	Application Server/OMI
Priority (client column)	1-999 (1 most urgent)	Priority 0-999. 0 most urgent. 0 is mapped to 1 in InTouch.
Name (client column)	Alarm name = Tag name.	Object.attribute
Comment (client comment)	Separate alarm comment, which is different from the tag comment.	Object short description or alarm message where available.
Group	Alarm group allows client-side filtering. Sub-groups must be on same InTouch.	No alarm group. But Area provides mappable concept. Sub-Areas can be on different nodes.
State	No behavioral differences between Application Server/OMI and InTouch HMI. Four states, which are combinations of ACK/UNACK and ALARM/RTN. Optionally, a fifth state, Latched, can be enabled for the Galaxy. • UNACK/ALARM (usually displayed as UNACK) • ACK/ALARM (usually displayed as ACK) • UNACK/RTN (usually displayed as UNACK_RTN) • ACK/RTN (usually displayed as ACK_RTN) • LATCHED (optional) These states have a 1:1 correspondence with states of the Alarm feature, which keeps track of whether the alarm is InAlarm and IsAcked. Alarms in the state ACK/RTN are not shown in the SUMMARY alarm display because they do not need any further action from the operator unless Latched is enabled for the Galaxy. But, all four states appear in the HISTORY display, and in the Alarm Database. If the Latched state is enabled, ACK/RTN alarms remain visible in the alarm display until the operator sends a Dismiss command.	
Value, CheckValue	Only static values sent with alarm message.	Static values and dynamic references are provided.
Ack	No behavioral differences between Application Server/OMI and InTouch HMI. All alarms are sent to the client and require acknowledgement regardless of priority.	
Dismiss	If the Latched alarm feature has been enabled for the Galaxy, all alarms in LATCHED state require a dismiss command.	
History	No behavioral differences between Application Server/OMI and InTouch HMI. Alarm state changes are logged to event history and shown on historical client.	

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configuring Plant State-Based Alarms

You can map alarm modes on a per-Galaxy basis to different plant operational states to control how alarms are reported. Five plant states are pre-defined, and have default alarm states associated with them:

Plant State	Default Alarm State	Available Alarm States
Running	Enable	Enable
Maintenance	Disable	Enable / Silence / Disable
Startup	Silence	Enable / Silence / Disable
Shutdown	Disable	Enable / Silence / Disable
Testing	Silence	Enable / Silence / Disable

You can define new plant states, rename plant states, or remove existing plant states, except the "Running" state (you can, however, rename "Running"). The alarm state for Running is read-only and cannot be changed from Enable.

After you have defined plant state-based alarm configurations for the Galaxy, you can assign plant state-based alarming to area objects in the Galaxy. This is done by setting the PlantState attribute for each area object that will use plant state-based alarming. The area object will automatically update its PlantAlarmMode attribute to match the alarm state that is set for the PlantState currently assigned to it.

Attribute	Definition
PlantState	The name of the currently assigned plant state (Running, Maintenance, etc.).
PlantStateAlarmMode	The alarm state of the assigned plant state (enable, silence, or disable). This attribute is read-only at run time.

Mapping Alarm Modes to Plant States

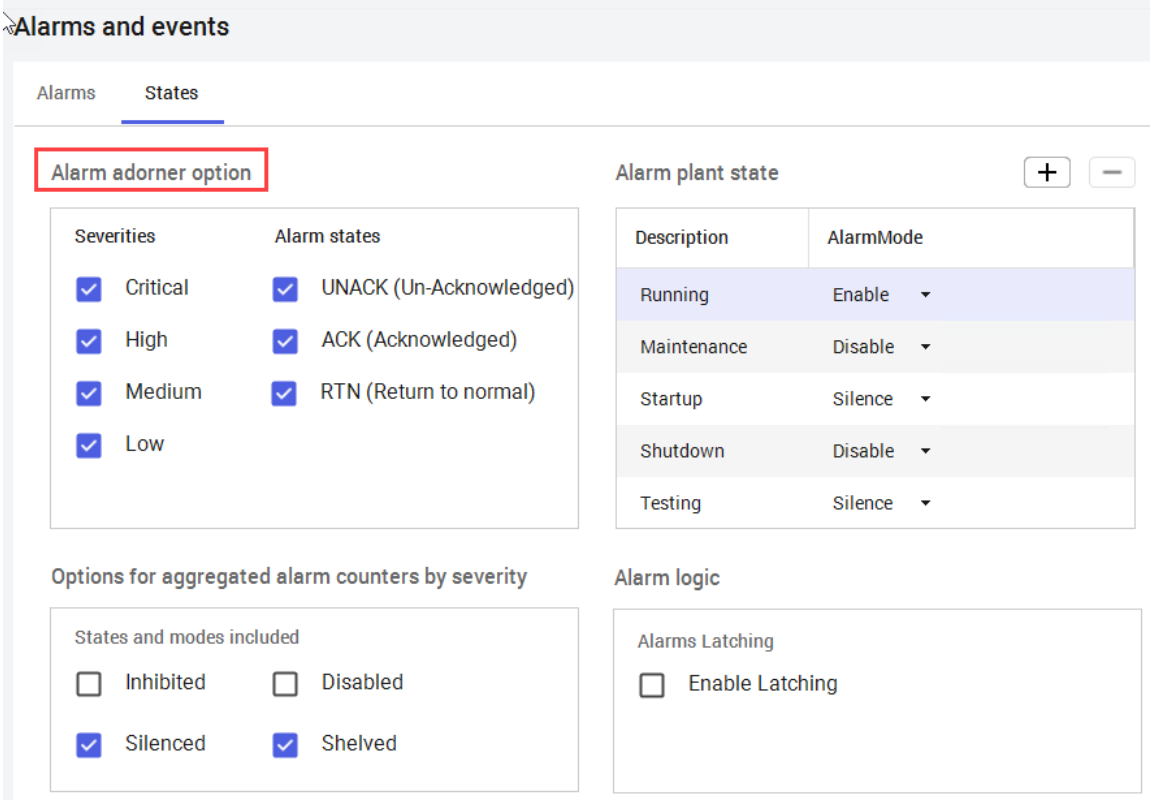
Define plant states and map them to alarm modes in the IDE. To access the configuration dialog:

1. From the ribbon, select **Galaxy,** then select **Configure.**
2. Select **Galaxy**, then select **Alarms**. The **Alarms** page appears.

Note: Settings in this dialog are not shared across Galaxies in a multi-Galaxy environment. Each Galaxy in the environment will have its own alarms configuration.

To use the default values, you do not have to set anything in this dialog. Simply enable plant state-based alarms for each area object that is to use this feature. See Configuring State-Based Alarming on an Area Object for additional information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Change Alarm Modes and Modify Plant States

- To change the AlarmMode value, click the **AlarmMode** associated with the Plant State that you want to change. Then, select the new value (enable, silence, or disable).

  Note: **AlarmMode** for Running is read-only and cannot be changed.

- To add a new Plant State, Click the **Add** (+) button. A new row is created at the bottom of the table.

  Enter the **Description** of the new plant state. The maximum length is 64 characters.

  Select a value for the **AlarmMode** (enable, silence, or disable).

- To delete an existing Plant State, click the **Remove (-)** button. The selected plant state and its associated value are removed from the table.

  Note: You cannot delete the first row of the **Alarm plant state** table. Running is the default PlantState **Description** in the first row.

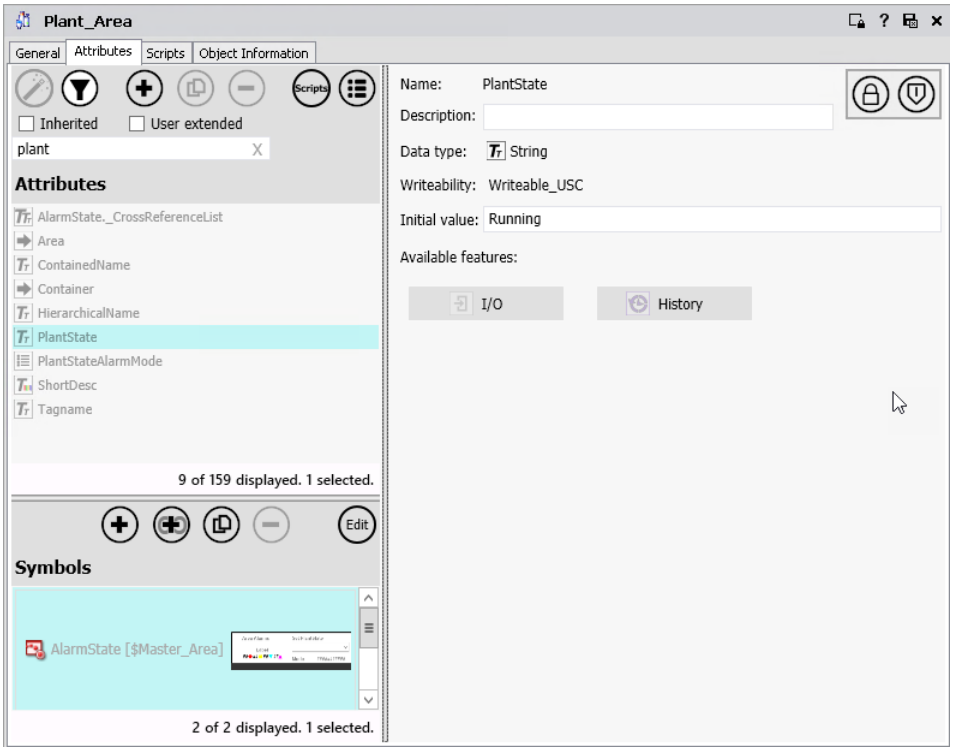
- To change the description of an existing PlantState, double click the plant state you want to rename, then enter the new description. The maximum length is 64 characters.

  Note: All Plant States, including Running, can be renamed.

Configuring State-Based Alarming on an Area Object

State-based alarm modes apply to String attributes, with writeability set to Writeable_USC (these cannot be changed). To use state-based alarming for one or more areas, you must configure the PlantState attribute for each area that will use plant state-based alarming.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Configure plant state alarming

1. Open the area object and click the Attributes page.
2. Select the PlantState attribute.
3. Enable the **I/O** feature and select Read, Read/Write, or Write.
4. Enable the **History** feature and configure it.
5. Select the PlantStateAlarmMode.
6. Repeat steps 3 and 4 to configure the **I/O** and **History** features.

Viewing Plant State-Based Alarms at Run Time

You can view plant state-based alarming at runtime through clients such as Object Viewer and InTouch applications in WindowViewer.

User Access and Security During Run Time

Operators with the appropriate permissions can change the PlantState of an area through a run time client. An operator may have to change PlantState from Startup to Running, for example. Scripts can also be used to implement modifications.

To change a PlantState, the operator enters a string to match one of the defined PlantState values (for example, "Running"). Before a change to PlantState can be implemented, the system checks that the user belongs to a role that has the permission "Can Modify Plant State of an Area" and is therefore authorized to make the change. If the operator enters a string that does not match a defined PlantState, the change is rejected.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

At run time, the AlarmMode attribute is read-only. Therefore, the AlarmMode for a PlantState cannot be changed through a run time client (for example, changing from enable to silence). It can only be changed through the IDE.

If an area is assigned to a PlantState and the PlantState is deleted through the IDE, the area will remain in that PlantState until it is changed. If AlarmMode for the deleted PlantState is anything other than enable, the AlarmMode will change to enable.

Note: Operators with appropriate permissions can use the AlarmModeCmd attribute to change AlarmMode of an area at run time. However, the AlarmModeCmd can only be used to set a more restrictive condition than the AlarmMode of the area's corresponding PlantState. For more information about changing alarm modes, see Enabling, Silencing, and Disabling Alarms.

Configuring Alarm Historization, Mapping, and Shelving Priority Ranges

The **Alarms and Events Configuration** dialog lets you:

- Map alarm severities to priority ranges.
- Enable alarm shelving by priority range.
- Enable or disable historization of both alarm severities and event types. All alarms are historized by default.

Default Alarm Mapping and Historization Values

Severity	Description	Shelve	Historize	From Priority Range	To Priority Range
1	Critical	N	Y	1	250
2	High	N	Y	251	500
3	Medium	Y	Y	501	750
4	Low	Y	Y	751	999

Shelving, historization and priority ranges are configurable.

Mapping Alarm Severity to Priority

- Severity 1 starts at priority 1 by default. Severities can be mapped to priorities in ascending or descending order. For example, severity 1 can map to priority range 1–250 or it can map to priority range 999-751.
- Severity 4 ends at 999 by default, but 999 is not required to be the end-of-range number. For example, severity 4 can end at 900, leaving a priority range above 900 unmapped to any severity level.
- A priority outside the configured priority ranges will return severity 0, not mapped. Since the priority is unmapped, it cannot be historized.

Note: Mapped alarm severities are not shared across galaxies in a multi-galaxy environment. Each galaxy in the environment has its own priority to severity mapping.

Configuring Historization for Alarms and Events

Historization is configurable for both events and alarms. Event types and the default historization setting for each type are:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

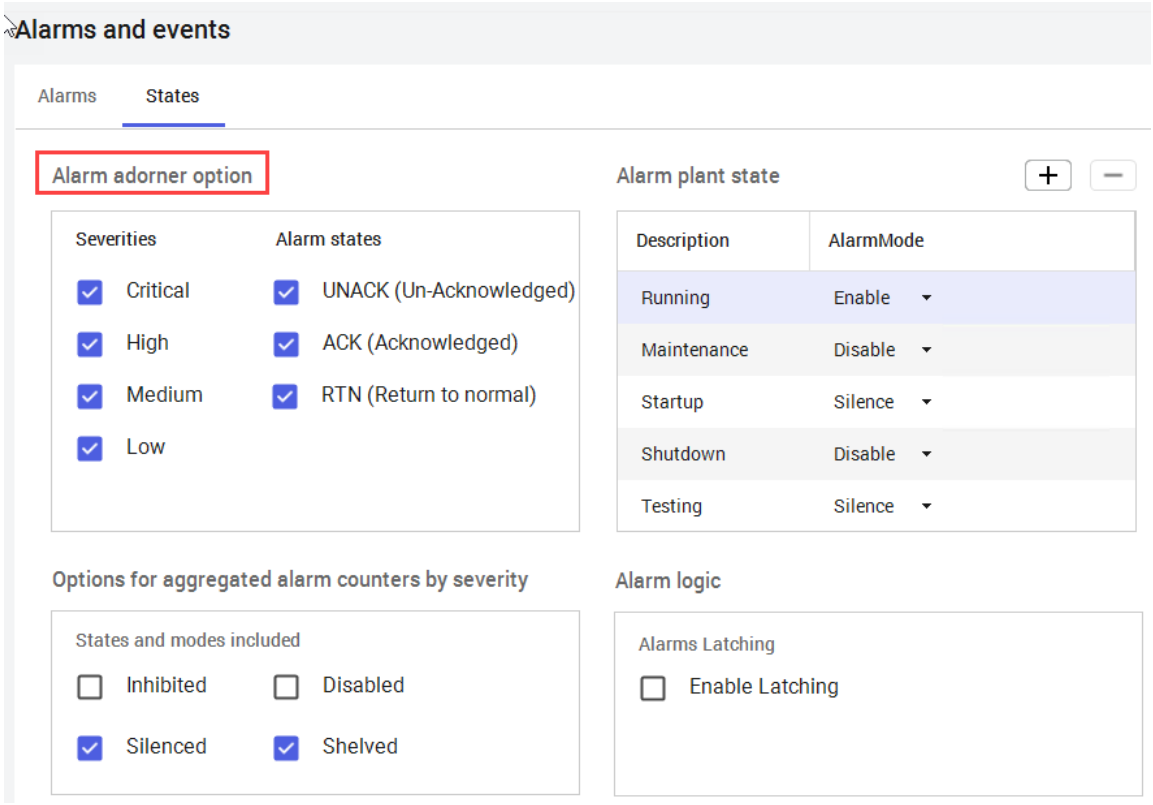
Default Event Type Historization Values		
Types	**Description**	**Historize**
1	System	Y
2	Application	Y
3	User	N

Note: In prior versions of Application Server, historization of alarms and events was configured through the Alarm DB Logger Manager (an InTouch component) and utilized the SQL database "WWALMDB." The method described here replaces the Alarm DB Logger method.

To configure alarm and event mapping

1. From the ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then select **Alarms**.

   The **Alarms** page appears.



3. Configure alarms:
   a. Edit the priority range for each severity level, or accept the default values.
   b. Accept the default (enabled) historization or click to clear the check box and disable historization for each severity level.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

4. Configure events: Accept the default historization values, or click to clear the check box and disable historization for each event type.

If you want to change the default Alarm Border indicator images shown in the **Alarms and Events Configuration** dialog, see "Changing Alarm Border Indicator Icons" in the *Creating and Managing Industrial Graphics User Guide*.

Monitoring Alarm Severities at Run Time

Monitor alarms by severity, expressed as integers 1–4, using clients such as Object Viewer and AVEVA OMI ViewApps. You can monitor individual alarms and you can monitor alarms aggregated by containment level. For more information about aggregating alarm states, see Aggregating Alarm State Information.

Typically, alarm priorities and priority to severity mapping are not changed during run time, but it is possible to change alarm priority configuration and severity mapping without closing the client application.

Understanding How Alarms are Ranked at Run Time

The most important alarm has the lowest severity number, but other criteria are taken into account when ranking alarms by urgency at run time.

Alarm Mode	Enabled is more urgent than silenced.
Alarm Shelved	FALSE (not shelved) is more urgent than TRUE (shelved).
InAlarm	TRUE (InAlarm) is more urgent than FALSE (normal).
Acked	FALSE (unacked) is more urgent than TRUE (acked).
Severity Level	1 is most urgent; 4 is least urgent.

Using Aggregated Alarm Information

Alarm aggregation lets you see all the active alarms on a containing object, such as an area object, as well as all the active alarms on its contained objects. This provides an efficient way at run time to identify if there are any active alarms (in the InAlarm state), and the location and status of alarms, for example, unacknowledged and unacknowledged - return to normal. This makes it possible to follow a trail from one level to the next to find the underlying cause of a complex object's alarms.

To obtain aggregated alarm severity status information:

1. Map alarm severity levels to priority ranges. For more information, see Configuring Alarm Historization, Mapping, and Shelving Priority Ranges.
2. Configure alarms on objects. For more information, see Configuring Alarms.
3. Enable alarm aggregation by Area. For more information, see Configuring Alarm State Aggregation
4. View aggregated alarm status information by means of run-time clients and applications. For more information, see Monitoring Alarm State Information at Run Time.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

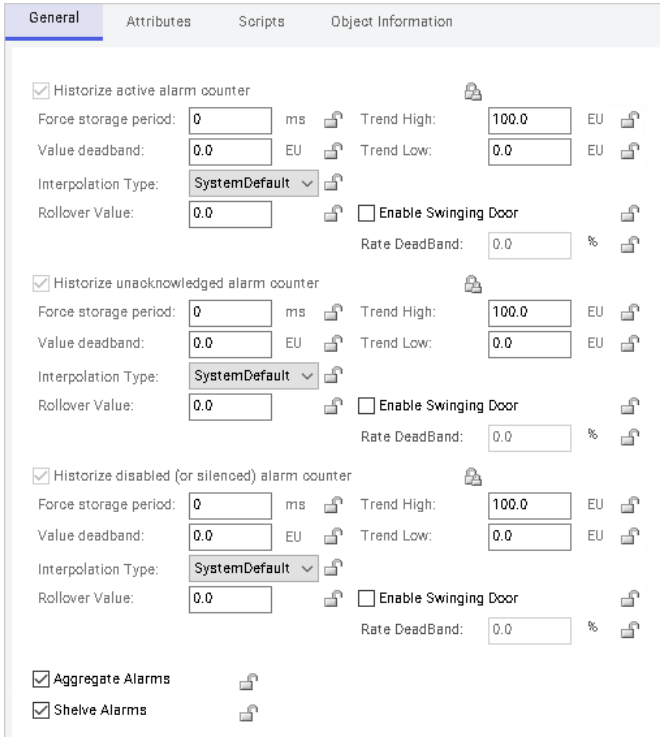
Configuring Alarm State Aggregation

Configuring alarm state aggregation consists of normal alarm configuration procedures plus an added step of enabling the aggregation feature on each relevant Area object.

Note: Alarm aggregation is enabled by default on Area objects. Alarm aggregation cannot be disabled on application objects, such User Defined objects.

To configure alarm state aggregation

1. Accept the default settings or configure alarm severities for each of the four severity levels you want to aggregate. This is a global configuration.
2. Configure an object with one or more alarms. All alarms configured on all objects will be aggregated.
3. Set the AlarmModeCmd of the object to enabled or silenced, but not to disabled.
4. Set the AlarmModeCmd of at least one alarm on the object to enabled or silenced, but not to disabled.
5. Set the AlarmModeCmd of the Area object to enabled or silenced, but not to disabled.
6. Check the **Aggregate Alarms** checkbox on the Area object editor to enable alarm aggregation. This sets the value of the AlarmAggregationStateCmd attribute to True.



Aggregating Alarm State Information

Run-time clients and applications, such as Object Viewer, AVEVA OMI ViewApps, and InTouch HMI applications, can show alarm aggregation statuses.

You can use animations such as the alarm border animation with Situational Awareness Library graphics or with Industrial Graphics in both Managed InTouch applications and AVEVA OMI ViewApps, and you can add alarm aggregation displays to navigation items in AVEVA OMI ViewApps

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Alarm aggregation functionality can be described for an object, for an area, and for an attribute.

Object	Aggregation represents all alarms on the object, on all contained objects, and on their descendents down to the lowest level of the model view. Alarm aggregation values on child objects are added to the values of the parent object or objects. All objects have this alarm aggregation functionality.
Area	Aggregation represents the alarms on the Area object itself, on all objects assigned to the area, and on all sub-Areas, down to the lowest level of the model view. If the Area's AlarmMode is silenced, all alarms on all Objects in that Area will be silenced. For more information about setting alarm modes, see the following topics: • Understanding Alarms • Setting Alarm State with Object Attributes • Enabling, Silencing, and Disabling Alarms.
Attribute	Aggregation represents all the alarms on the attribute itself. This is the lowest level of aggregation.

Alarms are aggregated if they are in one of three states:

- UNACK_ALM
- ACK_ALM
- UNACK_RTN

Alarms in the ACK_RTN state are not aggregated.

Alarms in silenced mode are aggregated, even though they do not appear in a run-time client.

A set of five attributes provide run-time aggregated alarm status information:

AlarmMostUrgentSeverity	Displays the severity as an integer 1–4 of the most important current alarms on an object and its descendents. If no alarms are in the InAlarm state or waiting to be acknowledged, the value is 0.
AlarmMostUrgentMode	Displays the mode (enabled or silenced) of the most important current alarm. Alarms in disabled mode are not aggregated.
AlarmMostUrgentInAlarm	Displays the InAlarm status (true or false) of the most important current alarm.
AlarmMostUrgentAcked	Displays the acknowledgement status (true or false) of the most important current alarm.
AlarmCntsBySeverity	In AVEVA OMI and InTouch HMI run-time client applications, this attribute is used to show the number of alarms by status and

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

severity. In the Object Viewer, this attribute displays an array that shows:

- The total number of active alarms (UNACK_ALM + UNACK_RTN + ACK_ALM) at each severity level.
- The UNACK_ALM alarm count.
- The UNACK_RTN alarm count.
- A decimal representation if any active alarm bits are set on the local object.

  Note: By default, **AlarmCntsBySeverity** includes shelved and silenced alarms, but does not include hidden or disabled alarms. You can change this by changing the **AggregatedAlarmStatesIncluded** filter. For more information, see About the AggregatedAlarmStatesIncluded Attribute

About the AlarmCntsBySeverity Attribute

AlarmCntsBySeverity is a 13 element array, divided into three groups of four, plus one additional element to indicate local alarm severity and status. The first group of four elements shows the number of aggregated active alarms by severity (S1 through S4), the next group shows the aggregated number of unacknowledged alarms (UNACK_ALM) by severity, and the third group shows the aggregated number of return to normal (UNACK_RTN) alarms by severity. The final element of the array is a bit value that defines which, if any, of the alarms are on the selected (local) object. See Local Alarm Display in the Object Viewer for additional information.

You can determine the number of active, acknowledged alarms (ACK_ALM) by subtracting the number of unacknowledged alarms shown in the second group of elements from the active alarm count shown in the first group of elements.

Note: By default, **AlarmCntsBySeverity** includes shelved and silenced alarms, but does not include hidden or disabled alarms. You can change this by changing the **AggregatedAlarmStatesIncluded** filter. For more information, see About the AggregatedAlarmStatesIncluded Attribute

About the AggregatedAlarmStatesIncluded Attribute

AggregatedAlarmStatesIncluded is a four-element array that controls whether shelved, hidden, silenced, and/or disabled alarms are included in the values of the **AlarmCntsBySeverity** attribute. It is a galaxy-wide attribute. It is the same for all engines, areas, and objects.

The elements of **AggregatedAlarmStatesIncluded** are:

Element	Alarm condition	Default
1	Shelved	True
2	Silenced	True
3	Disabled	False
4	Hidden	False

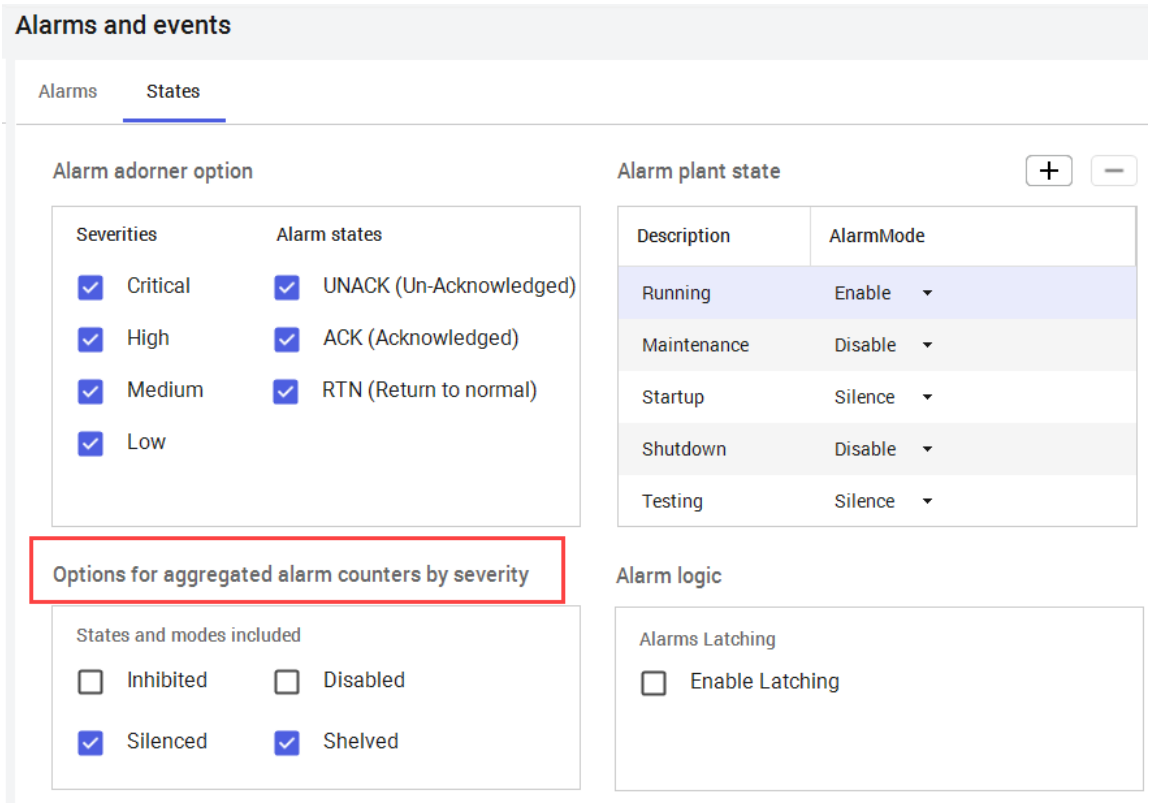
You can change the **AggregatedAlarmStatesIncluded** settings using the IDE. Follow the procedure below.

For more information about **AlarmCntsBySeverity**, see About the AlarmCntsBySeverity Attribute.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To configure the alarm states included in AlarmCntsBySeverity:

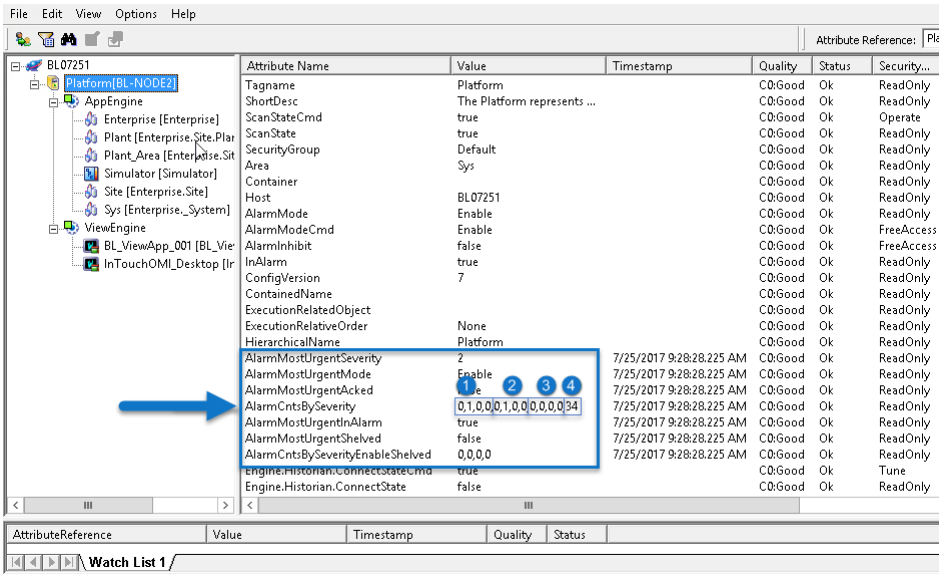
1. From the IDE main menu, select **Galaxy**, then **Configure > Galaxy - Alarms and events**. This opens the **Alarms and events** dialog.
2. Select the **States** tab.
3. In the **Options for aggregated alarms counters by severity** section, select the checkboxes for the alarm states to include in the **AlarmCntsBySeverity** totals and clear the checkboxes for those not to include.



Local Alarm Display in the Object Viewer

The Object Viewer shows thirteen values, separated by commas, for the **AlarmCntsBySeverity** attribute. The first twelve values (logically divided into groups of four) show how many alarms of each severity and type have been aggregated for the selected object or area:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



1. Number of active alarms, by severity.
2. Number of unacknowledged active alarms, by severity.
3. Number of unacknowledged alarms that have returned to normal, by severity.
4. Bit value that shows which alarm severity and status applies to the local object.

   This value is a sum of the bit values shown in the following table:

	← (LSB) Active (1)				UnAck_ALM (2)				UnAck_RTN (3) (MSB) →			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
Bit value	1	2	4	8	16	32	64	128	256	512	1024	2048

In the Object Viewer example shown above, the local object bit value (4) is 34. This is the sum of the active S2 bit value plus the UnAck_ALM S2 bit value. Other examples:

- If the local object has one or more active S1 alarms that are unacknowledged, the bit value is 17 (S1 Active Alarm = 1; S1 UnAck_ALM = 16).
- If the local object has one or more active S4 alarms that are UnAck_RTN, the bit value is 2056 (S4 Active Alarm = 8; S4 UnAck_RTN = 2048).
- If there are no local alarms, the bit value is 0.
- If there are multiple local alarms with different severities and unack statuses, the bit value is the sum of each bit that is set. For example, if there are two unacknowledged S1 alarms, but one has returned to normal, the bit value is 273 (two S1 Active Alarms = 1; one S1 UnAck_ALM = 16; one S1 UnAck_RTN = 256). Note that each bit that is set is only counted once, even if more than one alarm exists at that severity or status.

When alarm aggregation is shown on a navigation item in an AVEVA OMI ViewApp, the bit value is used by the navigation item to determine the alarm location (selected object or a contained object). See Aggregated Alarm Display in AVEVA OMI Navigation for additional information.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: By default, **AlarmCntsBySeverity** includes shelved and silenced alarms, but does not include hidden or disabled alarms. You can change this by changing the **AggregatedAlarmStatesIncluded** filter. For more information, see About the AggregatedAlarmStatesIncluded Attribute

Aggregated Alarm Display in AVEVA OMI Navigation

The **Alarms and events** dialog is used to configure which aggregated alarms are displayed on a Galaxy-wide basis. You can set whether or not display alarm information in individual AVEVA OMI ViewApps by enabling/disabling the **ShowAlarms** attribute. AVEVA OMI ViewApps display alarm indicators and alarm counts for aggregated alarms on NavigationApp controls (NavigationTree and BreadcrumbControl), if the **ShowAlarms** attribute is enabled in the ViewApp. To enable this attribute, select the **Show Alarms** option when you configure the control in the **Layout Editor**.

Note: Alarm indicators ("adorners") used in AVEVA OMI navigation are FrameworkElement Adorners, part of the Microsoft .NET Framework used in creating and configuring apps for AVEVA OMI ViewApps.

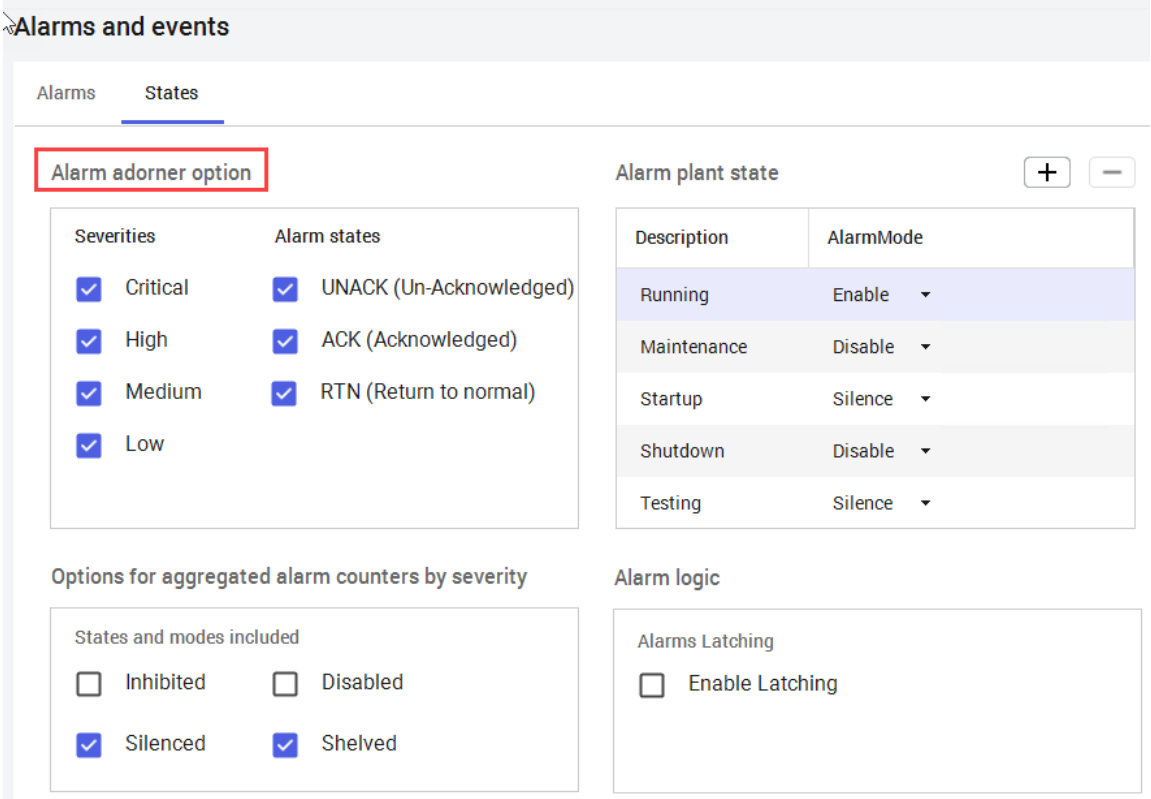
The opacity of the alarm indicator shows if the alarm is on the selected navigation node or on a contained object. When enabled, the following alarm severities and alarm statuses are shown:

- If the color of the alarm indicator is opaque (solid), the alarm is on the selected (local) object. If enabled, the severity (S1 to S4) and status of the alarm (UNACK_ALM, UNACK_RTN, ACK_ALM) is shown graphically, along with a count for each alarm severity that has been enabled.
- If the color of the alarm indicator has some transparency, the alarm is on a contained object.
- The color of the blinking line under the object name shows the highest-severity (most urgent) alarm.
- An indicator on the NavigationTree or BreadCrumbControl shows if the data quality of the object is bad. Bad quality may result if communication has been lost with the object.

You can also determine which navigation node or asset contains the alarm by selecting the alarm icon. This displays a tooltip that indicates the node(s) or asset(s) sending the alarm(s).

Set the **Alarm adorner option** in the **Alarms and events** dialog (**Galaxy** > **Configure** > **Galaxy - Alarms and events**, then select the **States** tab). This configures which aggregated alarms are displayed on NavigationApp controls and any custom apps that subscribe to the aggregated alarm count during run time.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The following **Alarm Adorner Options** are set through the dialog. The corresponding attribute settings are propagated to run-time nodes (all attribute names are prefixed by MyViewApp.Alarms). For additional information about these attributes, see About Alarm Adorner Attributes.

Alarm Severities to Display	**Attribute Name (Alarms Namespace)**	**Description**
Critical	ShowCriticalAlarms	When enabled, the ViewApp displays critical alarms
High	ShowHighAlarms	When enabled, the ViewApp displays high alarms
Medium	ShowMediumAlarms	When enabled, the ViewApp displays medium alarms
Low	ShowLowAlarms	When enabled, the ViewApp displays low alarms

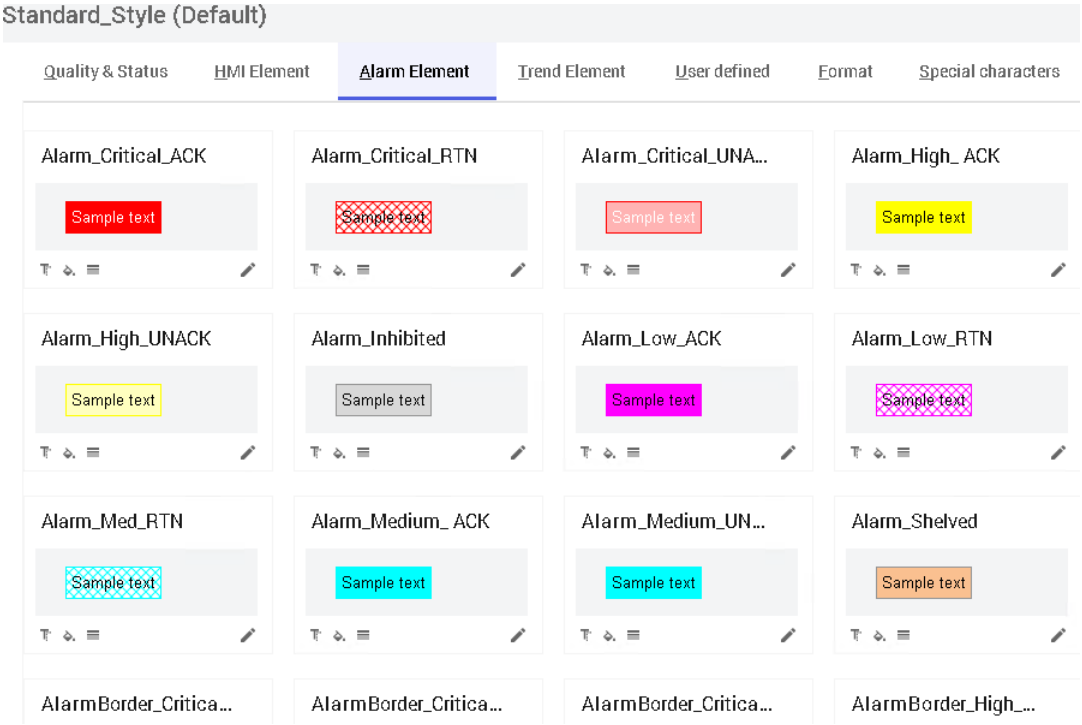
Alarm States to Display	**Attribute Name (Alarms Namespace)**	**Description**
UNACK (Unacknowledged)	ShowUnacknowledgedAlarm	When enabled, the ViewApp displays UNACK alarms
ACK (Acknowledged)	ShowAcknowledgedAlarms	When enabled, the ViewApp displays ACK alarms

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

RTN (Return to Normal)	ShowReturnToNormalAlarms	When enabled, the ViewApp displays RTN alarms
LATCHED	ShowLatchedAlarms	Latched alarm state must be enabled for the Galaxy for Latched alarms to show. When enabled, the ViewApp displays Latched alarms.

Alarm Indicator Appearance

The appearance of alarm indicators is configured through the **Alarm Element** tab of the **Galaxy Style** page, and uses the graphics that are shown on the **Alarms** page.



About Alarm Adorner Attributes

Alarm Adorner attributes are available only for use in AVEVA OMI ViewApps. All Alarm Adorner attributes operate in the MyViewApp.Alarms namespace. Attributes for the Alarm Adorner are specified with the MyViewApp.Alarms prefix in the form MyViewApp.Alarms.*attribute_name*. Alarm Adorner attributes cannot be used in an InTouch application because the MyViewApp.Alarms prefix is regarded as a configuration error.

Attribute Name	Data Type	Attribute Type	Default Value	Description
ShowCriticalAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays critical alarms
ShowHighAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays high alarms

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Attribute Name	Data Type	Attribute Type	Default Value	Description
ShowMediumAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays medium alarms
ShowLowAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays low alarms
ShowUnackedAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays UNACK alarms
ShowAckedAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays ACK alarms
ShowReturnToNormal Alarms	Boolean	Read/ Write	True	If True, the ViewApp displays RTN alarms

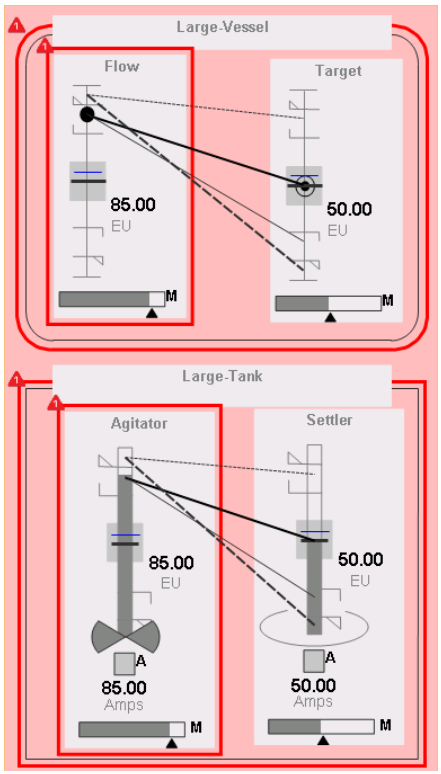
Monitoring Alarm State Information at Run Time

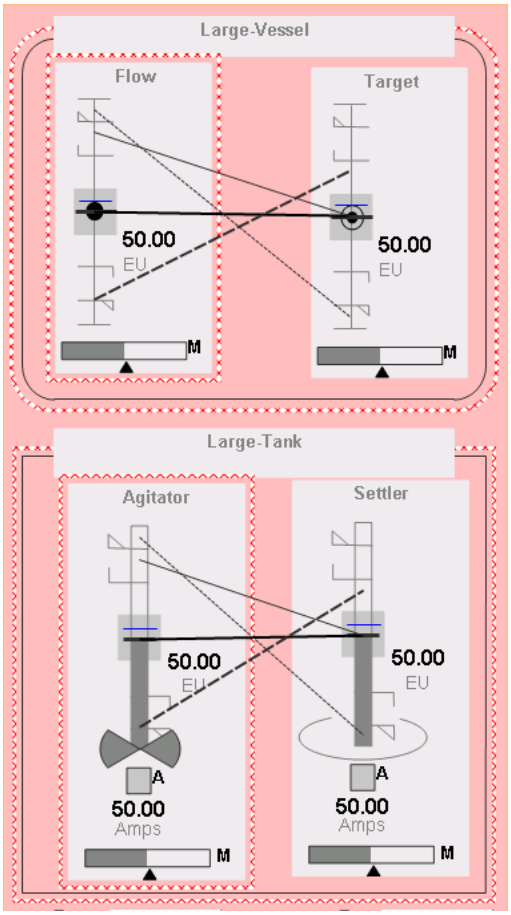
You can obtain run-time alarm state information, including aggregated alarm information, using the Object Viewer or InTouch Tag Viewer. You can visualize the objects in an AVEVA OMI ViewApp or InTouch HMI application using the AlarmApp for AVEVA OMI or Alarm Client for InTouch HMI, along with the alarm border animation to represent the aggregated alarms at each level of containment. You can also enable aggregated alarm information for the NavTree

Note: AlarmApp is not supported in InTouch HMI applications. Similarly, AlarmClient is not supported in AVEVA OMI ViewApps. Use AlarmApp for AVEVA OMI ViewApps to allow alarm aggregation to follow selected objects and areas.

An alarm border animation applied to a group of graphics can indicate a priority 1 alarm with a red border, flashing or solid for different alarms. The alarm border animation can also indicate a return to normal (RTN) for the same graphics.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





Alarm clients configured in InTouch applications will display the severity of each alarm in the **User1** field. You can change "User1" to a more descriptive heading, such as "Severity".

For information about configuring alarm animations, see "Configuring an Alarm Border Animation" in the *Creating and Managing Industrial Graphics User Guide*.

About graphic elements

This topic explains how to work with the common features of graphic elements.

Graphic elements are basic shapes and controls you can use to create a graphic. You can:

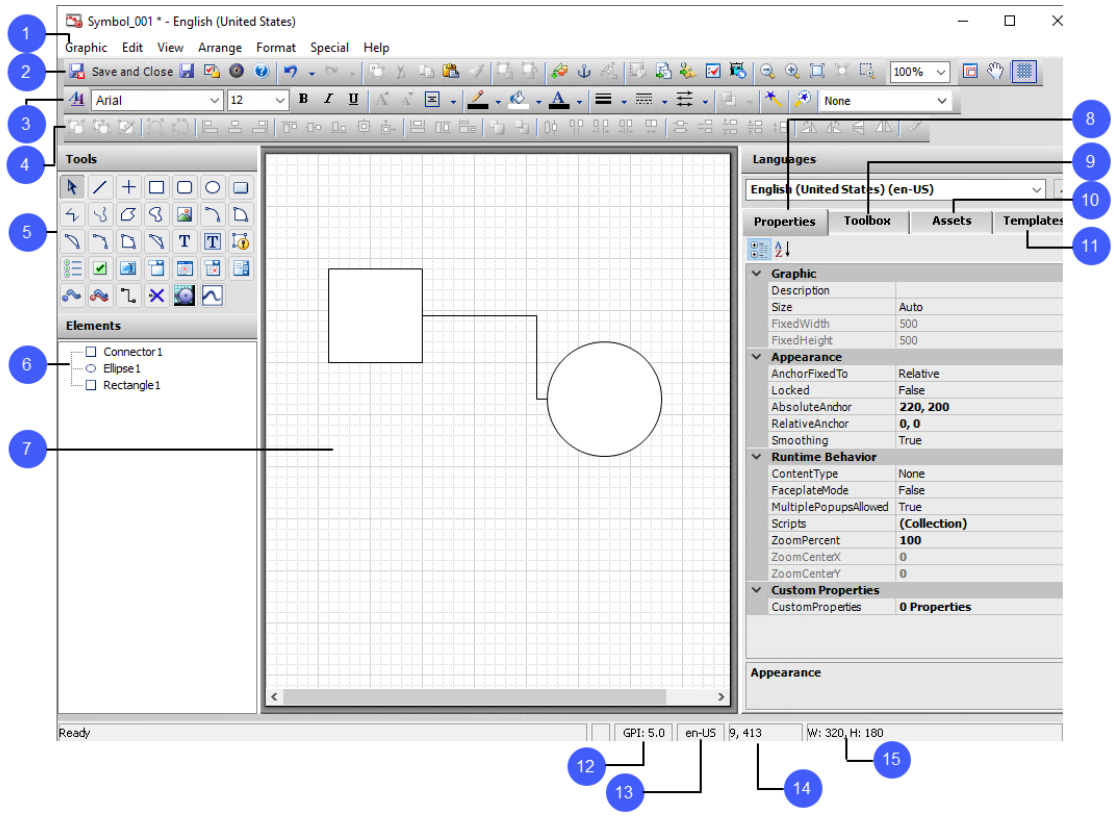
- Draw an element by selecting an element from the **Tools** panel, placing it on the canvas, and then configuring its properties.
- Select one or more elements on the canvas with the mouse or from the **Elements** list.
- Edit certain elements in a special way called inline editing.
- Copy, cut, paste, and duplicate elements.
- Move elements around on the canvas.
- Align elements to each other.
- Change the spacing between elements.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Resize elements.
- Change the z-order of elements to change which elements appear on top of others when they overlap.
- Rotate elements.
- Change the origin of elements to specify around which point the elements are rotated.
- Flip elements on their horizontal or vertical axis.
- Lock elements to stop them being moved or changed.
- Undo and redo any number of changes made previously to the graphic.
- Create groups of elements to bind them together.
- Create a path graphic from multiple open line elements.
- Import an SVG (Scalable Vector Graphic) graphic.

About the Graphic Editor

The figure below shows the various components of the Industrial Graphic Editor to create and configure graphic elements and graphics. You check a graphic and then double-click it to begin editing. The Industrial Graphic Editor opens and shows the graphic you selected on the canvas area.



1	The **Menu Bar** shows options with drop-down lists of commands to modify the graphic on the canvas.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2	The **Command** bar includes commands to save, save and close, and close the Industrial Graphic Editor.
3	The **Configuration** bar includes options to configure the visual appearance of text and lines that are part of the graphic on the canvas.
4	The **Placement** bar includes options to align selected graphic elements or graphics on the canvas.
5	The **Tools** pane contains a set of icons that represent graphic elements. A graphic element is selected from the **Tools** pane and then dragged and dropped onto the canvas for editing.
6	The **Elements** area lists the current set of graphic elements or graphics on the canvas in a hierarchical view.
7	The **Canvas** area shows graphic elements or other content that belong to the graphic currently being edited.
8	The **Properties** grid lists the Graphic, Appearance, Runtime Behavior, and Custom Properties of the graphic element or graphic selected on the Canvas. Each property contains a data entry field to assign a value to the property.
9	The **Toolbox** tab includes a set of folders containing screen profiles, layouts, graphics, apps, and ViewApp namespaces.
10	The **Assets** tab shows the derived instances of objects listed in the **Model** view of a ViewApp.
11	The **Templates** tab shows the contents of the **Template folder**.
12	The **GP**I field shows the calculated Graphic Performance Index of the graphic currently being edited. The GPI is a value from 0 to 5 that indicates load-time performance of each piece of content that belongs to a graphic.
13	The **Language** field shows the languages configured for the selected graphic on the canvas.
14	Current cursor position by pixel width and height from the origin of the canvas (0,0), which is the top left corner of the canvas area.
15	Width and height in pixels of the selected graphic element on the canvas.

About the graphic element property commonalities

Some properties are common to most graphic elements, such as fill, line styles, and visibility. You can:

- Edit the name of an element.
- Edit the fill properties of an element.
- Edit the line properties of an element.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Edit the text properties of an element.
- Set the style.
- Set the transparency level of an element.
- Tweaking colors and style for an element's gradient style.
- Enable and disable elements for runtime interaction.
- Change the visibility of an element.
- Change the tab order of an element.
- Use the Format Painter to format elements.
- Edit the general properties of a graphic.

About graphic-specific and element-specific properties

You can configure graphic-specific and element-specific properties.

You can configure:

- General properties of a graphic.
- Radius of rounded rectangles.
- Shape and end appearance of lines and H/V lines.
- Auto-sizing and word-wrapping in text boxes.
- Image-specific properties.
- Button-specific properties.
- Control points and tension in curves.
- Angles in pies, chords, and arcs.
- Status elements.
- Windows common controls.

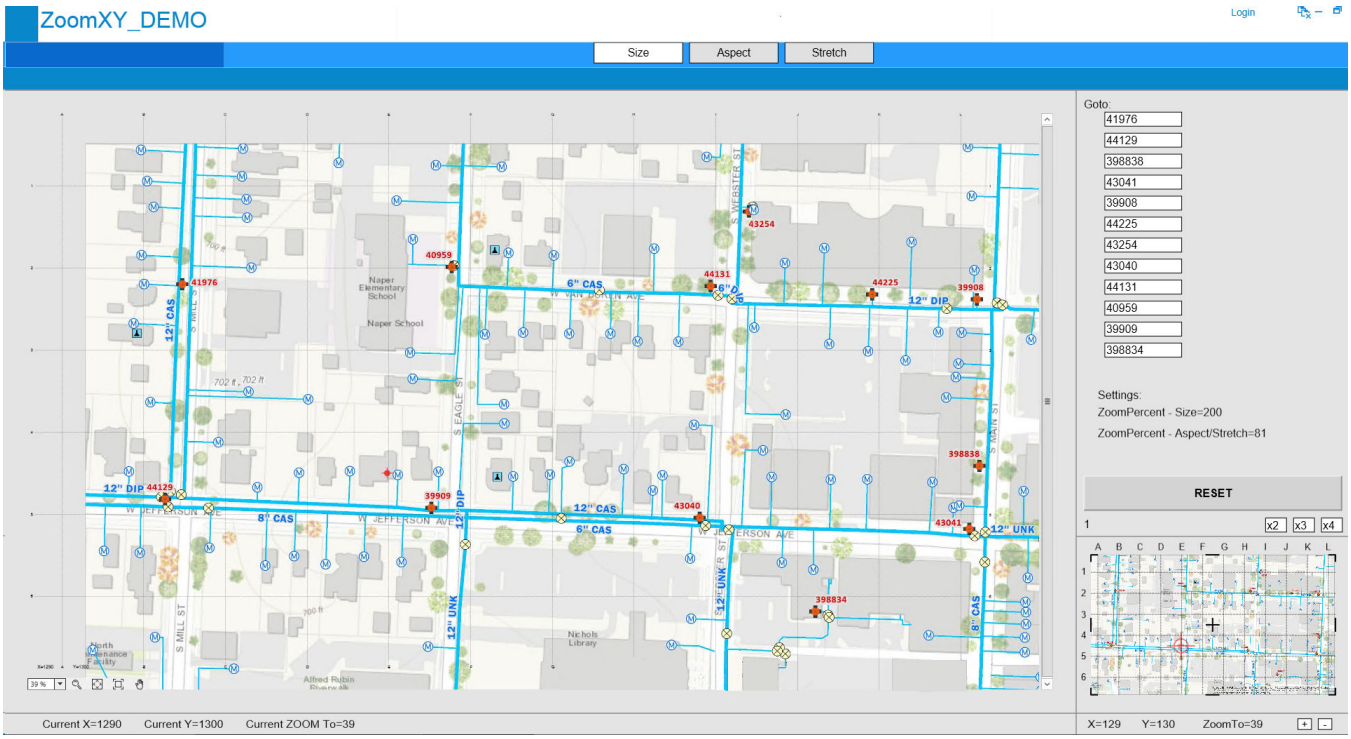
For more information about these topics, see Edit graphic-specific and element-specific properties in the Industrial Graphic Editor help.

About zoom percentage

Graphics shown in a running ViewApp can be zoomed in to show more detail using scripting. Instead of zooming in from the existing center point of the pane, a script can specify a zoom area center point within the boundaries of a graphic. The ability to place the zoom area center point anywhere within the boundaries of a graphic restricts the zoom area to a specific portion of the graphic to show more detail.

The following graphic shows a graphic of a map within a running ViewApp. The red crosses represent the center points of the different zoom areas of the graphic. The buttons shown at the right of the graphic include action scripts that set XY coordinate center points of the zoom areas. When an operator selects a button, the graphic is moved within the pane to set the selected zoom area to the center of the pane. The map can be zoomed in or out.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Set the zoom percentage

There are several distinct tasks that must be completed to enable programmatic zooming in a ViewApp. The following list shows the recommended order these tasks should be completed.

1. Determine the paired X and Y coordinate values that will serve as the center points of the programmatic zoom areas of the graphic.
2. Assign values to layout pane properties that affect programmatic zooming. See About Layout pane property effects on zoom behavior.
3. Create a set of attributes with the Namespace Editor that will be used in scripts and animation. See Set zoom percentage attributes.
4. Write a set of scripts and assign animations to enable programmatic zooming. See Minimal set of scripts and animations to enable programmatic zooming of graphics

Specify Zoom Center Point Coordinates of a Graphic

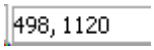
Before you create scripts and animations to set programmatic zooming of a graphic, you must determine the different X,Y coordinates within a graphic that you want to assign as the center points of the zoom areas.

To determine the zoom center point coordinates of a graphic

1. Open the graphic in the Graphic editor.
2. Place the mouse cursor directly over the point in the graphic that you want to set as the center point of a zoom area.

   The current X and Y cursor values of the cursor location are shown in the status bar beneath the canvas area of the Graphic editor.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. Save the coordinates of each location to specify them in action scripts.

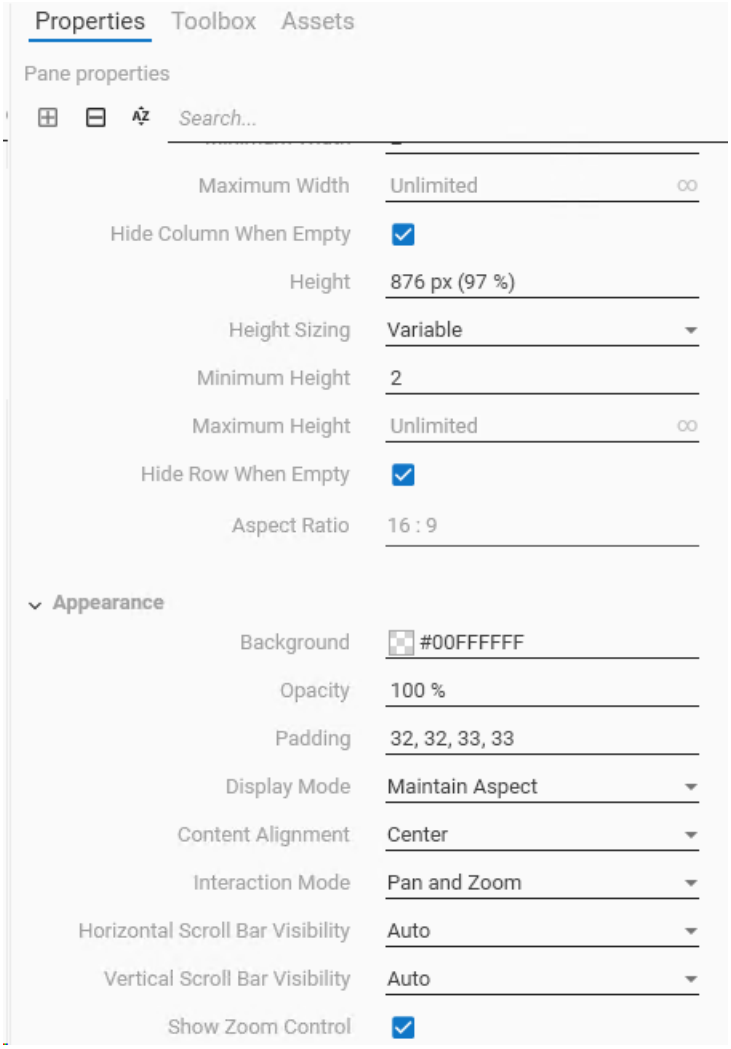
About Layout pane property effects on zoom behavior

Layout pane properties specified during design time determine some aspects of zoom behavior during runtime.

- The values assigned to the **Display Mode** property determine the placement of the graphic within a pane when a graphic is zoomed larger than the dimensions of a pane and whether a graphic retains its original aspect ratio.

  Important: The zoom percent range varies based on the value assigned to the **Display Mode** property. When **Display Mode** is set to **Maintain Size**, the zoom percent range is from 10 percent to 5000 percent. When **Display Mode** is set to **Maintain Aspect** or **Stretch**, the zoom percent range is from 100 percent to 5000 percent.

- The values assigned to the **Content Alignment** property determines the placement of graphic within a pane when zoomed larger than a pane in at least one dimension.
- Panning and zooming should be enabled for a pane and show the pan and zoom control icons during runtime.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The following table lists important pane properties that will affect programmatic zoom control of graphics.

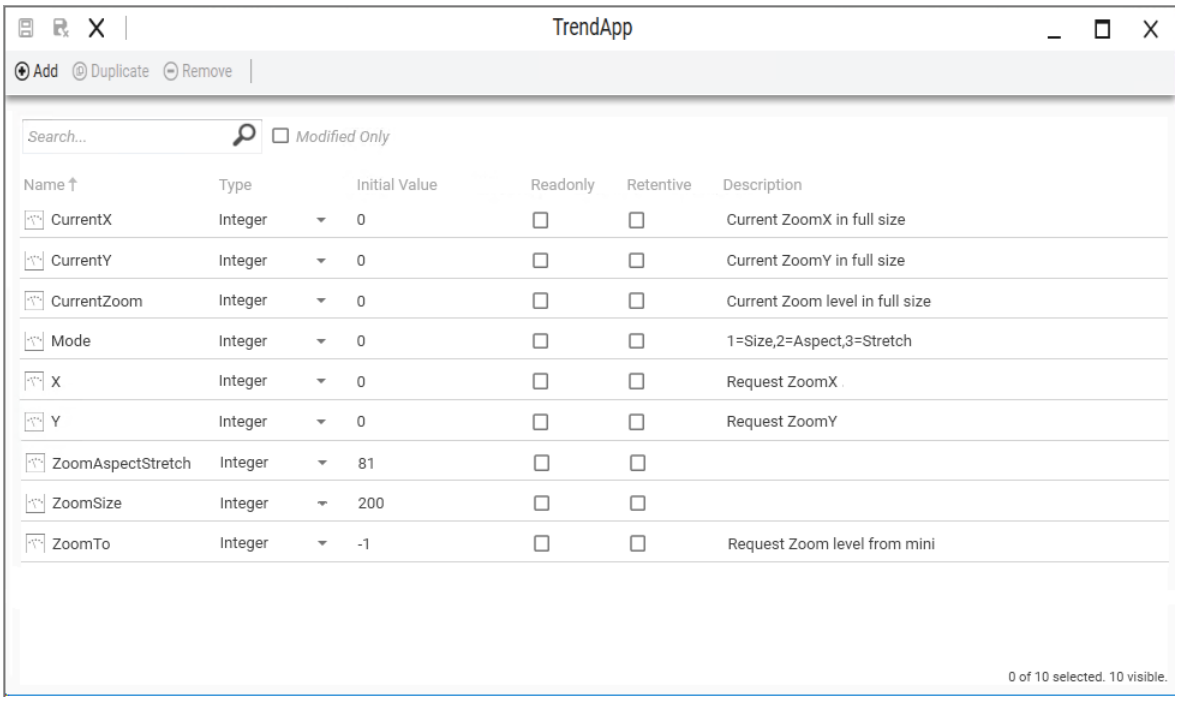
Pane Property	Description
Display Mode	Assigned value determines how a graphic is resized to fit the current pane size. For more detail about the different **Display Mode** values, see Set pane properties: display mode. • **Maintain Aspect** is the default value to retain the aspect ratio of the graphic when it is zoomed larger or smaller. A graphic's left and right boundaries will align with the pane or its top and bottom boundaries will align with the pane. A graphic is truncated in any dimension that is larger than the pane's width or height. • **Maintain Size** places a graphic within the pane based on the assigned **Alignment** and **Padding** property values when the content is smaller than the pane. When the size of a graphic exceeds the pane's boundary, the graphic retains its aspect ratio and is centered in the graphic and truncated in any dimension beyond the pane boundary. • **Stretch** resizes the graphic regardless of its aspect ratio to completely fit within the dimensions of the pane.
Content Alignment	Assigned value determines the alignment of the graphic within a pane area. For more information about the different **Content Alignment** values, see Set pane properties: display mode. **Center** is the default to place the center of the graphic at the center of the pane area.
Interaction Mode	Assigned values determines whether the user can can pan or zoom the pane area. **Pan and Zoom** is the default to enable panning and zooming in a pane during runtime.
Show Zoom Control	Assigned value determines if the pan and zoom controls are shown in a pane during runtime. The recommended value is to show both controls during runtime.

Set zoom percentage attributes

The following figure shows an example set of attributes created with the Namespace editor to control zoom behavior with scripts. In this example, the MyZoom Galaxy namespace has been created to contain the zoom attributes.

These attributes can be used in action scripts and a variety of animations to select a zoom level and center point of a zoom area during runtime.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The following table describes a suggested set of attributes that provide programmatic zoom levels based on the example attributes shown in the figure above.

Attribute	**Script or Animation**	**Purpose**
CurrentX	Value Display animation	An integer value that shows the current X value of the center point of a zoom area.
CurrentY	Value Display animation	An integer value that shows the current Y value of the center point of a zoom area.
CurrentZoom	Value Display animation	An integer values that shows the current zoom level.
Mode	Action Script	An integer value that specifies the pane's display mode. • **Maintain Size**=`MyViewApp.MyZoom.Mode==1` • **Maintain Aspect**=`MyViewApp.MyZoom.Mode==2` • **Stretch**=`MyViewApp.MyZoom.Mode==3`
X	Action Script	An integer value that sets the X coordinate value of the center point of the zoom area.
Y	Action Script	An integer value that sets the Y coordinate value of the center point of the zoom area.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

ZoomAspect Stretch	Action Script	An integer that sets the zoom percentage when the value of the **Mode** attribute is set to **Maintain Aspect** or **Stretch.**
ZoomSize	Scripts and User Input animation	An integer that sets the zoom percentage when the value of the **Mode** attribute is set to **Maintain Size**.

ZoomTo	Scripts and User Input animation	An integer that sets the zoom percentage.
		The zoom range minimum or maximum percentage values can be specified by using ZoomTo in an animation expression.
		`MyViewApp.MyZoom.ZoomTo<10`
		`MyViewApp.MyZoom.ZoomTo>4999`
		Zoom in and zoom out buttons are disabled when the ZoomTo attribute expression is true.

For more information about using these attributes in scripts and animation, see Minimal set of scripts and animations to enable programmatic zooming of graphics.

Minimal set of scripts and animations to enable programmatic zooming of graphics

As part of building a ViewApp that incorporates user controls to resize graphics during runtime, scripting and animation must be associated with the user controls.

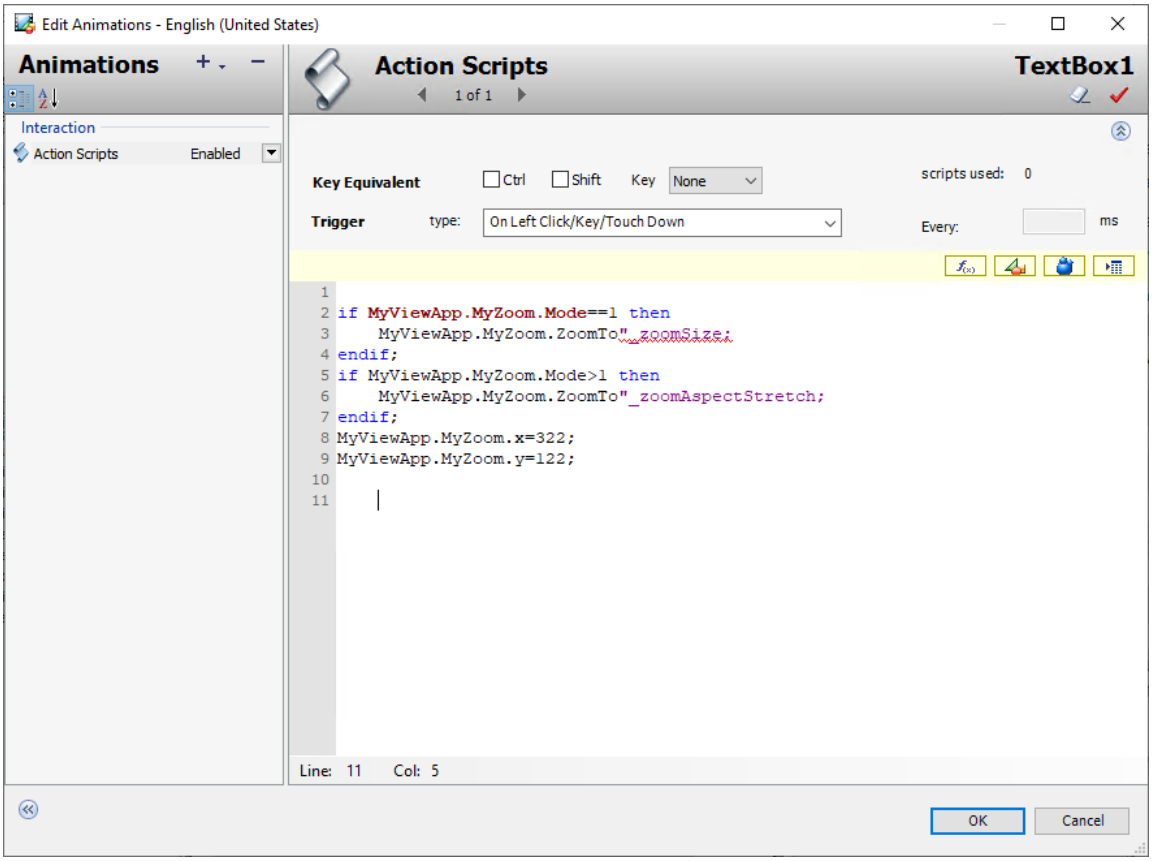
Note: This topic describes a minimal set of scripts and animations to enable programmatic zooming of graphics. More complex scripting can be used to provide additional zoom features or displays.

Create an Action Script for Each Zoom Center Point

The following figure shows an example of an action script to set the center point of a zoom area on a graphic. In this example, the script determines the current display mode of the pane and then sets a zoom percentage. The script then sets the X and Y coordinates of a center point of the zoom area that has been previously mapped on the graphic.

Typically, the action script is associated with a button that the user selects to move to the center point of the zoom area of the graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

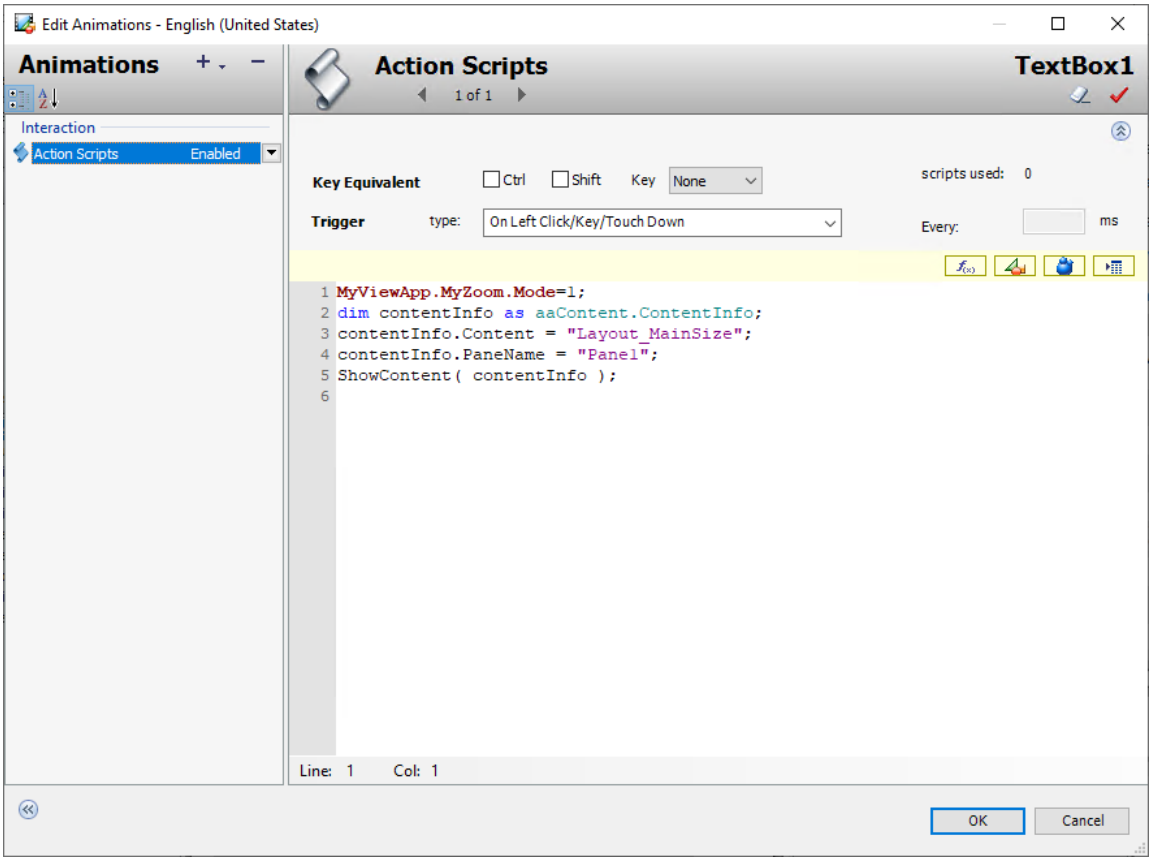


Important: In scripts, the attribute that sets the zoom percentage value must be declared before the attributes that specify the X and Y coordinates of the zoom area. The X and Y coordinates can be specified in any order.

Create an Action Script to Select Different Layouts Based on the Display Mode Property Value

This type of script is optional. If you want to change layouts containing different graphics during runtime based on a selected Display Mode, create an action script that includes the ShowContent method. In this example, a different layout is used for each possible Display Mode property value, which are associated with buttons.

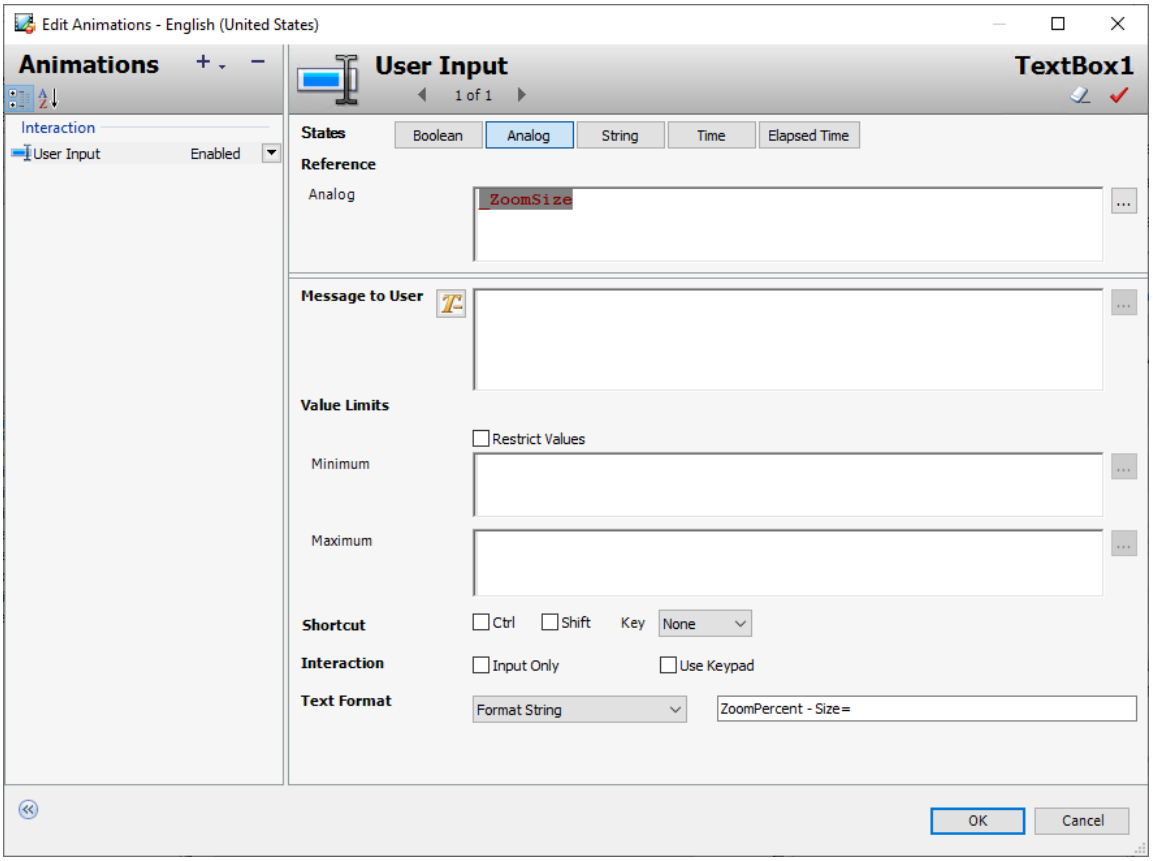
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Create User Input Animation to Set Zoom Values

The following figure shows an example of User Input animation to set the zoom level of a graphic. In this example, the user selects a data entry field during runtime.

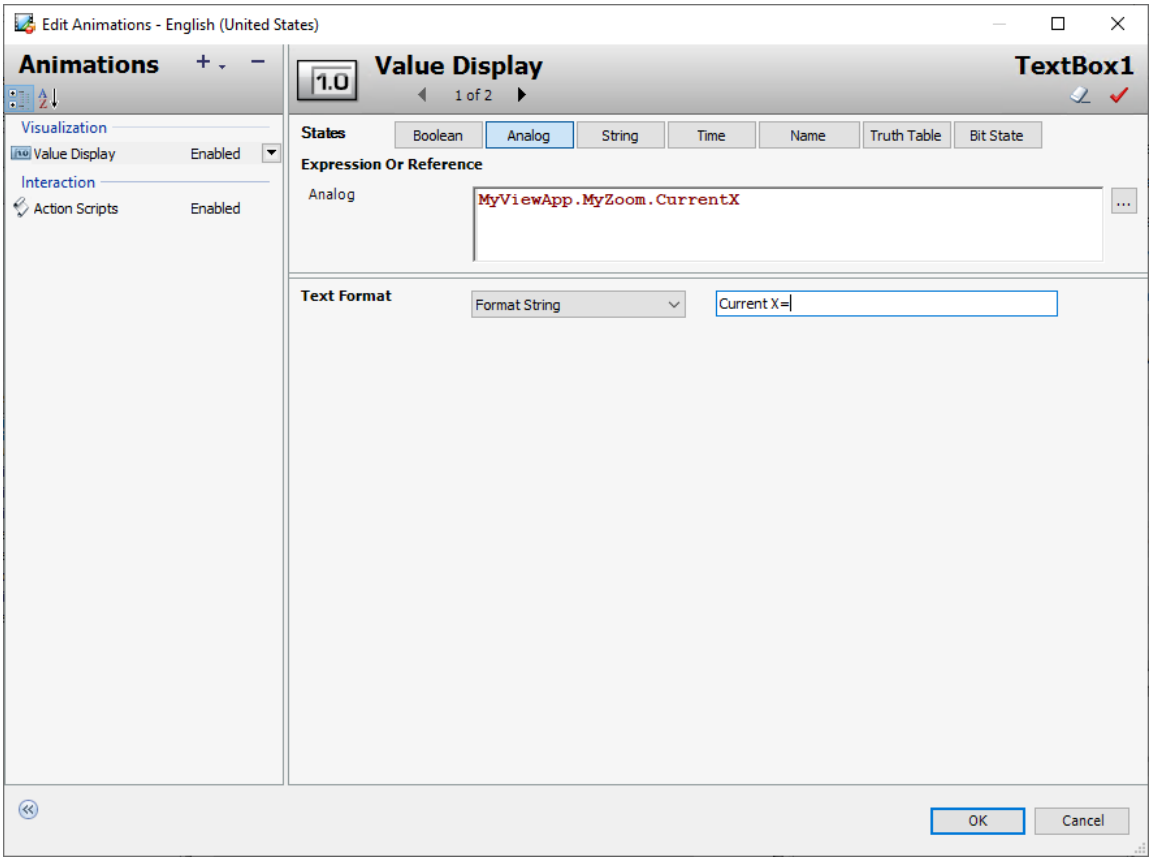
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Create Value Display Animation to Show Current Graphic Coordinates

You may want to show the current zoom area coordinates of your graphic during runtime. The following figure shows an example of Value Display animation that shows the current X coordinate value of the center point of a zoom area. You can also use Value Display animation to show the current zoom level of the graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Monitoring and Showing Quality and Status

You can configure your graphic to show non-good status and quality of attributes or tags in different ways:

- A status element shows a specific icon depending on the quality and status of configured attributes, tags or elements.
- The text, fill, or line appearance of elements is overridden depending on the quality and status of the attributes and tags they reference.
- Elements are drawn with an outline depending on the quality and status of the attributes they reference.

Quality and status elements might not be supported by all HMIs. Refer to "Working with the Industrial Graphic Editor" in your HMI help for more information.

Using Status Elements

Status elements show a specified graphic depending on the quality and status of:

- Attributes and tags configured for specific animated elements.
- One or more specified attributes or tags.

You can assign status elements to an animation in three steps:

1. Draw the status element on the canvas.
2. Associate the status element with animated elements on the canvas and/or attributes that provide the quality and status data to be monitored.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3. If needed, configure the appearance of the status element.

Drawing the Status Element on the Canvas

You can easily place a status element on the canvas to show an icon that indicates quality and status of attributes or tags contained in selected animated elements and/or specified attributes and tags.

You do this as you would with any other element. For more information, see Drawing and Dragging Elements.

Configuring the Status Element

You can associate the status element with:

- Animated elements that use attributes and tags that provide the quality and status that is to be monitored.
- Attributes and tags that provide the quality and status to be monitored.

In both cases, the appearance is set by the settings in your application. For example, in AVEVA ViewApps it is set in the **Quality and Status** tab of the **Configure Galaxy Style Library** dialog box.

For more information on how to configure this animation, see Configuring Animation for a Status Element.

Setting the Appearance of a Status Element

You can set the appearance of a status element depending on the quality and status of its referenced attributes and tags and/or attributes and tags used in its referenced elements.

You can also preview the appearance of a status element. For more information, see Previewing all Status Appearances, Previewing all Status Appearances.

You can reset the appearance of a status element to its default. For more information, see Resetting an Override Appearance to its Default, Resetting an Override Appearance to its Default.

Overriding Element Appearance Depending on Quality and Status of its Attributes

You can configure any animated element to appear differently depending on the quality and status of its associated attributes and tags if your HMI application supports it.

For animated elements, you can:

- Override the appearance of the text font, style, and blinking.
- Override the appearance of the fill style and blinking.
- Override the appearance of the line style, weight, pattern, and blinking.
- Preview all status appearances in one dialog box.
- Reset the status appearances to their defaults.
- Use an outline to indicate a specified status or quality.

Note: Instead of overriding the appearance of elements on the canvas, you can use a status element. The status element shows an icon representing quality and status of monitored attributes and tags.

Overriding the Text Appearance of Elements to Indicate Non-Good Status or Quality

If your application supports it, you can configure the text appearance of animated elements with attributes that have non-good status or quality are overridden with a specific text appearance.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Overriding the Fill Appearance of Elements to Indicate Non-Good Status or Quality

If your HMI application supports it, you can configure the fill appearance of animated elements with attributes that have non-good status or quality are overridden with a specific fill appearance.

Overriding the Line Appearance of Elements to Indicate Non-Good Status or Quality

You can configure a Galaxy so that the line appearance of animated elements with attributes that have non-good status or quality are overridden with a specific line appearance.

To override the line appearance of elements specified by a Status element

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select **Enable Quality and Status Display**.
5. Select the edit icon for a status or quality.
6. Click the **Line** tab.
7. To override the line pattern, select the **Line Pattern** checkbox and select a line pattern from the adjacent list.
8. To override the line weight, select the **Line Weight** checkbox and type a new line weight in the adjacent box. Alternatively, you can increment or decrement the line weight by clicking the adjacent boxes.
9. To override the line color:
    a. Select **Line Color** checkbox.
    b. Click the color box.
    c. Select a line color from the **Select Line Color** dialog box. For more information, see Setting Style.
10. To override the line blink behavior:
    a. Select **Blink**.
    b. Select a blinking speed from the **Speed** list (Slow, Medium, or Fast).
    c. Click the color box.
    d. Select a line blink color from the **Select Blink Color** dialog box. For more information, see Setting Style.
11. Click **Apply**.

Adding Outlines to Elements to Indicate Non-Good Status or Quality

You can configure the Galaxy so that animated elements with attributes that have non-good status or quality are shown with an outline.

To add outlines to elements to indicate non-good status or quality

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select **Enable Quality and Status Display**.
5. Click the **Outline (OI)** tab.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

6. Select the edit icon for a status or quality.
7. Select **Show Outline**.
8. To set the line pattern, select **Line Pattern** and select a line pattern from the dropdown list.
9. To set the line weight, select **Line Weight** and enter a line weight. Alternatively, you can increment or decrement the line weight by clicking the adjacent boxes.
10. To set the line style:
    a. Click the color box next to **Line Color**.
    b. Select a line color from the **Select Line Color** dialog box. For more information, see Setting Style.
11. To set the line blink behavior:
    a. Select **Blink**.
    b. Select a blinking speed from the **Speed** list (Slow, Medium, or Fast).
    c. Click the color box.
    d. Select a line blink color from the **Select Blink Color** dialog box. For more information, see Setting Style.
12. Click **Apply**.

Previewing all Status Appearances

You can preview the appearance of all status overrides by showing the **Status Legend** dialog box.

To preview all override appearances

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Click **Preview Legend**. The **Status Legend** dialog box appears.
5. Click **Close**.

Resetting an Override Appearance to its Default

For any status, you can reset the default appearance:

- Text, fill, and line overrides.
- Outline settings.
- Status element settings.

To reset a status or quality to its default appearance

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select the edit icon for a status or quality.
5. Click **Reset to default**. All text, fill, and line overrides, status element icons, and outline settings are reset to their defaults.

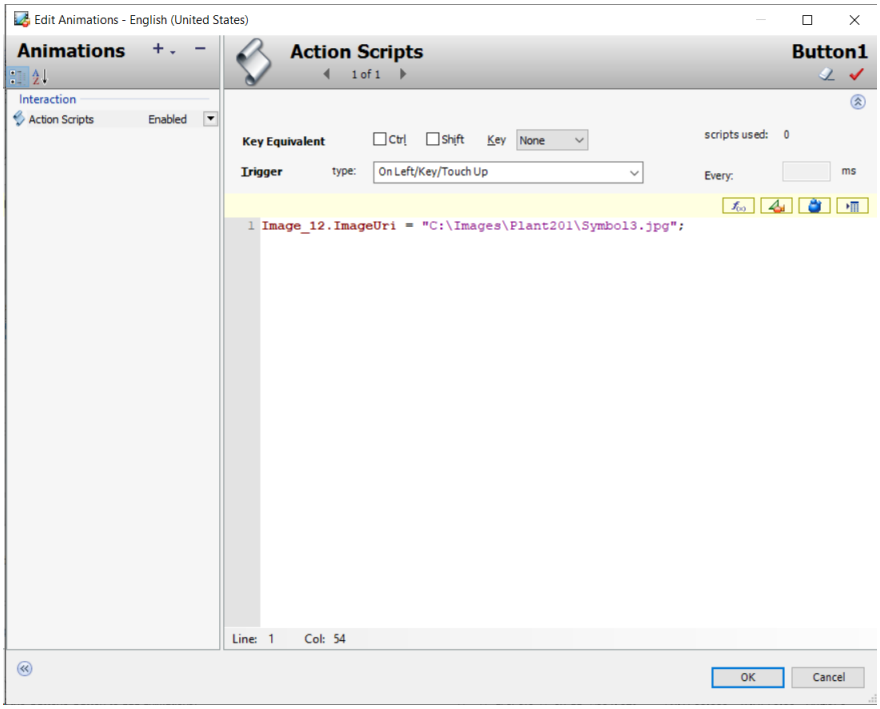
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Change images at runtime

You can use the ImageUri property in an image scripts to change an image at runtime through scripting. This property is only available for image scripts. The syntax is:

<*owningImage*>.ImageUri = <*path to image to be shown*>

See the example below.



Use animation

You can use animations to change the appearance of graphic elements at runtime. Animations are driven by data that comes from attribute values and expressions as well as element properties.

You can use:

- **Visualization animations** such as visibility, fill style, line style, text style, blinking, percent fill horizontal, percent fill vertical, horizontal location, vertical location, width, height, orientation, value display or tooltip.
- **Interaction animations** such as disable, user input, horizontal slider, vertical slider, pushbutton, action script, show graphic or hide graphic.
- **Element-specific animations** for the Status element and Windows common control elements.

Each element in your Industrial graphic can have one or more animations. You can disable and enable individual animations. You can also cut, copy and paste animations between elements. Only animations supported by the target element are pasted.

You can also substitute references and strings in animations.

For detailed information and instructions on animations, see Animate graphic elements in the Industrial Graphic Editor help.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: Not all animations are available for all element types. Some animations do not make logical sense, such as line style with a text element. You cannot select or copy these animation combinations.

Create and manage graphics

You can configure Industrial Graphics and Situational Awareness Library graphics to visualize data in an AVEVA OMI ViewApp.

You use the Industrial Graphic Editor to create graphics from basic elements, such as rectangles, lines, and text elements. You can also use the Industrial Graphic Editor to embed and configure Industrial Graphics from the Visualization folder library of graphics .

After you create a graphic, you can embed it into another graphic or place it in a layout pane to be displayed at runtime.

You can embed a graphic in a template or instance of an object, or into an InTouch window using WindowMaker, providing several ways to visualize object-specific information quickly and easily. Embedding a graphic in a template means that you can update one graphic and cascade the changes throughout your ViewApp.

Depending on your development requirements, you can select where and how to store Industrial Graphics.

- Store graphics in the Visualization folder if you want to define them as a standard that you can re-use, such as a generic valve graphic. You can store Industrial Graphics here if you want to use them only in the InTouch HMI or in AVEVA OMI VIewApps.
- Store graphics as AutomationObject templates if you want to use them in multiple instances at runtime. For example, you can create a valve graphic contained in an AutomationObject template that represents the functionality of a valve type on your plant floor.
- Store graphics as AutomationObject instances if you want to use them in only one specific object instance. For example, an AutomationObject instance that can be assigned a very specific piece of machinery as a graphic.

The topics in this section discuss some general topics and describe some actions which have aspects specific to OMI. For general information and procedures about creating and managing graphics, see Create and manage industrial graphics in the Industrial Graphic Editor help.

Create a graphic

You can create a new graphic:

- In the Visualization folder, for generic graphics that you frequently use in different situations. For example, a valve graphic.
- From a template. Do this if the graphic applies to any object instance created from that template. An example is a graphic representing a type of tank when your facility has multiple tanks of that type.
- From an object instance. Do this if the graphic is particular to that object and you are unlikely to re-use it in any other situation.

Creating a new graphic adds it to your graphics library. You can then manage and use the graphic. For example, you can:

- Edit the graphic

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Rename it
- Copy it
- Create new folders -- groups of related graphics, such as ones to use in a particular application or that represent related devices
- Move graphics between folders
- Rename folders

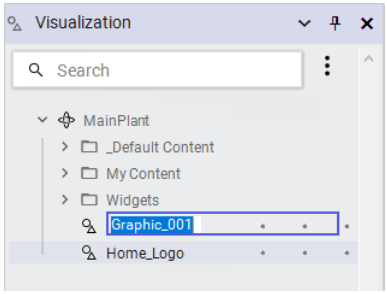
Create graphics in the visualization folder

You can create a new graphic in the Visualization folder. It is then listed in the Visualization folder with a proposed default name. You can:

- Rename the graphic.
- Move the graphic.
- Edit the graphic with the Industrial Graphic Editor.

To create a new graphic from the IDE

1. On the **Home** ribbon, in the **Create** area, select **Graphics**. The **Visualization folder** appears and a new graphic is listed.



Note: You can also press **CTRL + SHIFT + H** to create a new Industrial graphic or right-click in the **Visualization folder** and select **New** and **Graphic** from the shortcut menu.

2. Rename the graphic.

   Names must be unique within the entire Visualization folder hierarchy. Valid characters for graphic names include alphanumeric characters, #, and _ (underscore). Graphic names cannot contain spaces and the graphic name cannot begin with the $character.

3. Double-click the graphic name. The Industrial Graphic Editor appears.
4. Draw the graphic.

Create graphics in AutomationObject templates

You can create a graphic from the Object Editor **Attributes** tab in an AutomationObject template. Creating a graphic this way automatically associates the new graphic with the AutomationObject.

To create a new graphic for an AutomationObject template

1. Open the AutomationObject template. Click the **Attributes** tab. Any local and inherited graphics are listed.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. In the **Content** area, click the **Add** (+) icon.
3. Give the new graphic a name. Names must be unique. Valid characters for graphic names include alphanumeric characters, $, #, and _ (underscore). Graphic names cannot include spaces and cannot begin with the $ character.
4. If needed, type the description of the graphic in the **Description** box.
5. Make sure the graphic is selected and click the **Edit** icon. The Industrial Graphic Editor opens.
6. Draw the graphic.

Create graphics in AutomationObject instances

You can create a graphic from the **Attributes** tab in an AutomationObject instance. Creating a graphic this way automatically associates it with the AutomationObject instance.

Note: AutomationObjects can also inherit graphics from their parent template. You can only view an inherited graphic in read-only mode. Inherited graphics can only be removed or edited in the template where they are defined.

To create a new graphic for an AutomationObject instance

1. Open the AutomationObject instance. Click the **Attributes** tab. Any local and inherited graphics are listed.
2. In the **Content** area, click the **Add** (+) icon.
3. Give the new graphic a name. Names must be unique. Valid characters for graphic names include alphanumeric characters, $, #, and _ (underscore). Names cannot include spaces and cannot begin with the $ character.
4. If needed, type the description of the graphic in the **Description** box.
5. Make sure the graphic is selected and click the **Edit** icon. The Industrial Graphic Editor opens.
6. Draw the graphic.

Graphic Change Propagation

If the graphic is hosted by the Visualization folder and edited, all graphics hosted by AutomationObject templates and instances that contain embedded instances of this graphic are also updated.

If the graphic is hosted by an AutomationObject and edited, all graphics hosted by derived AutomationObjects are also updated.

Organizing Graphics

Use the library available to your HMI/SCADA software to organize your graphics by creating a folder hierarchy as you would with files and folders in Windows Explorer. You can move graphics around within the folder hierarchy and copy or rename them as needed.

For more information about organizing graphics, see the help file for your HMI/SCADA software.

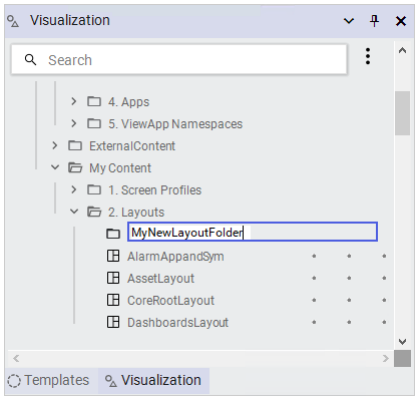
Create folders in the visualization folder

You can create folders in the **Visualization folder** to organize your graphics. For example, you can create a folder called "Valves" to store different valve graphics.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To create a folder in the Visualization folder

1. Open the **Visualization folder**.
2. Select the folder under which you want to create a new folder. Select the Galaxy name if there are currently no folders.
3. On the **Home** ribbon, in the **Create** area, select **Folder**, then **VIsualization folder**. A new folder is created with a default name.
4. Rename the new folder as needed.



Use graphic thumbnails

When you place your mouse cursor over an Industrial Graphic in the Visualization folder, a thumbnail view of the graphic's displays. This allows you to preview the graphic without opening it.

The graphic thumbnail updates automatically when a change is made. However, thumbnails of graphics that contain embedded instances of the base graphic will not automatically update. You must manually do so by right clicking the graphic and selecting **Update Thumbnail**.

You can update graphic thumbnails from any of the following levels:

- individual graphic
- folder level
- update all graphics from the root of the Visualization folder

The time it takes to refresh a folder or the Visualization folder is proportional to the number of graphics each contains. This may take several minutes.

Note: If you have imported a newer version of a client control already embedded in a graphic, restarting WindowMaker and updating the graphic thumbnail will not update the contents of the newly imported control. To update the graphic with the contents of the new control, edit and save the graphic. The graphic thumbnail will then reflect the contents of the new control.

Move graphics between folders

You can move graphics from one folder in the Visualization folder to another. Moving graphics between folders does not affect their functionality.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To move graphics between folders

1. Open the Visualization folder.
2. Locate the graphic you want to move.
3. Do either of the following:
   - Drag the graphic to the folder you want to place it in. The graphic moves to the new folder.
   - To move the graphic to the top level in the Visualization folder hierarchy, drag the graphic to the Galaxy name icon.

Rename graphics

You can rename a graphic at any time. Renaming a graphic does not affect its functionality.

Graphic names must be unique within the entire hierarchy of the Visualization folder.

To rename a graphic

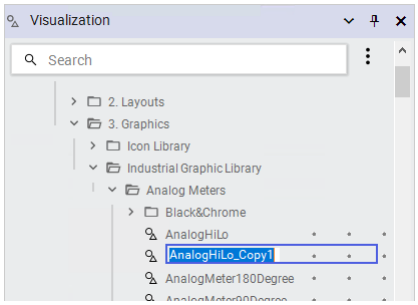
1. Open the Visualization folder.
2. Select the graphic to rename.
3. In the **Edit** area of the **Home** ribbon, click **Rename**. The graphic name is in edit mode.
4. Type a new unique name for the graphic and click **Enter.**

Copy graphics

You can create copies of graphics in the Visualization folder. The copies are suffixed with "_Copy1", "_Copy2", and so on.

To create a copy of a graphic

1. Select the graphic to copy.
2. In the **Edit** area of the **Home** ribbon, click **Duplicate**. A copy of the graphic is created.



3. If needed, type a new name for the graphic.

Delete a graphic

You can delete a graphic that you no longer need. Deleting a graphic removes it completely from the Industrial Graphic Editor.

- When you delete a graphic, you are shown where the graphic is used. This allows you to understand the impact of deleting the graphic before you actually delete it.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- You cannot delete a graphic that someone else has open for editing or left checked out.
- If you delete a graphic that is embedded in another graphic, it shows a Not Found message.

To delete a graphic

1. Select the graphic or graphics that you want to delete.
2. On the **Home** ribbon, in the **Edit** area, click **Delete**.

   The **Confirm Delete** dialog box appears, showing the number of objects that reference each graphic you are deleting.
3. To see the places that a graphic is being used, click the down arrow for that graphic at the right of the **Referenced** column.
4. To delete the graphic(s), click **Delete**.

Viewing a Graphic in Read-Only Mode

You can view a graphic in read-only mode if you don’t want to edit it, or if it is checked out by somebody else.

If you open a graphic in read-only mode, you have access to all functions in the Industrial Graphic Editor that don't change the graphic.

To view a graphic in read-only mode

1. Select the graphic that you want to view in read-only mode.
2. Click **Open Read-Only**. The selected graphic opens in the Industrial Graphic Editor.

Rename graphic folders

You can rename a graphic folder at any time. Renaming a folder does not affect the functionality of any graphics it contains.

To rename a folder in the Visualization folder

1. Open the Visualization folder.
2. Select the folder to rename.
3. On the **Home** ribbon, in the **Edit** area, click **Rename**. The folder name is in edit mode.
4. Type a new unique name for the folder and click **Enter**.

Delete graphic folders

You can delete a graphic folder in the Visualization folder at any time.

You can only delete folders that do not contain any graphics. Move the graphics to another folder or delete them before deleting the folder.

To delete a folder from the Visualization folder

1. Open the Visualization folder.
2. Select the folder to delete.
3. On the **Home** ribbon, in the **Edit** are, click **Delete**. When a confirmation message appears, click **Yes**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Move graphic folders

You can move a graphic folder within the Visualization folder. If you move a folder, all graphics and folders it contains are also moved.

To move a folder in the Visualization folder

1. Open the Visualization folder.
2. Locate the folder to move.
3. Do either of the following:
   - Drag the folder to the folder you want to place it in. The folder is moved to the new location.
   - To move the folder to the top level in the hierarchy, drag it to the Galaxy name icon.

Customize graphic folders

You can hide or show graphic folders after creating them. You can do this only for first-level folders--those directly beneath the Galaxy in the hierarchy.

To customize graphic folders

1. On the ribbon, select **Galaxy**.
2. Select **Configure**, then **Integrated development environment**, then **Folders**.
3. In the **Folders** pane, click **Visualization**.
4. Do one or more of the following:
   - To hide or show all folders, clear or select the checkbox for **Folder list**.
   - Clear the checkbox for any folder you want to hide.
   - If a folder is hidden and you want to display it, select its checkbox.
5. When you are done with your changes, select **Save**.

   Folders are shown or hidden, depending on your settings.

Creating Multiple Configurations of a Graphic

The Symbol Wizard Editor is a feature of the Industrial Graphic Editor to create multiple configurations of a graphic. A graphic configuration represents different visual or functional variations of a graphic.

Graphic configurations are created using layers containing associated graphic elements, custom properties, and named scripts. Based on graphic properties and possible values of these properties, rules are applied that specify when a layer is part of a graphic configuration.

For more information about using Symbol Wizards to create multiple configurations of a graphic, see Working with Symbol Wizards.

Understanding Visual and Functional Graphic Configurations

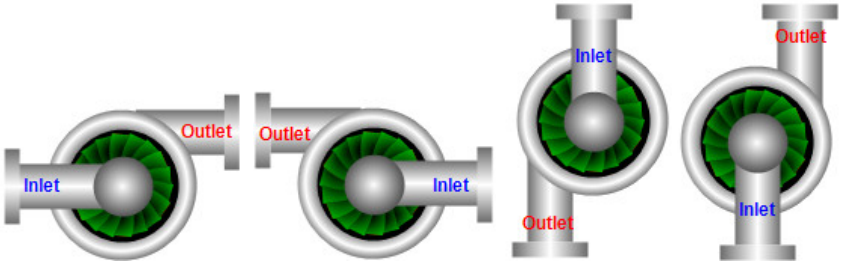
Standard Industrial Graphics show reasonably realistic views of process objects. These graphics can be modified with the Symbol Wizard to incorporate multiple visual configurations in a graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Situational Awareness Library graphics are a special set of Industrial Graphics that are available for use in your HMI/SCADA software, and are designed using the Symbol Wizard Editor. Their design cannot be changed. However, you can select Wizard Options from the Symbol Wizard Editor to select the configurations that are incorporated into each graphic's design.

Visual Graphic Configurations

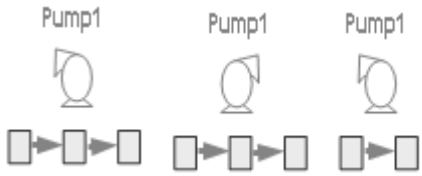
Using an example of a centrifugal pump with separate inlet and outlet pipes, there are four practical visual configurations. The pump's blade housing is common and appears in all possible configurations. But, the pump's inlet and outlet pipes can be placed at the left or right in a horizontal direction or at the top or bottom when the pump is oriented vertically.



Orientation is the visual property that identifies the different configurations of a pump graphic. The attributes associated with the Orientation property are left, right, top, and bottom.

Functional Graphic Configurations

Situational Awareness Library graphics include functional properties in addition to visual properties. For example, a multi-stage pump graphic includes a Wizard Option to select either a five-stage, three-stage, or single stage pump in addition to a visual Orientation property to select left or right pump configurations.



About graphic scaling

By default, a graphic attempts to scale to the size of its hosting pane and maintain the graphic's horizontal and vertical aspect ratio. The size of a graphic created using the Industrial Graphic Editor is determined by a bounding rectangle around all graphic elements and embedded graphics that compose the graphic.

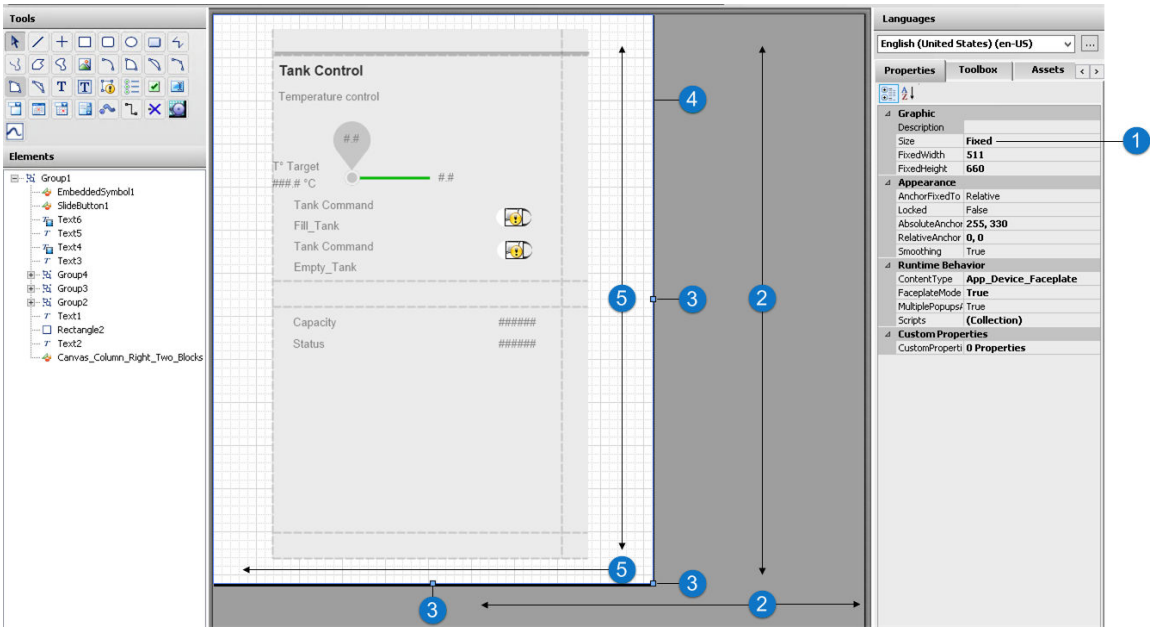
When a graphic is placed in a layout pane, if the graphic's bounding rectangle exceeds the size of its hosting pane, the bounding rectangle becomes smaller by reducing the original size of its contents. If the graphic's bounding rectangle is smaller than the pane size, the bounding rectangle increases and enlarges the original size of its contents.

Graphics that have a defined fixed size render in runtime identically to graphics without a fixed size. A graphic's size adjusts to the size of the pane while maintaining its aspect ratio as appropriate. Most graphics that have a configured fixed size are typically placed in a specific target pane of a known size that enables them to be the rendered exactly as designed in the Industrial Graphic Editor without any required resizing during configuration or runtime.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About the Industrial Graphic Editor

After placing a graphic or graphic elements on the canvas of the Industrial Graphic Editor, you can set several properties to specify a fixed size for the graphic. The following screen shot shows how to adjust the fixed size of a graphic.



1	The **Size** property is set to **Auto** by default, which automatically sizes graphics. Set **Size** to **Fixed** to create fixed size graphics. When set to **Fixed**, the **FixedWidth** and **FixedHeight** properties appear, which show the dimensions of the fixed area of a graphic in pixels.
	You can enter the graphic's width and height directly in the **FixedWidth** and **FixedHeight** fields.
	• **FixedWidth**
	  • Default value = 500
	  • Maximum value = 7680
	  • Minimum value = 0
	• **FixedHeight**
	  • Default value = 500
	  • Maximum value = 4320
	  • Minimum value = 0
	You can also change the width and height of the graphic's fixed size area by moving the boundary with its grab handles.

2	The gray area represents that part of the Industrial Graphic Editor canvas that is outside the boundary of the fixed size area of the graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3	Grab handles on the fixed size boundary can be moved to change the size of the fixed area of a graphic. The **FixedWidth** and **FixedHeight** properties update to show the current width and height when you move the boundary. Grab handles are placed at the following locations on the fixed size boundary: • Top right corner • Center right side • Bottom right corner • Bottom center
4	The thick black line shows the boundary between the Industrial Graphic Editor canvas within the fixed size area of the graphic set by the graphic's **FixedWidth** and **FixedHeight** properties and the canvas area outside the fixed dimension area.
5	The white area of the Industrial Graphic Editor canvas is within the fixed size boundary of the graphic.

You can place graphic elements from the Industrial Graphic Editor's **Tools** section or embed graphics as you would normally do when the **Size** property is set to **Auto**. As long as the graphics elements or embedded graphics are placed inside the fixed size boundary, the graphic will fit the size of a pane perfectly. Graphics within the graphic maintain the same size, scale, spacing, and white spaces around the graphics on the canvas specified during design time.

Create a fixed-size graphic for a ViewApp pane with a known size

This topic describes the general sequence of steps to create a fixed size graphic designed to fit a ViewApp pane with a known size. This workflow is typical of creating content that will fit exactly within a pane while editing a ViewApp with the ViewApp Editor.

Important:

The size of a graphic in runtime is affected by the resolution of the screen that will show the running ViewApp and the specified width and height of the layout pane in which the graphic is placed. Also, the layout **Display Mode** property includes a set of options that determines how content is resized during runtime to fit within the size of a layout pane.

The following procedure describes the steps to create a fixed size graphic that will fit within a pane of a known size. You should create a fixed size graphic whose width and height matches the resolution of the screen profile.You should set the the dimensions of the hosting pane to the same size as the graphic.

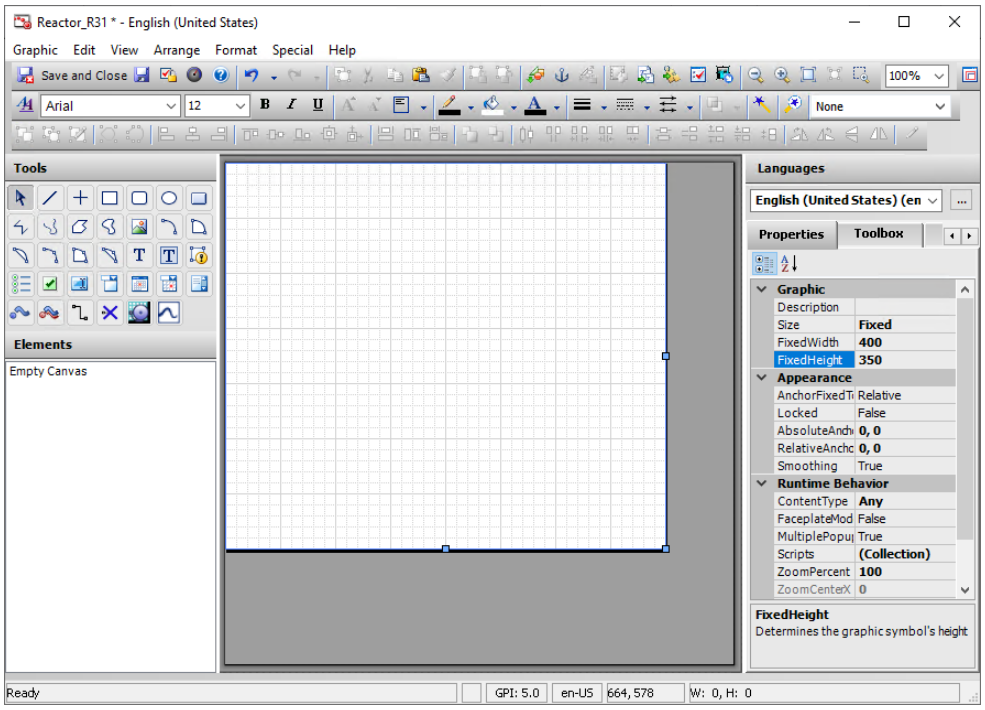
To create a fixed size graphic that fits within a pane of a known size

1. Open the ViewApp in the ViewApp Editor that will contain the fixed size graphic.
2. Right-click the pane that will contain the fixed size graphic and select **Create Graphic**.

   A dialog box appears requesting a name for the graphic to be created.
3. Assign a name to the graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The Industrial Graphic Editor opens. and the white portion of the canvas shows the dimensions of the pane where you created a graphic. The **Size** property is set to **Fixed** and the **FixedWidth** and **FixedHeight** properties show the dimensions of the pane in pixels.



4. Place graphic elements and graphic to create a graphic.
5. Save and close the graphic to return to the ViewApp Editor.

   The graphic you created appears in the pane where you created the graphic. The graphic should fit into the pane at the same size as it was designed in the Industrial Graphic Editor.
6. Save the ViewApp and deploy it.
7. Open the deployed ViewApp

   In runtime, the ViewApp will show the graphic you created at the same size as it was originally designed.

   The rendered graphic shows:

   - The graphic fits in the pane at the same size as it was designed in the Industrial Graphic Editor.
   - The graphic's size does not change in runtime.
   - Any blank spaces around the graphic are maintained and appear in the graphic during runtime.

Create a fixed-size graphic for a ViewApp pane size

This topic describes the general procedure to create a fixed size graphic that will fit a pane of a known size even if the pane does not exist when the fixed size graphic is created.

This workflow is typically the case where two different employees work on a ViewApp to update it to reflect changes in a production environment. The first employee is usually a user who creates a fixed size graphic. A second employee will drag and drop the graphic in a pane while editing the ViewApp with the ViewApp Editor.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To create a fixed size graphic that will fit a pane with a known size

1. The first employee creates a new graphic and opens it in the Industrial Graphic Editor.

   The entire canvas area of the Industrial Graphic Editor is enabled to place graphic elements and graphics.
2. Configure the graphic to a known pane size.
   a. Select the **Graphic** property **Size** and set it to **Fixed**.

      The **FixedWidth** and **FixedHeight** properties appear after the **Size** property is set to **Fixed**.
   b. Update the values of **Fixed Width** and **Fixed Height** to the pixel dimensions of a pane with a known size.

      You can also move the grab handles on the fixed size boundary to set the dimensions of the graphic.
3. Place the graphic elements and other graphics within the fixed size area of the graphic.
4. If you want white space around the graphic, adjust the width and height of the graphic to show a white margin around the graphic elements and graphics that compose your graphic.
5. Save and close the graphic.
6. The second employee creates a layout to be used in the ViewApp with a pane that is the same size as the fixed size graphic.
7. Open the ViewApp with the ViewApp Editor.
8. Assign a Screen Profile and the layout containing the pane to the ViewApp.
9. Select a screen that contains the layout with the pane of a known size.
10. Select the fixed size graphic created by the first employee and drag and drop it in the pane.

    The configured graphic appears in the selected pane of the ViewApp when it is edited by the ViewApp Editor. The graphic preview shows:

    - The graphic fits in the targeted pane with the same size as it was designed in the Industrial Graphic Editor.
    - If necessary, the graphic can scale larger or smaller to fit the pane.
    - Any white space placed around the graphic while it was created is maintained when the graphic is placed in a pane.
11. Save the ViewApp and deploy it
12. Open the running ViewApp.

    The running ViewApp shows the fixed size graphic in the target pane at the same size as it was created in the Industrial Graphic Editor and configured in the ViewApp Editor. The rendered fixed size graphic shows:

    - The graphic fits in the pane with the same size as it was designed in the Industrial Graphic Editor
    - Any white space placed around the fixed size graphic is maintained.

About fixed-size graphics when pane is resized

Graphics created for a layout pane obtain size dimension from the pane's size by default. The values assigned to the **FixedHeight** and **FixedWidth** properties automatically match the height and width of the pane. Graphics that are designed with its elements placed within the dimension boundaries will fit in the pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When the dimensions of a graphic are changed while editing in the ViewApp Editor, the following behaviors are observed:

Fixed Size Graphic Size is Greater Than Pane Size

- When the graphic size is larger than hosted pane size, elements placed outside the pane are included in the configuration view. The graphic's size is reduced to fit it entirely within the dimensions of the pane.
- When the **FixedWidth** and **FixedHeigh**t properties of a graphic match the pane size, but there are graphic elements beyond the graphic's fixed width and height boundaries, the graphic's size is reduced to fit the graphic elements outside the fixed size area within the dimensions of the pane.

Fixed Sized Graphic Size is Smaller Than Pane Size

When a graphic's fixed size is smaller than the hosted pane size, the graphic's size is increased to fit the pane size. The sizes of all of the graphic's graphic elements are increased.

Fixed Size Graphic Embedded in Another Graphic

- The embedded fixed size graphic maintains its size configured at design time.
- There will not be any clipping of any white space provided around primitives or embedded graphics present in the graphic.
- There will not be any scaling of graphics. An exception to this is if the embedded graphic was resized at design time, the graphic will look scaled.

About fixed-size graphics during runtime

The following list summarizes the key behaviors of fixed size graphics during runtime.

- Graphics that have the **Size** property set to **Fixed** render all graphic elements placed within the fixed size boundary at their configured fixed size at runtime.
  - Graphics do not change size or aspect ratio within the pane that is hosting the graphic.
  - Graphics configured to fit exactly within a hosting pane render to fit exactly within the size of a pane. The graphic elements in the graphic render at the same size and placement as designed in the Industrial Graphic Editor.
  - The white space around primitives or embedded graphics present in the graphic maintains its size and placement within the graphic.
- Fixed size graphics change size or aspect ratio to fit the hosting pane during runtime in the following cases:
  - When the graphic's size is larger or smaller than the hosting pane.
  - When graphicsize is the same as the hosting pane but there are graphics that are placed outside the fixed size boundary configured for the graphic. This includes graphics that are placed on negative coordinates.
  - When graphic size is the same as the hosting pane, but graphics that are placed on the boundary of the graphic marked by the size configured for the graphic.
- Embedded graphics follow the same behavior as standard fixed size graphics.

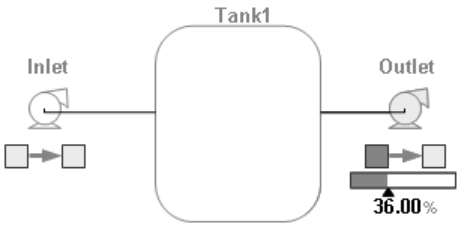
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

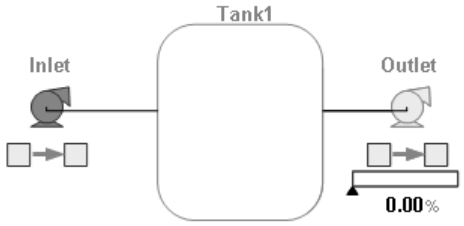
Create faceplate graphics

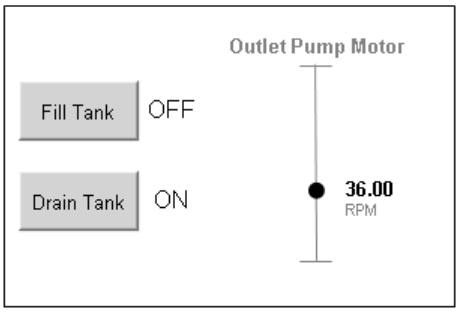
A faceplate graphic typically shows visual elements for operators to monitor and manage equipment represented by graphics in a running application. Operators can select buttons in a faceplate graphic to turn equipment on or off. A meter or other type of graphic can be embedded in a faceplate to show the current measured value of a selected piece of equipment during runtime.

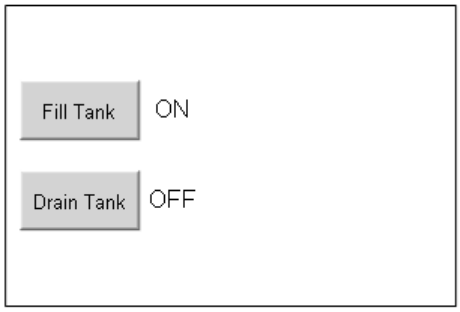
Graphics that have been enabled for faceplate compatibility mode enable operators to manage a set of similar graphics using a single faceplate. Instead of creating a separate faceplate for each graphic, a faceplate can be adapted for different graphic configurations by showing or hiding graphic elements or embedded graphics in the faceplate itself.

For example, you can create a faceplate graphic to fill or drain a tank. The faceplate graphic shows an embedded RPM meter when a SA_MultiStagePump graphic is configured as an outlet pump with a variable speed motor controller. The same faceplate can be used without a meter when another SA_MultiStagePump graphic represents a fixed speed inlet pump that runs without an output controller.









In the example of the embedded RPM meter in the faceplate graphic to fill or drain the tank shown above, two conditions must be met to hide the RPM meter:

- The embedded RPM meter graphic's **FaceplateMode** property must be set to true.
- The embedded RPM meter graphic must have a custom property with a reference to an invalid value.

Important: Only AVEVA OMI ViewApps support faceplate compatibility mode. Managed and traditional InTouch applications do not support faceplate mode.

About faceplate graphics creation

This section describes the general steps to create a faceplate graphic, which consist of the following:

- Determine the graphics on a faceplate that should be shown or hidden to support different graphic configurations.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Determine the valid and invalid reference states that will show or hide graphics on a faceplate.
- During design time, set faceplate compatibility mode at the root level or canvas of the graphic.

Determine the Graphics that Run in Faceplate Compatibility Mode

Consider the graphics that will represent the equipment to be controlled and monitored from a faceplate. Determine the unique qualities of each piece of equipment that will need to be shown on the faceplate by a graphic. These unique qualities are the object attributes, such as a PV attribute or ON/OFF attribute. The presence of these attributes based on the configuration of the graphic determine how the faceplate is built.

Determine the Valid and Invalid Reference States that Show or Hide Faceplate Graphics

Determine the reference states when faceplate graphics need to be shown or hidden for all configurations of equipment represented by the graphics.

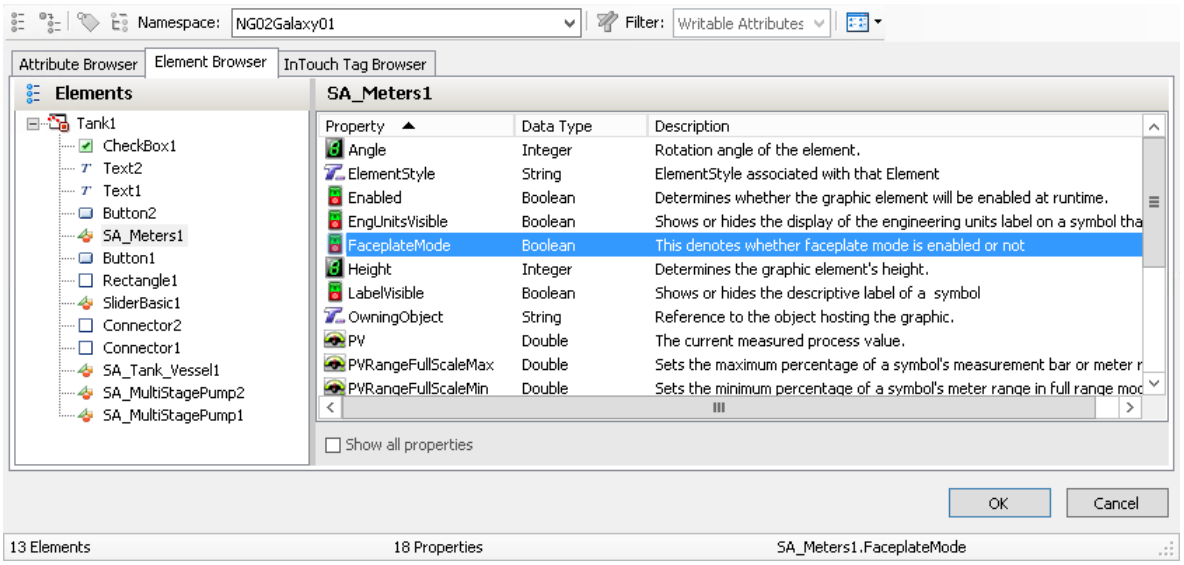
In the example of using a faceplate to manage fixed-speed and variable speed pumps, a meter should be hidden when an operator is managing a fixed-speed pump. When running a fixed speed pump, a reference to a meter's **PV** custom property is invalid and will hide the meter when it is running in faceplate compatibility mode.

Set Faceplate Compatibility Mode

Set faceplate compatibility mode for those graphics that appear on a faceplate that must be hidden to support a specific equipment configuration. Faceplate compatibility mode can be set on graphic groups and the root level or canvas of a graphic. For more information about the inheritance rules of faceplate compatibility mode for grouped or nested graphics, see About graphics visibility during runtime.

In the example of using a faceplate to manage fixed-speed and variable speed pumps, the meter should be hidden on the faceplate when an operator is managing a fixed-speed pump.

Graphics include the Boolean **FaceplateMode** property. When a graphic's **FaceplateMode** property is set to True, the graphic is hidden when it contains an invalid reference to a target. In the following example, a CheckBox graphic element toggles the state **FaceplateMode** property of a meter graphic to enable or disable faceplate mode compatibility.



When the operator selects a fixed-speed motor from the faceplate, the meter's **PV** custom property reference becomes invalid, which will hide the meter when its **FaceplateMode** property is True.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About graphics visibility during runtime

Two conditions determine if a graphic can be shown or hidden during runtime using faceplate compatibility mode. The following table shows the conditions to show or hide a graphic on a faceplate:

FaceplateMode Property	Valid Reference	Invalid Reference
True	Show Graphic	Hide Graphic
False	Show Graphic	Show Graphic

- The **FaceplateMode** Runtime Behavior property must be set to True to show or hide a graphic in faceplate compatibility mode. The **FaceplateMode** default value is False, which shows a graphic under all reference conditions.
- A faceplate graphic is hidden if **FaceplateMode** is True and an invalid reference is made from a graphic to a target attribute from the following:
  - **Graphic element**

    When **FaceplateMode** is True and a graphic element has an invalid reference through animation or script, only that graphic element is hidden within a graphic. If the graphic element is part of a graphic embedded into another graphic, the embedded graphic still appears although the graphic element with the invalid reference is hidden.
  - **Embedded graphic**

    When **FaceplateMode** is True and an embedded graphic has an invalid reference, then the entire graphic will be hidden. To show or hide an embedded graphic based on the value of **FaceplateMode,** a reference must be made to the entire graphic that is embedded, not just to a graphic element that belongs to the graphic.
  - **Expressions**

    Animation expressions, script expressions, or custom property expressions are not evaluated if there is at least one invalid expression.
  - **Animations**

    If an animation has an invalid reference, then the graphic associated with the animation will be hidden in faceplate mode. If an animation has a valid reference then its animation will execute. Animations with valid references are evaluated even if the graphic with which the animation is associated is hidden by faceplate mode. This is because a graphic element may be associated with several animations, some which may be valid while others are invalid.
  - **Scripts**

    Named scripts, action scripts, data change scripts value scripts, and On show/On hide scripts, do not run if a script has at least one invalid reference.
  - **Quality and status icon**

    Global status and quality ignores invalid references and does not show any status and quality indications related to them.
  - **Error and warning messages**

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When a graphic is set to faceplate compatibility, no error or warning messages are logged for invalid references. The data status graphic element is hidden if it contains an invalid reference.

- **Undeployed GR node**

  In faceplate compatibility mode, objects show a Configuration Error status when the GR node is undeployed. Any graphic that shows a Configuration Error message is hidden. The graphic becomes visible again when the GR node is redeployed.

- **Stopped App engine**

  In faceplate compatibility mode, objects show a Communication Error status when an App engine is undeployed. Any graphic that shows a Communication Error message remains visible and is not hidden.

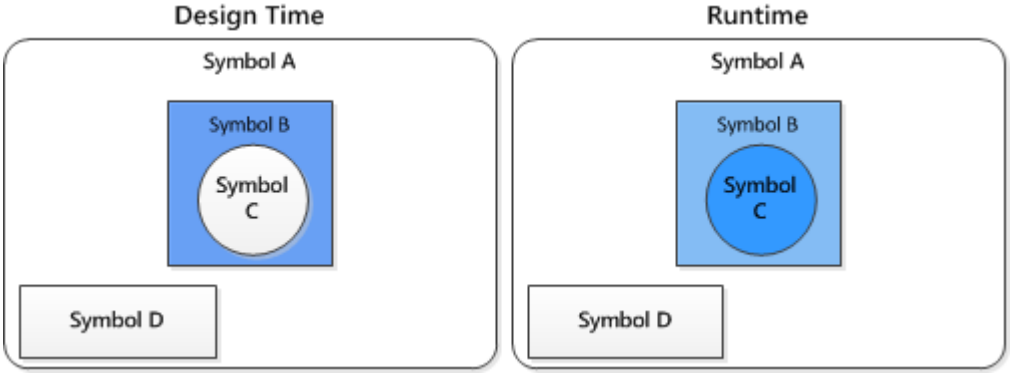
Runtime Behavior of Faceplate Mode

Graphics configured for faceplate compatibility mode follow defined inheritance rules during runtime.

Embedded Graphics

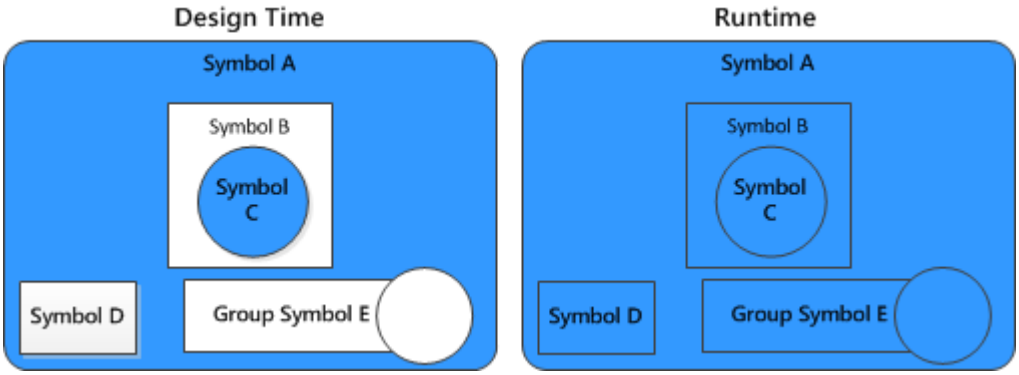
Embedded graphics inherit faceplate compatibility from the parent to child graphic during runtime. But, faceplate compatibility mode is not inherited from the child to the parent embedded graphic.

In the following figure, Symbol B is set to faceplate compatibility during design time. During runtime, Symbol C embedded within Symbol B inherits faceplate compatibility.



Graphic Groups

If a parent graphic is set to faceplate compatibility mode, then all its child embedded graphics and graphic groups inherit faceplate compatibility mode at runtime.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About settings attributes

Settings attributes typically enable or disable visibility options of Situational Awareness Library graphics and some Viewapp global settings. The Settings attributes are assigned default values and load when a ViewApp starts.

Settings attributes can be set from faceplate graphics. You can also embed the **WW_App_Function** widget in a graphic and use it to control settings attributes. The WW_App_Function widget is included in Galaxies created from the Default.cab, as part of the **ViewApp UI Library** in the Visualization folder.

The sample **$InTouchOMI_ViewApp**, which is associated with the **WWD_L_Desktop** layout, uses the WW_App_Function widget. A User Input animation has been configured to set the value of the MyViewApp.Settings.TrendDuration attribute. You can configure other attributes similarly. Refer to the sample ViewApp for more information.

Important: Settings attributes are predefined and available only for AVEVA OMI ViewApps. Settings attributes cannot be included in an InTouch application.

All Settings attributes operate in the MyViewApp.Settings namespace. The names of Settings attributes are specified with the MyViewApp.Settings prefix in the form MyViewApp.Settings.*attribute_name*.

Attribute Name	Data Type	Attribute Type	Initial Value	Description
TrendDuration	Integer	Read/Write	15	The length of a trend duration in minutes.
GridLines	Boolean	Read/Write	True	Shows or hides grid lines on a pen trend
PenTrendVisible	Boolean	Read/Write	True	Shows or hides all pen trend lines of a trend graphic
PVVisible	Boolean	Read/Write	True	Shows or hides the process value of a Situational Awareness Library graphic.
PVOptimal RangeVisible	Boolean	Read/Write	True	Shows or hides the optimal range indicator on a Situational Awareness Library graphic.
PVOperating AutoScale	Boolean	Read/Write	False	Enables or disables meter auto scaling on a Situational Awareness Library graphic.
SPVisible	Boolean	Read/Write	True	Shows or hides a Setpoint on a Situational Awareness Library graphic.
TrackerVisible	Boolean	Read/Write	True	Shows or hides a tracker line on the meter of a Situational Awareness Library graphic.
LabelVisible	Boolean	Read/Write	True	Shows or hides the label on a Situational Awareness Library graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

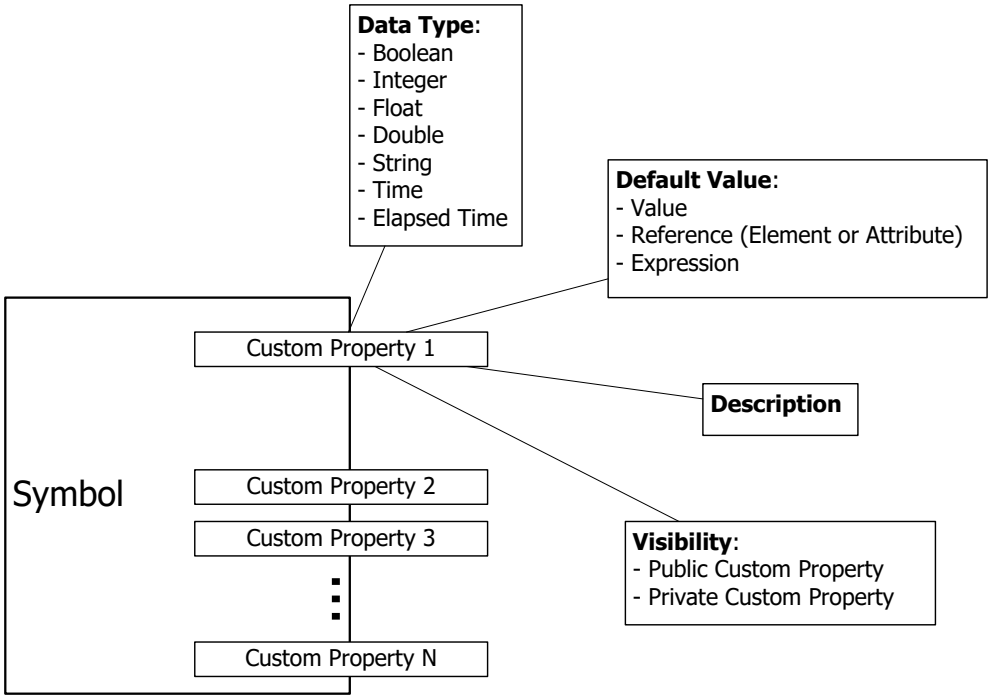
EngUnits Visible	Boolean	Read/Write	True	Shows or hides the engineering units of a Situational Awareness Library graphic.
ControllerOP Visible	Boolean	Read/Write	True	Shows or hides the Controller Output for a Situational Awareness Library graphic.
Controller OPCmd Visible	Boolean	Read/Write	True	Shows or hides the Output Command for a Situational Awareness Library graphics.
AlarmIndicator Visible	Boolean	Read/Write	True	Shows or hides an alarm indicator on a Situational Awareness Library graphic.
ChartRange Increment	Integer	Read/Write	5	Sets the precision of the incremental range of chart values shown as measurement lines of Situational Awareness Library Dashboard graphics.
SafetyVisible	Boolean	Read/Write	False	Shows or hides a safety indicator on a Situational Awareness Library graphic.
EnableDisable Sound	Boolean	Read/Write	True	Enables or disables alarm sounds via a Situational Awareness Library graphic to alert the operator of an alarm condition. To function properly, this attribute must be connected to a PLC with an integrated audio system.
ShowControl Links	Boolean	Read/Write	True	Shows or hides the control links of a Situational Awareness Library graphic.
ShowToolTips	Boolean	Read/Write	True	Shows or hides the tooltips of a graphic.
ToggleSPTag	Boolean	Read/Write	False	Toggle between a Setpoint or tagname value on a Situational Awareness Library graphic.
DisableHorn	Boolean	Read/Write	False	Enable or disable an external alarm horn via a Situational Awareness Library graphic. To function properly, this attribute must be connected to a PLC with an integrated audio system. This feature is typically used in a control room to allow the operator to troubleshoot the alarm condition without the distraction of the horn blaring.
AckAll	Boolean	Read/Write	False	Enable or disable the acknowledgement of all alarms.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

OptimizeFor Touch	Boolean	Read/Write	False	Enable or disable optimization of touch gestures at runtime.
ReadOnly	Boolean	Read Only	False	Enable or disable read-only ViewApps at runtime. If false, users have the ability to change attribute values in the ViewApp at runtime. If true, the ViewApp and its attributes are read-only. The ReadOnly property value is assigned by editing a ViewEngine object with the Object Editor. To change the value, you must first undeploy the ViewEngine.

About custom properties for graphics

You can configure and use custom properties to extend the functionality of graphics and use them in combination with InTouch tags. You can use binding with custom properties to dynamically change the reference of a custom property.



You can associate custom properties with functionality you want exposed and that you want to be reusable. You can also use custom properties to connect an embedded Industrial graphic to InTouch tags.

For example, possible uses for using custom properties are:

- A "TankLevel" custom property of type Writable Attribute can be given a value of "me.pv".
- A "MaxFillLevel" Custom Property of type Expression can be given a value of "Me.MaxCapacity - 200".

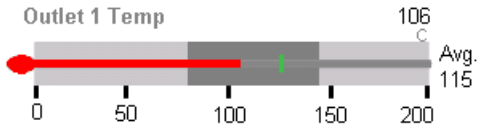
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For general information about creating and managing custom properties, see Custom properties in the Industrial Graphic Editor help. For some examples of using custom properties to show historical data, see the topics below.

Use custom properties to show historical summary data

You can add a custom property to reference historical data collected over a specified period during runtime. The Historian can transform this data to create a set of analog or state statistics that can be shown by Industrial graphic animation during runtime.

For example, consider an example of a temperature meter graphic that shows an optimal temperature range and you would like to know what the average temperature has been over the last 15 minutes. You can add a Value Display animation that shows the average temperature derived from analog temperature data saved in the Historian and referenced by a custom property.



Analog statistical summary data

A custom property analog reference can subscribe to statistics from analog data collected over a defined summary period and saved to the Historian. The following table lists the analog historical statistical data that can be specified for a custom property.

Analog Statistical Data	**Description**
Average	A time-weighted average calculated from values within a summary period. The average is calculated at the end of the summary period.
Count	A value count calculated from values within a summary period. The count is calculated at the end of the summary period.
First	The first value that occurs within a summary period based on the actual timestamp within the summary period.
Integral	An integral value calculated from values within a summary period. The integral is calculated at the end of the summary period.
Maximum	The first maximum value that occurs within a summary period.
Minimum	The first minimum value that occurs within a summary period.
PercentGood	The ratio of labeled "good" quality data to all data within the summary period. The ratio is expressed as a percentage in the range 0 to 100. PercentGood is calculated at the end of the summary period.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Analog Statistical Data	Description
StdDev	Time weighted standard deviation calculated from values within a summary period. The value is calculated using time weighted sums (Integrals) and time weighted sums of squares (IntegralOfSquares) values.
Last	The last value that occurred in the summary period based on the actual timestamp within the summary period.

State statistical summary data

The Historian stores and retrieves values, where every value gets stored with some timestamp and associated quality. This triplet of value, timestamp, and quality is called VTQ and constitutes the smallest addressable piece of data in the Historian data model.

State summary statistics summarize the states of a value. Four different state value data types are possible: analog (integer), Boolean, string, and Null.

Note: Possible Boolean state values are 0 or 1. True or False strings are not supported.

A custom property state reference can subscribe to state statistics from the Historian as static text, an expression or reference, an aggregate function name, minutes, and state value.

The Historian returns the VTQ for one cycle of a specified state. The quality returned is always OpcQuality. The time returned is always the summary period start time. Value and Time differ based on the aggregate function.

The following table lists state historical statistical data that can be specified for a custom property.

State Statistical Data	Description
Average	Average time a state occurred and completed within a summary period. A partial state within a summary period is ignored for an average calculation. (StateTimeAvgContained)
Minimum	Minimum time a state occurred and completed within a summary period. A partial state is ignored. (StateTimeMinContained)
Maximum	Maximum time a state occurred within a summary period. (StateTimeMax)
Count	Number of times a state occurred and completed within a summary period. A partial state is not counted. (StateCountContained).
Percent	Percentage of the summary period that a state occurred. (StateTimePercent)
Total	Total time a state occurred within a summary period. (StateTimeTotal)

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Historical summary period

An Industrial graphic custom property can show historical statistics from Historian data over a defined summary period. During design time, you must specify the summary period to collect Historian summary data by entering values for the **Duration** and **StartTime** options shown on the **Edit Custom Properties** dialog box.

These assigned values are passed as input parameters of the new custom property. A value in minutes must be assigned to the **Duration** option.

The **StartTime** option can be left blank. Auto refresh is applied if a **StartTime** value is not specified.

- If a start time is not specified, then the start and end times of the summary period are calculated as:

  Start Time = Current Time - Duration

  End Time = Current Time

- If a start time is specified, then the start and end times of the summary period are calculated as:

  Start Time = StartTime option value

  End Time = StartTime + Duration

The **Duration** option can accept a negative number when a start time is specified. When **Duration** is assigned a negative number, the start time input parameter value becomes the end time of the summary period. The start time is calculated using the formula shown below:

End Time = StartTime assigned option value

Start Time = End Time + Duration (in this case it is negative value)

Duration can accept values from 1 minute to 10080 minutes, which is one week. **StartTime** must be within datetime Min and Max Value. During runtime, history summary data is auto refreshed at an interval that is 25 percent of its duration length when a **StartTime** value is not specified.

Show statistical summary data

The following procedure explains how to specify custom property options to subscribe to historical statistical data. Before completing this procedure, you must have object or other data that is saved to the Historian and enabled for statistics.

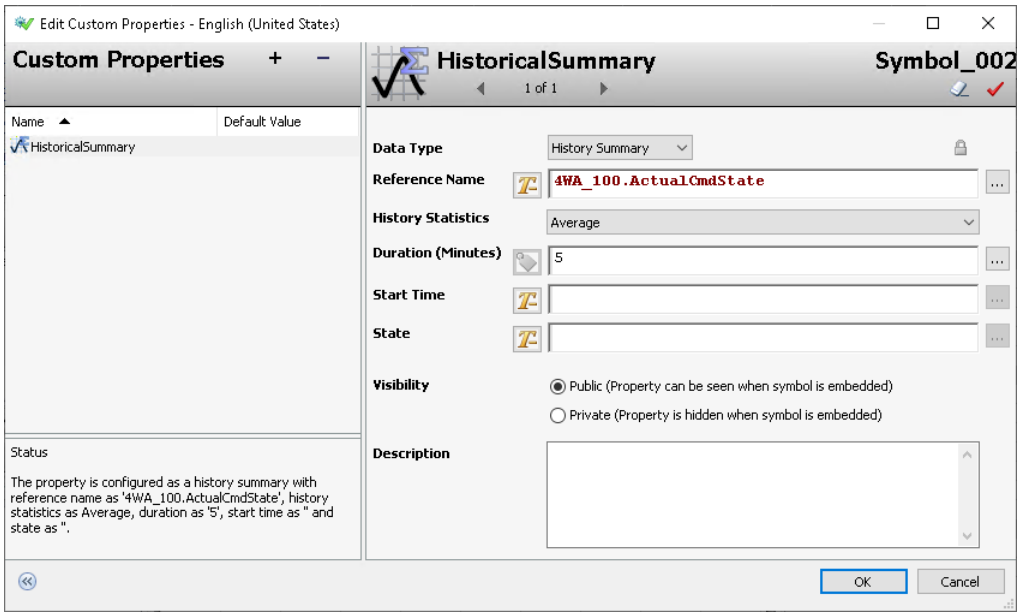
To show historical summary data using custom properties

1. Open the Industrial Graphic Editor.
2. Add a custom property to an Industrial graphic.
   a. Click the canvas to cancel any selected elements.
   b. On the **Special** menu, click **Custom Properties**. The **Edit Custom Properties** dialog box appears.
   c. Click the **Add** icon. A new line is added in the custom properties list.
   d. Type a name for the new historical summary custom property and press **Enter**.
   e. You can see the names of the graphic and custom property in the header of the dialog box.
3. Select **HistorySummary** from the **Data Type** field.

   Important: The History Summary data type only works with Application Server object attributes intended for AVEVA OMI ViewApps or InTouch HMI managed applications.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The **Edit Custom Properties** dialog box updates to show fields specific to the **HistorySummary** data type.



4. Enter a reference to data saved in the Historian in the **Reference Name** field.

   The icon to the left of the **Reference Name** field toggles input to the field as **Static Text** or **Expression or Reference** mode.

   Auto-Detect

   The Historian server is auto-detected from the AppEngine on which the reference attribute is running. For example, if the **Reference Name** field is set to UDO.UDA1, the reference is set to the Historian server name configured for the AppEngine on which UDO is running.

   Expression

   When an expression or reference is typed in the **Reference Name** field, a connection is made to the specified Historian Server. The reference can be an external reference like an object attribute or a custom property.

   Note: A reference cannot be made to historical data from an InTouch tag.

5. Select the type of historical statistics by selecting an option from the drop-down list of **History Statistics.**

   Average is the default type of historical statistic. The following table shows the historical statistics options for analog and state summary data.

Historical Statistics	Analog Historical Data	State Historical Data
Average	I	I
First	I	
Minimum	I	I
Maximum	I	I
Count	I	I

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Historical Statistics	Analog Historical Data	State Historical Data
StdDev	I	
Integral	I	
PercentGood	I	
Percent		I
Total		I
Last	I	

6. Set the length of the summary historical period in minutes by entering a value in the **Duration** field.

   Acceptable **Duration** values are from 1 to 10080 and the default is 5. Duration can be specified as an integer, an expression, or a reference. For more information about possible **Duration** values, see Historical summary period.

7. Set the start time of the of the summary period in the **StartTime** field.

   A start time can be specified as static text, an expression, or a reference. The default start time is the current time.

   A time for **StartTime** is optional and can be left blank. Auto refresh is applied if a **StartTime** value is not specified.

   For more information about setting a start time, see Historical summary period.

8. Set the type of Historian summary data in the **State** field.

   A **State** value can be expressed as an integer constant, static text, an expression, or a reference.

   If a string value is provided, then string state summary data is queried from the Historian. If an integer value is entered, the Historian query is for analog state summary data. If a Boolean state summary value is provided, the **State** value must be 0 or 1.

   State can be left empty. If empty, the default query is for analog summary data.

   To get summary historical data for a Null state, enter "NULL" in the **State** field. The query checks for OpcQuality equal to opcnull and StringValue "NULL" in the result.

Use binding in custom properties

Application Server object scripting supports a type called "Indirect". It enables you to bind a variable to a reference and read and write to it. This is done using the BindTo() method.

Note: The BindTo() method binds a variable to a reference as long as the graphic is shown.

For example, the local script variable ptr is defined and bound to the reference ud1_001.Int1.

```
dim ptr as indirect;
ptr.BindTo("ud1_001.Int1");
```

Within the same script you can use the indirect variable pointer to read from and write to the attribute ud1_001.Int1.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Industrial Graphics also use scripting in the same way as the scripting of Application Server.

However, as an Industrial graphic can be embedded into an InTouch window and run anonymously, the time it takes to connect to the reference can be longer than one scan cycle.

For that reason, you cannot use the indirect variable immediately after it is bound to a reference to read from and write to it.

```
dim ptr as indirect;
ptr.BindTo("ud1_001.Int1");
ptr = 42;
```

In the example, the value 42 cannot be written to the reference ud1_001.Int1 as the binding takes longer.

To avoid this problem, you can modify your Industrial graphic script, using the Industrial Graphic Script Editor to write the value after it is ensured that the binding is complete. The completion of the binding is indicated by the quality of the indirect variable.

- Using artificial delays in the script is not recommended as this may result in script timeouts and would in turn affect UI refresh. For example, do not use a while loop that increments a variable from 1 to <60000. This could cause the runtime application (OMI or InTouch) to go blank blank or become unresponsive while the script is executing.
- Instead of artificial delays, allow the whole script to continue execution across multiple cycles, without any artificial delays in the script, and set the script's expression to false once the quality of all the attributes is good.

Create a named script, with a trigger set to ‘While True’ and query for the quality and use the indirect variable to read from and write to the reference when its quality is good. This script will try again every second until the indirect attribute returns with good quality. If the quality is good, then the script exits from the while loop. Once the quality is confirmed, you can use the indirect variable reference in other scripts or button animations.

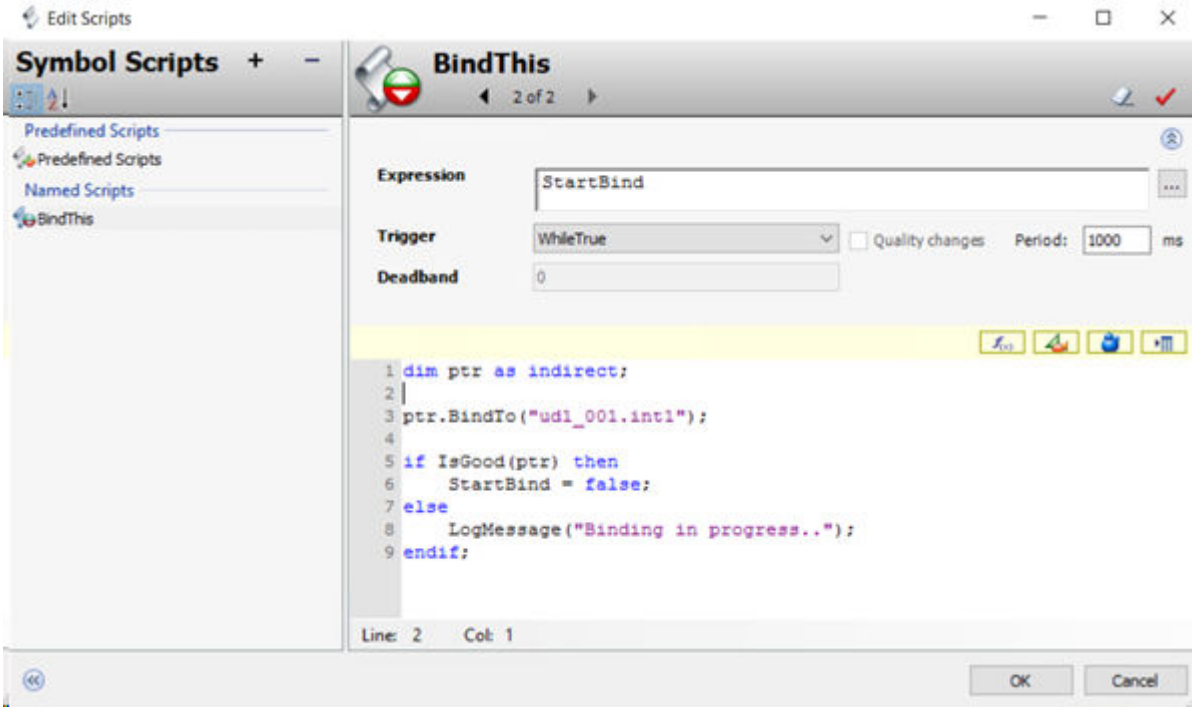
Note: Make sure to include an exit condition in your script, so that the script does not hang if the binding cannot be made.

The following example script shows you how to do this:

- Expression: Create a Boolean custom property StartBind with an initial value of True.
- Trigger: WhileTrue
- Period: 1000 ms

```
dim ptr as indirect;
ptr.BindTo("ud1_001.Int1");
if IsGood(ptr)then {if quality not good}
        StartBind = false;
else
        LogMessage("Bidnding in progress..");
endif;
```

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: Similar behavior can occur when you try to bind to a reference of an object that is hosted on a different AppEngine.

Changing the Expression or Reference of a Custom Property at Run Time

You can change the expression or reference of a custom property at run time by calling the SetCustomPropertyValue() method on the graphic using a client script:
`SetCustomPropertyValue(System.String name, System.String value, System.Boolean isConstant);`

You can select this method using the Element Browser from within the Industrial Graphic Editor.

This method has three parameters:

- *name* - Name of the custom property to be modified on the graphic. This parameter is of type string, and it can be a reference or a constant.
- *value* - The new value to be set. This parameter is of type string, and it can be an expression, reference, or constant. If the value is given in quotes ("), then the value is considered a constant. If the value is given without quotes, then the value of the expression is considered a reference.
- *isConstant* - A flag that indicates whether the new value will be evaluated as a constant or a reference. This parameter is of type Boolean. If it is set to True (1), then the new value will be treated as a constant. If it is set to False (0), then the new value will be treated as a reference. This parameter only applies when the *value* parameter is a reference or constant and the custom property specified in the *name* parameter is a string or time type. This parameter has no meaning if the custom property is an integer, float, Boolean, or double type.

Note: The *isConstant* parameter does not override the type of input for the *value* parameter. The *value* parameter itself can be either a constant or a reference depending on whether it is enclosed in quotes. The *isConstant* parameter is only determining how the actual value (coming from the *value* parameter) is evaluated.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The whole expression or reference of the custom property is replaced with the new value, regardless if it is overridden or not. No partial replacement is supported.

Only public custom properties on the graphic can be changed.

When the method executes, it overrides any modifications done by previous IOSetRemoteReference() calls from a native InTouch script.

For an example of configuring the custom property as a reference, say you have a Motor_001 object with the string attribute name "State" that stores the current state of the motor ("Running" or "Stopped"). You also have an Industrial graphic that has the string data type custom property "MotorState." The following script code will set the MotorState custom property to Motor_001.State in run-time:

```
GraphicA.SetCustomPropertyValue("MotorState","Motor_001.State",False);
```

As a result of the call, the function will set the string custom property GraphicA.MotorState to "Motor_001.State" as a reference. The string custom property GraphicA.MotorStatus will resolve that reference and update its value with the reference value ("Running" or "Stopped").

For an example of configuring the custom property as a constant, say you have a Motor_001 object with the Boolean attribute name "State" that reflects the current state of the motor (True or False). You also have an Industrial graphic that has the string data type custom property "MotorState." The following script code causes the MotorState custom property to hold the state of equipment—"Running" or "Stopped"—as text based on the value returned for Motor_001:

```
IF Motor_001.State THEN
   GraphicA.SetCustomPropertyValue("MotorState","Running",True);
ELSE
   GraphicA.SetCustomPropertyValue("MotorState","Stopped",True);
ENDIF;
```

As a result of the call, the function will set the string custom property GraphicA.MotorState to "Running" or "Stopped," depending on the value of Motor_001.State.

About using an External Content item

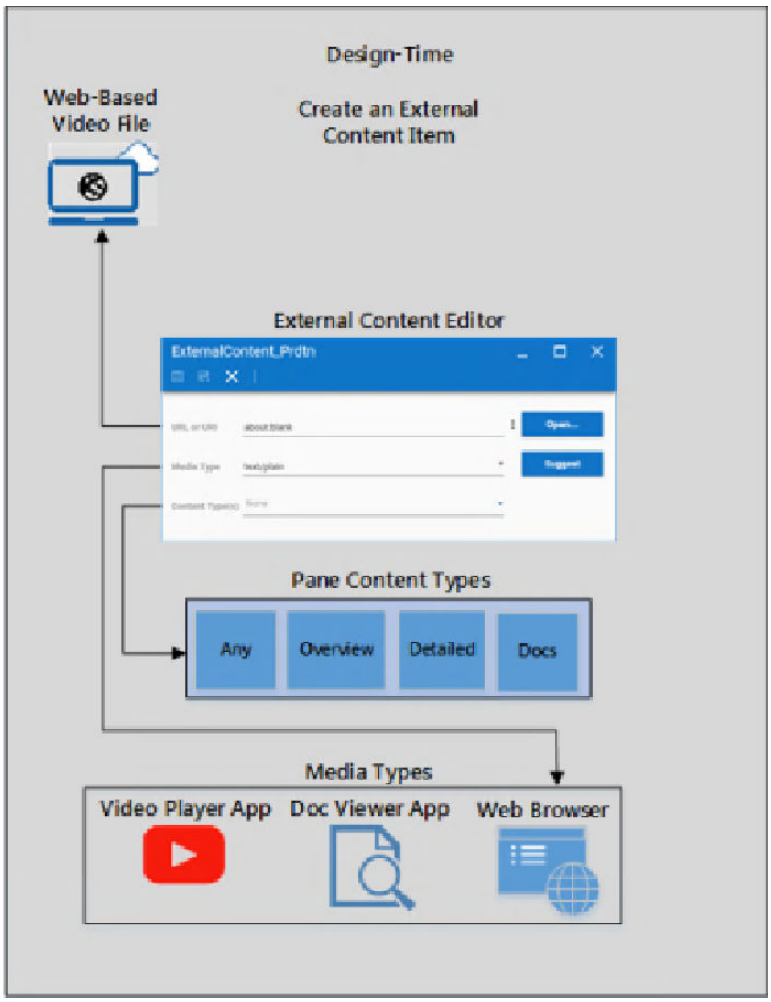
External content refers to media content outside of a Galaxy, such as an html document, that can be linked to a ViewApp and displayed during runtime.

External content is associated with a ViewApp by placing an External Content item on a layout pane during design time. An External Content item is created and then configured with the External Content editor to identify the location of documents and their content types. An External Content item is managed within the Visualization folder hierarchy of content items, which is visible in the System Platform IDE and in the **Toolbox** or **Assets** areas of the Layout and ViewApp editors.

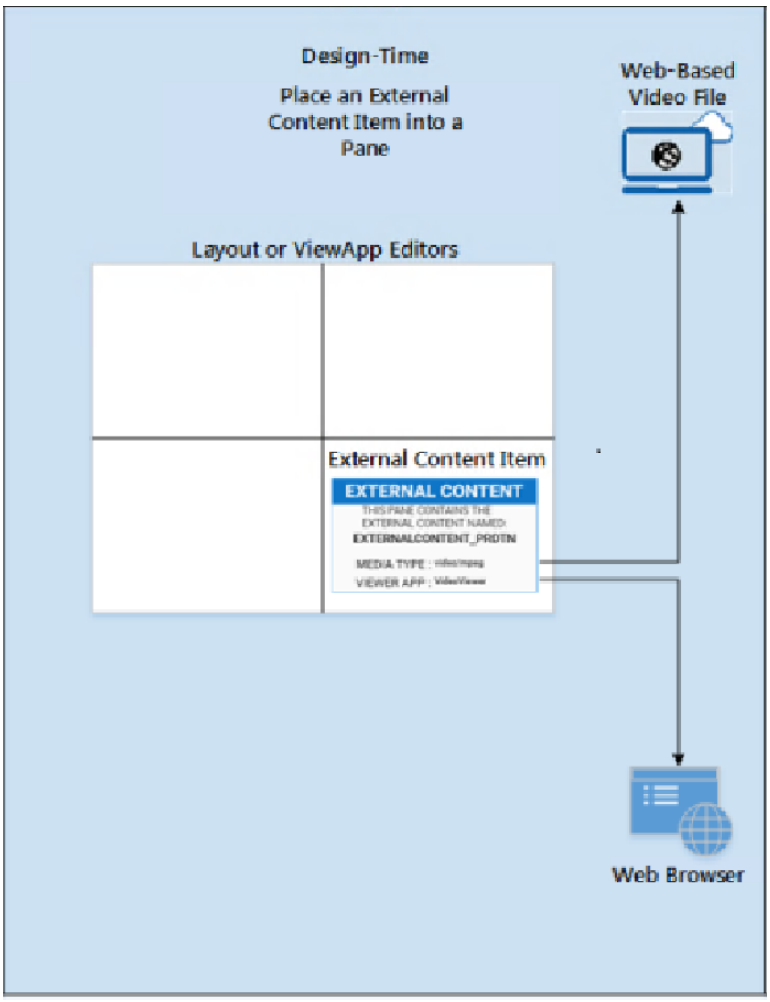
External content example

The following figures summarize using an External Content item to point to a video that operators can display while managing a running ViewApp. An External Content item is configured during design time to identify the URL of the web server and the type of media. The External Content item is selected from the **Toolbox** and placed onto a layout pane.

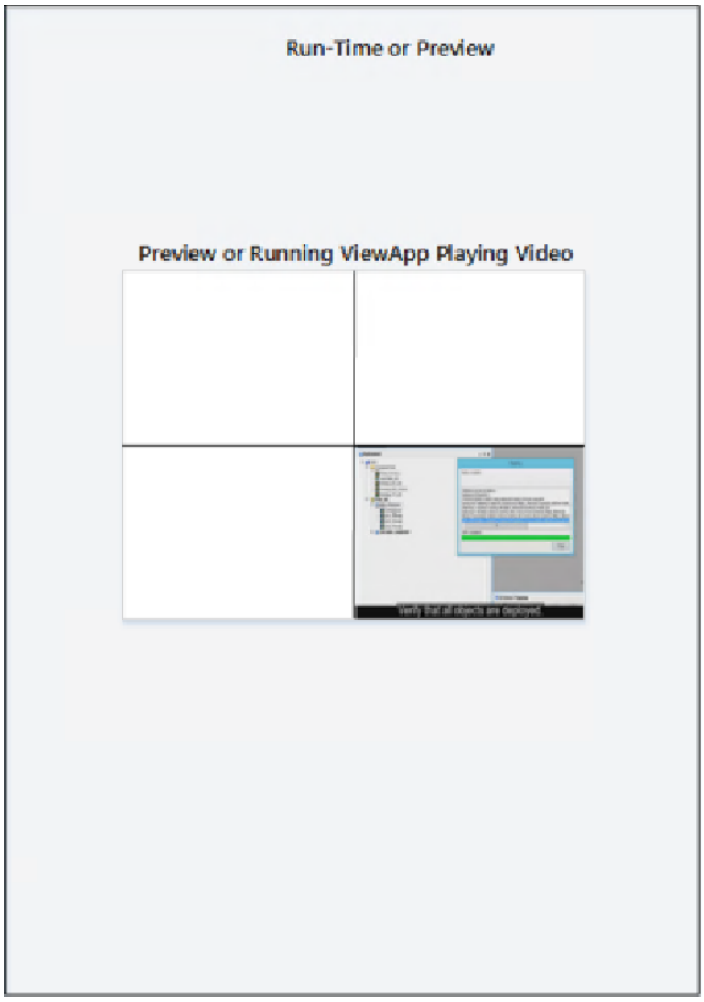
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Create an External Content item

You create an External Content item from the **Visualization folder** of the System Platform IDE. Initially, the External Content item defaults to a set of values that can be changed using the External Content editor.

To create an External Content item

1. Open the IDE and select the **Visualization folder** tab.
2. Select a folder within the **Visualization folder** if you want to create a new External Content item at a specific location.
3. On the **Home** ribbon, in the **Create** area, select **External Content**.

   You can also create an External Content item by other methods:

   - **Keyboard Shortcut**

     Press Ctrl + Shift + C

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

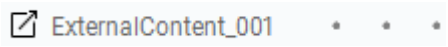
- **Accelerator Keys**

  Press Alt + G + N + C

- **Shortcut Menu**

  Right-click a folder of the **Visualization folder**. Select **New**, then **External Content**.

A new **External Content** item is created in the **Visualization folder.**



The name of the new item is ExternalContent followed by a three-digit number.

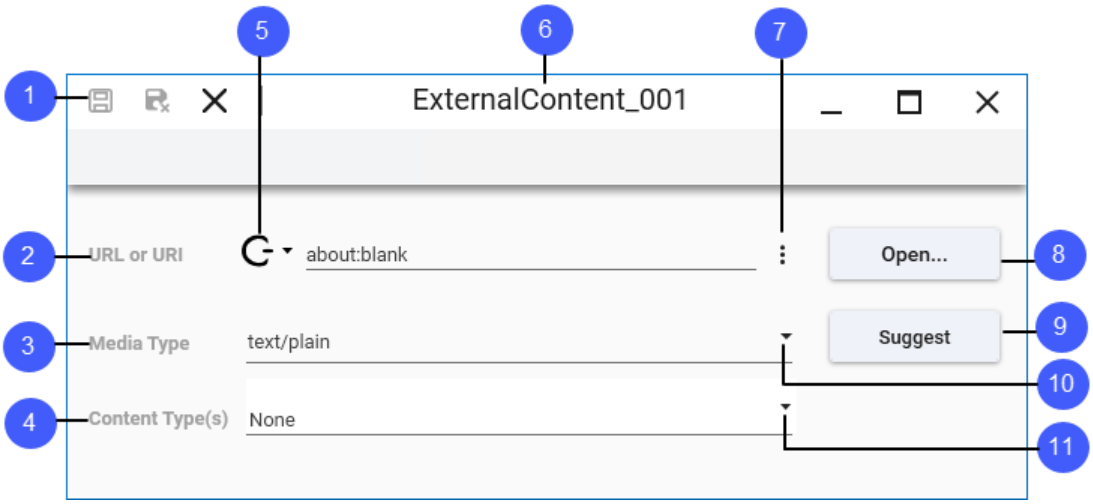
4. Rename the External Content item.

After you create an External Content item, it must be configured to associate it to specific media content. See About the External Content editor.

About the External Content editor

You use the External Content editor to configure an External Content item. Before you configure an External Content item, you must know:

- Location of the external media
- Type of media
- Content type(s) of the layout pane that can host the External Content item



1	Commands to save, save and close, and close the External Content editor.
2	URL or URI that specifies the location of external media using a standard format. A warning message appears if the external content is located on the local computer hosting System Platform.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3	A media type is a two-part identifier that specifies the type of application required to process or view remote content. A media type can be entered in the field or selected from a drop-down list. A warning message appears if a media type is specified that does not have an associated viewer application.
4	Content type assigned to the external media that enables ViewApp algorithms to place content in specific panes during runtime.
5	Icon with drop-down menu to select if the URL/URI field will be a static string or a reference/expression. The icon changes, depending on this selection. Constant Reference If you select **Reset** from the drop-down menu, any value entered in the field is discarded, and the type reverts to Constant with a default value of "about:blank."
6	Name of the External Content item.
7	Drop-down list with options to browse for an external content file or specify a URL using HTTP or HTTPs. When URI/URL is set to **Reference or Expression**, the drop-down list changes to the Galaxy Browser icon (**...**), and you can use it to select references.
8	**Open** validates the location of external media specified in the **URL or URI** field, if a **Constant** value has been selected for the field. An attempt is made to display the external media in an application assigned as the default by the operating system, not the app specified for the external content media type. The **Open** button is disabled if a **Reference or Expression** has been selected. A warning message appears if the content cannot be found at the location specified in the **URL or URI** field.
9	**Suggest** is optional. When selected, it automatically selects a media type based on the entry in the **URL or URI** field, if a **Constant** value has been selected for the field. The entered value can be changed if the suggestion does not match the expected media type. The **Suggest** button is disabled if a **Reference or Expression** has been selected. **Note**: Ensure the suggested media type is correct. The external media does not appear if the media type is incorrect.
10	Drop-down list of commonly used media types. Media types that have apps in the Galaxy capable of servicing them are shown in bold text.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

11	Drop-down list of content types that can be used at preview\runtime to assign the external content to panes of a matching content type. **Note**: None is not a content type. Instead, it simply means a content type has not been assigned.

Configuration Prerequisites

Before starting the steps to configure an External Content item, ensure the following prerequisites have been met:

- If you are placing media files on a remote computer in your network, ensure connectivity between the computer where the ViewApp will run and the computer where the remote media is stored.

  The example screenshot shown above shows a sample configuration with the URI set to a standard Windows mapped drive. For this configuration to work successfully, it is important that the computer to which the ViewApp is launched from has the same drive letter mapped to a location having the referenced content.

- If you intend to show videos within a running ViewApp on a computer running a supported version of Windows Server 2012, ensure the Desktop Experience feature is enabled. Desktop Experience includes required media features that are not enabled by default on the different versions of Windows Server 2012 supported by System Platform.

Configure an External Content item

Each External Content item has three properties that must be assigned values:

- **URL or URI**
- **Media Type**
- **Content Type(s)**

An External Content item is modified from the External Content editor. The editor can be opened by double-clicking an item from the Visualization folder. Also, an External Content item that has been placed onto a layout pane can be edited from the Layout and ViewApp editors by the selecting the item from the Actions list.

To configure an External Content Item

1. Open the IDE and select the **Visualization** tab to show the External Content items available in the Galaxy.
2. Double click an External Content item to open it in the External Content editor.
3. If you are using a static string for this **URL or URI** value, enter the value. The value you enter is the location of the external content specified by a URI-formatted string. A media location must be specified.
   - You can browse for the media file by selecting the vertical dots icon to the right of the field. A drop-down list includes a **Browse for file** option that enables you to browse your network and select a media file. The **URL or URI** field updates and shows a formatted URI path to the file.
   - You can also enter a URL to a web site by selecting the **http://** or **https://** options from the drop-down list. The **http://** and **https://** options assist the user when entering URLs by prefixing the URLs with the selection. See More Information About Specifying a URI for information about the format of a URI string.
4. Select the **Constant/Reference** icon if you will enter a reference in the URL or URI field, and select **Reference or Expression**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- The vertical dots to the right of the data entry field will change to the Galaxy Browser (**...**) icon, and you can browse for the reference you want to use.
- If you choose to use a **Reference or Expression**, the **Open** and **Suggest** buttons are disabled.

5. If you are using a static string (Constant), select the **Open** button to validate the value you entered.

   The **Open** button attempts to launch the appropriate media application and display the external content specified in the **URL or URI** field using the default application assigned from the operating system. For example, an external video file will play in Windows Media Player.

   The **Open** button is disabled if you are using a **Reference or Expression**, and the External Content object subscribes to the reference at runtime.

6. Select the type of media from the drop-down list of the **Media Type** field.

   The value you enter is the MIME type of the content. To be viewable, the media type must be supported by an app within the galaxy. A media type must be specified.

   A media type is a two-part name that identifies the type of app required to display the external media within a running ViewApp. The bold media type names listed in the **Media Type** field's drop-down list have apps that can play the external media. Users may also type in a media type that is not available from the drop-down list. However, for the external content to be displayed at runtime, an app must be imported that can service the exact media type.

   If you want, selecting the **Suggest** button updates the **Media Type** field with a suggested media type based on the file extension of the external media entered in the **URL or URI** field. Also, it is based on file extension if it's a file in files system (file:\\) or based on the header from server response if it's a http or https.

   The **Suggest** button is disabled if you are using a **Reference or Expression**.

   See Specify media types for External Content for information about supported media types.

7. Select the type of content from the drop-down list of the **Content Type(s)** field.

   The value you enter is used by content-placement algorithms within AVEVA OMI during runtime to determine the placement of content when AutoFill or ShowContent calls are invoked.

   See Set pane properties: content types for the types of content that can be assigned to layout panes.

8. Save your changes and exit from the External Content item.

Specify the URL of External Content

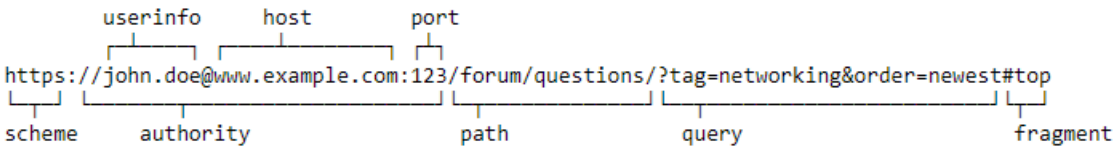
In the context of External Content, a Uniform Resource Identifier (URI) is a string that identifies the pathway to specific content that can appear in a running ViewApp.

All URIs adhere to predefined syntax rules.

- Generic URI syntax consists of a hierarchical sequence of components

```
URI = scheme:[//authority]path[?query][#fragment]
```

Example



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Each URI begins with a scheme name followed by a colon

  Examples of common scheme names include http:, https:, and file:

- An optional authority component preceded by two slashes (//)
  - Use information component consisting of a user name and optional password preceded by a colon followed by an at symbol (@)

    //username:password@

  - Host subcomponent consisting of a registered hostname or IP address
- A path component consisting of a sequence of path segments separated by a slash (/). A path is always defined for a URI, though the defined path may be empty
- An optional query component preceded by a question mark (?), containing a query string of non-hierarchical data.

For more information about the format of a URI, see https://tools.ietf.org/html/rfc3986

Example URIs

The following examples show common URIs that identify External Content within a Galaxy

- ftp://ftp.is.co.za/rfc/rfc1808.txt -- ftp scheme for File Transfer Protocol services
- file:///Z:/Media/Runtime%20Language%20Switching.mp4
- http://www.exampleserver.com/documents/index.html

Specify media types for External Content

A media type is a standard two-part string analogous to a MIME type that identifies external content file types and their format. The media type is used to identify external content and the type of app required to display media during runtime.

A media type consists of a type and a subtype, which is further structured into a tree.

type "/" [tree "."] subtype ["+" suffix] *[";" parameter]

For more information about the format of media types, see https://www.iana.org/assignments/media-types/media-types.xhtml

System Platform provides a default text/html media type that includes an associated app to display web content.

The drop-down list of the **Media Type** field shows media types in bold text to indicate a viewer app is available in the Galaxy that can service external content with these media types. Other listed media types that appear in plain text require an app to be created and imported into a Galaxy to play the specified media.

Link an External Content item to an object

After you have created and configured an External Content item, you can link it to an object template.

External Content items can be linked only to objects as opposed to graphics, which can be linked to or owned by an object. All External Content items reside in the Visualization folder. A single External Content item can be linked to multiple objects.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To link an External Content item to an object template

1. Open an object template in the **Object Editor**.
2. Select the **Attributes** tab.
3. In the **Content** pane, select the **Link Content** button.

   The Galaxy Browser opens.
4. Navigate to the folder that contains the **External Content** item to be linked.
5. Select the item, then click **OK**.

   The item is added to the object and appears in the **Content** tab.

   Note: After linking an External Content item, you can modify it by selecting it and then selecting **Edit**. However, any changes you make will apply to all objects that link to the item.
6. Save your editing changes to the object template and exit from the Object editor.
7. Create an instance of the of the linked object template.
8. Create a layout which contains pane with a Content Type that matches the content type of the External Content item.
9. Add a navigation tree to the layout
10. Create a ViewApp that incorporates the layout.
11. Deploy the ViewApp.
12. In runtime, navigate to the object or any instance created from template.

    The linked External Content item will auto-fill the matching pane to show the external content.

Associate External Content to a ViewApp

After you configure an External Content item, you must place it onto a pane of a layout incorporated into a ViewApp. You can place an External Content item into a pane from either the Layout or ViewApp editors. External Content items are listed in the **Toolbox** or **Assets** tabs of either editor.

Note: You cannot select an External Content item from the Industrial Graphic Editor.

The External Content item must have been assigned a valid location for a document and a viewer application must be associated with the assigned document media type.

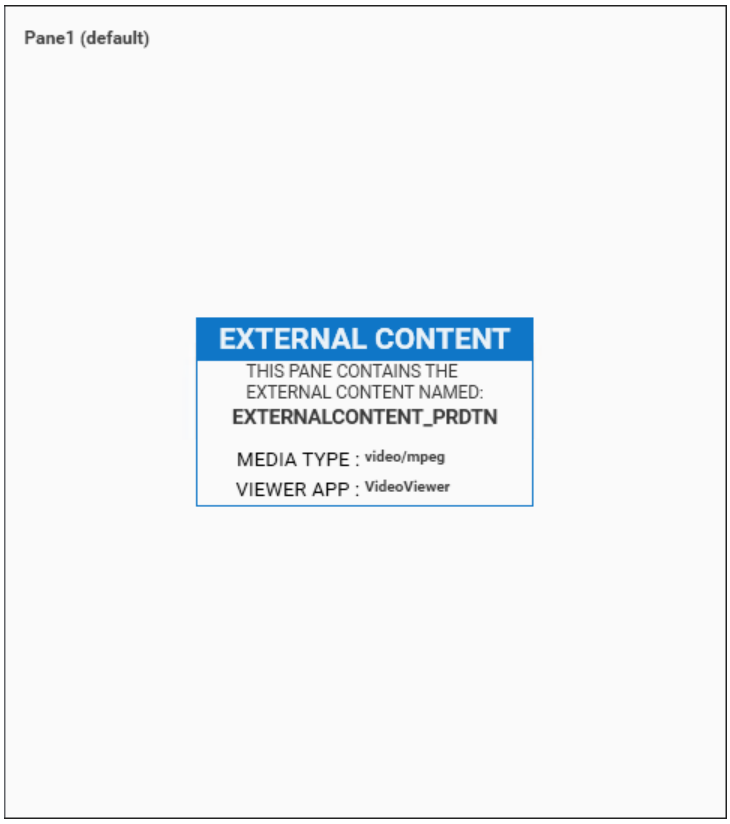
If you attempt to place an External Content item onto a pane that that does not have an assigned viewer application to display the media type, a warning message appears after placing the item onto a pane. The ViewApp will not show the document, although the ViewApp still continues to run during runtime.

To associate external content to a ViewApp

1. Open the System Platform IDE and select the **Toolbox** tab to see the list of layouts.
2. Select a layout that you want to insert external content in and open it in the Layout editor.
3. Select the **Toolbox** tab of the Layout editor to show the list of content accessible from the Galaxy.
4. Locate the External Content item to place in the layout.
5. External Content thumbnails appear beneath of the list of content shown in the **Toolbox** list.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

6. Drag an External Content thumbnail from the **Toolbox** area onto the desired layout pane.



   The External Content thumbnail appears in the pane containing the name of the External Content item, the media type of the external content, and the viewer app that will display the media during runtime. Also, the layout **Actions** area shows the name of the External Content item and the pane that it was placed in.

Editing an External Content Item After Being Placed onto a Pane

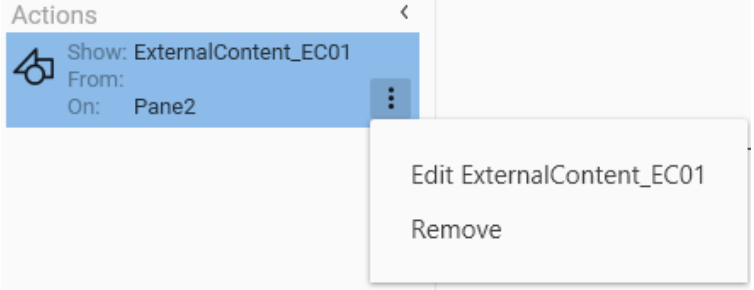
An External Content item can be modified after being placed in a layout or a ViewApp, which automatically propagates the changes

To edit an External Content item in a layout

1. If necessary, open the layout containing an External Content item.
2. Locate the item in the layout **Actions** list.
3. Select the **Options** icon shown at the right of the item in the **Actions** list.

   A drop-down list shows an Edit External Content command.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



4. Select the edit command.

   The External Content editor opens the item for editing.

5. Update the values assigned to the item and save your changes.

   The External Content thumbnail in the pane updates to reflect the changes made to the item.

Manage external content

An External Content item can be managed from the IDE using standard options from the right-click shortcut menu or from the **Edit** options of the IDE menu bar.

A ViewApp containing apps designed to display external content can be exported from one Galaxy and imported to a ViewApp in another Galaxy. The External Content editor is capable of recognizing a new media type from the imported app.

An External Content item can be:

- Exported to an aaPKG file
- Imported from an aaPKG file
- Included in a Galaxy backup
- Restored from a Galaxy backup
- Included in an export of layouts or ViewApps that have External Content items associated with them

Create an app for external content

Apps that are designed to load external content when a ViewApp starts must be able to recognize a URI. App developers also need to include a list of the Media Types that can be serviced by the app. An app's supported media types must be specified in an AppManifest.xml file. For more information about specifying media types in the AppManifest.xml file, see the *AVEVA OMI Developer Guide*.

Note: When there is more than one application in a Galaxy capable of servicing a specific media type, the more recent imported application will be used by the ViewApp.Users cannot select an application for a media type.

After importing the app, these Media Types will be added to an internally maintained Media Types registry. Panes will refer to this registry when trying to open External Content. The app that services the Media Type associated to the External Content will first be launched and then the external content loaded.

The overall steps in the workflow to import an app are described in the following list.

1. Create an AVEVA OMI app that recognizes a URI and supports one or more defined media types.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. Enclose the app in an ArchestrA aaPKG file and export it to a location accessible to your development computer.
3. Open the System Platform IDE.
4. Import an app in an aaPKG file that is designed to load and display a specific type of external media.
5. Create an External Content item.
6. Edit the External Content item.

   The **Media Type** field of the External Content item lists the media types supported by the imported app. The External Content editor must be closed and reopened if an app containing an unrecognized media type is imported while the External Content editor is open. Reopening the External Content editor lists the new media type.

Script editor

AVEVA OMI supports two types of scripts:

- Simple scripts can perform assignments, comparisons, simple math functions, and similar actions.
- Complex scripts can perform logical operations using conditional branching with IF-THEN-ELSE type control structures.

For detailed information about scripting and available script functions, see located in the *Application Server Help*.

Add a script to a graphic

You can associate scripts with your graphics. Scripts can add animation to a graphic or its elements that can be executed in runtime.

Caution: If you configure scripts that affect more than element and graphic animation, the script processing may affect performance.

You can:

- Configure the predefined scripts of a graphic.
- Add named scripts to a graphic.
- Edit existing named or predefined scripts in a graphic.
- Rename named scripts in a graphic.
- Remove named scripts from a graphic.
- Substitute references in named or predefined scripts.
- Use element methods in named or predefined scripts.

The autocomplete features available in the Application Server QuickScript editor are also available in the Industrial Graphic Editor script editor. For more information about Application Server scripting, see *the Application Server Scripting Guide*.

The topics in this section describe using the SignedWrite() function for secured and verified writes. For information about all other aspects of associating scripts with graphics, see .

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Working with the SignedWrite() function for secured and verified writes

This section provides information about the SignedWrite() function, SignedWrite() runtime behavior, scripting tips, and in-depth script examples.

For SignedWrite() scripting information including script syntax, parameters, and basic script examples, see the *Application Server Scripting Guide*.

You can write to an Automation Object attribute that is configured for Secured Write or Verified Write security classification by means of the Industrial Graphics SignedWrite() script function.

The SignedWrite() function can be used only in client scripts, not in Application Object scripts, and only on Attributes that have been configured for Secured Write or Verified Write. Attempting to use the function on an Attribute not so configured will result in an error message at runtime.

SignedWrite() runtime behavior

At runtime, the SignedWrite() function does the following:

1. Checks the target Attribute for Signed Write or Verified Write configuration.

   If not so configured, the following error message appears: **Operation Failed. Attribute does not have the correct security classification.**

2. Checks Galaxy security.

   If the Galaxy is not secured, the following error message appears: **Operation Failed. Galaxy is not secured.**

3. There are several ways to write to an Attribute configured with Secured Write or Verified Write security classification, it is possible to have multiple SignedWrite() and other Secured/Verified writes pending from the same script, or even from multiple scripts running side-by-side.

4. Determines which dialog is required—Secured Write or Verified Write—and pops up the appropriate dialog box.

   If Smart Cards are enabled, the function displays different versions of the Secured Write and Verified write dialog boxes.

5. Lists the predefined comments, if any, from the configured Predefined Comments list. Up to 20 comments are supported.

6. Enables comment editing if the Comment_Is_Editable parameter is configured and comment enforcement is other than PredefinedOnly.

7. Acquires the user credentials and authenticates them.

   If the user credentials are invalid, an error message appears. The function will attempt the write only if the credentials are valid.

8. Checks if comment enforcement is mandatory, and displays an error message if the comment is empty.

9. Performs the write if user credentials are valid and the comment entry satisfies the comment enforcement parameter.

10. Provides a return status.

11. Following a Secured or Verified Write a security Event is written to the event log, including the signee name, verifier name, if any, Type of write: "Secured Write" or "Verified Write", Comment, if any entered by user, Reason Description, if any provided, Field Attribute description, if any, or the Short Description of the Application Object, if no Field Attribute description exists.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Each call to SignedWrite() is distinct from any other. The success or failure of any individual write does not affect other attempted writes.

Entering user credentials for SignedWrite() is distinct from logging on to the client application. The user can modify attributes configured with Secured or Verified Write even if another user is logged on, without affecting the session of the logged-on user.

SignedWrite() script execution sequence at runtime

The SignedWrite() function goes into a queue and the script continues executing. The function is queued for operator entry. The script may complete prior to the operator completing the Secured or Verified Write operation.

By contrast, the SignedAlarmAck() script function executes completely synchronously, and waits for user input before proceeding to the next line in the script.

SignedWrite() scripting tips

Using Bound References in SignedWrite()

If the Attribute parameter string evaluates to the name of a Custom Property and that Custom Property is a bound reference to an Attribute, the SignedWrite() function will write to that indicated Attribute. The Attribute must have the security classification of Secured Write or Verified Write.

The SignedWrite() function supports Custom Properties that are nested bound references. That is, if the string evaluates to the name of a Custom Property and that Custom Property is a bound reference to another Custom Property which itself is a bound reference, the SignedWrite() function will follow through the chain of bound references until it finds an item that is a value. If that item is an Attribute that has the security classification of Secured Write or Verified Write, the SignedWrite() function will write to that item.

Using SignedWrite() in WhileTrue, WhileFalse, or Periodic Type Scripts

Using the SignedWrite() function with WhileTrue, WhileFalse, or Periodic type scripts can repeatedly execute the script, causing another secured write dialog box to pop up with each trigger. We do not recommend using the SignedWrite() function with WhileTrue, WhileFalse, or Periodic types.

Using SignedWrite() with OnShow and OnHide Scripts

We do not recommend using the SignedWrite() function with OnShow and OnHide scripts. This can cause issues with window functionality, including the window title bar, windows losing correct focus, and windows opening on top of one another.

Examples of using the attribute parameter in the SignedWrite() function

Working from the overall syntax of the SignedWrite() function, the script examples in the following table illustrate a number of approaches to using the Attribute parameter in the SignedWrite() function.

The following String, Boolean, and Integer user defined attribute conditions apply to the script examples:

- User Defined object UD1_001
- String attribute UD1_001.udString1 with the value "WW". Security classification is set to Secured Write.
- Boolean attribute UD1_001.secBool1 with the value false. Security classification is set to Secured Write.
- Integer attribute UD1_001.secInt1 with the value 24. Security classification is set to Secured Write.
- User Defined object UD2_002

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

String attribute UD2_002.udString2 with a value "UD1_001.udString1" The following custom property conditions apply to the script examples:

- String custom property CP1 with a reference to UD1_001.udString1
- String custom property CP2 with a value "UD1_001.udString1"
- String custom property CP3 with a value "UD1_001"
- Boolean custom property CP4 with a reference to UD1_001.secBool1
- Integer custom property CP5 with a reference to UD1_001.secInt1
- String custom property CP6 with a reference to an attribute on Owning Object UD1_001 as me.udString1

Script Example	Function and Result
`SignedWrite("CP1", "AVEVA", "using redirect", true, 0, null);`	Uses the CP1 reference UD1_001.udString1 and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
`SignedWrite(CP2, "AVEVA", "using string value", true, 0, null);`	Resolves CP2 string value "UD1_001.udString1" to a reference and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
`SignedWrite(CP3+".udString1", "AVEVA", "using string expression", true, 0, null);`	Resolves the string "UD1_001.udString1" to a reference and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
`SignedWrite("UD1_001.udString1", "AVEVA", "using constant string", true, 0, null);`	Resolves the string "UD1_001.udString1" to a reference and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
`SignedWrite(UD2_002.udString2, "AVEVA", "using attribute containing string", true, 0, null);`	Resolves the UD2_002.udString2 string value "UD1_001.udString1" to a reference and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
`SignedWrite("CP" + "1", "AVEVA", "using redirect from string expression", true, 0, null);`	Resolves the expression to "CP1" to use the CP1 reference UD1_001.udString1 and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Script Example	Function and Result
`SignedWrite("CP4", true, "using redirect", true, 0, null);`	Uses the CP4 reference UD1_001.secBool1 and pokes to it the value true. Result: The value in UD1_001.secBool1 will change from false to true
`SignedWrite("CP5", 37, "using redirect", true, 0, null);`	Uses the CP5 reference UD1_001.secInt1 and pokes to it the value 37. Result: The value in UD1_001.secInt1 will change from 24 to 37
`SignedWrite("CP6", "AVEVA", "using redirect using relative reference", true, 0, null);`	Uses the CP6 reference me.udString1 and resolves it to UD1_001.udString1 and pokes to it the value "AVEVA". Result: That the value in UD1_001.udString1 will change from "WW" to "AVEVA"

Configure the SignedWrite() script function

You can create a dashboard application to automate routine use of Secured and Verified Write by means of the SignedWrite() function.

To configure the SignedWrite() script function

1. Open the System Platform IDE.
2. Create a graphic and associate it with an attribute configured with Secured Write or Verified Write. For more information on associating attributes with graphic, see the Application Server User Guide, "Creating and Working with UDAs" topic.
3. Add the SignedWrite script function to the graphic. The following editor detail shows the buttons configured with scripts in the applied example:
4. Configure the scripted functionality you require. Scripts for the buttons shown in the example are as follows:
   a. Hard-coded DataUDO.SecUDA: The following example sets the value of 23 to DataUDO.SecUDA. The user optionally can enter a comment, but no pre-defined comment list is available.

      ```
      DataUDO.RetStatus=SignedWrite("DataUDO.SecUDA", 23, "Set the Value", True, 0, null);
      ```

   b. Attribute Pointer has DataUDO.SecUDA: The source to be written to is passed as a parameter to the function. Attribute_Pointer is a custom property whose value is set to DataUDO.SecUDA.

      The following example sets the value of 23 to DataUDO.SecUDA. The user optionally can enter a comment, but no pre-defined comment list is available.

      ```
      DataUDO.RetStatus=SignedWrite(Attribute_Pointer, 23, "Set the Value", True, 0, null);
      ```

   c. Attribute Pointer and Pre-Defined List: The pre-defined comment list is an array. This example extends the functionality of example b to force the user to enter a comment (Comment_Enforcement parameter set to 1) and also presents a pre-defined set of comments linked to the DataUDO.PreDefComments[ ] array.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The following example will set the value of 23 to DataUDO.SecUDA. The user must enter a comment and may use one from the pre-defined comment list.

```
DataUDO.RetStatus=SignedWrite(Attribute_Pointer, 23, "Set the Value", True, 1,
DataUDO.PreDefComments[]);
```

d. Variable Array: The predefined list is a pointer to an array. This example extends the functionality of example c to force the user to enter a comment (Comment_Enforcement parameter set to 1) and also presents a predefined set of comments linked to DataUDO.PreDefComments[ ] array.

The value of custom property CP1 is "DataUDO.PreDefComments[ ]".

The following example will set the value of 23 to DataUDO.SecUDA. The user must enter a comment and may use one from the pre-defined comment list.

```
dim xInd as Indirect;
xInd.BindTo(CP1);
DataUDO.RetStatus=SignedWrite(Attribute_Pointer, 23, "Set the Value", True,
1, xInd);
```

e. All Parameters Variable: The predefined list array is built into the script. All parameters are passed as variables.

The following example will set the value of 23 to DataUDO.SecUDA . The user must enter a comment and may use one from the pre-defined comments list.

```
dim MyList[5] as string;
MyList[1] = "Batch Accepted";
MyList[2] = "Batch Rejected";
MyList[3] = "Batch on Hold";
MyList[4] = "Batch Resumed";
MyList[5] = DataUDO.PreDefComments[4];
DataUDO.RetStatus=SignedWrite(Attribute_Pointer, SignedWrite_Value_Ptr,
SignedWrite_Reason, Enable_Edit_Comment, Comment_Options, MyList[]);
```

How to write to an attribute configured for Secured Write or Verified Write security classification

There are several ways to write to an Attribute configured for Secured Write or Verified Write security classification.

- Any assignment in a script that sets the value of the Attribute, such as

  ```
  A=B;
  ```

  where A references an Attribute that is configured for Secured Write or Verified write security classification.
- Any action on an animation graphic that alters the value of an Attribute that has Signed Write or Verified Write security configuration, such as a user input, a slider, an up/down button on a counter, and other such actions.
- A script that uses the SignedWrite() function.

For information specific to the SignedWrite() function, see Working with the SignedWrite() function for secured and verified writes.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Example of Changing Element Properties using Scripts

You can change some properties of elements using scripting. This lets you configure additional run-time behavior to your elements in addition to design-time animation of those elements.

When you write scripts for the graphic or for one of its elements, you can use your HMI's attribute/tag browser to show and select a:

- Property of an element
- Custom property of the graphic

If a reference is not unique, the following order applies:

1. Dimensioned variable references
2. Graphic properties references
3. Custom property references
4. Object attribute references

To select an element property or a graphic custom property

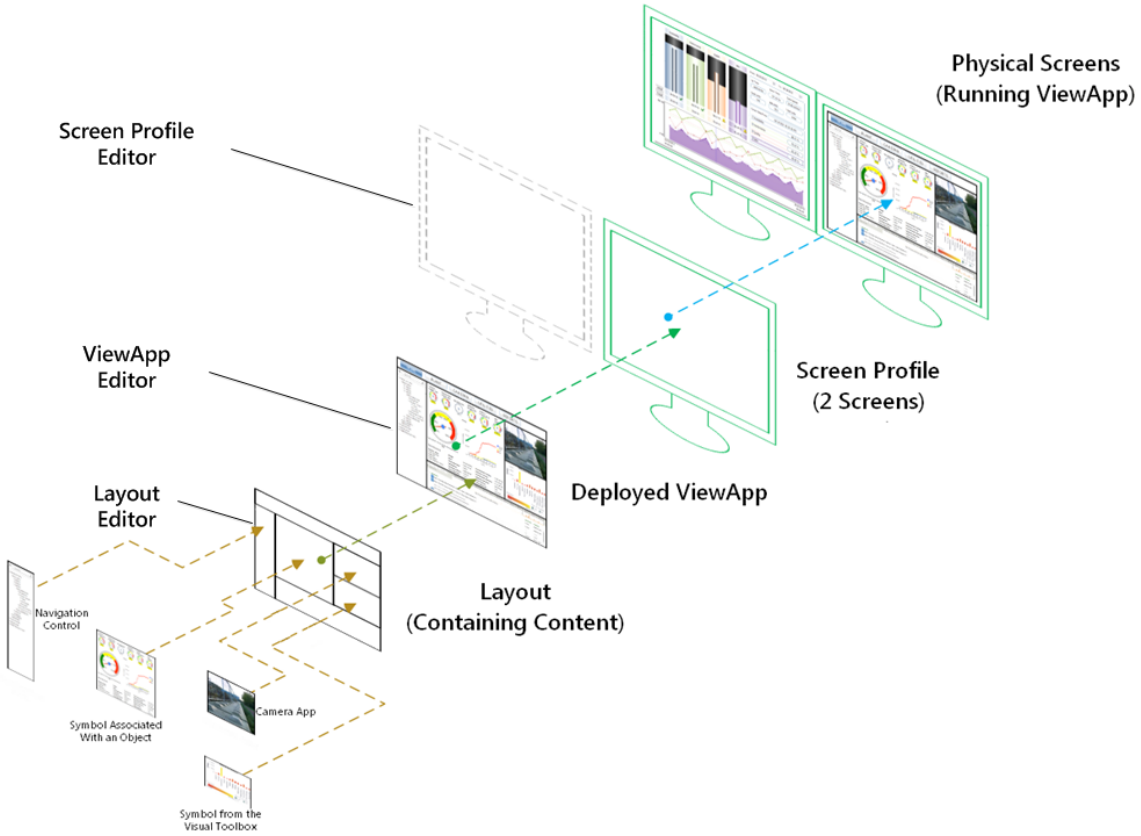
1. Open your HMI's attribute/tag browser. It shows the element names and the properties of the selected element.
2. Select an element or graphic from the list. The right pane shows the accessible properties of the selected element or graphic.
3. Select a property from the right pane and click **OK**. The reference appears in the script window.

ViewApp editor

Typically, you build standard objects, graphics, and controls during the design phase of your project. These standards are reusable components that can be inserted into a ViewApp in an almost assembly line process. The following figure shows the major steps of assembling a ViewApp with objects, screen profiles, layouts, and content with the ViewApp editor.

See ViewApp for information about using the ViewApp object.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



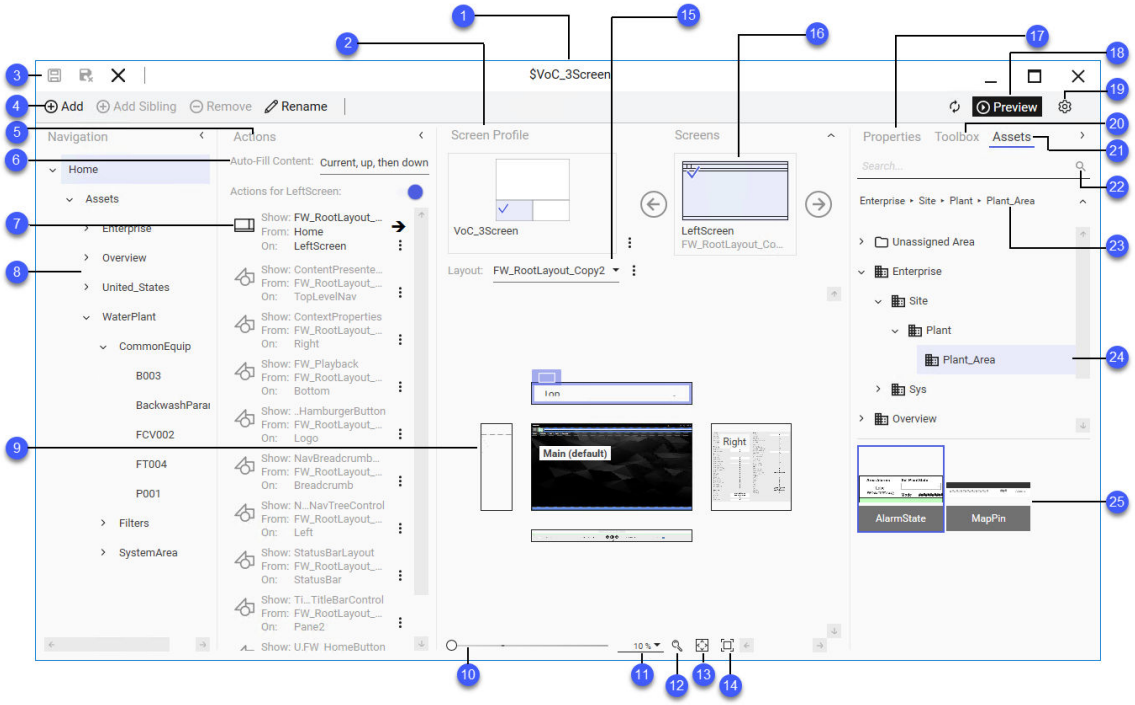
Note: Navigation workflows determine how content appears in ViewApp panes at different nodes within a navigation model. How to configure navigation is discussed separately in About ViewApp navigation.

About the ViewApp Editor

You use the ViewApp Editor to integrate all of the component parts to create a ViewApp. The ViewApp Editor incorporates the functionality to add, replace, and edit screen profiles, layouts, and content.

The figure below shows the various components of the ViewApp Editor

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



1	Selected Screen Profile	Name and thumbnail representation of the current screen profile associated with the ViewApp being edited. The screen thumbnail with a green background is the active screen in the profile. When a screen profile contains multiple screens, individual screens can be selected. When an individual screen is selected, the screen name appears in the **Actions** list and the layout assigned to the screen is shown in the editing area of the ViewApp Editor.
2	Custom Navigation Commands	Commands to manage custom navigation items placed in the navigation tree. These commands enable you to add, remove, and rename navigation items included in the ViewApp's navigation model.
3	Title Bar	The Title Bar shows the name of the ViewApp being edited. An asterisk (*) immediately after the ViewApp name indicates the edited ViewApp contains changes that have not been saved. The phrase Read Only appears when the ViewApp is open in the ViewApp Editor in read only mode.
4	Command Bar	Commands to save, save and close, and close the ViewApp currently edited with the ViewApp Editor.
5	Actions List	List of navigation actions associated with navigation items and the panes of a layout.
6	Auto-Fill Content	Drop-down list of Auto-Fill options that determine the different types of navigation modes to search for valid content that can be shown in a pane.
7	Action Items	Each action item lists the name of content or container placed in a pane. The action item also shows the source and the pane within the layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

8	Navigation Tree	Hierarchical tree view of navigation items associated with objects listed in the Model view of the Template folder. Custom navigation can be used to add or remove items from the navigation tree.
9	Pane Content	Thumbnail representation of the content shown in a pane based on the navigation item selected in the navigation tree.
10	Zoom Percent Slider	Moving the Zoom Percent Slider left or right adjusts the zoom percentage of the layout shown in the ViewApp Editor..
11	Zoom Percent Picker	Drop-down list of zoom percentages to set the size of the layout shown in the ViewApp Editor..
12	Zoom Box	The Zoom Box enables a selected area to be zoomed to the full width or height of the current editing area of the ViewApp Editor. The cursor shows a magnifying glass when the Zoom Box is selected. Drawing a rectangle by mouse or by touch within the editing area shows the content within the rectangle at the full width or height of the current editing area.
13	Zoom to Fit	Adjusts the size of the layout to fit entirely within the current area of the ViewApp Editor
14	Zoom to 100%	Sets the size of content shown in a ViewApp Editor to its actual size
15	Layout Selector	Displays the name of the layout assigned to the selected screen. Clicking the arrow at the right of the Layout name lists all available Layouts in the Galaxy. Selecting another layout within the list assigns it to the selected screen.
16	Screens List	Thumbnail list of all screens in the screen profile with an overlay of the layout assigned to the screen.
17	Properties grid	Shows content properties of the graphics or controls currently showing on the selected pane. If a pane is empty, no properties are listed. If the selected pane contains a graphic, the name of the graphic and its pane location appear as content properties. If the selected pane contains a control, the control's properties are listed.
18	Preview	Opens a preview window so you can see what the ViewApp will look like at runtime without having to deploy it. For more information , see About previewing a ViewApp.
19	Additional settings	Opens the Settings dialog box, where you can set additional values that affect the behavior of this ViewApp. For more information, see Set additional properties for a ViewApp.
20	Toolbox Tab	Shows the hierarchical folder structure of the Visualization folder. Selecting a folder shows thumbnails of the content within the folder.
21	Assets Tab	Shows a hierarchical list of the objects shown in the Model view of the IDE. Selecting an object from the list shows the thumbnail of any associated content.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

22	Search Field	Field to enter a search string to search for properties, graphics, or objects by name.
23	Breadcrumb	When the **Toolbox** tab is selected, a breadcrumb shows the list of graphic folder names to the content selected from the list. If the **Assets** tab is selected, a breadcrumb shows the serial list of Areas or other objects to the selected item from the list.
24	Selected Object	Object selected from the list of assets.
25	Associated Content	Content associated with the selected object from the assets list.

Configure the ViewApp Editor

You can use a set of common controls to adjust the editing areas of the ViewApp Editor while you are working on a ViewApp.

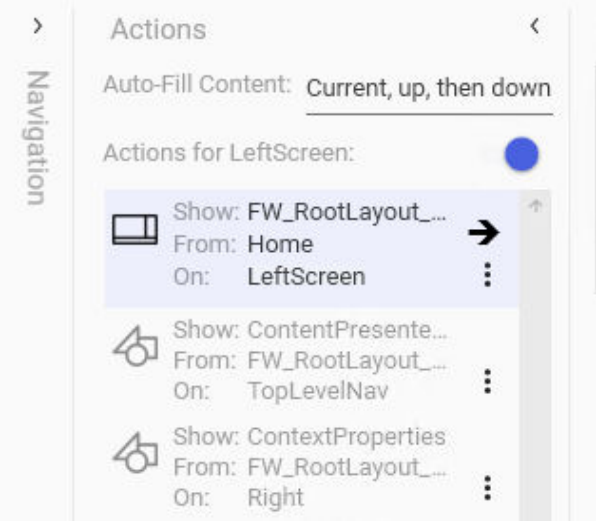
Adjust the Size of ViewApp Editor Areas

The ViewApp Editor consists of a set of columns that show a navigation hierarchy, an Action List, a graphic of the layout being edited, and a tabbed area at the right to show properties, assets, and content within the Visualization and Template folders.

Dividers are vertical or horizontal lines that separate the different areas of the ViewApp Editor. You can resize a ViewApp Editor area by moving its divider with a drag and drop mouse action or by a touch gesture.

Show or Hide ViewApp Editor Fields

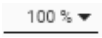
A chevron icon placed near the name of a ViewApp Editor field indicates the field can be expanded or hidden by selecting the chevron.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Change the Viewing Size of a ViewApp in the ViewApp Editor

The middle area of the ViewApp Editor show a graphic representation of the ViewApp currently being edited. A set of pan and zoom controls appear at the bottom of the field to change the size of the Layout graphic. Panning or zooming a layout is useful when the layout is designed for a much larger or smaller screen than the editing area of the ViewApp Editor.

	Moving the slider left or right changes the zoom level of the layout graphic shown in the ViewApp Editor. You can also move the slider by rotating the wheel of your mouse.
	The Zoom Percentage picker shows a drop-down list of zoom percentages. You select an option from the list to resize the layout graphic to the selected percentage.
	The Zoom Box enables a selected area to be zoomed to the full width or height of the current editing area of the ViewApp Editor. The cursor shows a magnifying glass when the Zoom Box is selected. Drawing a rectangle by mouse or by touch within the editing area shows the content within the rectangle at the full width or height of the current editing area.
	The Zoom to Fit control changes the size of the layout graphic to fit entirely within the current editing area of the ViewApp Editor.
	The Zoom to 100% control changes the size of the layout graphic to the actual size of the layout.
	You can pan to different parts of the editing area of the ViewApp Editor with the Pan tool. The cursor changes to a hand when pan mode is selected.

About ViewApp and derived templates

You begin working with a ViewApp by creating a derived template from the $ViewApp base template in the Template folder of the System Platform IDE. After creating a derived template from the $ViewApp template, you can begin editing your ViewApp with the ViewApp Editor.

Unlike other objects that are created from templates, you cannot create and edit an instance of a ViewApp. All of your editing is done in the derived template. An instance is created automatically when you assign the derived ViewApp template to a ViewEngine or WebViewEngine.

Before creating a ViewApp, you should have created the screen profiles and layouts that will be integrated into your ViewApp.

- Screen Profile Editor, see About screen profiles
- Layout Editor, see AppFramework layouts

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Create a ViewApp

You create a ViewApp from a derived template of the $ViewApp base template. When you start the ViewApp Editor for the first time after creating a Viewapp, the **ViewApp Initialization** wizard appears to select a screen profile and layouts that will be associated with the ViewApp.

Note: Before you create a ViewApp you should have created the screen profiles and layouts that will contain the content of your ViewApp.

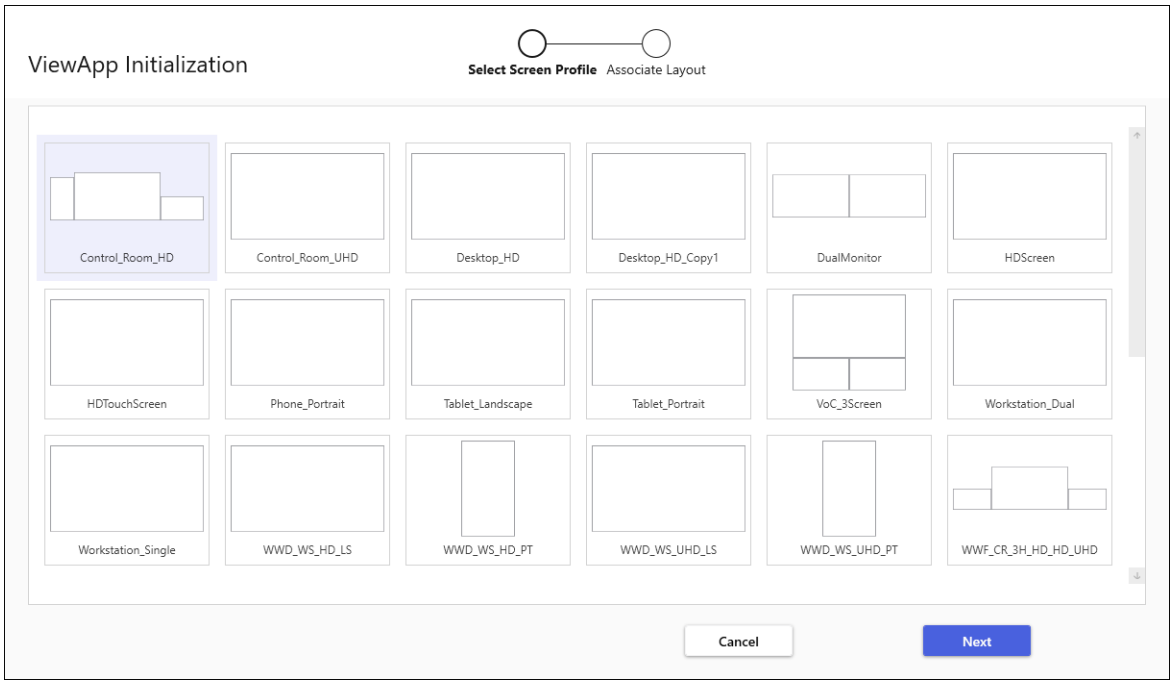
To create a ViewApp

1. Open the IDE.
2. Select the **Template folder** tab and expand the **System** folder to show the $ViewApp object.
3. Create a derived template of the $ViewApp object.
4. Assign a ViewApp name to the derived template you created.

   Note: Immediately after creating a ViewApp derived template, a red icon appears indicating no screen profiles and layouts have been assigned to the ViewApp.

5. Double-click on the ViewApp icon to start the **ViewApp Initialization** wizard.

   The wizard opens and shows the thumbnails of the screen profiles present in the Galaxy. Each thumbnail shows the configuration of screens that belong to the screen profile.



6. Select a screen profile to associate with your ViewApp and select **Next**.

   The wizard shows a thumbnail of the screen profile you selected and a list of layouts present in the Galaxy. Each layout appears as a thumbnail graphic showing its pane organization .

7. Associate a layout to one or more screens in the Screen Profile.
   a. Select a screen from the screen profile.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The screen border turns green and the name of the screen appears beneath the icon when it is selected.

b. Select a layout to associate with the selected screen.

Note: At least one screen in the screen profile must be associated with a layout before continuing to the ViewApp Editor.

8. Select **Finish** to close the Wizard and open the ViewApp editor.

For more information about the different components of the ViewApp editor, see About the ViewApp Editor.

Manage a ViewApp

You can perform routine tasks to manage a ViewApp from the Template folder. Management tasks include:

- Delete a ViewApp
- Rename a ViewApp
- Export a ViewApp
- Import a ViewApp

Delete a ViewApp

You can delete a ViewApp from the Templates pane of the IDE.

To delete a ViewApp

1. Open the IDE and display the **Templates** pane.
2. Navigate to the folder containing the ViewApp template to delete.
3. If the ViewApp is open for editing, close it.
4. Select the ViewApp you want to delete.
5. Press the **Delete** key or right-click the ViewApp and select **Delete** from the shortcut menu.

   The **Delete** dialog box appears to confirm your delete request.

   Click **Yes** to delete the ViewApp.
6. Verify the ViewApp no longer appears in the folder list.

Rename a ViewApp

You can rename a derived ViewApp by selecting it in the Templates pane.

After you enter a new ViewApp name, the name is validated to ensure it meets naming requirements. If it is a duplicate of an existing name or breaks any name rules, a message appears indicating there is a problem with the name.

Pressing the **Esc** key cancels a rename request and the Viewapp reverts to its original name. Committing an empty name cancels the rename request.

To rename a ViewApp

1. Open the IDE and display the **Templates** pane.
2. Navigate to the folder containing the ViewApp template to rename.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

3. If the ViewApp is open for editing, close it.
4. Select the ViewApp to rename.
5. Click the ViewApp name or press **F2** to enter edit mode. A blue background appears around the name.
6. Type the new name.
7. Commit the new name by clicking outside of the name.

   If the new name is valid, it appears in the Template folder. If the name is invalid, you see a dialog box that describes the problem.
8. If you typed an invalid name, click **OK** and enter another name.

Export a ViewApp

You can export one or more ViewApps from the Templates folder to an aaPKG file. You can then import the ViewApps from the export file to another Galaxy.

To export a ViewApp

1. Open the IDE and display the **Templates** pane.
2. Select one or more ViewApps to export.
3. On the **Home** ribbon, in the **Export** area, select **Selected**, then **As package.**

   You can also export a ViewApp from the shortcut menu: right-click the ViewApp name, select **Export**, and then **Selected object(s)**.

   The **ViewApp Export Preferences** dialog box appears with an option to export any graphics and other objects included with the ViewApp.
4. Select whether or not to export objects included in the View App and click **OK**.

   The **Export Selected objects** dialog box appears with a folder tree to specify where the exported file should be saved and a field to name the export file.
5. Select the folder to save the export file.
6. Assign a name to the export file.

   The default export file name is the name of the first selected ViewApp from the Template Toolbox.
7. Click **Save**.

   A horizontal bar shows the progress of the ViewApps being loaded into the export file.
8. Verify the export aaPKG file has been placed in the folder you specified.

Import a ViewApp

You can import ViewApp to the Templates folder of the System Platform IDE. Exported ViewApps are saved in an aaPKG file.

During the import process, you have the choice to keep or overwrite any ViewApps with the same name currently in the Template folder with the imported ViewApps.

To import one or more ViewApps

1. Open the IDE and display the **Templates** pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. Select the **Galaxy** ribbon, then select **Import**.
3. Select **Objects**, then **From package**.

   The I**mport Objects from package** dialog box appears to specify the folder containing a ViewApp export file.
4. Go to the folder where the aaPKG file containing ViewApps is located.
5. Select the aaPKG file and click **Open**.

   The **Import Preferences** dialog box opens with options to import ViewApps. You cannot have two ViewApps with the same name or more than one copy of the same version of a profile in the same Galaxy. When you import a ViewApp, you can choose options from the **Import Preferences** dialog box to control how you want naming and version conflicts handled.
6. Click **OK** to import the ViewApps.

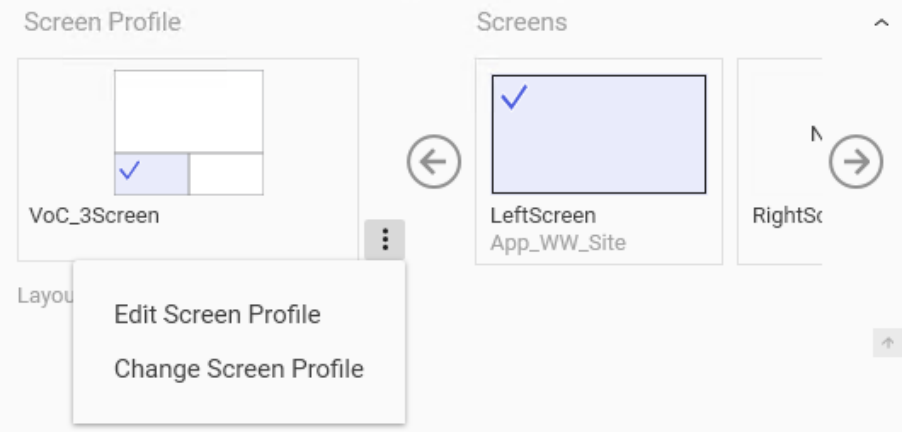
   A progress bar shows the ViewApps being imported into the Templates folder. After finishing the import, you should see the ViewApps in the list of Templates folder objects.

Configure a ViewApp

Typically, you create screen profiles and layouts before integrating them into a ViewApp with the ViewApp Initialization Wizard. The ViewApp Editor provides the capability to replace the current screen profile or edit it by opening the Screen Profile Editor from within the ViewApp Editor. You can also remove a layout associated with a screen or edit it by opening the Layout Editor from within the ViewApp Editor. These topics describe how to manage the screen profiles and layouts associated with a ViewApp from the ViewApp Editor.

Manage screen profiles in a ViewApp

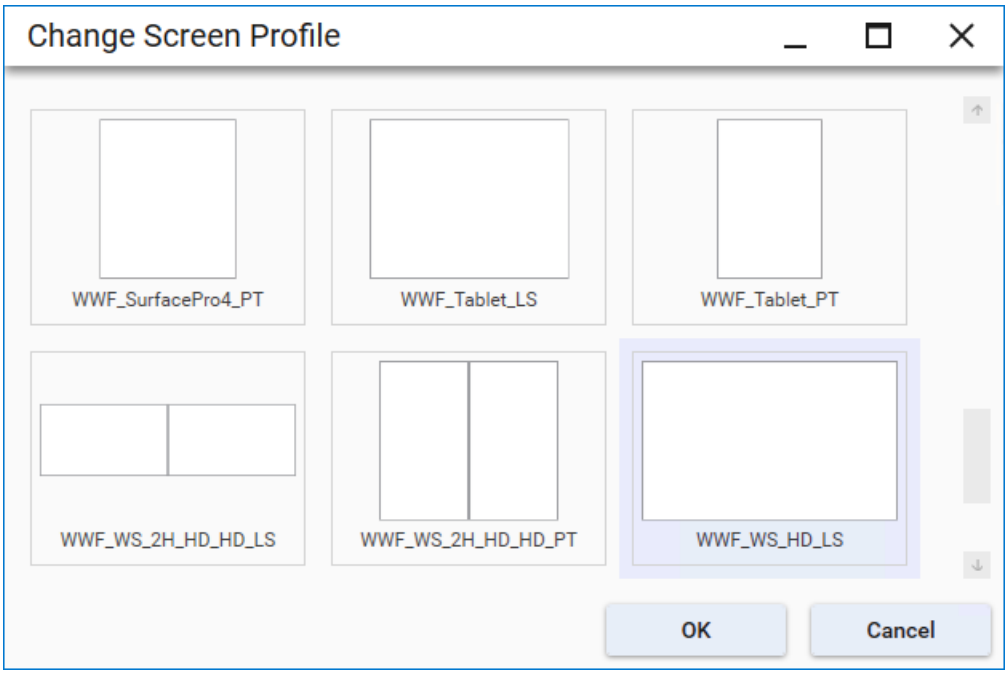
You select a screen profile with the Initialization Wizard the first time you open the ViewApp Editor. Thereafter, the ViewApp editor shows the screen profile you selected. You can change screen profiles or edit the one you selected within the ViewApp Editor by selecting the **Screen Profile** menu button.



If you select **Edit Screen Profile**, the Screen Profile Editor opens with the current screen profile selected for editing. You can make changes to the screen profile and save it. The Screen Profile Editor closes and returns you to the System Platform IDE.

If you select **Change Screen Profile**, a pop-up window shows thumbnails of other screen profiles defined in the Galaxy.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Select another screen profile from the list and click **OK** to replace the current screen profile with the one you selected.

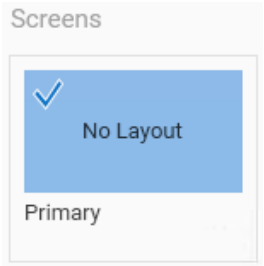
Manage layouts in a ViewApp

You select one or more layouts with the **ViewApp Initialization** wizard the first time you open the ViewApp Editor. Thereafter, the ViewApp editor shows the layouts you selected. You can edit the current selected layout or remove it within the ViewApp Editor by selecting options from the **Layout** menu button.

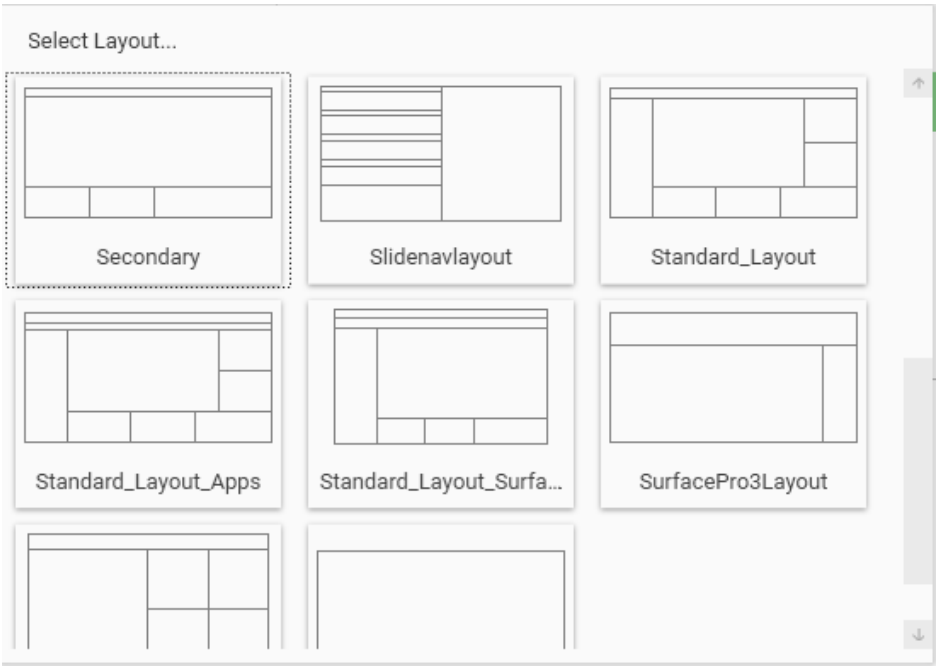
If you select **Edit Layout**, the Layout Editor opens with the current layout shown in the Layout Editor selected for editing. You can make changes to the layout and save it. The Layout Editor closes and returns you to the System Platform IDE.

If you select **Remove Layout**, the layout for the selected screen in the ViewApp Editor is removed. The thumbnail of the layout you removed no longer appears within the selected screen and you see a message that no layout is assigned to the screen. Also, the thumbnail of the layout shown in the ViewApp Editor indicates that it is empty.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



You can also change layouts by selecting another layout from the **Layout** field with the ViewApp Editor. Clicking the arrow in the **Layout** field shows a pop-up menu of all layouts defined in the Galaxy.



Select another layout from the list to replace the current or deleted layout with the one you selected. The selected screen and the layout thumbnail are updated with the layout you selected.

Set additional properties for a ViewApp

The additional settings available for a ViewApp determine the timing of certain actions taken when the ViewApp is open. The available settings are:

- **OnShowTimeoutExtendMs**: This extends the default timeout period for the OnShow script when the ViewApp is opened. The amount you enter is added to the default timeout of 1000 msec (one second). So, the default value of 5000 msec actually sets the OnShow timeout to six seconds. This is the maximum amount of time that the OnShow script will wait this long for the ViewApp's data to be bound. The behavior is:
  - If the data is bound before the timeout period is reached, the OnShow script executes as soon as the data is available.
  - If not all of the data is bound, the OnShow script executes when the timeout period is reached.
- **DataPollingIntervalMs**: How often (in msec) the ViewApp will poll for new data. The default is 1000 msec (one sec).

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **UseLanguageSettingFontOnly**: If this is set to True. when the ViewApp is opened in the OMI web client, it will use only the font that is set for the language the user has selected in the OMI web client. This must be one of the languages defined on the Galaxy configuration **Languages** dialog. This font will override all other font settings for any element of the ViewApp. The default value is False.

To set additional values for a ViewApp

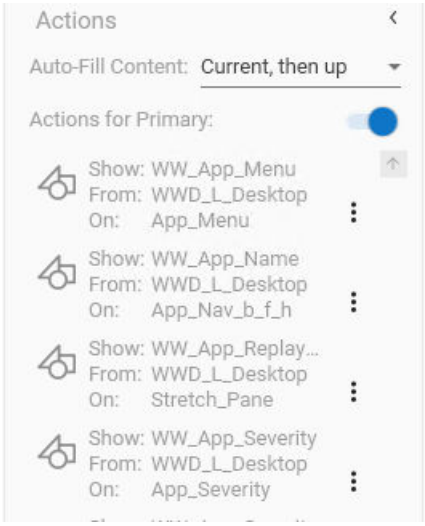
1. On the ViewApp editor, click the **Additional settings** icon at the top right. The settings dialog box opens.
2. On the **Settings** dialog box, enter the desired values in JSON format, like this:

```
{
"Web": {
"OnShowTimeoutExtendMs" : 5000,
"DataPollingIntervalMs" : 1000,
"UseLanguageSettingFontOnly" : False
}
}
```

3. When you are done, select **OK** to save your changes and close the **Settings** dialog box.

Add content to a ViewApp

You can use a drag and drop mouse action or touch gesture to add content to a layout within the ViewApp Editor. Assigning content to a pane adds a Show Content action to the ViewApp's navigation Action List, which coordinates the automatic filling of panes during runtime.



Content from the **Toolbox** or **Assets** tabs at the right of the ViewApp Editor can be dragged on a pane of the layout. Content can include:

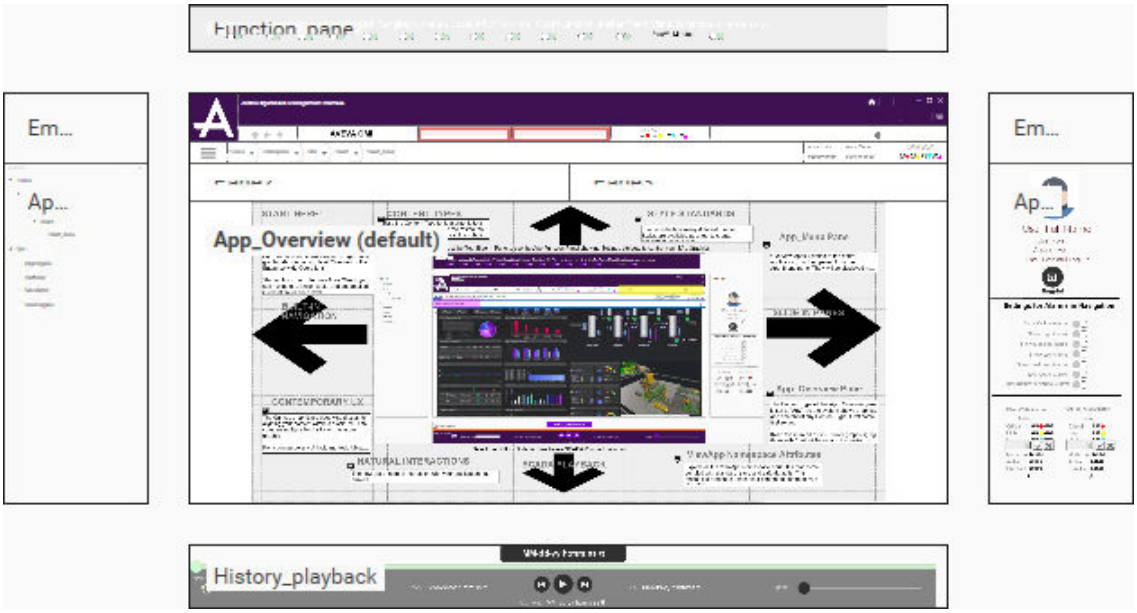
- Graphic
- Layout
- Control

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- App Instance

While content is dragged over a layout, the background color of the target pane changes to indicate the pane will receive the content when it is dropped. The shape of a mouse cursor also changes to indicate the content can be dropped on the pane. After content has been dropped, a thumbnail appears within the pane representing its content.

Note: You can also add content to a pane using a touch device by selecting and holding the content to move it over the target pane.

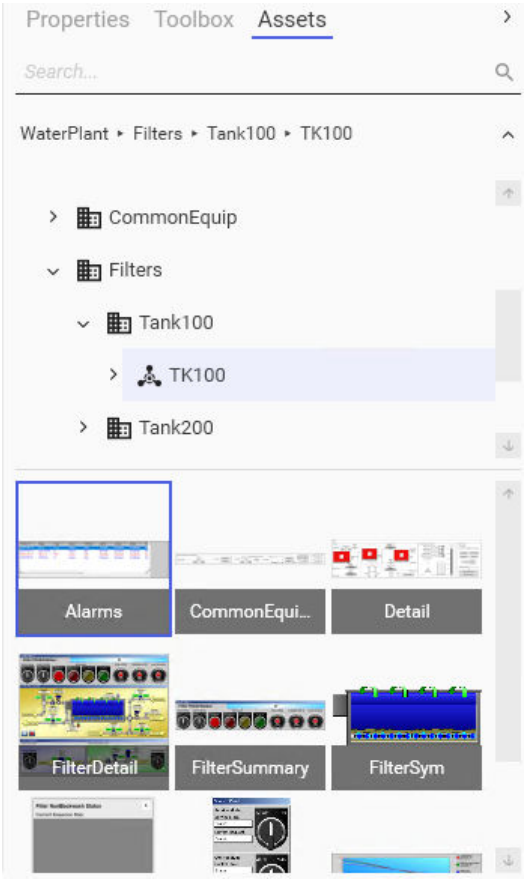


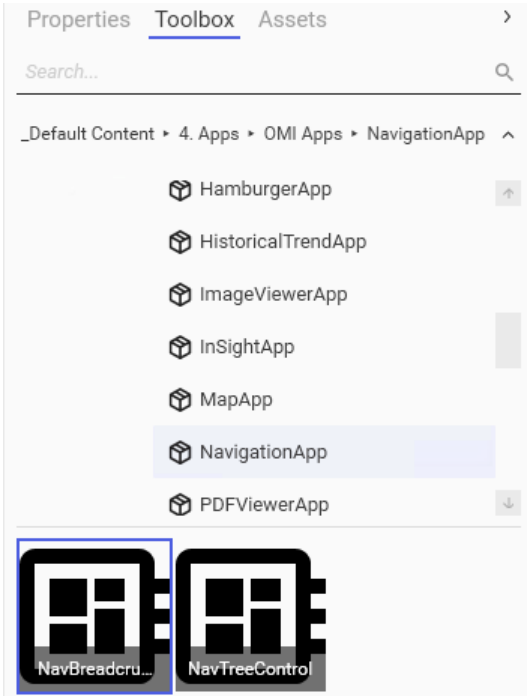
Browse content to add to a ViewApp

You can browse the **Assets** and **Toolbox** tabs to find content that can be added to a layout within the ViewApp Editor. Each item shown in the **Assets** view represents an Area or an Asset.

Each item shown in the **Toolbox** tree represents either a folder or a control library found in the Visualization folder. Selecting an item from a browse tree selects the item and displays any content belonging to that item in the content area beneath the browse list.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.





If a listed item contains sub-items, an expand arrow appears to the left of the item. Clicking or touching the expand arrow retrieves and displays the sub-items. The current selection does not change until an item is selected. If the number of items exceeds the amount of vertical space in the content area of the ViewApp Editor, a scrollbar appears to scroll the list .

If there is insufficient horizontal space to display an item name, a horizontal scrollbar appears at the bottom of the tree to enable you to scroll the information into view.

Search for content

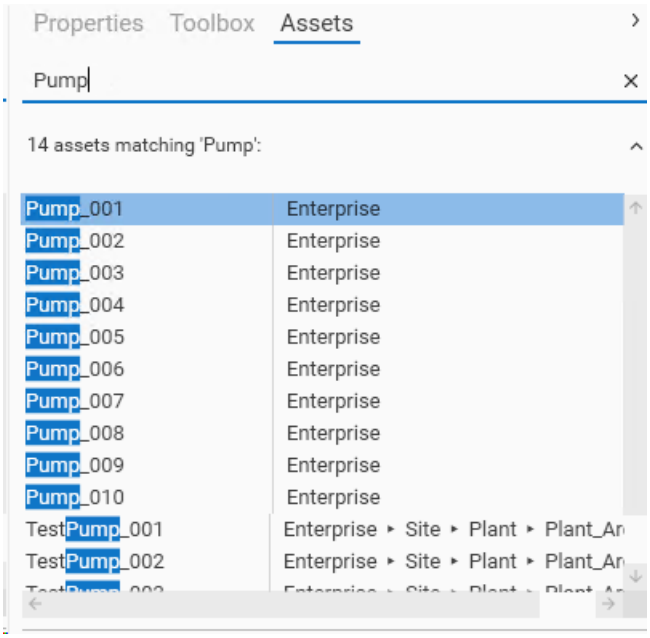
You can search within the currently displayed toolbox folders or assets by entering a search term in the **Search** field. Clicking anywhere in the **Search** field hides the hint text and magnifying glass in preparation to enter a search string.

During the period when you are entering a search string, an X appears in the **Search** field to indicate the string can be deleted. If the X or the Esc key is pressed, the string is cleared from the **Search** field and no search is conducted.

Asset Searches

An Assets search is applied across the entire Galaxy. The search results show the first batch of Areas and assets containing the search term in a multi-column list displayed in ascending alphabetical order. A message beneath the **Search** field indicates the number of matching items found by the search.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

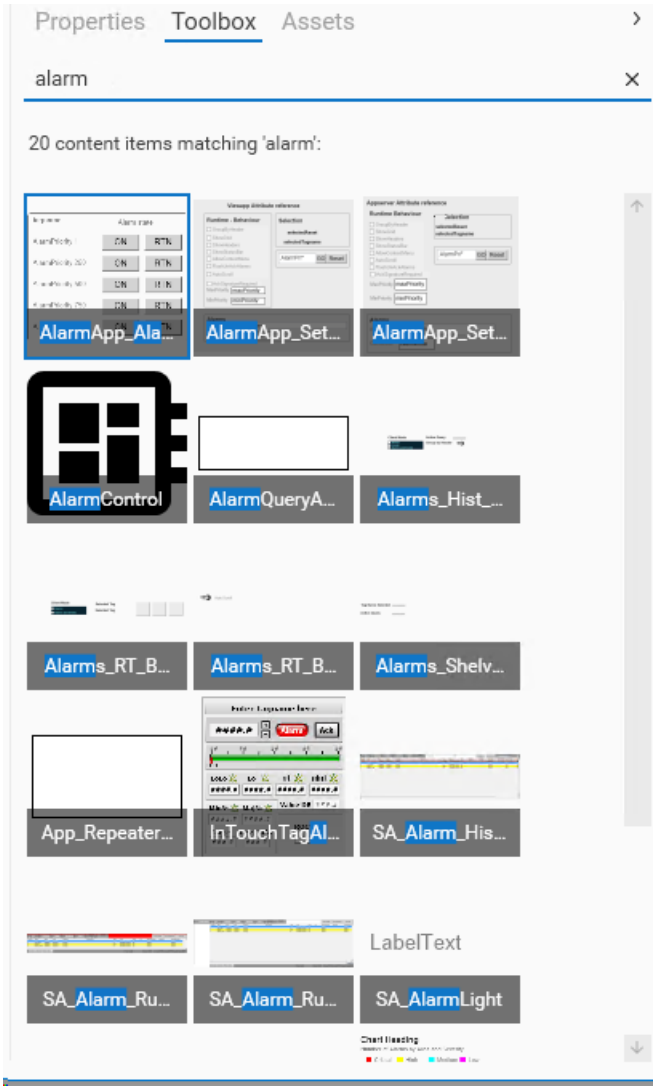


The search results show the matching search term highlighted within their names. You can select an item from the search results list. Selecting an item updates the content view with all content that is owned or linked to the asset you selected.

Toolbox Searches

A Toolbox search is applied across the entire Visualization folder. The search results show thumbnails of content that match the search term in the content area by ascending, alphabetical order. A message beneath the **Search** field indicates the number of matching items found by the search.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



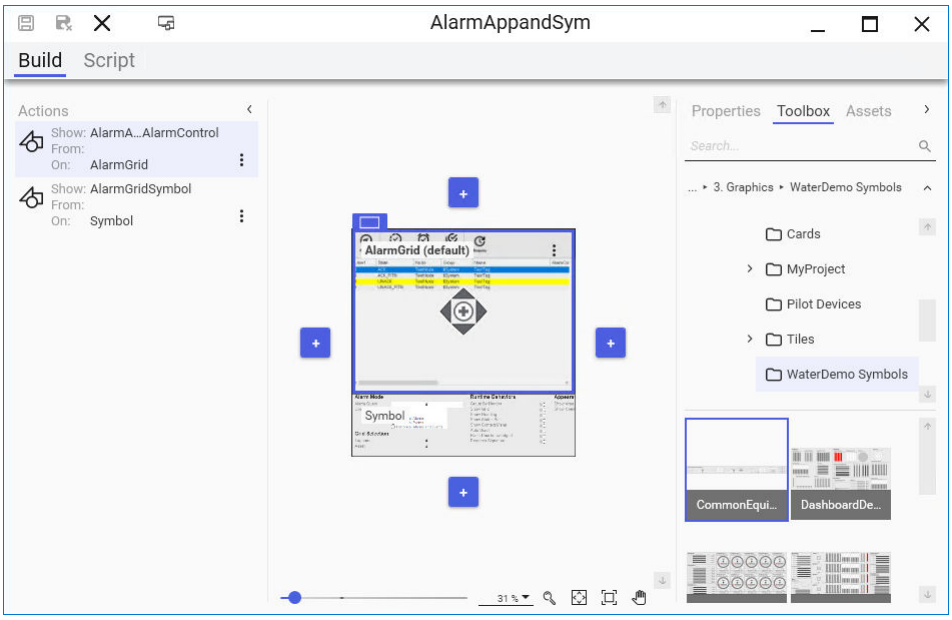
Search results items in the content area can be selected and dropped in a layout pane.

About adding content to an empty pane

If you add content into an empty pane, then,

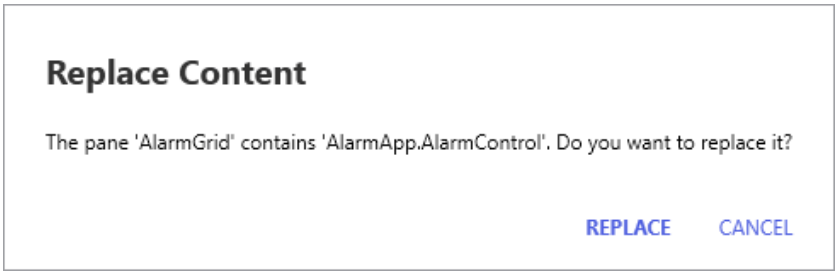
- The **Content** property is set to the name of the content added to the pane
- The **Pane** property is set to the name of the pane hosting the content.
- The **Actions** list updates to show an entry containing the name of the content and the name of the pane hosting the content
- A preview of the content appears in the pane

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: If a layout is added to a pane within the same layout being edited, the cursor changes appearance to indicate the layout cannot be added.

If the target pane being receiving content is a single content pane and already contains content, your see a dialog box asking whether the current content should be replaced or not.



Select **Replace** to update the **Content** property. The thumbnail of the new content appears in the pane.

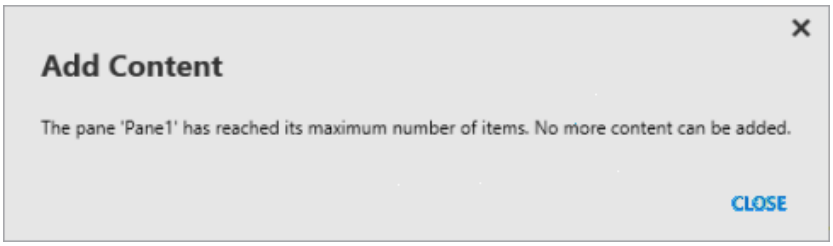
Select **Cancel** to cancel the add content process. The existing content remains in the pane.

Add content to multi-content panes

If a pane is defined as a multi-content pane, dropping a second content item adds the content and the pane is selected. The **Pane** property updates to show the name of the most recent content added to the pane.

If you attempt to drop a content item that will exceed the maximum limit set for the pane, the add content action will fail. A dialog box appears notifying you that the maximum number of content items has been reached and no more content can be added to the pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

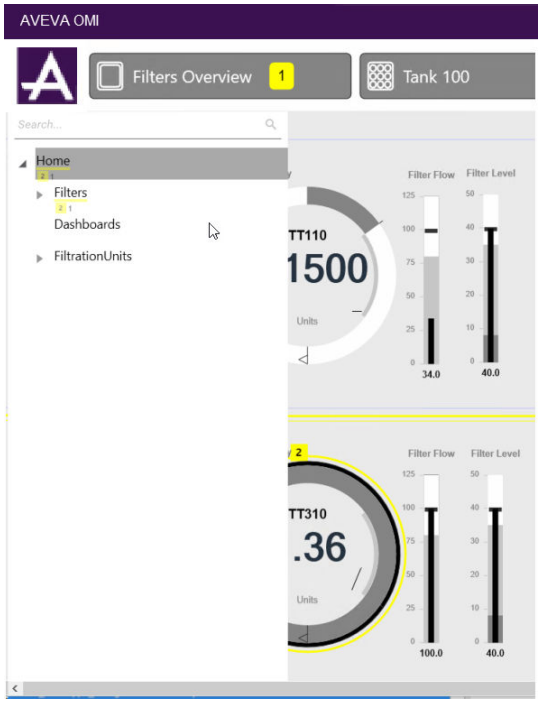


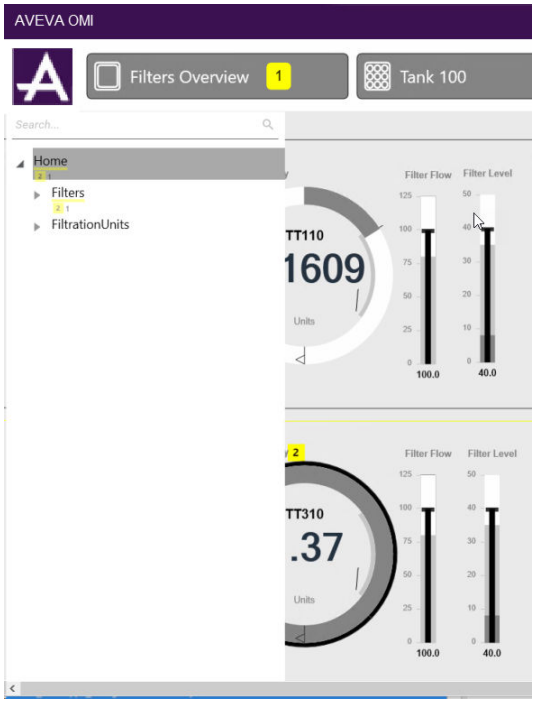
Show or hide ViewApp navigation items

During runtime, ViewApp navigation items can be shown or hidden based on the access level assigned to, and/or user roles selected for, the navigation items. The navigation item will be visible only to logged in users who satisfy each of these conditions, if configured".

Controlling the display of content in a ViewApp by access levels or roles is an effective way of preventing access to a ViewApp's sensitive controls from unauthorized users. The following figures show a ViewApp to monitor tanks and filters in a water treatment plant. The ViewApp includes a **Dashboards** navigation item containing controls that regulate pump flow between tanks, which is the responsibility of supervisors in the plant's control room. Operators only need to see the status of filters and filtration units as part of their work responsibilities.

Navigation item security has been implemented for the **Dashboards** navigation item. The figure on the left shows the ViewApp navigation item shown to supervisors who need access to **Dashboards** controls. The ViewApp on the right hides the **Dashboards** navigation item from operators.





About navigation item security

Security properties can be configured differently to implement various types of navigation item security.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Security	Navigation Item Property Configuration	Navigation Item Visibility Requirements
No Security	**Access Level**=Not Configured **User Roles**=Not Configured	Navigation item always visible
Access Level Only Security	**Access Level**=Integer <*N*>in the range 0-9999 **User Roles**=Not Configured	• Security must be configured in the Galaxy • User must log in to the ViewApp • Logged-in user must have an access level equal to or higher than the specified value <*N*> of the **Access Level** property. • The logged-in user can belong to any role
User Roles Only Security	**Access Level**=Not Configured **User Roles**=*Role1*,*Role2*,*Role3*	• Security must be configured in the Galaxy • User must log in to the ViewApp • Logged-in user can be assigned any access level • Logged in user must be a member of at least one of the specified user roles.
Access Level and User Roles security	**Access Level**=Integer <*N*>in the range 0-9999 **User Roles**=*Role1*,*Role2*,*Role3*.	• Security must be configured in the Galaxy, • User must log in to the ViewApp • Logged-in user must have an access level equal to or higher than the specified value <*N*> of the **Access Level** property. • Logged in user must be a member of at least one of the specified user roles.

Configure navigation item security

Before you can configure navigation security in a ViewApp, the following prerequisite tasks must be completed:

- Security must be configured for the Galaxy to authenticate users by their user names and passwords as part of the ViewApp login process.
- Security roles must be assigned to those users who will interact with a running ViewApp. Each user must be assigned an access level and user role. For more information about implementing security for users by role and access name, see About Galaxy security.

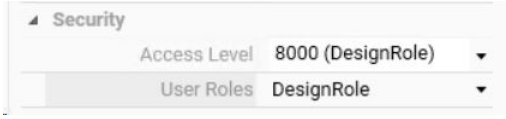
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Decide how you want to implement security. Navigation items containing secure, restricted content must be assigned an access level, or user role, or both.

The following procedure explains how to assign security to a navigation item associated with content that must be secured

To configure navigation item security

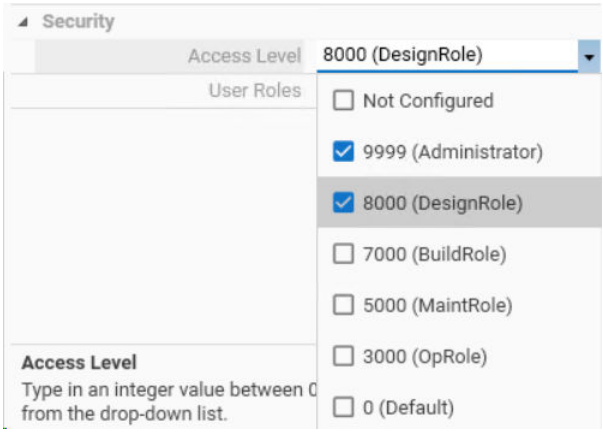
1. Open the ViewApp with the ViewApp Editor.
2. Expand the **Navigation** area of the ViewApp Editor to show the navigation tree listing the items you want to apply security.
3. Select an item from the navigation tree that you want to apply navigation security to.
4. Select the **Properties** tab in the ViewApp Editor to show the **Access Level** and **User Roles** properties.

   The following steps of this procedure vary based on how you want to implement navigation item security.

   - If you only want to implement **Access Level** security, omit steps 7 and 8.
   - If you only want to implement **User Role** security, omit steps 5 and 6.
   - If you want to implement both **Access Level** and **User Role** security, complete all steps.

5. Select the **Access Level** property field to show a drop-down list of access levels by user roles.

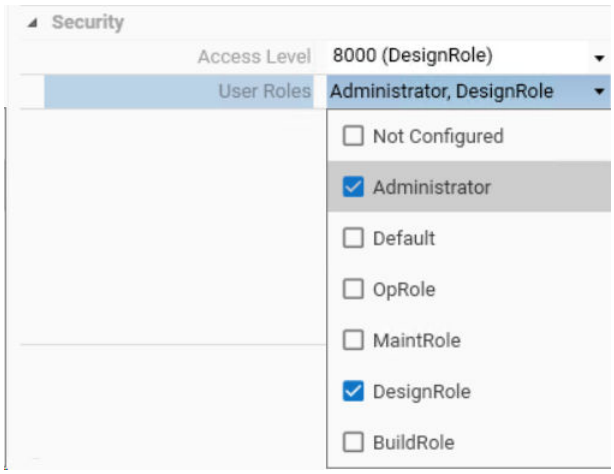
   

6. Select an access level from the list.

   Note: You can also enter an access level less than 9999 directly in the field without selecting an access level from the list. Also, any listed access level greater than the one you selected is also selected in the list.

7. Select the **User Roles** property field to show a drop-down list of user roles.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



8. Select one or more user roles from the list.

   The selected roles appear in the **User Roles** field in a comma delimited list.

9. Save your changes.

User permissions and navigation item security

The comparison of a navigation item's access level to the logged on user's access level follows inheritance rules. If a parent navigation item is not visible during runtime because its access level is greater than the logged on user's access level, all child navigation items are also hidden, even if the child navigation items have been assigned access levels less than the user's access levels.

If the user cannot access a navigation item during runtime, then the content from the next accessible navigation item is displayed based on the selected Auto-Fill mode. A comparison is made between the user's access level and the next navigation item's access level in the Auto-Fill sequence. If the user is authorized to see the content, it appears in the ViewApp. Otherwise, the Auto-Fill sequence continues until content is shown from an authorized navigation item or the last item in the Auto-Fill sequence is reached.

About previewing a ViewApp

Building an AVEVA OMI ViewApp typically involves significant internal testing before placing a ViewApp into production. Builders must follow an iterative process of modifying a ViewApp and then viewing the running ViewApp to assess the effects of development changes. To run a ViewApp, the Galaxy must be deployed. This repetitive edit, deploy, and assessment process is lengthy and labor intensive.

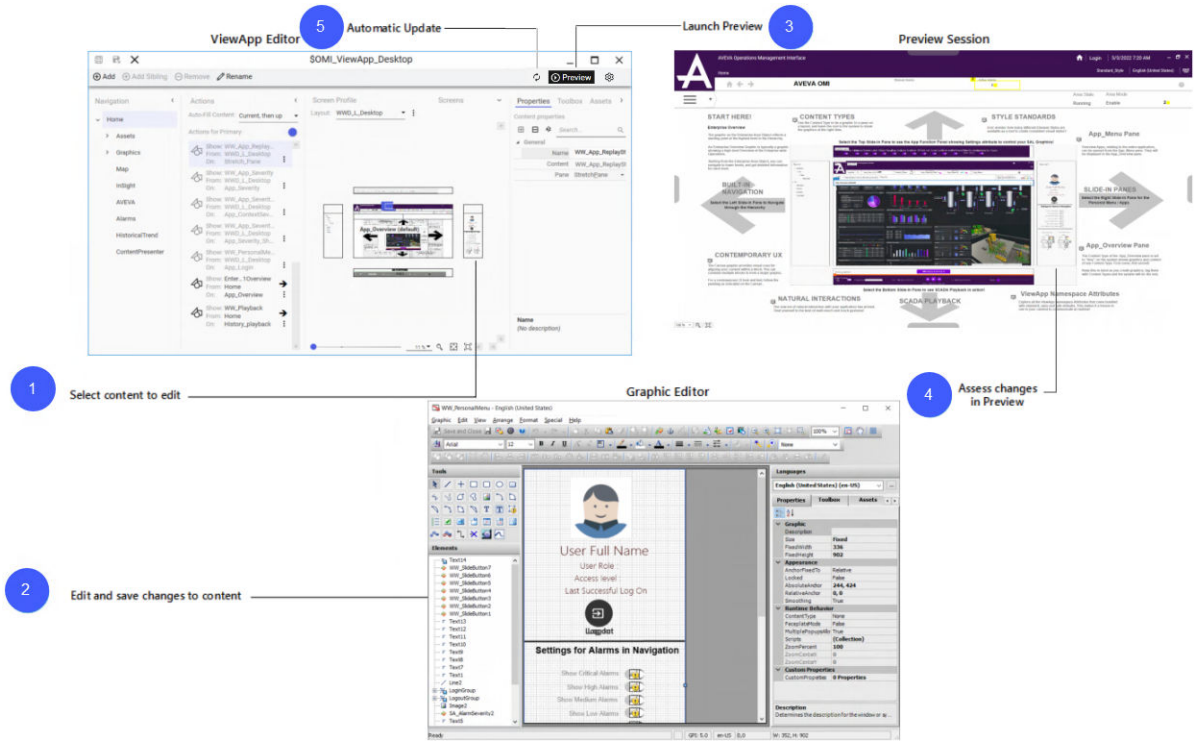
AVEVA OMI now provides a preview session, which enables builders to quickly assess their changes in a running ViewApp without deploying the ViewApp. A preview launches immediately and runs in a separate window to show the current content being edited.

The following figure shows the major steps in an example workflow to edit a graphic within a ViewApp and then launch a preview session to assess the editing changes in the ViewApp. Select the **Preview** button to preview the ViewApp in a native ViewApp. Preview sessions can remain open while you continue editing the ViewApp.

With some exceptions, when changes are made to the ViewApp, the preview refreshes to show the updates that were made. You can toggle the **Auto Update** icon next to the **Preview** button to activate or prevent updates to a preview session.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: You cannot preview what a ViewApp will look like in the OMI web client. While the OMI web client display of a ViewApp is generally very similar to the desktop client display, to see exactly how a ViewApp looks in the web client, you must assign it to a WebViewEngine, deploy it, and open it in the web client. If you make changes to the ViewApp while it is open, refresh the web page to see their effect in the web client.



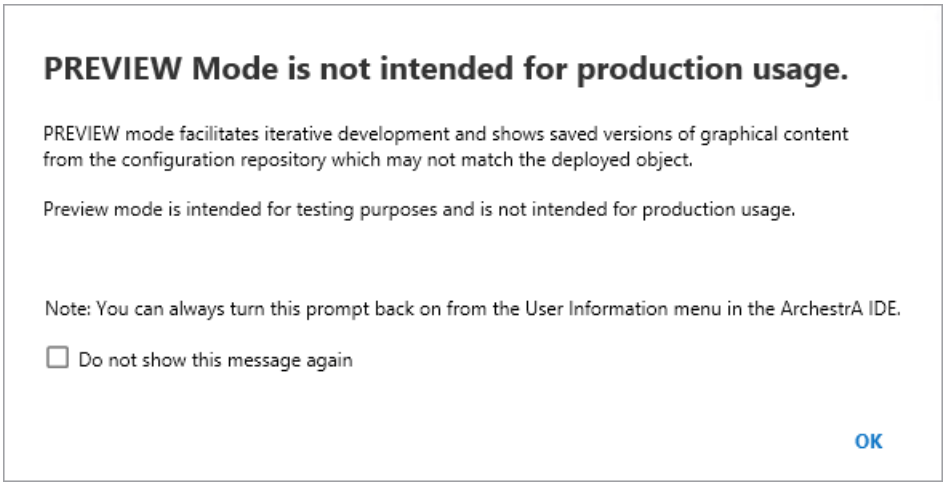
Preview a ViewApp

The following procedures explain how to launch and close a ViewApp preview. For a complete list of content and ViewApp functions supported in a preview session, see About ViewApp previews at different stages.

Before you request a preview, you should decide if it is acceptable that alarm and historical data are not updated after you make changes to your ViewApp.

A message appears when you initially request a preview that explains preview mode is intended for testing purposes only and not intended for production usage.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



A second message appears if a WinPlatform instance is not deployed for the local computer where the preview request is made. The message provides options to accept a preview that may not show accurate historical and alarm data, or cancel the preview request. If you want to ensure that accurate data appears in a preview, deploy a WinPlatform instance for the local computer before submitting a preview request.

Both messages include a check box to hide the message when the check box is selected. If you want to restore a message that you hid, select the appropriate preview prompt option from the **Configure User Information** dialog. For more information, see Configure User Preferences.

To preview a ViewApp

1. Decide how you want to run your preview session.
   - If you want to ensure that all live data in a preview will be updated, deploy the WinPlatform instance for the local computer in which the ViewApp preview will run.
   - If you do not want to deploy the WinPlatform instance on the local computer and are willing to run a preview that does not update live data, select **OK** when you see the warning message.
2. Open a ViewApp in the ViewApp editor.
3. Select **Preview** shown in the upper right corner of the ViewApp editor.

   The first message appears with a caution that preview mode is not intended for production usage.
4. Click **OK**.

   If you did not deploy the ViewApp's Galaxy WinPlatform object, you see a second warning message about the integrity of historical and alarm data shown in the preview.
5. Click **OK** to start the preview session.

   The preview session launches from the computer hosting the System Platform IDE. The preview window opens and shows the current focus of the ViewApp as it is edited in the ViewApp editor.

   Note: When previewing a ViewApp for the first time after launching the System Platform IDE, it takes several seconds for the **Preview** button to become enabled while the ViewApp editor is preparing the preview data.

   If a Preview is active and you have made changes to the ViewApp without saving them, the caption on the **Preview** button changes to Update:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

6. Make changes to one or more component parts of the ViewApp.

   You can select the **Preview** button any time during an editing session. After you select **Preview** again:

   - If the ViewApp contains unsaved changes, the ViewApp will be saved and the new configuration is passed to Preview.
   - If Auto Update is off and some contents (layout, graphics) have been changed, the changes will be passed to Preview.
   - If the navigation node in ViewApp Editor has been changed, the new navigation node will be passed to Preview.
   - If the current selected screen has been changed, the current selected screen will be activated in Preview.

7. Select **Save**.

   The word **Preview** appears on the button and the preview updates to show the changes you made to the ViewApp.

   Note: If you have multiple instances of the same content in a ViewApp, only the edited instance is updated in the preview. Instances of the same content in other panes appear unchanged in the preview.

8. Open the preview window and verify the ViewApp is running.

To close a ViewApp preview

You can also close a ViewApp preview session by doing any of the following:

- Close the System Platform IDE by selecting the **Close** button.
- Close the System Platform IDE by selecting **Exit** from the System Platform IDE **Galaxy** menu bar item.
- Close the ViewApp editor by selecting the **Close** button.
- Close the edited ViewApp by selecting **Save and Close** from the ViewApp editor.
- Close the edited ViewApp by selecting **Close** from the ViewApp editor.
- Select the **Close** button of the preview window.

Other atypical editing actions will also close a preview session:

- Disconnect from the Galaxy Repository
- Restart the computer running the System Platform IDE.
- Change Galaxies from the System Platform IDE.
- Change Galaxy security from the System Platform IDE

About ViewApp previews at different stages

When a builder requests a preview of the ViewApp open in the ViewApp editor, the initial preview shows the current content of the ViewApp's constituent components. The preview shows similar runtime data that would appear in a deployed ViewApp.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Initial Preview

The following list describes the components of a ViewApp that appear in an initial preview request.

- Graphics
  - Graphics from the Visualization folder
  - Embedded graphics
  - Graphics currently placed in ViewApp panes
  - Graphics associated with objects
  - Data values associated with graphic custom properties, animations, or attributes.
- Objects
  - Attributes assigned to an object
  - Graphics associated with an object
  - Hierarchy of objects in the navigation model.
- Screen Profile associated with the ViewApp
- Layout associated with the ViewApp
- Alarms
  - Active alarms from any runtime node.
  - Alarm severity and acknowledgement status
- Apps
  - Standard apps placed in ViewApp panes
  - Custom apps placed in ViewApp panes
- Current selected item of the ViewApp's navigation model
- Current values assigned to the ViewApp namespace
- Navigation security by assigned Access Levels and and content types.
- Galaxy Styles
  - element styles
  - Format Styles
  - Quality and Status Overrides
- ViewApp language
- Historical playback
  - Displays real-time data from attributes whose data is stored in the Historian.
  - Playback controls can be modified by the user from the preview session.
  - Displays historical data from the period set by the playback controls.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Preview Updates

After editing changes have been made to ViewApp content, the running preview updates to show the changes in the ViewApp after the changes have been saved. Not all changes to a ViewApp necessarily update the preview session.

If a preview session is open, then the following types of editing changes will not appear until the ViewApp editor is closed and opened again.

- Change in language settings
- Change to a Screen-Profile
- Change in Asset hierarchy
- Change to an alias-name
- Change to the object order in the hierarchy shown by the Model view

The remainder of this topic describes the supported updates to a preview session by the various components that can be included as ViewApp content.

Graphics

- Visual or functional changes to embedded graphics.
- Changes to the content type or Access Level assigned to the pane in which the graphic is placed.

Apps

A preview session updates when the following changes are made to apps or controls embedded in a ViewApp.

- Change the properties of apps embedded in a ViewApp.
- Rename an app.
- Delete an app

  Note: After renaming or deleting an app, all preview sessions show a popup dialog that indicates the preview session needs to be shutdown or restarted.
- Import an aaPKG file containing one or more new apps and place the imported apps on a layout pane.
- Delete and import an app with updated versions of DLL files with the same name as the original deleted app.
- Change the global configuration values of the InSightApp, Historical TrendApp, and the AlarmApp controls.

ViewApp Navigation

A preview session updates content rendering when the following changes are made to a ViewApp navigation model:

- Change the autofill mode of a navigation item.
- Add or remove custom navigation items.
- Change the position of a custom navigation item in the hierarchy of the navigation model.
- Change the reference that navigation item points to
- Change the property of the navigation item
- Add or remove the content of a navigation item.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Change the current selected navigation item

ViewApp Layout

A preview session updates when the following changes are made to one or more layouts associated with a ViewApp:

- Delete a layout
- Rename a layout
- Add, delete, merge, or resize a layout pane.

  Note: An updated preview shows any panes that have been added or deleted. A message appears when a pane is deleted indicating the content associated with the deleted pane cannot be found.
- Change layout properties.
- Change the properties of a pane within a layout.
- Add or delete a pane from a layout.
- Change the content associated with a layout pane.
- Change properties of content associated with a layout pane.

ViewApp Namespace

A preview session updates when changes are made to a ViewApp Namespace or the attributes associated with it:

- Add, delete, or modify attributes associated with a ViewApp Namespace.
- Rename a ViewApp Namespace.

Historical Playback

A preview updates to show real-time or historical data when the playback controls are adjusted by the user.

Global Styles

Changes to a ViewApp's assigned Galaxy Styles (element styles, Format Styles, and Quality and Status Overrides) will be updated automatically If the WinPlatform is deployed at the IDE node. The ViewApp editor must be closed and reopened again if the WinPlatform is not deployed to show any changes to Global Styles in the preview session.

Alarms

A running preview updates to show the following alarms changes if the computer hosting the preview has its WinPlatform deployed. For more information about the requirements to see historical and alarm data in a preview, see Preview a ViewApp.

- Active alarm monitoring from any runtime node.
- Alarms can be acknowledged or shelved in an active preview session.

Import aaPKG File

A preview session does not refresh during the period when an aaPKG file is being imported. After the import is completed, the preview session refreshes with the changes in the Galaxy.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Switch ViewApp language at runtime

AVEVA OMI language features enable you to develop ViewApps that can be switched to another language at runtime. To enable runtime language switching, you must:

- Configure multiple languages for the ViewApp.
- Export your application text for offline translation.
- Translate one or more exported dictionary files.
- Import one or more translated dictionary files.

This section describes the features and procedures for configuring languages and enabling runtime language switching.

Define and configure galaxy languages

System Platform language features enable you to develop applications that can be switched to another language at run time. To enable runtime language switching, you must:

- Configure multiple languages for the application.
- Export your application text for offline translation.
- Translate one or more exported dictionary files.
- Import one or more translated dictionary files into the Galaxy.

You can set the Galaxy default language and you can define additional languages for purposes of switching graphic and alarm languages at runtime.

The default language is set and languages are added only at the Galaxy level.

Graphics language switching

Runtime language switching applies to all portions of the graphics or animations that you create or configure using the Industrial Graphic Editor or InTouch WindowMaker. The language to display is set when the application is being shown at runtime.

You can only view translations at design time for Industrial Graphics in the Graphic Editor. You cannot switch languages in WindowMaker at design time to see the translations.

About localization workflow

The typical workflow for adding translations to a Galaxy consists of the following:

1. From the **Galaxy** menu, select **Export**, select **Localization**., then choose which option you want to use. If you did not select any objects in the IDE before opening the **Export** menu, the options for choosing **selected** graphics, alarm fields, and internationalized strings will not be active:
    - All internationalized strings
    - Internationalized strings from selected objects (inactive if no objects were selected)
    - Selected Industrial Graphics (inactive if no objects were selected)
    - All Industrial Graphics

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Selected Alarm Fields (inactive if no objects were selected)
- All Alarm Fields

2. When exporting Industrial Graphics or internationalized strings, select the language you are exporting for translation and a destination folder.

   When exporting alarm fields, select as appropriate:

   - Export all alarm messages to a single file
   - Create one translation file per area
   - You can also limit the export to only alarm messages that have not yet been translated.

     Note: For applications that only target sub-areas of the Galaxy, exporting by Area is a more optimal workflow.

   When exporting internationalized strings, you can also limit the export to strings that have not yet been translated.

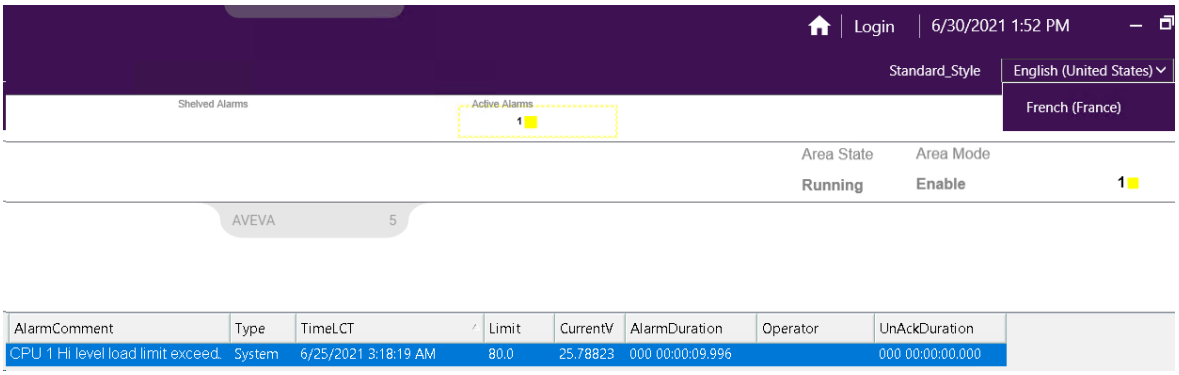
3. After exporting the file, add translations for each field to be localized in the file.
4. Import the translated file(s) back into the Galaxy.
5. For InTouch HMI, import the translated language files into InTouch.
   - In stand-alone InTouch, you can perform a single import operation.
   - In managed InTouch, you must import the language file into each managed InTouch application.
   - For InTouch applications that only target subareas of the Galaxy, exporting by Area is a more optimal workflow.

   Since AVEVA OMI applications access the imported files directly from Application Server, there is no additional import required for AVEVA OMI applications.

Alarm app language switching

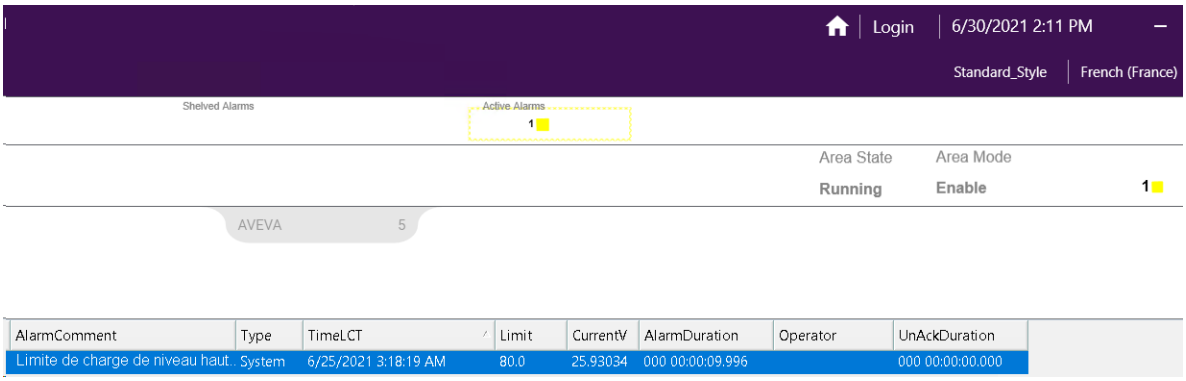
The Alarm app supports alarm comment language switching, which enables you to switch the language of alarm comments that appear in the Alarm app while a View app is running. During runtime, several methods can be used to switch the language of alarm comments that appear in the ViewApp. The following example shows how to switch to another language using the TitleBar app.

Alarm Comment Shown in the Default English Language



Alarm Comment Shown in French by Switching the Language from the TitleBar App

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The following table shows the major steps to enable alarm comment language switching and references for more information.

Alarm Comment Translation Workflows	**For More Information, See**
Configure another language for the Galaxy	Configure languages for a Galaxy
Export alarm comments in a single file to be translated	Exporting Alarm Comments for Offline Translation (see Export alarm comments for offline translation on page 506)
Translate alarm comments into another language	Translate exported alarm comment language files
Import the translated alarm comments	Import translated alarm comment language files
Configure the language switching method	Testing Language Switching Functionality at Run Time

If you change an alarm comment after exporting the alarm comments, you must re-export the alarm comments. The translated alarm comments imported into InTouch and displayed at runtime do not change dynamically when the alarm comment is edited.

Configure languages for a Galaxy

The language settings of the Galaxy control which languages are available to graphics and alarm and event messages. You cannot add a language at the graphic or attribute level. Languages are only added at the Galaxy level using the IDE.

When you open a graphic for editing using the Graphic Editor, open a managed InTouch application in WindowMaker, or open an AVEVA OMI ViewApp, the language settings are retrieved from the Galaxy.

For example, you configure English and French languages for the Galaxy. You open graphic G1 in the Graphic Editor. graphic G1 is now configured with English and French languages. Using the IDE, you add German to the list of configured languages. You must close the Graphic Editor and open graphic G1 again to see the German language available for the graphic.

For more information about language switching in Industrial Graphics, see the *Creating and Managing Industrial Graphics User Guide*.

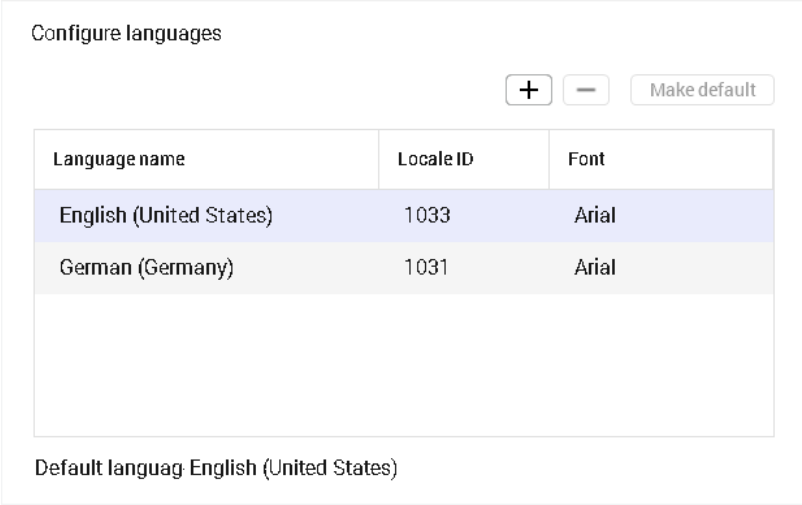
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Add a language to a Galaxy

Every Galaxy is associated with a base language. You must configure any additional languages that you want to be supported by the Galaxy.

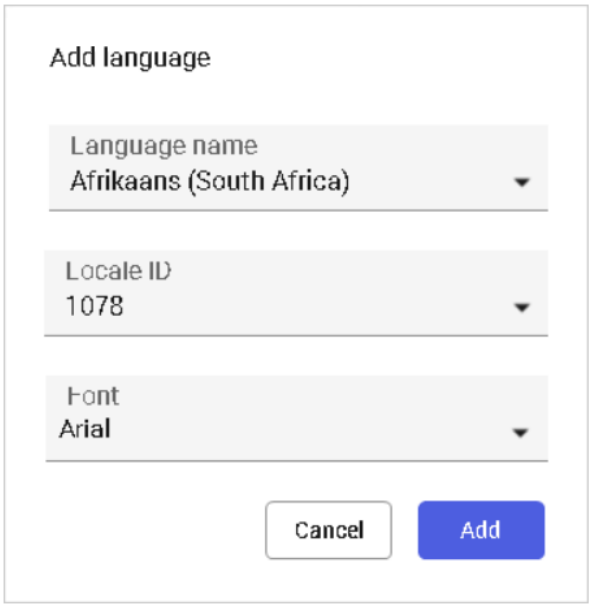
To add a language to a Galaxy

1. Using the System Platform IDE, open the Galaxy that you want to add a language to.
2. From the ribbon, select **Galaxy**, then select **Configure**.
3. Select **Galaxy**, then **Languages**. The **Configure Languages** dialog box appears.



The **Configure Languages** dialog box shows the base (default) language of the Galaxy.

4. Click the **Add [+]** button. The **Add Language** dialog box appears.



5. Specify the language and font for the translated text.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- In the **Language name** or the **Locale ID** list, select the language to add. If you select the language by name, the corresponding locale ID appears in the **Locale ID** list, and vice versa.
- In the **Font** list, select a font.

6. Click **Add** to close the **Add Language** dialog box. The language you selected now appears in the **Configure Languages** dialog box.
7. To add more languages, repeat steps 4 through 6.
8. To specify a particular language as the default, select the row in the list and then click **Make Default**.
9. When you are done, click **Save**.

Remove a language from a Galaxy

You cannot remove the default language. At least one language must be configured for a Galaxy.

To remove a language for a Galaxy

1. Using the System Platform IDE, open the Galaxy from which you want to remove a language.
2. From the ribbon, select **Galaxy**, then select **Configure**.
3. Select **Galaxy**, then **Languages**. The **Configure Languages** dialog box appears.
4. Select the language to remove and then click the **Remove[-]** button.
5. When the **Confirm Delete** dialog box appears, click **Delete**.
   - The dialog box provides an additional option to delete the alarm comment translations for the language selected. Leave the check box empty (unselected) if you want to keep the alarm comment translations for the selected language, but just want to remove the language selection option from the galaxy.
6. Click **Delete** to confirm deletion of the selected language (and the alarm comment translations for that language if selected)**.**

Modify the font for a language

The default font for all languages is Arial. You can only set the font for a language at the time you add it. To change the font setting for a previously configured language, you must first delete it and then re-add it as described in Add a language to a Galaxy.

You cannot edit the font for the default language. To change the font for the default language:

1. Set a different language as the default.
2. Delete the language for which you want to change the font.
3. Add back the language and set the font.
4. Set the language you just added as the default.

The configured font for a language is used in design time when a specific translation is supplied for a language. It is also propagated to all translations for secondary languages when the fonts are not specifically overridden by you.

For example, if you create a text element and provide translation for the secondary language without modifying the font, the text is shown using the font from the Galaxy. However, if you manually modify the font, then the text will always use the chosen font. This same behavior also applies to runtime.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For an Industrial graphic to show the updated font from the Galaxy in an InTouch ViewApp, WindowViewer must be restarted. No notification is provided to WindowViewer when the Galaxy font changes.

All managed InTouch applications are updated to use the new font for the selected language.

Change the default language for a Galaxy

When you change the default language for the Galaxy:

- The new default language is shown when the Graphic Editor opens.
- The base language for translation export is changed to the new default language.
- You cannot import translations for that language.
- Graphics opened in design time use the default language when showing text that does not have a specific translation in a secondary language.
- Graphics shown at runtime use the default language if specific translated values are not provided for the currently viewed language.
- Thumbnails for graphics are shown using the default language.

To change the default language of a Galaxy

1. From the ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Languages**. The **Configure Languages** dialog box appears.
3. Select the language you want to set as the new default language.
4. Click **Make Default**.

   The selected language is now the default language for the Galaxy.

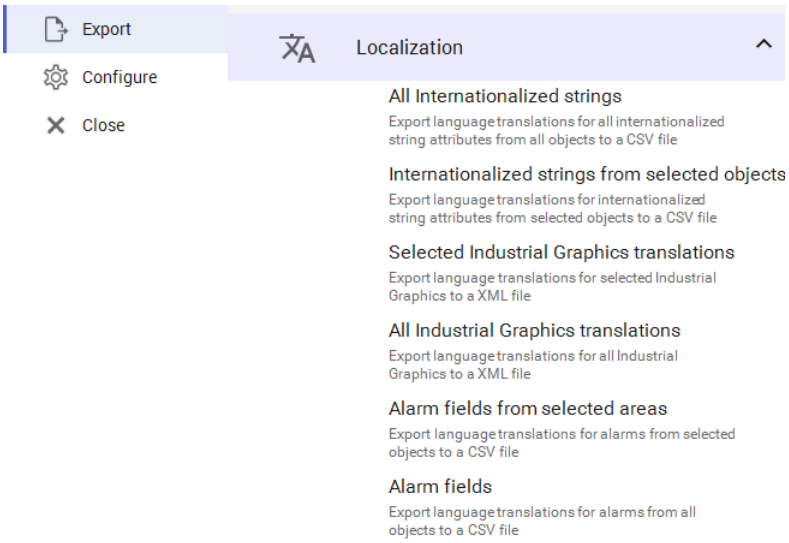
Export files for offline translation

Typically, text strings are sent out for bulk translation. You can export translation files and modify them using a text editor or a spreadsheet program like Microsoft Excel.

Application Server has the following export options for localization:

- All Internationalized strings
- Internationalized strings from selected objects (one or more objects must be selected before this option can be used)
- Selected Industrial Graphics (select the graphics you want to localize before you use this option)
- All Industrial Graphics
- Alarm Fields from selected areas (select the area objects you want to localize before you use this option)
- All Alarm Fields

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



If you make changes to your application after you export your dictionary files, you must export the dictionary file again. For more information, see Export text to an existing dictionary file.

You can only export the text strings for one language at a time. You cannot export Industrial Graphics text strings for the default language.

In order to localize a string attribute, it must be of type InternationalizedString. Not all attributes are this type. For a list of InternationalizedString attributes, see Attributes of Type InternationalizedString.

Each graphic in a Galaxy is only exported one time, no matter how many times it is referenced.

When you export the text, you specify a folder for the dictionary files. Creating a new folder to export phrases for each language makes it easy to manage dictionary files. For example, ...\Galaxy1\My German Files\.

Also, alarm message files are exported as tab-delimited text files and have the following convention: Galaxy_*<GalaxyName>*_*<LanguageID>*.txt. If you will be exporting language data for different objects at different times, use separate target directories to prevent subsequent exports from overwriting the first export.

When you export the dictionary for an application, you can copy and paste the text file into Microsoft Excel.

Types of language dictionary files

Export operations to support localization can create the following types of dictionary files:

- Dictionary file (type XML) for all Visualization folder graphics, template graphics, and AutomationObject graphics. The file naming convention for graphics is: *<GalaxyName>*_*<LanguageID>*.xml. For example, if the Galaxy name is TestSample and the language being exported is French (Language ID = 1036) then the file name is TestSample_1036.xml.
- Dictionary file (type TXT) for internationalized string attributes. The file naming convention for internationalized strings is: Galaxy_*<GalaxyName>*_*<LanguageID>*_Internationalized_strings.txt
- Dictionary file (type TXT) for alarm messages, group names, state messages, and event messages. The file naming convention for alarm messages is: Galaxy_*<GalaxyName>*_*<LanguageID>.*Alarm_Comments.txt
- Dictionary file (type XML) for each managed InTouch application.
- Dictionary file for each SmartSymbol (type XML) in managed InTouch applications.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: Alarm export files are text files and are therefore named and formatted differently from graphic dictionary files, which are XML files. For further information see Export alarm comments for offline translation.

Export language data for all Industrial Graphics in a Galaxy

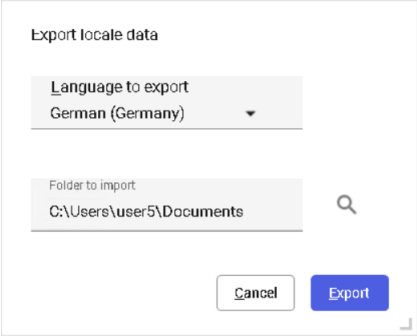
You can export language data for all Industrial Graphics in a Galaxy at one time. The export contains language data for:

- Visualization folder graphics.
- Graphics contained in AutomationObject templates, except for an $InTouchViewApp template.
- Graphics contained in AutomationObject instances.

The export operation only applies to graphics that are checked in.

To export language data for all Industrial Graphics

1. Using the System Platform IDE, open the Galaxy for which you want to export graphic text.
2. From the ribbon, select **Galaxy**, then select **Export**.
3. Select **Localization**, then **All Industrial Graphics**. The **Export Locale Data** dialog box appears.



4. Configure the export settings.
   - In the **Languages to export** list, select the language dictionary to export. The default language is not listed.
   - In the **Path** box, type the folder to which you want to export the dictionary. Click the browse icon to select an existing folder or create a new folder.
5. Click **Export**. The export progress is shown as the XML file is created.
6. When file creation is complete, click **Close**.

Export language data for specific objects

You can export language data for the following types of Galaxy objects:

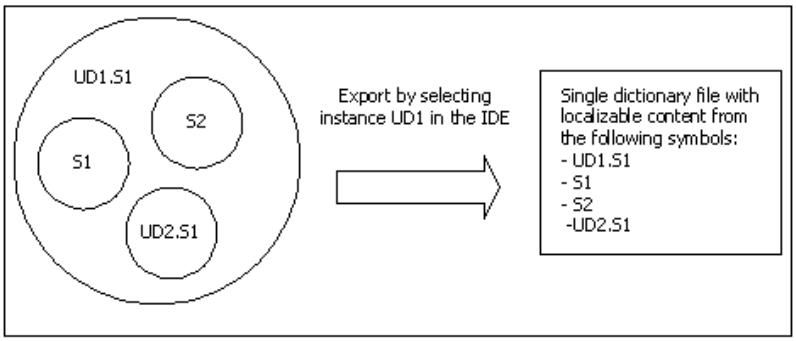
- Visualization folder graphics
- AutomationObject templates and instances
- $InTouchViewApp templates and instances.

All graphics in an instance or template are exported.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

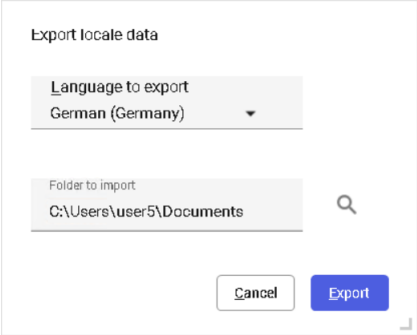
When you export the language data for a graphic, the language data for all graphics referenced by the graphic is also exported. The exported graphics can be referenced through direct embedding or through a show graphic animation.

For example, a Galaxy has Symbol1 and Symbol2 defined in the Visualization folder. There are two instances called UserDefined1 and UserDefined2. UserDefined1 includes Symbol1. UserDefined2 includes Symbol1 and Symbol2. The instance graphic UserDefined1.Symbol1 embeds Symbol1 and Symbol2 from the Visualization folder and one instance graphic UserDefined2.Symbol1. If you select the instance UserDefined1 and export the language data, then the language data would also be exported for the graphics Symbol1, Symbol2, and UserDefined2.Symbol1.



To export language data for specific objects

1. Using the System Platform IDE, open the Galaxy for which you want to export graphic text.
2. Select one or more objects to export.
3. From the ribbon, select **Galaxy**, then select **Export**.
4. Select **Localization,** and then select **Selected Industrial Graphics**. The **Export Locale Data** dialog box appears.



5. Configure the export settings.
   - In the **Language to export** list, select the language dictionary to export. The default language is not listed.
   - In the **Path** box, type the folder to which you want to export the dictionary. Click the browse icon to select an existing folder or create a new folder.
6. Click **Export**. The export progress is shown as the XML file is created.
7. When file creation is complete, click **Close**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Export graphic language data for a managed InTouch application

You export language data for a managed InTouch application using the IDE. You cannot export translations from within the InTouch HMI.

The export includes strings for:

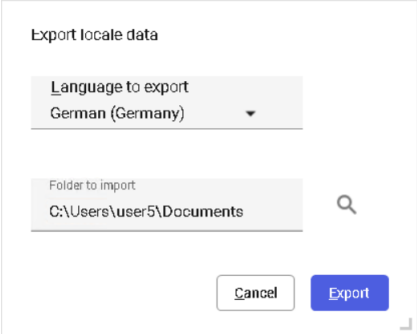
- All InTouch windows
- All SmartSymbols
- Industrial Graphics referenced by the $InTouchViewApp template

The export causes a cascade export of all referenced Industrial Graphics.

When you export language data for a managed InTouch application, the default language for the Galaxy is ignored. The InTouch default locale is always the installed InTouch locale. If the InTouch installed locale and the Galaxy default language are not the same, the export is skipped for the selected InTouchViewApp and a message is logged.

To export language data for a managed InTouch application

1. Using the System Platform IDE, open the Galaxy for which you want to export graphic text.
2. Select one or more managed InTouch applications.
3. From the ribbon, select **Galaxy**, then select **Export**.
4. Select **Localization,** and then select **Selected Industrial Graphics**. The **Export Locale Data** dialog box appears.



5. Configure the export settings.
   - In the **Language to export** list, select the language dictionary to export. The default language is not listed.
   - In the **Path** box, type the folder to which you want to export the dictionary. Click the browse icon to select an existing folder or create a new folder.
6. Click **Export**. The export progress is shown as the XML file is created.
7. When file creation is complete, click **Close**.

Export graphic language data for a published InTouch application

If you export languages for a published InTouch application, the following are not included:

- Embedded Industrial Graphics
- Custom property overrides

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- String overrides.

The export only includes native InTouch translations.

Export text to an existing dictionary file

If you change your application after you translate the text strings, you need to export the text again. For more information, see Export files for offline translation.

If you export a file with the same name to the same directory, you will be asked to confirm that you want to overwrite the existing file.

If you click **Yes**, the existing file is deleted and the current text for graphics in the Galaxy are exported as a new file.

Any existing translations for a graphic are reflected in the new .xml file.

Translate exported graphic language files

The procedures and tools for translating the exported language files are similar for both graphic text and for alarm comments. However, there are important differences. Procedures for each are described in this section.

Translate exported graphic text dictionary files

After you export the dictionary file containing your application text, use Microsoft Excel to edit the text.

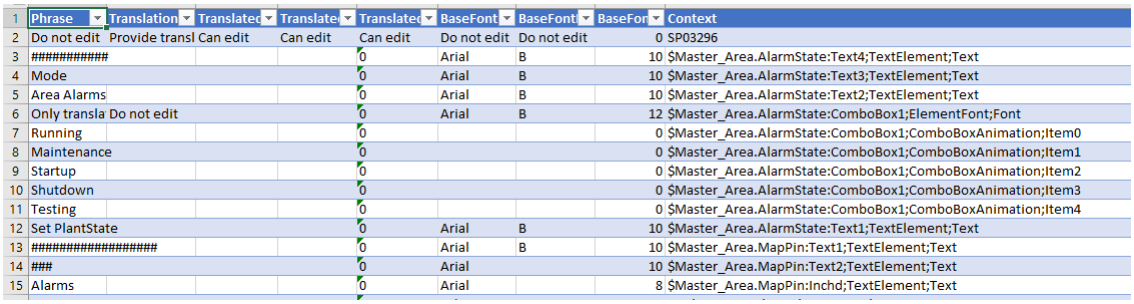
To translate an exported dictionary file

1. Open the XML file in Excel. The **Open XML** dialog box appears.
2. Click **As an XML** list, then click **OK**. A message may appear informing you that an XML schema will be created.
3. Click **OK**.

   The XML file opens in Excel with columns for the:

   - Phrases in your application.
   - Translated phrases from the translator.
   - Translated font name.
   - Translated font properties.
   - Translated font size.
   - Base font properties.
   - Base font size.
   - Context, phrase ID, language ID and foreign language ID.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



	Phrase	Translation	Translated	Translate	Translate	BaseFont	BaseFont	BaseFon	Context
2	Do not edit	Provide transl	Can edit	Can edit	Can edit	Do not edit	Do not edit	0	SP03296
3	##########				0	Arial	B	10	$Master_Area.AlarmState:Text4;TextElement;Text
4	Mode				0	Arial	B	10	$Master_Area.AlarmState:Text3;TextElement;Text
5	Area Alarms				0	Arial	B	10	$Master_Area.AlarmState:Text2;TextElement;Text
6	Only transla	Do not edit			0	Arial	B	12	$Master_Area.AlarmState:ComboBox1;ElementFont;Font
7	Running				0			0	$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item0
8	Maintenance				0			0	$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item1
9	Startup				0			0	$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item2
10	Shutdown				0			0	$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item3
11	Testing				0			0	$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item4
12	Set PlantState				0	Arial	B	10	$Master_Area.AlarmState:Text1;TextElement;Text
13	##################				0	Arial	B	10	$Master_Area.MapPin:Text1;TextElement;Text
14	###				0	Arial		10	$Master_Area.MapPin:Text2;TextElement;Text
15	Alarms				0	Arial		8	$Master_Area.MapPin:Inchd;TextElement;Text

Important: Only modify data in the **Translation**, **TranslatedFontSize**, **TranslatedFontName**, and **TranslatedFontProperty** columns. Do not change any column header. Do not insert or delete rows.

4. Type the language-specific text in the **Translation** column in the row that corresponds with the base language string in the **Phrase** column.
5. If necessary, change the font parameters for the translated strings. If you only provide a translation, the Galaxy-configured font for the language is used to render the text after the translation is imported. If you specify a font, it overrides the Galaxy-configured font.
   - In **TranslatedFontName** column, type the font name.
   - In the **TranslatedFontProperty** column, type the notation for the font properties:

     B = **bold**

     I = *italic*

     U = underline

     For example, if you want to the text to be bold, type **B** in the **TranslatedFontProperty** column. If you want the text to be bold and underlined, type **BU** in the **TranslatedFontProperty** column.
6. Save the file using XML Data as the file type.

Important: If you save as another file type, such as XML Spreadsheet, Excel changes the schema and the Galaxy cannot load the file. If you change the name of the XML file, the file will not import properly into the Galaxy.

Import translated language files

You can import alarm comment language files for re-export by area to facilitate the export of new, untranslated alarm comments, and to facilitate re-export of previously translated alarm comments that have been changed.

For graphic text, you must import the translated dictionary files for each language to enable runtime language switching for those languages. All dictionary files for a given language should be placed in the same folder.

You can import files for only one language at a time. When you import, you select the desired language and specify the files to import.

Import translated graphic dictionary files

All affected graphics and relevant objects are checked out before the import begins and are checked in when the import is done. If an affected graphic or object is already checked out, a message appears in the progress dialog box and the import is skipped for the checked-out object.

For a published InTouch application, only the native InTouch translations are imported.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

You can configure how you want Galaxy and graphic/object mismatches handled during the import.

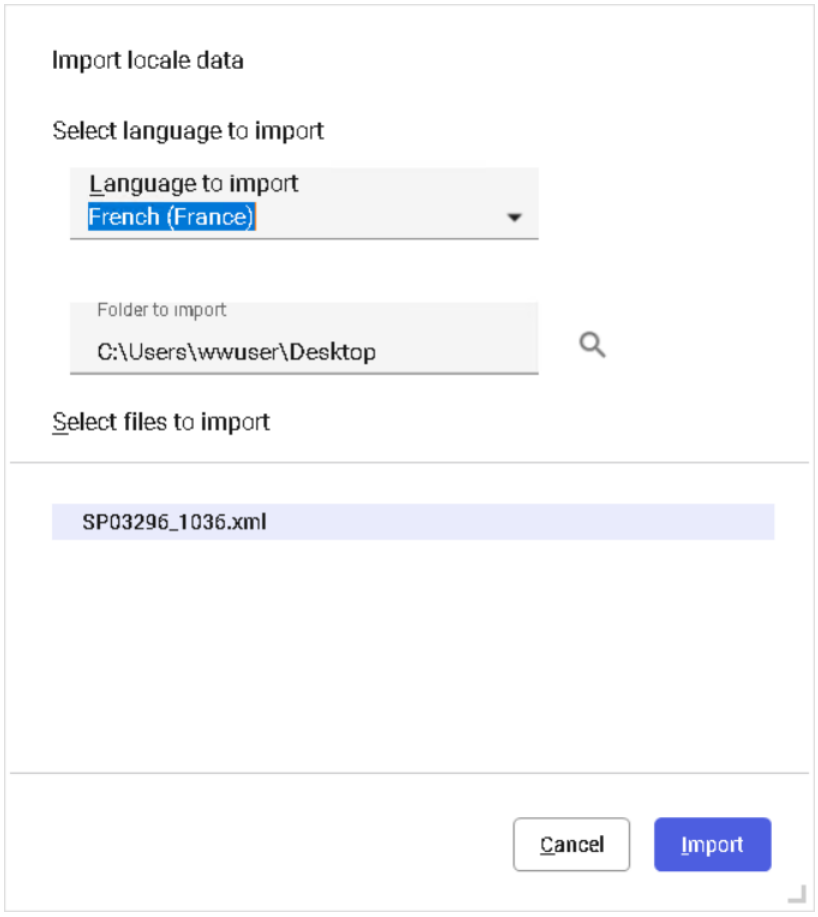
The import is skipped if:

- The default locale specified in the .xml translation file does not match the current default locale of the Galaxy.
- An InTouchViewApp's installed locale does not match the current default locale of the Galaxy.

Important: You cannot cancel the import after it starts.

To import a translated dictionary file

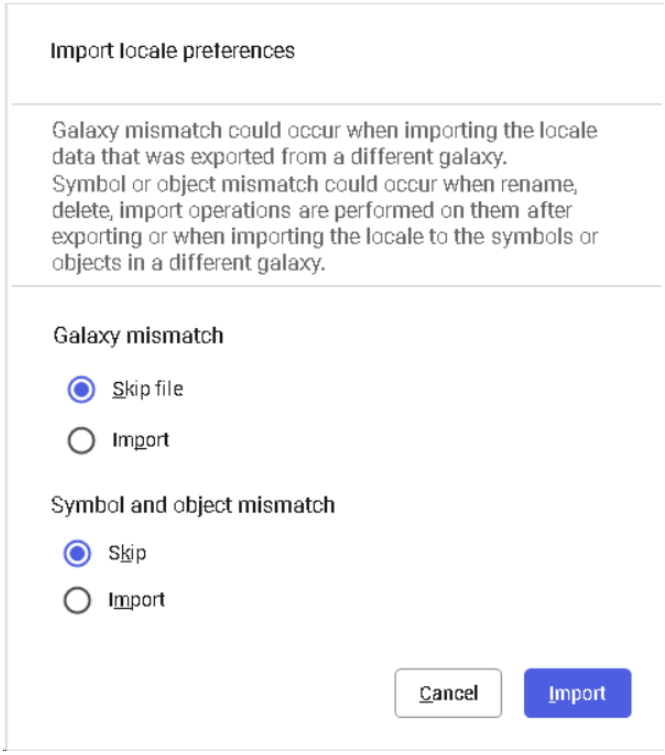
1. Using the System Platform IDE, open the Galaxy for which you want to import graphic text.
2. Close all editors and check in all Galaxy objects.
3. From the ribbon, select **Galaxy**, then select **Import**.
4. Select **Localization**, then select **Industrial Graphics translation**. The **Import locale data** dialog box appears.



5. Configure the import settings.
   - In the **Language to import** list, select the language dictionary to import.
   - In the **Folder to import** box, specify the folder that includes the dictionary file to import.
   - In the **Select files to Import** box, select the .xml files to import. Only files that include the current Galaxy name and the locale ID for the selected language are shown.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

6. Click **Import**. The **Import Locale Preferences** dialog box appears.



7. In the **Galaxy mismatch** area, configure how you want Galaxy mismatches handled. A Galaxy mismatch occurs when you try to import a translation file that was exported from a different Galaxy. The Galaxy name in the translation .xml file is used to match the current name of the Galaxy.
   - Click **Skip file** to skip all the files that do not contain the current Galaxy name in the file name.
   - Click **Import** to import all the selected files, regardless of what the Galaxy name is in the .xml filename.
8. In the **Symbol/Object Mismatch** area, configure how you want graphic and object mismatches handled. A graphic or object mismatch occurs when the name of the graphic and the internal ID (GObjectID) of the graphic do not match what is within the .xml file. Objects include AutomationObjects and InTouchViewApp objects.
   - Click **Skip** to skip the graphics and objects that have mismatched names or IDs in the .xml file
   - Click **Import** to import a graphic or object only if it has a matching name or matching ID. If the name resolves to one object, and the ID resolves to another object, then the import is skipped.

   For examples, see Examples of graphic or object mismatch handling during language imports.
9. Click **Import**. The import progress is shown.
10. Click **Close**. The **Import Language Dictionary Files** dialog box appears.
11. Click **Check In**. The check in progress is shown.
12. Click **Close**. A summary of the import is shown.
13. Click **OK**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Examples of graphic or object mismatch handling during language imports

The following table shows an example of handling mismatch conditions for a Visualization folder graphic. The bold name/ID is the matching name/ID in the .xml file and current Galaxy during import.

	Graphic Name and ID in the Galaxy while Exporting	**Graphic Name and ID in the Galaxy while Importing**	**Change made after export and before import**	**Skip Option Selected**	**Import Option Selected**
1	S1, 100	**S1, 100**	No change	Import to S1	Import to S1
2	S1, 100	S2, **100**	Rename S1 to S2	Skip	Import to S2
3	S1, 100	**S1**, 200	Delete S1, Create/ Import S1	Skip	Import to S1
4	S1, 100	**S1**, 200 S2, **100**	Rename S1 to S2 Create/Import S1	Skip	Ambiguous, skip import
5	S1, 100	No S1 and No 100 in the Galaxy	Deleted S1	Skip	Skip import

The following table shows an example of handling mismatch conditions for an AutomationObject graphic. The bold name/ID is the matching name/ID in the .xml file and current Galaxy during import.

	Graphic Name and ID in the Galaxy while Exporting	**Graphic Name and ID in the Galaxy while Importing**	**Object Name and ID in the Galaxy while Importing**	**Change made after export and before import**	**Skip Option Selected**	**Import Option Selected**
1	Pump1, 10	S1, 100	**Pump1, 10**	No change	Import to S1 only if Pump1 has S1 with an ID of 100	Import to S1 only if Pump1 has S1 with an id of 100

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

	Graphic Name and ID in the Galaxy while Exporting	Graphic Name and ID in the Galaxy while Importing	Object Name and ID in the Galaxy while Importing	Change made after export and before import	Skip Option Selected	Import Option Selected
2	Pump1, 10	S1, 100	Pump2, **10**	Rename Pump1 to Pump2	Skip	Skip import if S1 and 100 pointing to two different graphics in Pump2 (ambiguous) Import to S1 only if Pump2 has S1 or a graphic with id of 100.
3	Pump1, 10	S1, 100	**Pump1**, 20	Delete Pump1 Create and Import Pump1	Skip	Skip import if S1 and 100 pointing to two different graphics in Pump1 (ambiguous) Import to S1 only if Pump1 has S1 or a graphic with id of 100.
4	Pump1, 10	S1, 100	**Pump1**, 20 Pump2, **10**	Rename Pump1 to Pump2 Create and Import Pump1	Skip	Ambiguous, skip import
5	Pump1, 10	S1, 100	No Pump1 and No 10 in the galaxy	Deleted Pump1	Skip	Skip import S1

Language data handling for galaxy operations

If you import or export objects, all language data within the associated graphics is imported or exported.

You can export all objects in the Galaxy, selected instances/templates, or selected graphics from the Visualization folder. All language data is exported as part of the graphics definition, regardless of the languages configured in the Galaxy.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

You can import Galaxy objects that contain language data from the same Galaxy or a different Galaxy. All language data in the Industrial Graphics is imported, regardless of what languages are configured for the target Galaxy.

No Industrial Graphic or InTouchViewApp language data is read or modified during the import operation.

If you import an unmanaged InTouch application, the configured locales in the Galaxy are applied to the unmanaged InTouch application. No element translations in the InTouch application are removed during import, even though languages may no longer be visible in the InTouch HMI or available at runtime in WindowViewer.

Export alarm comments for offline translation

You typically send alarm comment text out for bulk translation. You can export alarm comments and modify them using a text editor or a spreadsheet program such as Microsoft Excel. After translation, the files can be re-imported to the Galaxy for runtime alarm comment language switching.

Guidelines and Recommendations

The following guidelines and recommendations will help you make best use of the alarm comment language switching feature:

Important: The exported file name is generated automatically and must not be changed.

Organize your export

Follow these recommendations when exporting alarm comments for translation:

- You must export alarm comment text for only one language at a time.
- When you export text, you must specify a folder to place the language files. We recommend that you create a separate folder for each language. For example, ...\Galaxy01\ChineseFiles\.
- For large Galaxies, exporting files for each area, after being translated and re-imported in the Galaxy, will perform faster when the alarm clients access those files.
- For small Galaxies, we recommend that you export the entire Galaxy for translation.

Translation file formatting and editing

Microsoft Excel is recommended for translation file formatting and editing, since it supports large worksheets and provides enhanced formatting capabilities to convert the text file into tabular format. See Export alarm comments from very large galaxies.

Reimport

- We highly recommend re-importing the translated files to the Galaxy after translation. Although re-import is not mandatory, you can use the re-import to optimize the export files.
- If you make changes to your alarm comments after you export comment text, you must export the alarm comments again. For this reason, we further recommend using the import functionality to reimport the language file after each translation. This practice will avoid the need to re-translate alarm comments you have already translated.

For further information and procedures, see the following sections:

Export alarm comments from very large galaxies

Exporting Alarm Comments by Area

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Import translated alarm comment language files

About the alarm comments language file

The alarm comment language switching feature exports alarm comments, messages and other data to a .txt file. That file can be edited directly, but working in a text editor is typically more difficult and prone to errors.

- All files export as Unicode. We recommend that you save the exported files as Unicode.
- We recommend that you edit the .txt file using Microsoft Excel. Be sure to save the edited file as a .txt file.

The language file name is not user-configurable. The file naming convention for alarm comment export is:

Galaxy_<GalaxyName>_<localeID>_Alarm_Comments.txt

or

Galaxy_<GalaxyName>_<AreaName>_<localeID>_Alarm_ Comments.txt

where the locale ID is the selected language decimal identification number.

For example, if the Galaxy name is "TestSample" and the exported language is Chinese, the language file name would be Galaxy_TestSample_2052_Alarm_Comments.txt. The locale ID for Chinese is 2052.

For example, if the Galaxy name is "TestSample" and the Area name is "Area001", and the exported language is German, the language file name would be Galaxy_TestSample_Area001_1031_Alarm_Comments.txt.

See Translate exported alarm comment language files for further information about alarm comment language files.

Export alarm comments from very large galaxies

You can export language data in one operation for all alarm comments in a Galaxy. However, very large Galaxies with numerous alarm comments require longer export processing time and can result in a very large .txt language file. For handling large Galaxies and a large number of alarm comments, we recommend that you export alarm comments by Area rather than exporting alarm comments for the entire Galaxy to a single language file. See Exporting Alarm Comments by Area.

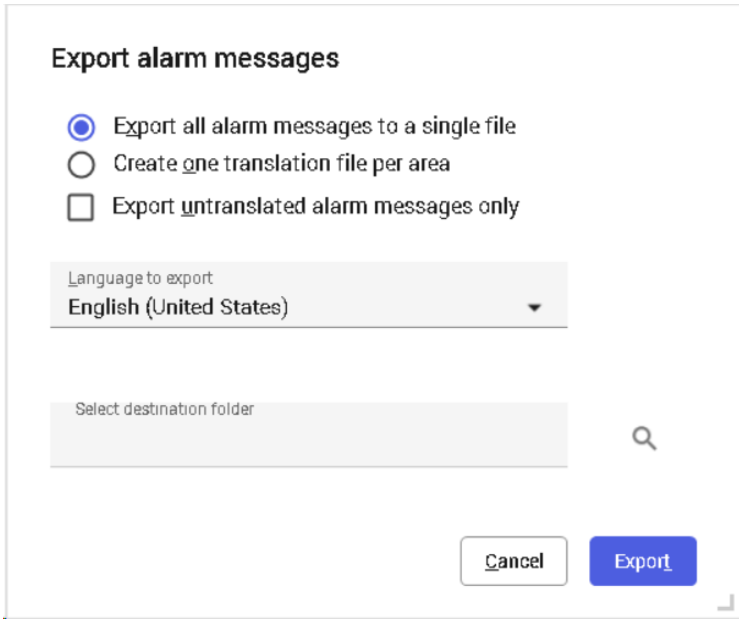
Export all galaxy alarm comments

You can export language data for all alarm comments in a Galaxy at one time. The export operation only applies to objects that are checked in.

To export all Galaxy alarm comments to a single file

1. Using the System Platform IDE, open the Galaxy in which you want to export alarm comments.
2. From the ribbon, select **Galaxy**, then select **Export**.
3. Select **Localization**, then select **Alarm Fields**. The **Export Alarm Messages** dialog box appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



4. Configure the export settings:
   a. Select the **Export all alarm messages to a single file** radio button.
   b. Check the **Export untranslated alarm messages only** check box if you have already exported, translated and re-imported your language file and you want to export only newer, untranslated messages. Otherwise, leave the check box unchecked.
   c. In the **Language to export** pull-down list, select the language to export. Make sure that you have configured the Galaxy for the language that you want to export, since the export file will contain the locale ID of the language you select. You can export only one language at a time.
   d. In the **Path** box, type the folder to which you want to export the language file. Click the browse icon to select an existing folder or create a new folder.
5. Click **Export**. The export progress box appears.
6. Click **Close** when the export completes.

Exporting Alarm Comments by Area

You can export all alarm comments by Area or you can export Alarm comments for a specific Area.

Using File Names

The *Galaxy_<GalaxyName>_<AreaName>_<localeID>_Alarm_ Comments*.txt file name convention applies to specific Area alarm comment exports. Area names will be included the file name. For example, given a Galaxy name "TestSystem", an Area name "Area001", and an export to Chinese (language designation 2052), the language file name would be Galaxy_TestSystem_Area001_2052_Alarm_Comments.txt.

Export objects not assigned to an area

System Objects typically are not assigned to an Area. These are:

- Platform Objects
- Device Integration Objects

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Engine Objects

For purposes of language export, these System Objects form their own "Area" or become notification distributors for the purpose of sending alarms. Exporting System Objects follows these important conventions:

- System Objects can only be exported from the **Galaxy** menu. They cannot be selected and exported using a context menu.
- System Objects can only be exported by exporting alarm comments for all areas.

To export alarm comments for all areas

1. Using the System Platform IDE, open the Galaxy for which you want to export alarm comments.
2. From the ribbon, select **Galaxy**, then select **Export**.
3. Select **Localization**, then select **Alarm Fields**. The **Export Alarm Messages** dialog box appears.
4. Configure the export settings:
   a. Select the **Export all alarm messages to a single file** radio button.
   b. Check the **Export untranslated alarm messages only** check box if you have already exported, translated and re-imported your language files by Area and you want to export only newer, untranslated messages. Otherwise, leave the check box unchecked.
   c. In the **Language to export** pull-down list, select the language to export. You can export only one language at a time.
   d. In the **Destination folder** box, type the folder to which you want to export the language file. Click the browse icon to select an existing folder or create a new folder.
5. Click **Export**. The export progress box appears.
6. Click **Close** when the export completes.

Re-export alarm comments

After exporting and translating alarm comments, you can re-import them into Application Server to help maintain up-to-date alarm comment language files.

After adding new alarm comments that require translation or after modifying existing, already translated alarm comments, you must re-export the language file to update the translations.

Exporting New Untranslated Alarm Comments

In the **Export Alarm Messages** dialog box, select the **Export untranslated alarm messages only** check box. This creates a file named _untranslated.txt. For example, exporting untranslated alarm comments in Chinese for a Galaxy name "TestSystem" would create a file named Galaxy_TestSystem_2052_Alarm_Comment_Untranslated.txt.

Export modified existing alarm comments

You can make changes to alarm comments that you previously exported, translated and imported back into Application Server. You must re-export the alarm comment language file(s) to update the translations.

In the **Export Alarm Messages** dialog box, select the same language option and folder you selected for the previous export. When asked, confirm that you want to overwrite the existing file.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Translate exported alarm comment language files

Alarm comments are exported to a .txt file, located in the selected directory. You can edit the files with a text editor, but we recommend Microsoft Excel as the editor. Using the format-as-table feature in Excel greatly simplifies editing the specific text to be translated.

To translate an exported alarm comment file

1. Open the .txt file in Excel. The Excel **Text Import Wizard** appears. Follow the wizard instructions to import the text file in tab delimited, general column data format. The file appears unformatted in Excel.

Tip: You can open the file in text editor, and then copy and paste the file contents into a blank Excel worksheet instead of using the import wizard. The results will be the same

---

	A	B	C	D	E	F	G	H	I	J	K	L
1	Unique Phrases:											
2												
3	PhraseId (	Default M	Translated Message (EDIT THIS COLUMN)									
4	3	AppEngine										
5	4	Enterprise										
6	5	OMI_ViewApp_Desktop										
7	6	Plant										
8	7	Plant_Area										
9	8	Platform										
10	9	Simulator										
11	10	Site										
12	11	Sys										
13	12	The AppEngine hosts and schedules execution of Application Objects, Areas, and Device Integration Objects.										
14	13	The OPCClient provides connectivity to OPC servers.										
15	15	ViewEngine										
16	16	Device write failed -										
17	17	Device write failed - Array index is out of range -										
18	18	Device write failed - Attribute not writable -										
19	19	Device write failed - Data value out of range -										
20	20	Device write failed - Incorrect data type -										
21	21	Device write failed - Invalid Attribute -										
22	22	Device write success -										
23	23	started										
24	24	started successfully.										
25	25	Stopped										
26	26	stopped successfully.										
27	27	Write access denied -										
28	28	Write access denied - Alarm ACK has already been done -										
29	29	Write access denied - Array index is out of range -										
30	30	Write access denied - Attribute not writable -										
31	31	Write access denied - Data value out of range -										
32	32	Write access denied - Incorrect data type -										
33	33	Write access denied - User does not have Alarm ACK privileges -										
34	34	Write access denied - User does not have permission to modify this type of attribute -										
35	35	Write access denied - User does not have permission to write to this type of attribute										
36	36	Write access denied - Users cannot modify configuration attributes while object is onscan -										
37	37	Write access denied - Verifier does not have permission to verify write for this type of attribute										
38	38	Write failed - Invalid Attribute -										
39	39	Write success -										
40												
41												
42	Alarms:											
43												
44	PhraseId (	Alarm (DO NOT EDIT)										
45	3	[_GROUP_NAME_] AppEngine										

2. Format the text as a table in Excel:
   a. Select all of the rows containing data in Column A through Column C.
   b. Select the **Home tab**, then select **Format as Table**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

c. Select the color pattern you want and click **OK** in the **Format as Table** dialog box. Excel will format the region to be edited. Resize the columns as needed.

	A	B	C
1	Column1	Column2	Column3
2	Unique Phrases:		
3			
4	PhraseId (DO NOT EDIT)	Default Message (DO NOT EDIT)	Translated Message (EDIT THIS COLUMN)
5	3	AppEngine	*<localized text message 1>*
6	4	Enterprise	*<localized text message 2>*
7	5	OMI_ViewApp_Desktop	*<localized text message 3>*
8	6	Plant	*<localized text message 4>*
9	7	Plant_Area	*<localized text message 5>*
10	8	Platform	*<localized text message 6>*
11	9	Simulator	*<localized text message 7>*
12	10	Site	*<localized text message 8>*
13	11	Sys	*<localized text message 9>*
14	12	The AppEngine hosts and schedules execution of Application Objects, Areas, and Device Integration Objects.	*<localized text message 10>*
15	13	The OPCClient provides connectivity to OPC servers.	*<localized text message 11>*
16	15	ViewEngine	*<localized text message 12>*
17	16	Device write failed -	*<localized text message 13>*
18	17	Device write failed - Array index is out of range -	*<localized text message 14>*
19	18	Device write failed - Attribute not writable	*<localized text message 15>*
20	19	Device write failed - Data value out of range -	*<localized text message 16>*
21	20	Device write failed - Incorrect data type -	*<localized text message 17>*
22	21	Device write failed - Invalid Attribute -	*<localized text message 18>*
23	22	Device write success -	*<localized text message 19>*
24	23	started	*<localized text message 20>*
25	24	started successfully.	*<localized text message 21>*
26	25	Stopped	*<localized text message 22>*
27	26	stopped successfully.	*<localized text message 23>*
28	27	Write access denied -	*<localized text message 24>*
29	28	Write access denied - Alarm ACK has already been done -	*<localized text message 25>*
30	29	Write access denied - Array index is out of range -	*<localized text message 26>*
31	30	Write access denied - Attribute not writable -	*<localized text message 27>*
32	31	Write access denied - Data value out of range -	*<localized text message 28>*
33	32	Write access denied - Incorrect data type -	*<localized text message 29>*
34	33	Write access denied - User does not have Alarm ACK privileges -	*<localized text message 30>*

The Excel display is divided into four regions:

- Column 1 contains Phrase ID numbers used internally in the language file. **Do not edit** this column.
- Column 2 contains the exported alarm messages. **Do not edit** this column.
- Column 3 contains the alarm message translations. Edit this column with your translation text.
- Below the alarm message text columns are the alarms and phrase IDs, which map to column 1 and are used internally in the language file. Do not edit this area.

3. Translate the alarm message text. Type the alarm message translations into the appropriate rows in column 3.

Important: Only edit the text in column 3, translations. No other text in the file may be edited. You can edit the original alarm comment text only within the System Platform IDE.

4. Save the file as a .txt file in the directory you originally selected for export.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Important: Do not change the file name or the file type, otherwise it will not import correctly into the Galaxy.

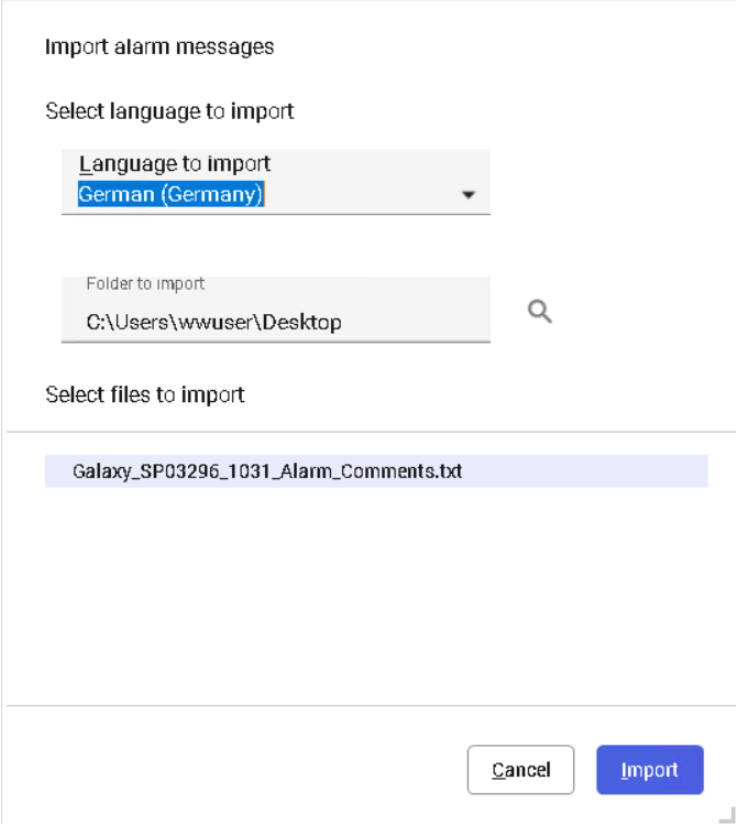
Import translated alarm comment language files

You can import translated alarm comment language files only from the **Galaxy** menu, not from the context menu for specific Areas.

To import alarm comment language files, you must first have exported alarm comments for translation of some or all of the alarm comments.

To import a translated alarm comment language file

1. Using the System Platform IDE, open the Galaxy in which you want to import translated alarm comments.
2. On the **Galaxy** menu, select **Import**, then select **Localization**, and then select the **Alarm Fields** menu option. The **Import Alarm Messages** dialog box appears.



3. Configure the import settings:
   a. In the **Language to import** pull-down list, select the language to import. You can import only one language at a time.
   b. In the **Folder to import** box, type or browse to the folder where you previously exported your language file. The available language files appear in the **Select files to import** list.
   c. Select a file or files to import.
4. Click **Import**. The import progress box appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

5. Click **Close** when the import completes.

Note: If an alarm name or message do not match, the alarm comment import will display a message for each failed alarm message import.

Testing the Language Switching Functionality at Run Time

We recommend that you test the run-time language switching functionality. Follow the procedures outlined in this chapter to enable run-time language switching in your application. Then import the appropriate language file into InTouch. The import step applies to both stand-alone and managed InTouch applications.

To test the language switching functionality

1. Open the application in WindowViewer.
2. On the **Special** menu, point to **Language**, and then click the name of the language to switch to.

   The information from the corresponding translated dictionary file (if one exists and has been imported into InTouch) loads and appears.
3. When you are done, click **Close**.

About predefined language attributes

AVEVA OMI includes a set of predefined Language attributes. These attributes identify the currently-selected language used by the ViewApp by name or by an ID number, and can be used in action scripts and animations to define actions that can change the current language and locale in a running ViewApp. These attributes can also be incorporated programmatically into an WPF-based app that can be added to a ViewApp. When used in an app, users can directly select a new language from a list of configured languages.

A typical runtime language workflow based on language attributes is for the user to select a graphic element shown in a ViewApp that includes an action script or animation containing a Language attribute. The script or animation switches to another pre-configured language. When implemented inside of an app, for example, in the TitleBar app, users can be presented with a drop-down list of all configured languages within the Galaxy, and can select one of the listed languages.

About attribute setting

All language attributes operate in the MyViewApp.Language namespace. Language attribute names are specified with the prefix MyViewApp.Language, in the form MyViewApp.Language.*attribute_name*.

When an attribute is set, it updates the other two attributes with their equivalent values. For example, setting CurrentLanguage will also set CurrentLCID with the LCID number and CurrentCultureInfo with the Culture Information that correspond to the attribute value set for CurrentLanguage.

Attribute Name	Data Type	Retentive	Attribute Type	Initial Value	Description
CurrentLanguage	String	True	Read/Write	None	String value of the language and locale, as specified for the .NET framework. For example, English-United States.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

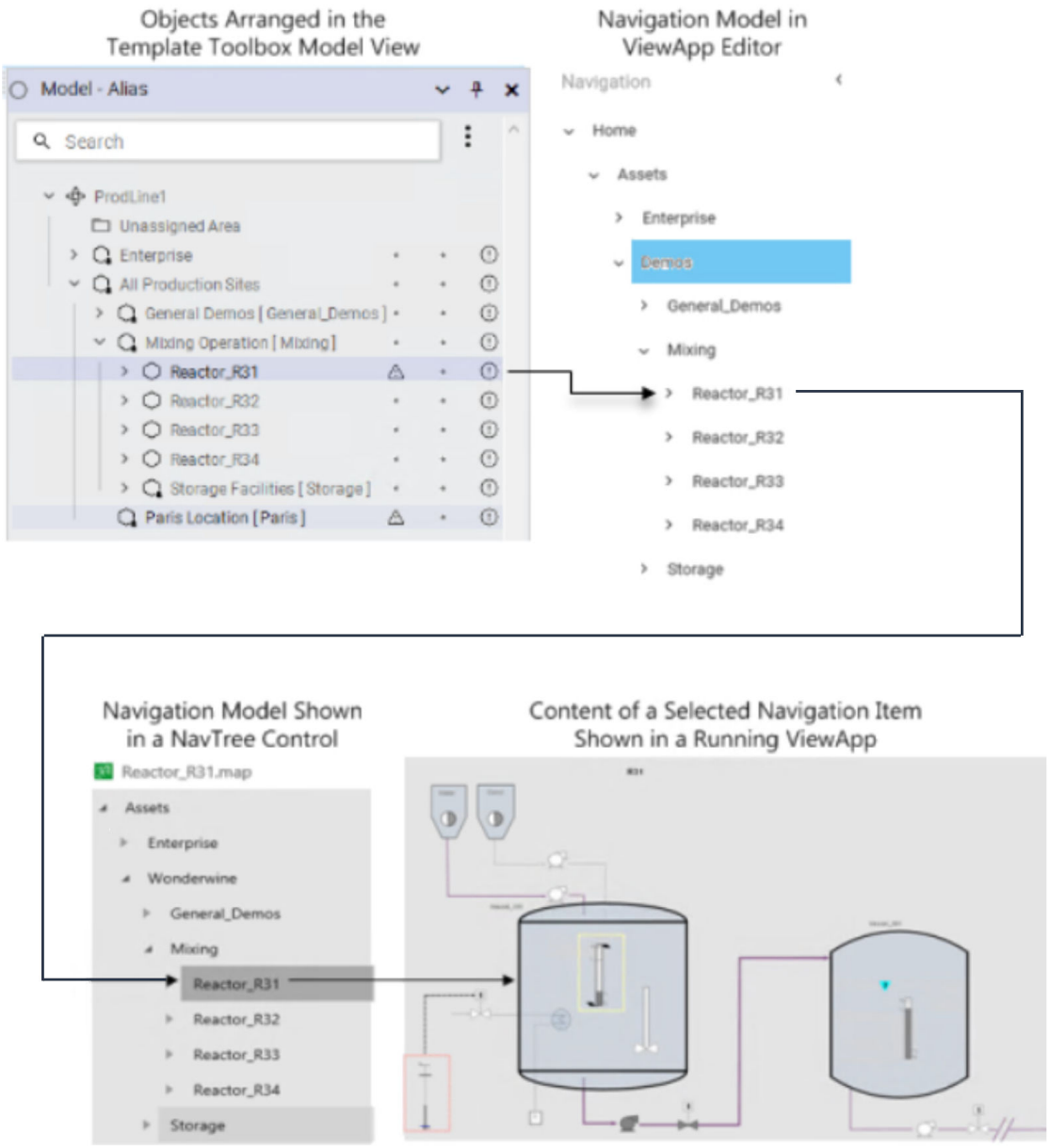
CurrentLCID	Integer	True	Read/Write	None	Integer value of the language and locale, as specified for the .NET framework. For example, 1033 (for English-United States).
CurrentCultureInfo	String	True	Read/Write	None	String value (abbreviated) of the language and locale, as specified for the .NET framework. For example, EN-US.

When an attribute is set, it updates the other two attributes with their equivalent values. For example, setting CurrentLanguage will also set CurrentLCID with the LCID number and CurrentCultureInfo with the Culture Information that correspond to the attribute value set for CurrentLanguage.

About ViewApp navigation

Navigation enables users to browse for and show information obtained from the constituent elements of a running ViewApp.The simplest type of navigation uses a visible component like a tree control that shows the hierarchical organization of objects or graphics used in the ViewApp. Selecting an item from the tree launches an action that shows graphics or other types of content that appear in one or more panes of the ViewApp's layout.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



ViewApp navigation consists of three essential components:

- Navigation model

  A navigation model is the view composition engine that drives the presentation of content from a layout or ViewApp to a screen viewed by a user. A navigation model defines a hierarchy of navigation items, references to visual content used by navigation aware controls, and the actions to execute when a navigation item is selected.

- Navigation item

  A navigation item is a unique element of the navigation model that contains the necessary information to show a visual representation of an item in various forms and triggers one or more navigation actions after being selected during runtime.

- Navigation actions

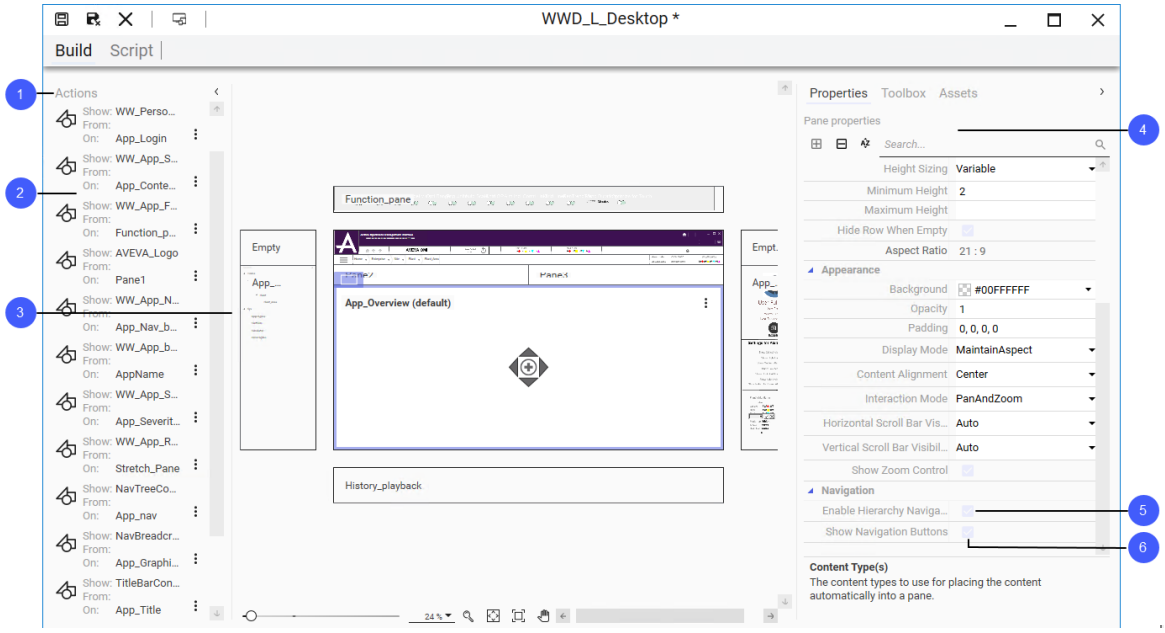
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Navigation behavior during runtime by navigation aware controls to show the information present in the navigation model.

The Layout and ViewApp editors provide integrated components and properties to configure navigation in a ViewApp. For more information about configuring navigation in each editor, see About the Layout Editor and navigation. and About the Layout Editor and navigation components.

About the Layout Editor and navigation

The following figure shows the major components and pane properties of the Layout Editor to configure navigation for a ViewApp.



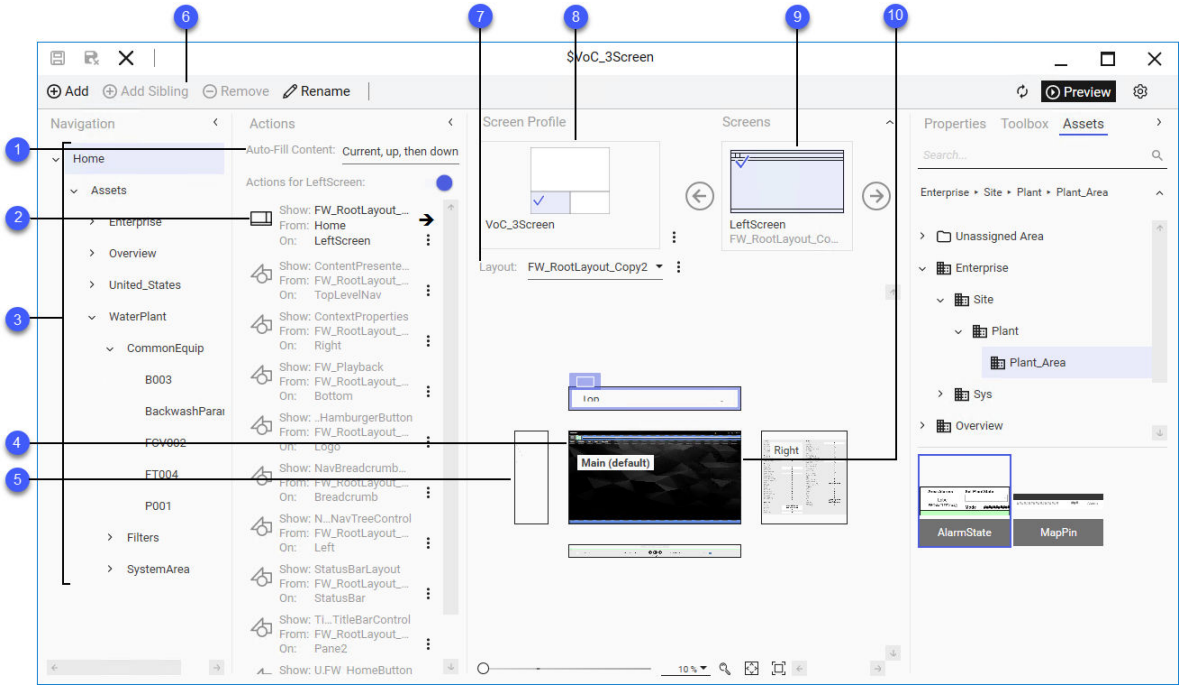
1	Actions List	Show navigation actions list associated with content added to the panes of a layout.
2	Actions	Navigation action of a piece of content added to a pane.
3	Navigation Control	Navigation tree control that shows the ViewApp's navigation model. During runtime, users can select a navigation item from the navigation control to show content.
4	Use For Auto-Fill	Although not shown in this graphic of the Layout Editor, the **Use For Auto-Fill** property determines if the pane can be filled with content automatically by Auto-Fill mode or not. Auto-Fill modes are selected from the ViewApp Editor.
5	Enable Hierarchy Navigation	Enables hierarchical navigation from the current context to show other content within a multi-content pane or neighboring navigation items by means of a vertical swipe gesture or by clicking on navigation buttons shown on the pane when **ShowNavigationButtons** property is set to true.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

6	Show Navigation Buttons	When set to true, shows graphic buttons on the pane to make a hierarchical or context navigation action while a ViewApp is running.

About the Layout Editor and navigation components

The following figure shows the major components of the ViewApp Editor to configure navigation for a ViewApp.



1	Auto-Fill Options	The **Auto-Fill Content** field includes a drop-down list of Auto-Fill options that set the possible navigation behaviors specific to the selected navigation item. For more information about Auto-Fill navigation, see About auto-fill navigation.
2	Actions List	The **Actions** field shows a list of content assigned to the panes of the layout applied to the selected screen. Each item in the **Actions** list includes a set of layout or content actions to remove or edit the pane's current content.
3	Navigation Model	The **Navigation** field shows the hierarchy of navigation items in the ViewApp. The entire set of navigation items is the navigation model, .which includes references to visual content and assigned navigation behavior when an item is selected.
4	Configuration View	Thumbnail representation of the content that appears in a pane while configuring ViewApp navigation. The icon dynamically updates as you add content to panes or assign navigation behaviors to items.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

5	Navigation Control	A Navigation tree control shows a visual representation of the ViewApp's navigation model with a set of selectable navigation items arranged in a hierarchical tree. During runtime, users can select a navigation item from the control to set the context of the ViewApp's view.
6	Command Bar	Commands to add, remove, or rename navigation items to create a custom navigation model of a ViewApp. For more information about creating custom navigation in a ViewApp, see About ViewApp custom navigation.
7	Layout	The **Layout** field shows the name of the layout applied to the selected screen in the screen profile. The **Layout** field contains a drop-down list of all layouts in the Galaxy. You can select another layout from the list to replace the current layout assigned to the screen. For more information about assigning another layout to the current screen from the ViewApp Editor, see Manage layouts in a ViewApp. The **Layout** field also includes commands to remove the current layout from the selected screen or edit the layout itself.
8	Screen Profile	The **Screen Profile** field shows an Icon view of the screens of the selected screen profile. A green screen background identifies the selected screen assigned to the ViewApp currently edited in the ViewApp Editor. The **Screen Profile** field also includes commands to remove the current screen profile from the selected screen or edit the screen profile itself.
9	Layout assignment to a screen	Visual representation of the selected layout applied to the selected screen in a screen profile. The names of the screen and layout appear beneath the icon image of the layout. For more information about assigning the initial layouts to the screens of a screen profile with the ViewApp Initialization wizard, see Create a ViewApp.
10	Content Pane	Content can be selected to appear in a pane by an assigned content type. If the assigned content type of a graphic, control, or other content matches the assigned content type of the pane, the content appears in the pane during runtime. For more information about content navigation, see About content types and navigation.

About ViewApp hierarchical navigation

When a ViewApp's navigation model is organized as a multi-level hierarchy of navigation items, users can make an action by mouse or touch to show content from another item or another piece of content in the navigation model.

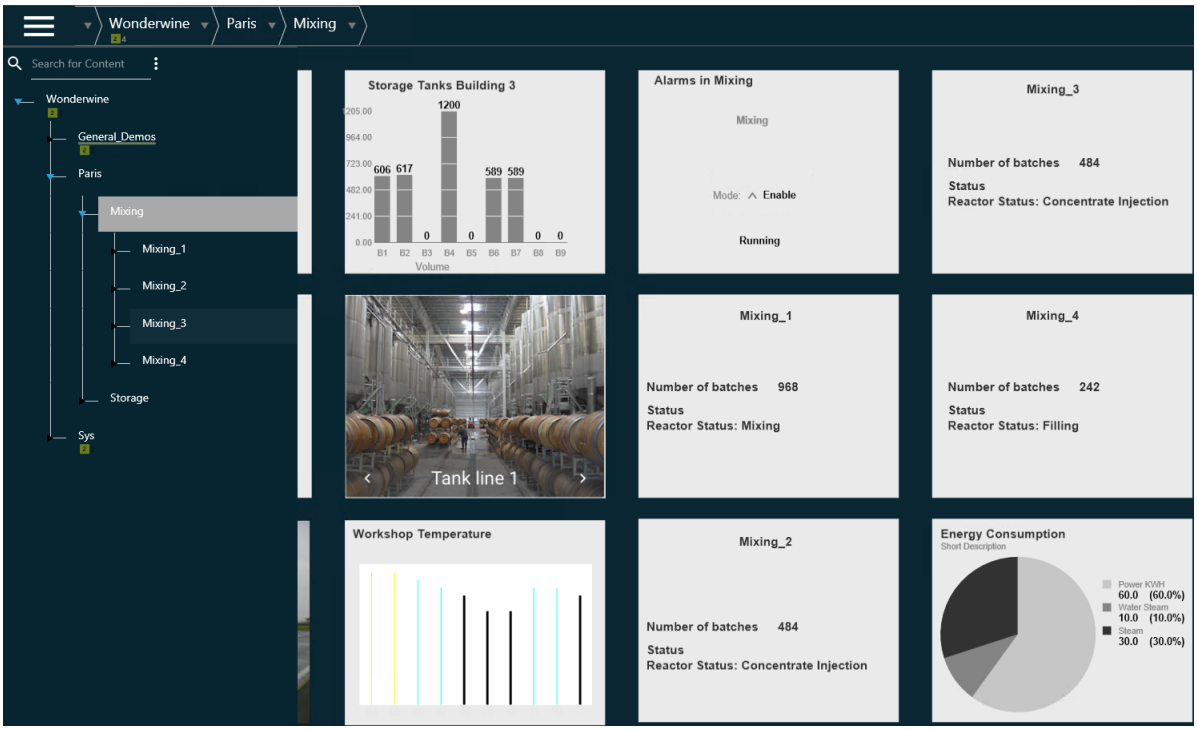
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

A layout can be configured with panes associated to each level in the hierarchy. Panes can be configured to support two types of hierarchical navigation behaviors:

- The user can make an action that navigates to the next item at the same level of the hierarchy and show content from that item.
- The user can make an action that navigates to the next piece of content within a multi-content pane.

Based on a pane property, tabs appear on the inner border of a pane that indicates hierarchical navigation is supported. When the user clicks or taps on a tab, the hierarchical navigation behavior is executed.

The following screen capture shows the behavior of hierarchical navigation. The current selected item is Mixing, which is a sibling item to Storage in the hierarchy. Users can show the content from these two sibling items by selecting the respective items shown in the navigation model. The Mixing navigation item is the parent of four child items.



The navigation breadcrumb shown above the navigation tree shows the selected Mixing item and its four child items with the vertical hierarchy represented as horizontal segments.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About ViewApp hierarchical navigation

To incorporate hierarchical navigation into a ViewApp, the following general configuration goals should be considered for your navigation model..

- Design the hierarchy of your navigation model to show similar content from similar pieces of equipment . For example, if you want to show data from a set of similar pumps, place the associated navigation items at the same sibling level in the model hierarchy. This enables the user to navigate quickly from pump to pump using hierarchical navigation.
- If you want to design a pane that shows increasingly detailed information from a single navigation item, create a stack or tabbed multi-content pane configured for hierarchical navigation. Place the content in order of detail to the different tabs or layers of the pane. Users can then drill-down through the content to see more detailed information with a simple hierarchical navigation action.

Each pane of a layout can be configured for hierarchical navigation. The following pane properties must configured within the Layout Editor to enable hierarchy navigation.

- **Enable Hierarchy Navigation**

  This pane property must be set to True to enable hierarchical navigation. The **Show Navigation Buttons** pane property is hidden unless **Enable Hierarchy Navigation** property is set to True.

- **Show Navigation Buttons**

  This property must be set to True to show navigation tabs on the pane at runtime. When enabled, users select the tabs by mouse or touch to to navigate to the next sibling navigation item in the hierarchy or other content in the same pane.

  If you want to use another visual element than the default tabs to trigger a hierarchical navigation action, set **Show Navigation Buttons** to False and create a replacement visual element that supports animation or action scripts. ViewApp Navigation attributes can be used to make hierarchical navigation actions when the user selects the visual element.. See Use navigation attributes in action scripts for a general procedure to incorporate custom hierarchical navigation in a ViewApp.

  Touch swipe gestures can be used to trigger hierarchical navigation actions. Set **Show Navigation Buttons** to False to hide the default hierarchical navigation tabs when a ViewApp is intended to be run on a touch device. On touch devices, swipe indicators appear within the pane that give users visual cues about the current swipe gestures that can be made.

Trigger hierarchical navigation actions

During runtime, users can use a mouse, a keyboard, or touch gestures to trigger hierarchical navigation actions.

Mouse

When the user clicks once on a navigation Item, the item is selected and its hierarchical commands are triggered. Clicking once on a hierarchical navigation tab within the inner border of a pane triggers the pane's navigation action.

Keyboard

Users can press keyboard up, down, left, or right arrows to move to different navigation items in the hierarchy. Pressing Enter triggers the hierarchical action associated with the selected item.

Touch

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Users tap a hierarchical navigation item to select it. If a pane is enabled for hierarchy navigation, users can swipe up to move to the previous sibling navigation item in the hierarchy. Swiping down moves to the next sibling item in the hierarchy. The pane shows swipe indicators to indicate the current valid swipe gestures within the pane and a count of how many pieces of previous sibling or next sibling content exists within the navigation item.

Layout panes can be filled with content using Auto-Fill mode. In a single layout the visible pane content can originate from different navigation items. Using hierarchy navigation in any pane uses the specific navigation item for the swiping behavior that placed that original content in the pane.

When a multi-content pane has hierarchy navigation enabled, the first content placed into the pane determines the navigation item, which may place future pieces of content into the pane. This ensures the multi-content pane only holds content coming from a single navigation item. With this restriction, an up down swiping behavior is not contingent on the content which is visible to the user, reducing confusion about the final navigation item after a swipe gesture to the next or previous sibling item.

When a multi-content pane is populated with multiple pieces of content, users can utilize content navigation to move through the multiple pieces of content in the pane. Swiping left in the pane moves to the previous sibling content. Swiping right moves to the next sibling pane content.

About auto-fill navigation

Auto-Fill is a navigation item behavior that controls how content is automatically placed in panes of a layout without manually configuring each view of a running ViewApp.

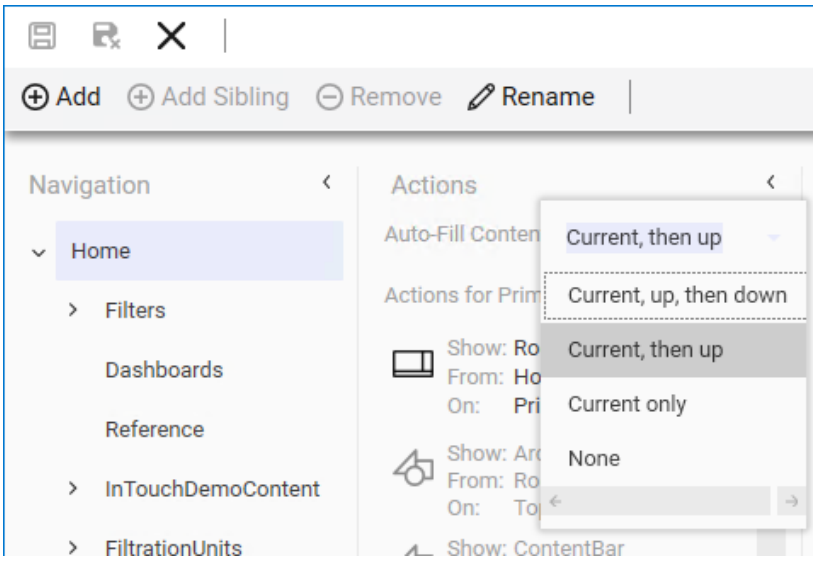
Before Auto-Fill navigation executes in a running ViewApp, appropriate layouts must be assigned to the available screens of a screen profile in a specific sequence.

- First, layouts are determined by examining the current navigation item to determine if there are any specific layouts that should be applied to screens.
- If any screens remain empty without layouts, the parent navigation item is queried to continue to search for layouts appropriate for screens.
- This process continues until the root node of the navigation hierarchy is reached, or all screens are assigned layouts.

After the appropriate layouts are determined, Auto-Fill navigation executes against the available layouts. If you link a layout to an asset and create a custom navigation item in a ViewApp with the navigation item set to the asset context, the autofill will attempt to fill the pane from the linked asset. You can override that autofill behavior by dragging and dropping content into the layout.

The ViewApp Editor contains a set of selectable Auto-Fill options to control how content is automatically placed in panes.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Note: The ViewApp's assigned layout panes must have the **Use For Auto-Fill** property enabled. For more information about layout pane properties, see Set pane properties 1.

- **None**

  In **None** mode, no content is automatically shown unless specific commands have been created to show content. You must assign all layouts for all screens and manually assign content to specific panes using drag and drop operations. Any commands in parent navigation nodes or child navigation nodes are not considered in assigning content to screens or panes. The **None** option is available only for custom navigation items.

- **Current only**

  In **Current only** mode, you must assign layouts to screens, but any navigation item content is automatically placed in the available panes using the best available pane algorithm.

- **Current, then up**

  In **Current, then up** mode, Auto-Fill searches for layouts that initially fill the screens prior to determining the content to be shown in specific panes. The current navigation item is inspected for any commands which place layouts or content full screen. For any screens that remain empty, Auto-Fill examines the parent navigation item for additional ShowLayout or ShowScreen commands to apply to empty screens.

  After content has been assigned to screens, any content residing on the current navigation item is shown to the screen using the best available pane algorithm for each individual piece of content. For any remaining empty panes, Auto-Fill looks up to the parent navigation item for additional commands or content, which apply to empty panes. This process continues until all panes are filled, or the root of the navigation hierarchy is reached. If Auto-Fill intends to replace a pane's content with the same content already in the pane, the command is skipped to avoid disturbing the existing content. In **Current, then up** mode, some panes may remain empty.

- **Current, up, then down**

  The **Current, up, then down** mode is identical to **Current, then up** mode with an additional action. If the Auto-Fill search reaches the root of the navigation hierarchy and there are still empty screens or panes, the first child of the current navigation item is inspected to see if it has content that applies to the available

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

screens or panes. This process continues down the navigation hierarchy by inspecting the first child navigation item at each level until arriving at a leaf node, then Auto-Fill stops.

About content types and navigation

Content types enable you to classify content within a ViewApp. Content assigned to a pane can be shown or hidden during runtime based on whether the assigned graphic content type matches the content type of the pane or not. This mutual association between a pane and a graphic content type is restrictive in that it limits the content that can appear in a pane to a specific subset content types. With some exceptions, a pane can only host defined content types from the available graphics that match the content type.

There are exceptions to content type matching:

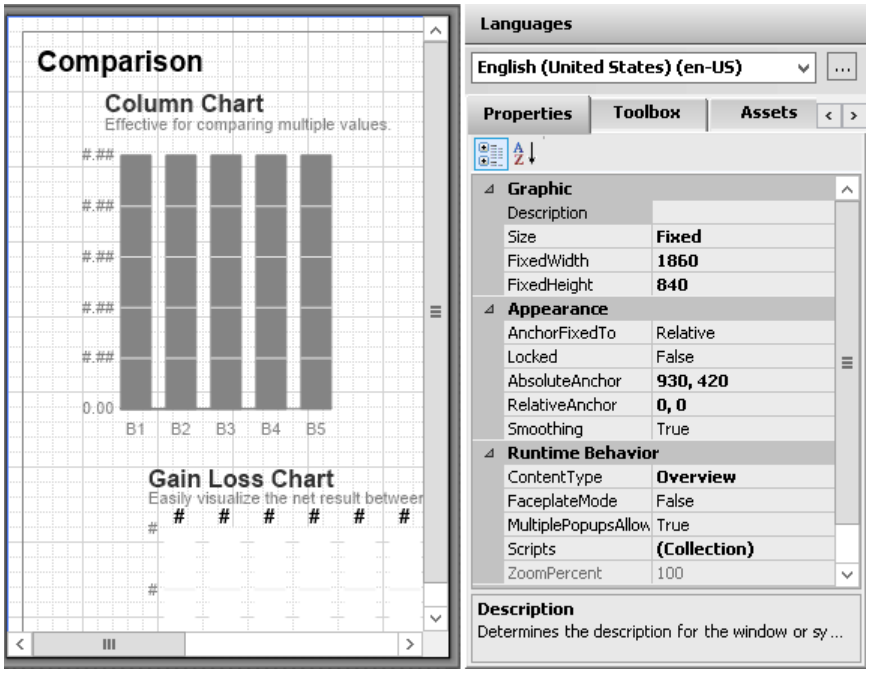
- Panes can be assigned the **Select All** content type, which matches any type of graphic content. The pane can show any type of graphic content.
- Graphics can be assigned the **All** content type, which matches any type of pane content. The graphic can be shown in any type of pane.
- Graphics and panes can be assigned multiple content types.

  The selected content types appear in a comma-delimited list in both the Industrial Graphic Editor and Layout Editor.

Assign a content type

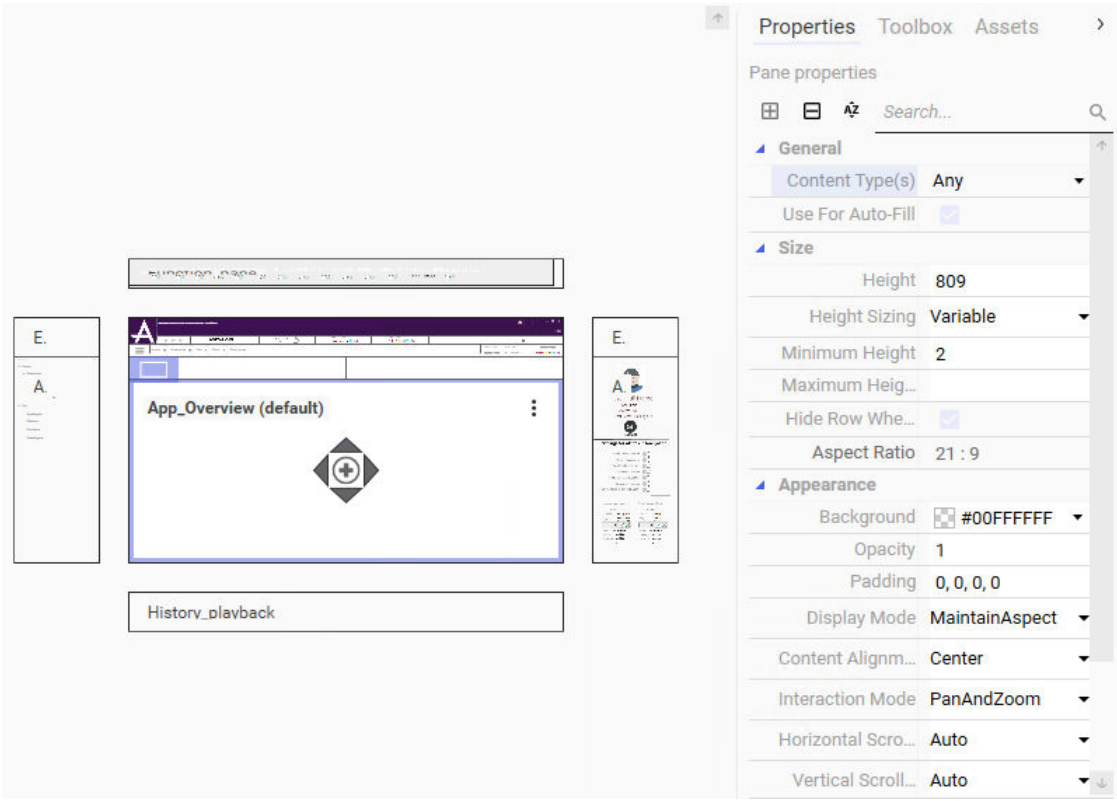
A content type can be assigned to a graphic by assigning a value to the **ContentType** property in the Industrial Graphic Editor. Similar content types can be assigned as a value of the **Content Types** pane property of the Layout Editor. Content is shown in the appropriate pane if the assigned value of the **ContentType** property matches the content type of the pane or a direct reference to the pane.

Graphics Editor ContentTypes Property



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Layout Editor Content Types Property



Action sequencing when content or other types are applied in ViewApp

Content and other types of ViewApp navigation follow a distinct sequence of steps to populate the available panes of a layout shown on a screen during runtime. When the user selects a navigation item and triggers its navigation actions, the following steps occur:

1. All screens within the ViewApp's selected screen profile are considered empty without content.
2. If the current navigation item specifies a layout for use on a screen, layouts are placed on the screens of the profile using inheritance. If any screens remain empty, the current navigation item's parent item is queried to determine if a layout is specified for any empty screen.
3. All panes belonging to the layouts assigned to screens are assumed to be empty before the composition engine assigns any content to the panes.
4. Content is shown in the appropriate pane based on content type matching between content and the pane or a direct reference to the pane.

   Layout panes containing pre-existing content are not considered in content type matching. But, a direct reference to a navigation action replaces pre-existing pane content.
5. Other panes without any assigned content are enumerated to retrieve their individual content types.
6. For each type of pane, the current level in the hierarchy is queried to see:
   a. If any explicit actions exist that specify a pane in the destination screen

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

b. If any implicit actions exist based on the type of content.

   Implicit navigation actions are triggered by an available pane that matches the content type of the content. A multi-content pane can be the destination of multiple implicit action calls while executing an Auto-Fill navigation action.

7. If any actions are found at that level of the navigation hierarchy, they are shown in the panes.
8. For any remaining empty panes, the current navigation item's parent is retrieved and the search starts again at step 3.
9. The search ends when content is found for all of the panes or the root of the of the navigation hierarchy is reached.

If there are multiple panes with the same content type, the pane chosen to show the content is selected based on the descending alphabetical order of the pane names.

About ViewApp custom navigation

The ViewApp Editor includes a set of custom navigation commands to create a navigation model for a ViewApp based on custom navigation items. Each custom navigation item can be assigned one or more pieces of content that appear in a specific pane during runtime.

Custom navigation gives users the ability to view specific types of content associated with navigation items that are not required to replicate the hierarchy of objects shown in the Model view of the Template folder. During runtime, the ViewApp's layout panes auto fill with the associated content when the user selects the custom navigation item.

Add custom navigation to a ViewApp

This topic describes the general steps to incorporate custom navigation into a ViewApp using a simple example of replacing the entire default navigation model with a set of custom navigation items.

The ViewApp Editor includes custom navigation commands shown on the menu bar.

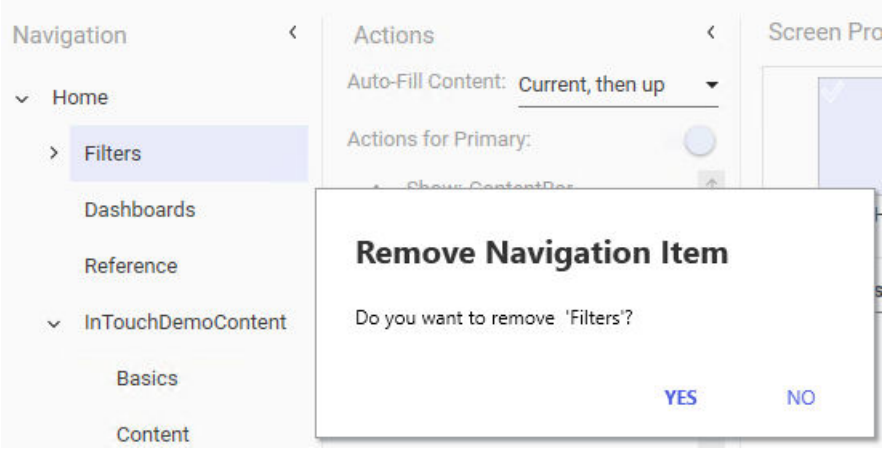
Add	Adds a custom navigation item as a child of the current selected item in the navigation model.
Add Sibling	Adds a custom navigation item as a sibling of the current selected item in the navigation model. Sibling items are at the same level within a navigation hierarchy.
Remove	Removes the selected navigation item and all child items beneath it in the navigation hierarchy.
Rename	Select to rename the current selected item. A blue background behind the current item name indicates the item can be renamed by typing over the current name.

The following procedure describes how to use these commands to create custom navigation items and associate content to them. See Configure properties for custom navigation items for a description of the steps to configure the properties of a custom navigation item.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To add custom navigation to a ViewApp

1. Create a ViewApp that includes a layout with a NavTree control that shows the hierarchy of default navigation items.
2. Select the immediate navigation item beneath the root item shown in the **Navigation** area of the ViewApp Editor.
3. Select **Remove** from the menu bar of the ViewApp Editor.
4. Select **Yes** to confirm that you want to remove navigation items.



5. Click **Add** from the menu bar to add a custom navigation item.

   The navigation item appears beneath the root as a child item. Blue text indicates the navigation item can be renamed.
6. Rename the item you added.
7. Select the custom navigation item you added.
8. Select the **Toolbox** or **Assets** tab to show their respective lists of graphics or objects.
9. Select a graphic or object that you want to associate with the custom navigation item and drag it to the pane you want to show this content when the custom navigation item is selected during runtime.

   The **Actions** list updates to show the content you associated with the custom navigation item.
10. Repeat these steps to build a complete navigation model with custom navigation items.

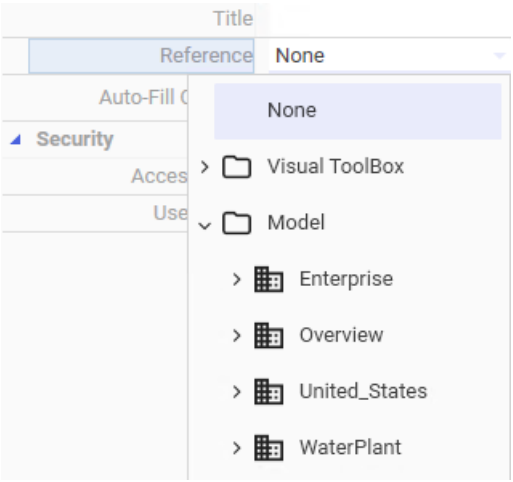
For information about configuring pane properties containing custom navigation content, see Configure properties for custom navigation items.

Configure properties for custom navigation items

After associating a piece of content to a custom navigation item, its properties need to be configured.

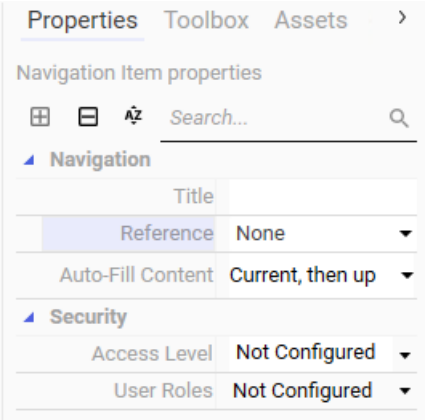
Custom Navigation Properties	Description
Title	Name assigned to the custom navigation item.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Reference	Reference node in the asset hierarchy that is the anchor point of a custom navigation item. Click at the right of the **Reference** field to show a drop-down list of assets defined for the ViewApp's navigation model.  Select an asset from the list. Any content associated with the asset you selected appears in the **Actions** list.
Auto-Fill Mode	The Auto-Fill mode assigned to the panes containing content assigned to the custom navigation item. Click at the right of the **Auto-Fill Mode** field to show a drop-down list of Auto-Fill modes. Select a mode from the list.
Access Level	Security access level assigned to content associated with the custom navigation item. Click at the right of the **Access Level** field to show a drop-down list of Access Levels for the different security roles defined in the Galaxy. Not Configured) 0 (Default) 9999 (Administrator) To view the content assigned to a custom navigation item, the runtime user role must be assigned an Access Level that meets or exceeds the the Access Level assigned to the content.
User Roles	User roles allowed to access content associated with the custom navigation item. Click at the right of the **User Roles** field to show a drop-down list of the roles defined in the Galaxy.

The following figure shows the default property values assigned to a custom navigation item.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



To edit custom navigation item properties

1. Open the ViewApp with the ViewApp Editor.
2. Select the pane containing custom navigation content.
3. Select the custom navigation item added to the the **Navigation** hierarchy.
4. Select the **Properties** grid of the ViewApp Editor.

   You should see the navigation item's properties
5. Set values to the properties.
6. Click **Save**.

ViewApp custom navigation rules

The following list describes some restrictions and behaviors when building a custom navigation model for a ViewApp.

- Add a new navigation item

  You cannot add a custom navigation item at the same level of the navigation hierarchy that contains a sibling navigation item with the same name.
- Configure a reference to an **Assets** or **Toolbox** folder

  When you configure a reference to an **Assets** or **Toolbox** folder, the name of the navigation item does not change and can be renamed if the new name does not match an existing sibling navigation item.

  You cannot add a new custom navigation item beneath a navigation item, which references an **Assets** or **Toolbox** folder. The **Add** menu option should be disabled.
- Asset hierarchy root node behavior

  You can select the root node of the asset hierarchy, as viewed from the **Template folder** of the System Platform IDE, for a reference to include asset nodes. The user can also select any child node in the asset hierarchy for the reference.
- Valid content in the **Visualization folder** hierarchy

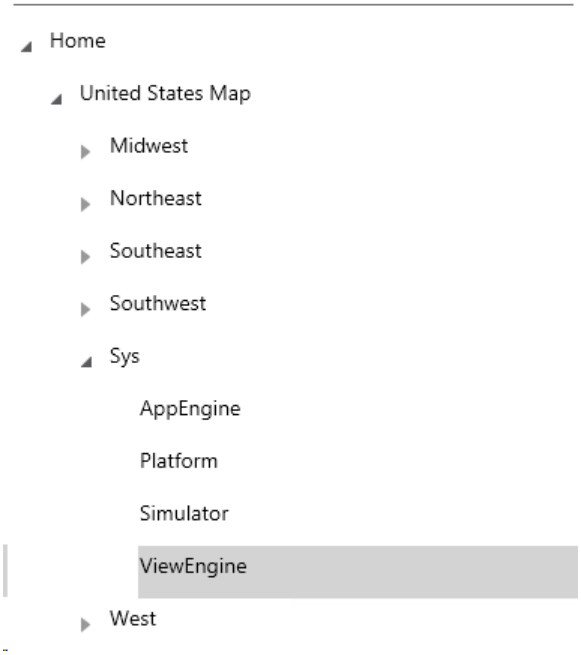
  You can select any folder in the **Visualization folder** hierarchy, including the root of the tree.
- Removing a Navigation Item

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

You can remove a navigation item from the hierarchy. All children items are removed if their parent is removed. A confirmation dialog box appears and requests confirmation that you want to remove the navigation item.

Hide assets in ViewApp navigation during runtime

System Platform includes the ability to hide assets in ViewApp navigation during runtime. The ability to hide assets from a ViewApp's navigation model is beneficial in cases where the assets themselves provide little value to an operator during runtime. An example is the case of child asset navigation items that are purely system assets in the Sys [System] folder, which appear in the navigation model of a ViewApp during runtime.



None of the system assets that appear in the Sys folder provide any meaningful content in a running ViewApp. The system assets within the Sys folder need to be hidden and not shown in a ViewApp's navigation model during runtime.

Hide assets from a ViewApp navigation model

The requirements to hide an asset from a ViewApp navigation model are listed below.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- The asset must be undeployed from the **Model** view to rename its alias name.
- The asset to be hidden must be a child navigation item in the ViewApp navigation model.
- You must change the alias name of the asset to start with an underscore.

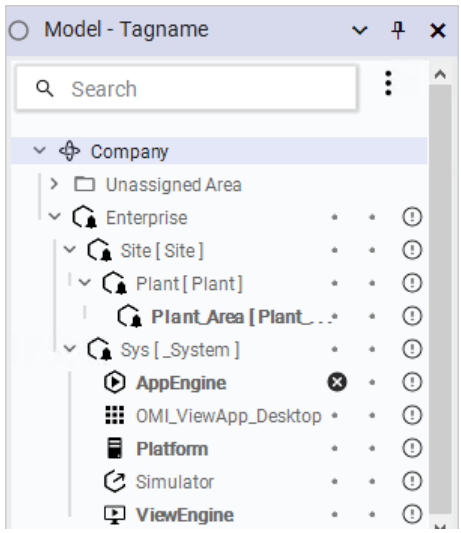
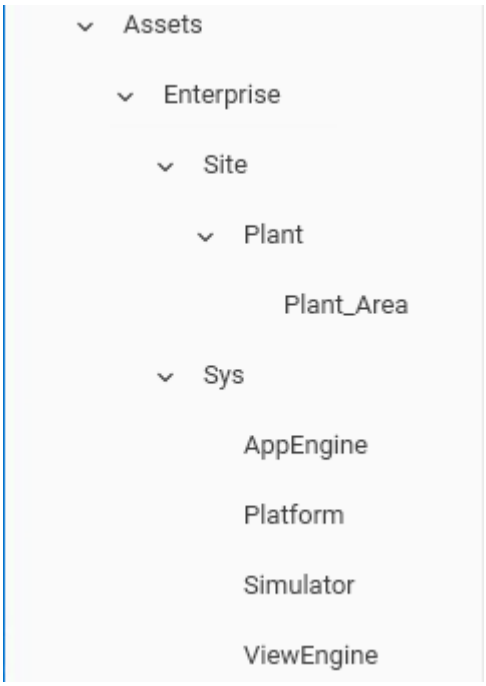
To Hide Assets from a ViewApp Navigation Model

1. Open the System Platform IDE and show the list of assets in the **Model** view at the lower left of the IDE.
2. If necessary, undeploy the child asset to be hidden.
3. Right-click the asset to show the shortcut menu.
4. Select the **Rename Alias** option from the menu.
5. Type an underscore character (_) as the first character of the asset's alias name and click **OK**.
6. Deploy the ViewApp.
7. Run the ViewApp to verify the renamed asset no longer appears in the navigation model.

   Also, the hidden asset does not appear when you run a ViewApp in preview mode.

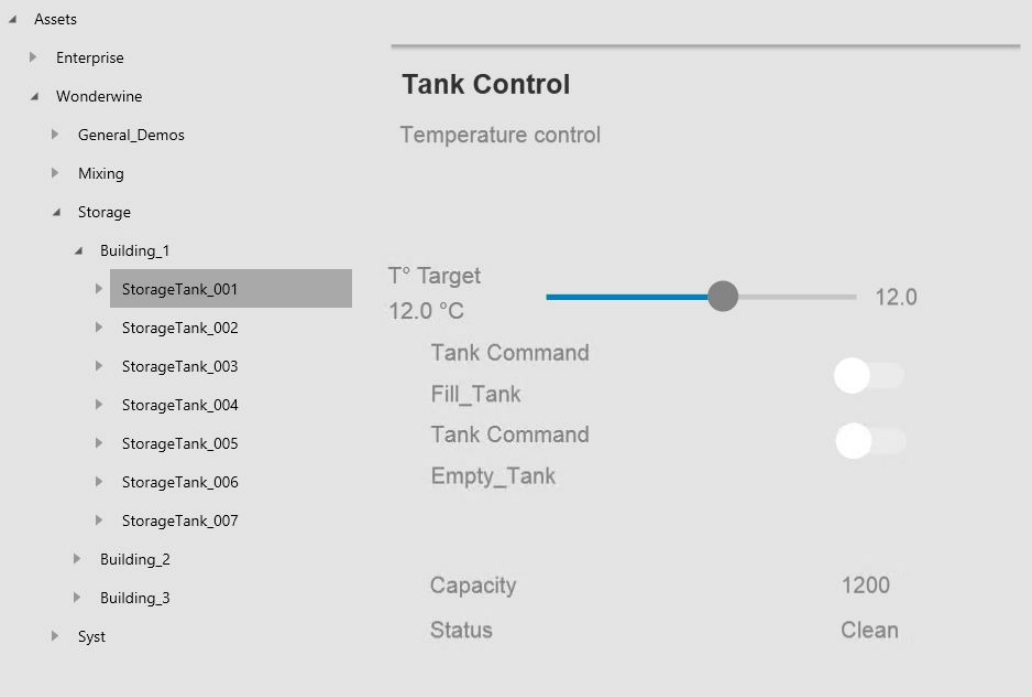
About ViewApp navigation hierarchical display

AVEVA OMI includes a NavTree app that shows a ViewApp's navigation items arranged in a hierarchical tree. During configuration time in the ViewApp Editor, the NavTree app shows this navigation item hierarchy based on the Model view of object instances. This view represents the navigation model of the ViewApp being edited.

Model View	NavTree App in ViewApp Editor
	

During runtime, users can select the chevron icon next to a navigation item to expand or collapse the tree of navigation items.When an item is selected from the tree, the item's navigation commands are executed and associated content appears in layout panes that meet the command requirements.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



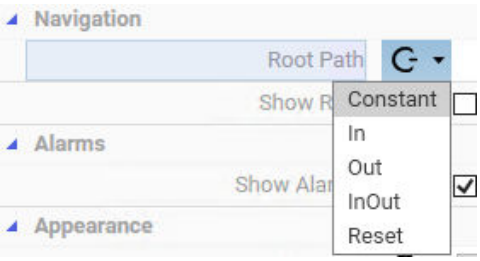
About display of controls in the ViewApp navigation hierarchy

Aside from the standard **Content** and **Pane** properties of all objects and graphics, the NavTree app includes sets of **Behavior**, **Alarms**, **Appearance**, and **Search** properties:

- Enable the **Show Root** property to show the root navigation item of the navigation model during runtime.
- Enable the **Show Alarms** property to show any active local and aggregated alarms on the individual items within the navigation model.
- Enable the **Show Search Bar** property to show a search field above the runtime Navigation tree to enter a search string. For more information about searching a ViewApp's navigation model during runtime, see *Search View App Content During Runtime* in the AVEVA OMI help.
- The **Appearance** properties determine the color, size, and font family of text shown on the navigation tree.

Each property contains a drop-down list of binding options. Some properties only provide a **Reset** option to reset a property value to its default.

In addition to a static binding, dynamic binding values can also be configured.

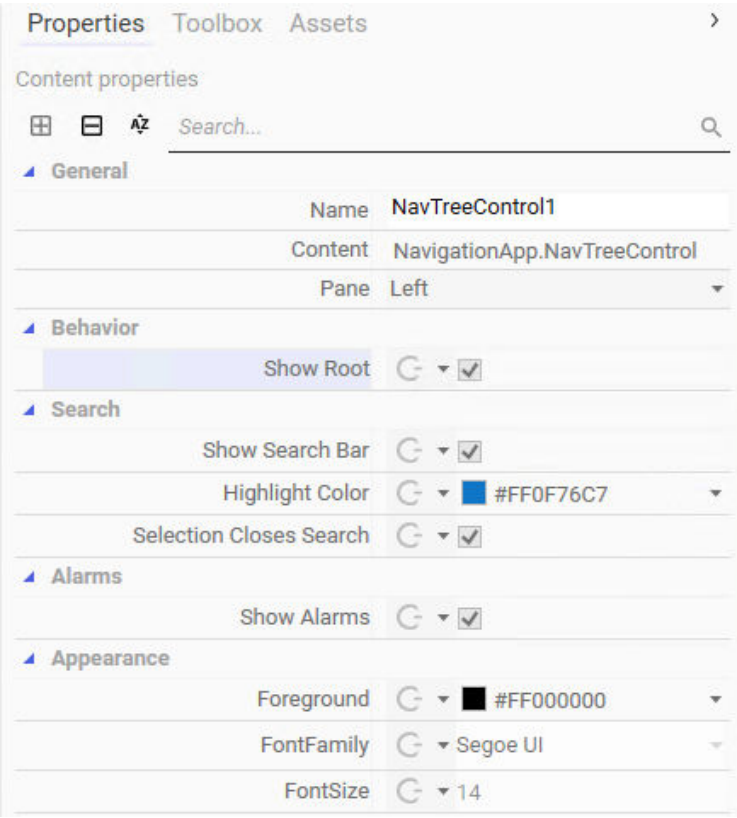


The type of control property /attribute binding can be specified by selecting a value from the drop-down list.

Constant	Static binding to the specified path. Enter a value in the data entry field.

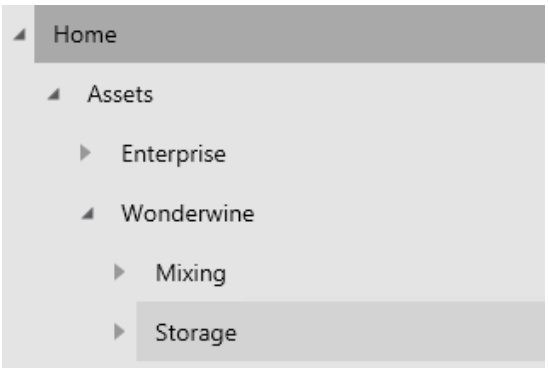
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

In	Dynamic root path with read only binding by the control to a reference.
Out	Dynamic root path with write only binding by the control to a reference.
InOut	Dynamic root path with read/write binding by the control to a reference.
Reset	Reset the current property to its default value.

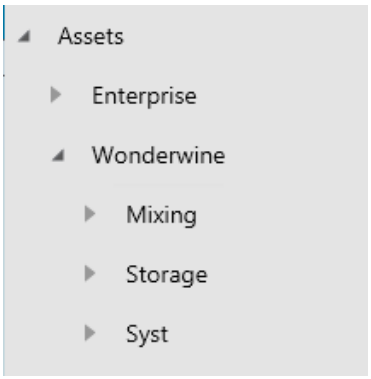


When **Show Root** is selected, the root navigation item appears at the top of the NavigationTree's hierarchical list of navigation items. The root navigation item is the default, selected startup item when a ViewApp starts.

Show Root = True



Show Root = False (Default)



The root navigation item is a custom navigation item, which supports the following common behaviors of all navigation items:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- The default name of the root navigation item is Home.
- The root navigation item can be renamed like any navigation item.
- The root navigation item can be a reference to an asset or a graphic item.
- The root navigation item can hold custom actions defining layouts and content in layouts.

The following behaviors are specific to the root navigation item:

- The root navigation item cannot be deleted.
- Sibling items to the root cannot be created.

When **Show Alarms** is selected, alarm counts from the **AlarmCntsBySeverity** attribute are displayed on any NavigationTree item with an active alarm. If the alarm has been aggregated from a contained object, alarms are shown at 50% opacity (transparency) on the containing object. If alarm is on the object, it is shown as a solid color. The alarm color and graphic reflect the severity of the alarm. See Aggregating Alarm State Information for additional information.

Set values for the color and font properties. Select the color property and then select the small triangle at the right of both property fields to show the color picker.

If you want to create event handler scripts, select the **Available Events** field to show a list of NavTreeControl events. For more information about creating event handler scripts, see Add an Event Handler Script.

About predefined navigation attributes

AVEVA OMI includes a set of predefined Navigation attributes. Read-only navigation attributes identify neighboring navigation items of the selected item by name or fully qualified path location. Read-write attributes can be used in action scripts and animations to define actions that can change the current navigation item in a running ViewApp. When the context of the ViewApp shifts from the current navigation item to a destination navigation item, the associated content of the destination item appears in the ViewApp.

A typical runtime navigation workflow based on Navigation attributes is for the user to select a graphic element shown in a ViewApp that includes an action script or animation containing Navigation attributes. The script or animation creates a navigation action based on the attributes to shift the view context to another navigation item in the ViewApp's navigation model.

Predefined navigation attribute rules

Navigation attributes are available only for use in AVEVA OMI ViewApps. Navigation attributes cannot be used in an InTouch application because the MyViewApp.Navigation prefix is regarded as a configuration error.

All Navigation attributes operate in the MyViewApp.Navigation namespace. Navigation attributes are specified with the MyViewApp.Navigation prefix in the form MyViewApp.Navigation.*attribute_name*.

Notes:

- Attribute names with *Path* return the fully qualified path of a navigation item within the navigation model.
- Attributes with the word *Title* in their names return the user friendly name set as the value of the **Title** property of a navigation item. When a navigation item's **Title** property is not configured, then the name of the navigation item is used instead.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Attribute Name	Data Type	Read/Write	Initial Value	Description
Current	String	Read/Write	None	Returns the name of the current internal name of a navigation item. Should always be assigned a value.
CurrentTitle	String	Read Only	None	**CurrentTitle** returns the name specified in the **Title** property of the current navigation item. If the navigation item's **Title** property is empty, **CurrentTitle** returns the current value.
CurrentPath	String	Read/Write	None	**CurrentPath** can be assigned the fully qualified path of a navigation item. When a value is set to **CurrentPath**, it is validated to see if the navigation item with the fully qualified path exists in the navigation model. If the navigation item exists, then the value is accepted. Otherwise, the previous value of **CurrentPath** is retained. A successful path assignment moves the navigation item to a new value and triggers Auto-Fill behaviors. Other Navigation attributes like Parent, NextSibling, PrevSibling, Current, CurrentAsset are updated.
CurrentAsset	String	Read/Write	None	**CurrentAsset** can be assigned the name of an asset if the current navigation item is an asset. When a value is set to **CurrentAsset**, it is validated to see if an asset with the given name exists in the navigation model. If an asset with given name exists, then the value is accepted. Otherwise, the previous value of **CurrentAsset** is retained. A successful asset assignment moves the navigation item to a new value and triggers Auto-Fill behaviors. Other attributes like Parent, NextSibling, PrevSibling, Current, CurrentAsset are updated.
CurrentAsset Title	String	Read Only	None	**CurrentAssetTitle** returns the user friendly name of the current asset. **CurrentAssetTitle** returns CurrentAsset if empty.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Attribute Name	Data Type	Read/Write	Initial Value	Description
CurrentAsset Path	String	Read Only	None	**CurrentAssetPath** returns the fully qualified name of the current asset path. **CurrentAssetPath** returns empty if the current asset is empty.
CurrentArea	String	Read/Write	None	**CurrentArea** can be assigned the name of the current asset's area if the current asset is valid. When **CurrentArea** is set to a value, the incoming value is validated to see if an area with the given name exists in the navigation model. If an area with given name exists, then the **CurrentArea** value is accepted. Otherwise the previous value of the **CurrentArea** is retained. A successful area assignment moves the navigation item to a new value and triggers Auto-Fill behaviors. Other Navigation attributes like Parent, NextSibling, PrevSibling, Current, CurrentAsset are updated.
CurrentArea Title	String	Read Only	Non	**CurrentAreaTitle** returns the user friendly name of the current asset's area. **CurrentAreaTitle** returns "CurrentArea" if empty.
CurrentArea Path	String	Read Only	None	**CurrentAreaPath** returns the fully qualified path of the current area. **CurrentAreaPath** returns empty if the current asset is empty.
CurrentRoot Area	String	Read Only	None	Returns the name of the current asset's Root area if the CurrentAsset is a valid asset. This is the area at the root of the Model Hierarchy (not the navigation hierarchy) which contains the CurrentAsset. This value is populated even if the Root Area node is not contained in the Navigation model.
CurrentRoot AreaTitle	String	Read Only	None	Returns the user friendly name of the current asset's Root area. If the root area is not part of the navigation model, this value will be empty.
CurrentRoot AreaPath	String	Read Only	None	Returns the fully qualified name of the current root area item using the ".Title" strings to form the path. Returns empty if

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Attribute Name	Data Type	Read/Write	Initial Value	Description
				the root area is not part of the Navigation model.
Root	String	Read Only	None	**Root** returns the internal name of the root item in the current navigation hierarchy.
RootTitle	String	Read Only	None	**RootTitle** returns the user friendly name of the root item in the current navigation hierarchy. If the root item's **Title** property is empty, **RootTitle** returns Root as its assigned value.
RootPath	String	Read Only	None	**RootPath** returns the fully qualified path of the root item using the Title strings to form the path.
Parent	String	Read Only	None	**Parent** returns the internal name of the parent navigation item in the current navigation hierarchy. If there is no parent item, **Parent** returns empty.
ParentTitle	String	Read Only	None	**ParentTitle** returns the user friendly name of the parent item in the current navigation hierarchy. If the parent item's **Title** property is empty, **ParentTitle** returns the Root value.
ParentPath	String	Read Only	None	**ParentPath** returns the fully qualified path of the parent item in the current navigation hierarchy. If there is no parent, **ParentPath** returns empty.
PrevSibling	String	Read Only	None	**PrevSibling** returns the internal name of the previous sibling item in the current navigation hierarchy. If there is no previous sibling, no value is associated with **PrevSibling**.
PrevSibling Title	String	Read Only	None	**PrevSiblingTitle** returns the friendly name of the previous sibling item in the current navigation hierarchy. If the previous sibling's **Title** property is empty, **PrevSiblingTitle** retains the current value.
PrevSibling Path	String	Read Only	None	**PrevSiblingPath** returns the fully qualified path of the previous sibling navigation item. No value is set to **PrevSiblingPath** if there is no previous sibling.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Attribute Name	Data Type	Read/Write	Initial Value	Description
PrevSibling Count	Integer	Read Only	None	Returns the number of previous sibling items remaining in the current navigation hierarchy.
NextSibling	String	Read Only	None	**NextSibling** returns the internal name of the next sibling item in the current navigation hierarchy. No value is set to **NextSibling** if there is no next sibling item.
NextSibling Title	String	Read Only	None	**NextSiblingTitle** returns the friendly name of the next sibling item in the current navigation hierarchy. If the next sibling's **Title** property is empty, **NextSiblingTitle** returns NextSibling as its assigned value.
NextSibling Path	String	Read Only	None	**NextSiblingPath** returns the fully qualified path of the next sibling navigation item. If there is no next sibling in the path, **NextSiblingPath** returns empty.
NextSibling Count	Integer	Read Only	None	Returns the number of next sibling items remaining in the current navigation hierarchy.
FirstChild	String	Read Only	None	Returns the internal name of the first child item in the current navigation hierarchy. If there is no first child, returns empty.
ChildCount	Integer	Read Only	None	Returns the number of child items in the current navigation hierarchy.
FirstChildPath	String	Read Only	None	Returns the fully qualified path of the first child, using the “Title” strings to form the path, for example, “\ \<rootname>\<nodename>”. If there is no first child, returns empty.
FirstChildTitle	String	Read Only	None	Returns the friendly name of the first child of the current navigation item. If there is no first child the string is empty.

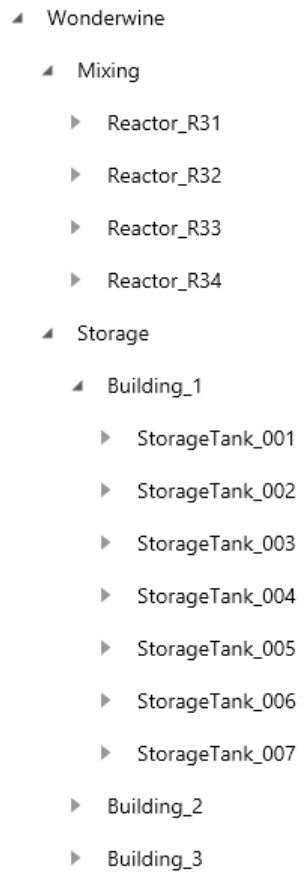
Use navigation attributes in action scripts

This example explains how to use Navigation attributes in action scripts associated with a graphic to enable users to shift the context of a ViewApp to sibling navigation items, or return to the root level item in the navigation tree.

An example ViewApp is designed to show the mixing and storage areas of a plant, which contain reactors and tanks. Users need the capability to quickly drill down within the ViewApp to see the details of individual reactors and tanks.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

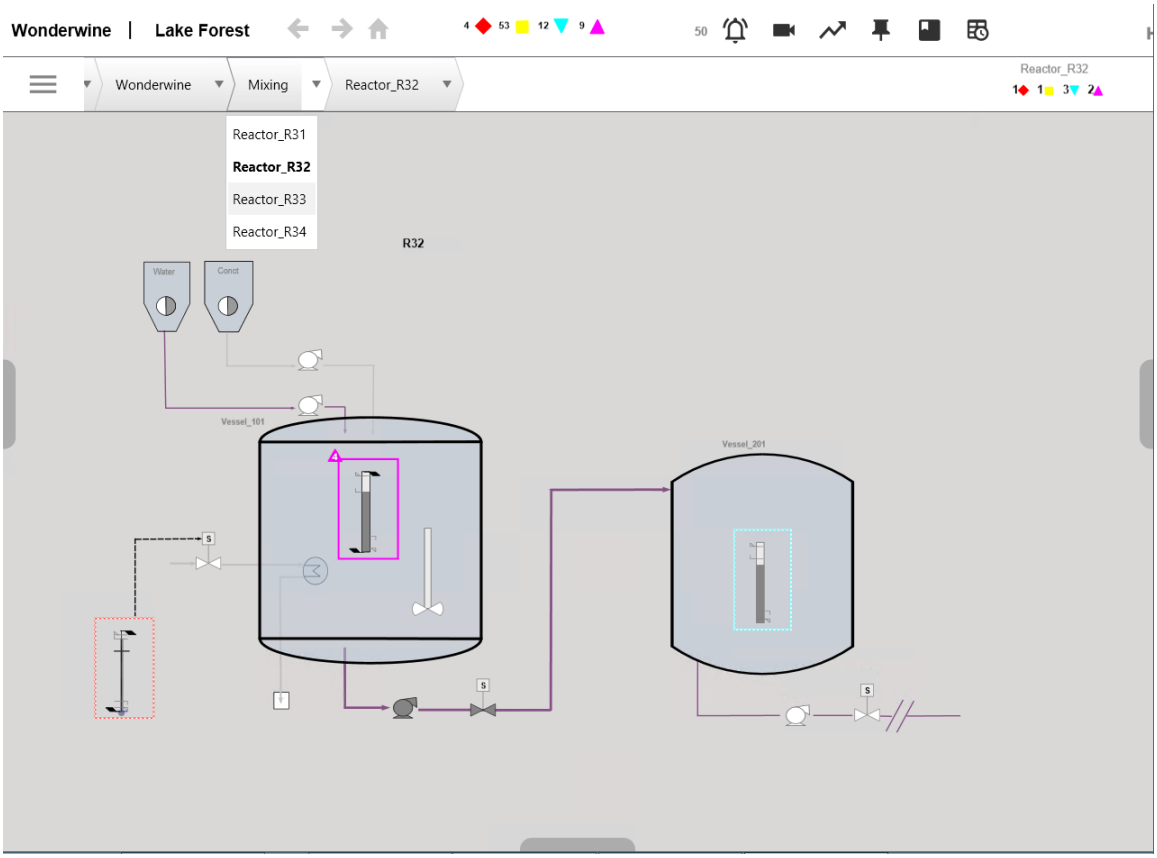
In the Model view of the IDE, the Mixing and Storage areas are sibling items in the navigation hierarchy. The Mixing area contains four reactors, which are siblings of each other. The Storage area contains three sibling buildings. Each building contains a set of storage tanks, which are siblings of each other. During runtime, the ViewApp's NavTree control shows the hierarchical relationships of the different items in the navigation model.



Users need the ability to shift quickly between sibling views. Also, users need to be able to immediately navigate to the root item in the navigation tree that shows an overview of the entire plant operation.

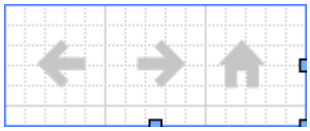
The graphic near the top of the ViewApp containing left/right arrows and a home icon is the visual element the user interacts with to navigate between sibling views and show the plant overview. The left arrow shows the previous item in a set of sibling items. The right arrow shows the next item. The home icon shows the plant overview.

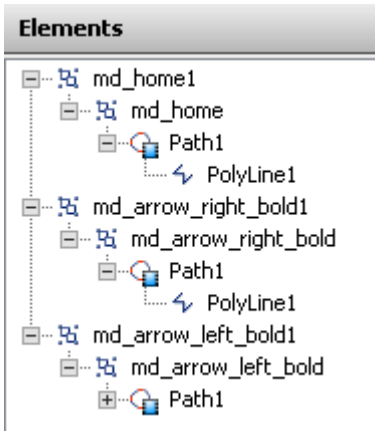
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



To set navigation pathways with Navigation attributes

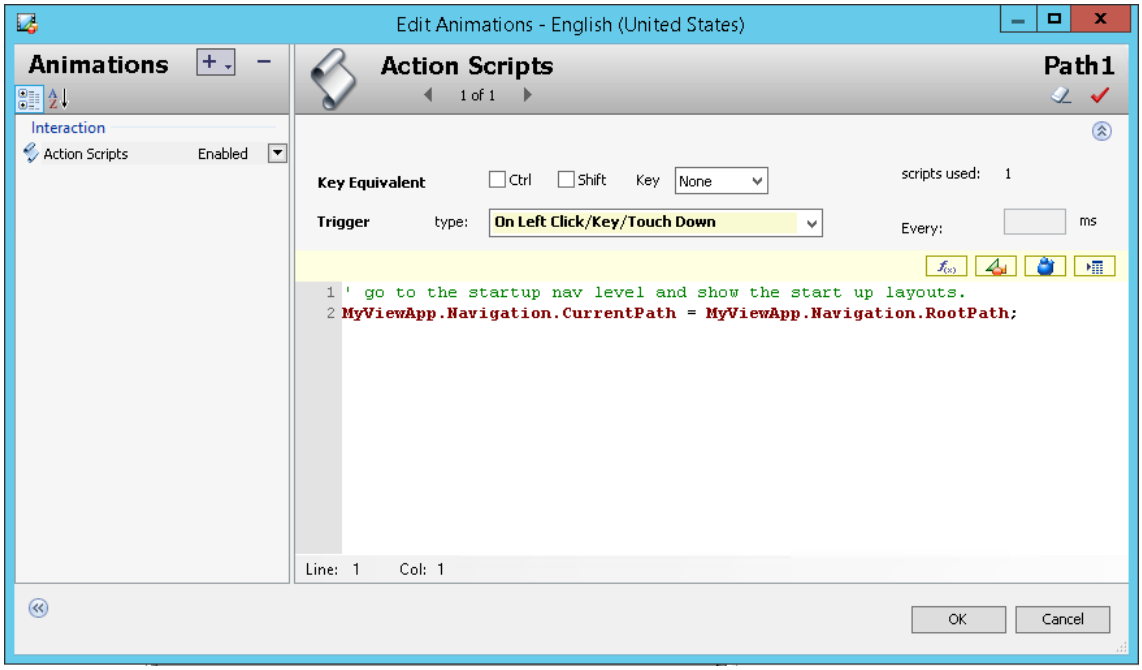
1. Create a graphic containing left/right arrows and a home icon graphic element.





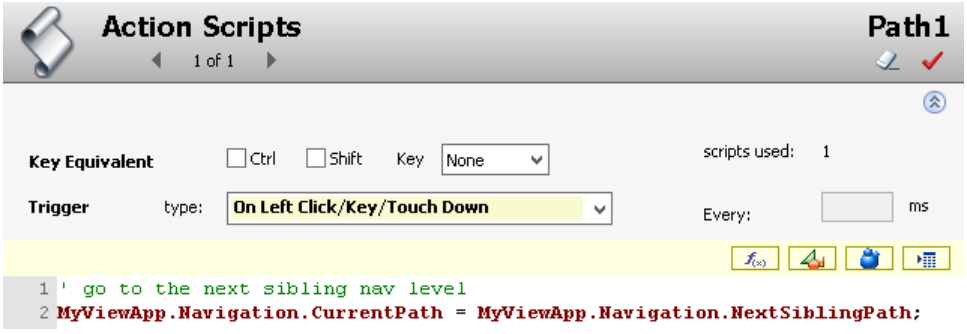
2. Select the Path element within the home graphic element to show the **Edit Animations** dialog box.
3. Select **Action Scripts** as the type of animation.
4. Enter a Navigation attribute expression to set the current navigation path to the root path.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

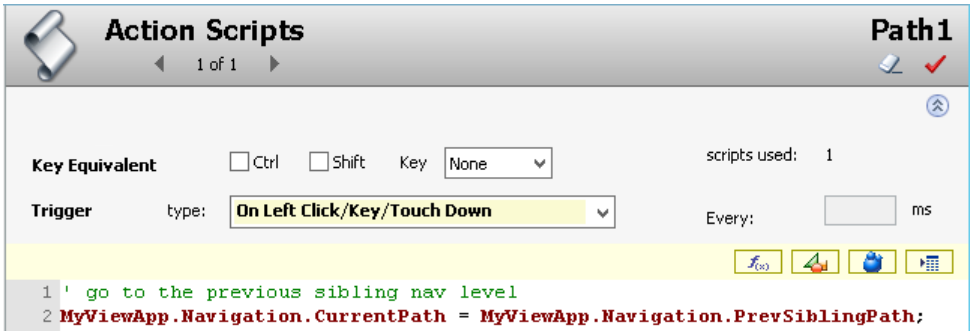


5. Repeat steps 2-4 to set actions scripts for the left/right arrow graphic elements.

Right arrow



Left arrow



6. Save your changes to the graphic.

About client controls embedded into a graphic

You can:

- Import and embed client controls into a graphic.

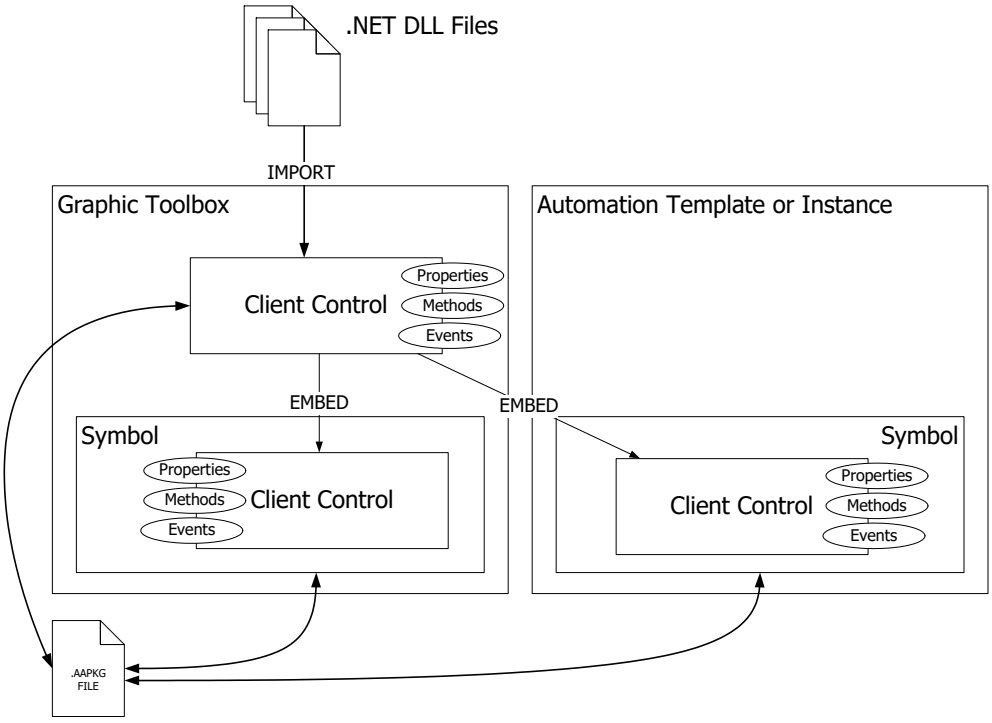
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- View and edit the properties of the client control.
- Bind the properties of the client control with attributes and element references.
- Configure scripts for client control.
- Animate client controls.
- Export a client control.
- Configure a client control with security.
- Ensure that dynamically loaded assemblies are included with the primary client control assembly when an application is deployed
- View additional client control information such as the files the client control uses and what objects and graphics are using the client control.

Client controls give you functionality contained in .NET controls you can use in graphics. To use this functionality, you must:

- Import the .DLL file that contains one or more client controls. The client control is imported into the Visualization folder.
- Browse and embed one or more of the client controls into a new or existing graphic. The client controls appear as elements.
- View and edit the exposed client control properties.
- Bind the client control properties to attributes, graphic custom properties or InTouch tags. Do this using data binding animation.
- Configure scripts for client control. Do this using the animation.

You can then use the graphic containing the embedded client control in a ViewApp.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Import client controls: options

You can import client controls into the Industrial Graphic Editor from .NET Dynamic Link Library (.DLL) files.

After importing client controls, you can organize them in the Industrial Graphic Editor as you would with Industrial Graphics. For more information, see Organizing Client Controls.

You can also import client controls that have previously been exported in an ArchestrA package (.aaPKG) file.

If you import a newer version of a client control that you are already using in your HMI software as embedded Industrial graphic, you need to restart the HMI software.

Import client controls

You can import one or more client controls from .DLL files. The client controls can be contained in one single .DLL file or span multiple files.

To import a client control, you must have security permissions to import graphic objects.

Note: If you select .NET .DLL files that do not contain client controls, the import process ignores these and continues at the next .DLL file.

To import client controls

1. On the IDE **Home** ribbon, select **Galaxy**, then select Import.
2. Select **Visualization**, then **Client Controls**. The **Import Client Control(s)** dialog box appears.
3. Select one or more .NET .DLL files that contain the client controls you want to import and click **Open**. The **Import Preferences** dialog box appears.
4. Select the appropriate option for the import and click **Import**. The **Import Client Control(s)** dialog box appears.
5. When the client controls are imported, click **Close.** The imported client controls appear in the Visualization folder.

Note: If the import fails, a message in the **Import Client Control(s)** dialog box indicates the error.

Example of installing the TagPicker control

To install the TagPicker client control

1. On the IDE **Home** ribbon, select **Galaxy**, then select Import.
2. Select **Visualization**, then **Client Controls**. The **Import Client Control(s)** dialog box appears.
3. Browse to the C:\Program Files (x86)\ArchestrA\Framework\Bin\ViewAppFramework\SharedLibs folder, select the aaHistClientTagPicker.dll file and click **Open**. The **Import Preferences** dialog box appears.
4. Select the appropriate option for the import and click **Import**.
5. When the import is complete, click **Close**.
6. Open the **Visualization folder** and expand the Galaxy node. aaTagPicker is listed as a client control.

Import previously exported client controls

You can import one or more previously exported client controls from an ArchestrA package file (.aaPKG). Previously the client controls may have been:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Exported without a graphic or an object instance or template.
- Embedded in a graphic and the graphic was exported.
- Embedded in a graphic and contained in an object instance or template and the object was exported.

To import a previously exported ArchestrA package containing one or more client controls

- Import the exported client controls the same way as you would import an AutomationObject (.aaPKG). For more information, see the *Application Server User's Guide*.

Organizing Client Controls

You can organize the client controls within the Visualization folder the same way as you would with Industrial Graphics. You can:

- Rename client controls.
- Move client controls in and out of folders.
- Delete client controls.

For more information, see Organizing Graphics.

Viewing and Changing the Properties of Client Controls

When you embed a client control into a graphic, the native properties of the client control are imported into the Properties Editor in the **Misc** group.

Also the element container of the client control has properties such as:

- Name
- X, Y, Width, Height, AbsoluteOrigin, RelativeOrigin, and Locked
- FillColor
- TextColor and Font
- Enabled, TabOrder, TabStop, and Visible

The element container properties override the native properties of the client control.

You can view and change the properties of the control in the Properties Editor.

To view or change the properties of a client control

1. Select the embedded client control on the canvas.
2. In the Properties Editor, locate a:
   - Container property in the property categories **Graphic, Appearance**, **Fill Style**, **Text Style** or **Runtime Behavior**.
   - Native property in the **Misc** property category.
3. View or change the located property.

To reset a client control back to its original size

- On the **Edit** menu, click **Control - Original Size**. The AutoSize property is set to False.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Example of changing a property of the TagPicker control

Install and embed the TagPicker Control into a graphic as described in:

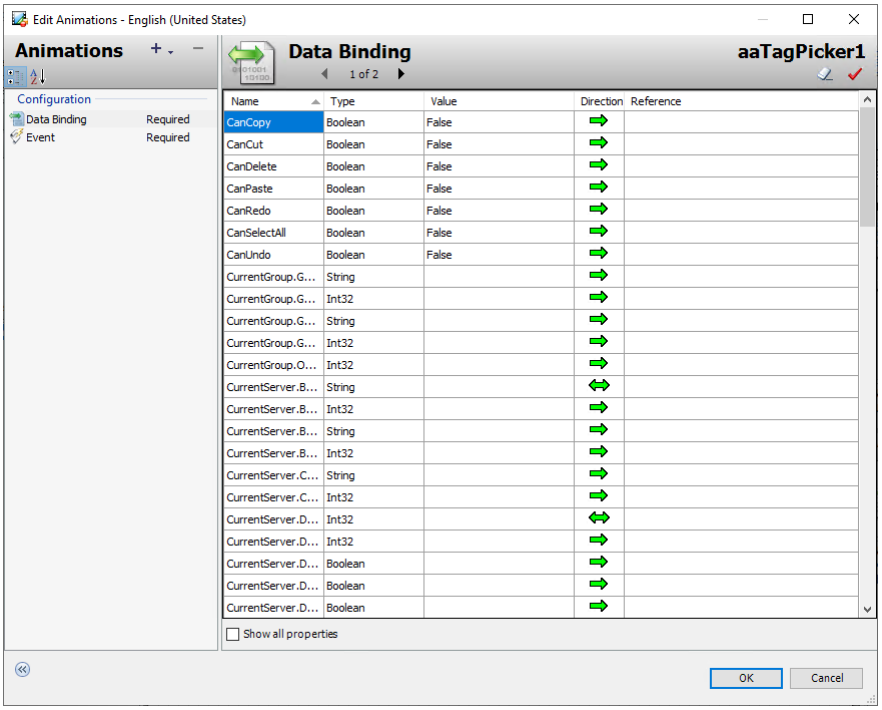
- Example of installing the TagPicker control.
- Embed the TagPicker client control.

In this example the "Tag Picker" caption of the TagPicker control is hidden.

1. Select the embedded TagPicker client control. The Properties Editor shows all properties of the client control.
2. In the **Misc** property category, locate the property **HideCaption**.
3. Assign the value True to it and click **Enter**. The caption "Tag Picker" of the TagPicker client control is hidden.

Binding Client Control Properties to Attributes or Element References

You can bind the properties of an embedded client control to attributes or element references. This lets you use attributes and element references as source and consumer of data for the client control properties. You do this with the Data Binding animation.



The Data Binding table contains the following information:

- **Name** - name of the client control property
- **Type** - the .NET data type of the property
- **Value** - the default value of the client control property
- **Direction** - indicates if the property is read/write or just read-only

read/write property

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

read-only property

write-only property

- **Reference** - the attribute or element reference the property is bound to

You cannot remove the **Data Binding** animation.

To bind a client control property with an attribute or element reference

1. Double-click the embedded client control on the canvas. The **Edit Animations** dialog box appears and the Data Binding animation is selected by default.
2. Locate the client control property that you want to bind with an attribute or element reference.
3. Double-click the **Reference** box.
4. Do one of the following:
   - Type an attribute or element reference.
   - Browse for an attribute or element reference by clicking the **Browse** button.
5. Repeat above for any other properties you want to bind.
6. Click **OK**.

Example of data binding in the TagPicker control

Install and embed the TagPicker control into a graphic as described in:

- Example of installing the TagPicker control.
- Embed the TagPicker client control.

In this example, the Boolean graphic custom property **HCV** controls the visibility of the ActiveFactory TagPicker caption.

Do the following:

1. Create a Boolean custom property and rename it **HCV**.
2. In the Industrial Graphic Editor, double-click the embedded ActiveFactory TagPicker control.
3. From the list of properties in the **Data Binding** configuration area, locate the **HideCaption** property.
4. Double-click the **Reference** box of the **HideCaption** property.
5. In your HMI's attribute/tag browser, select the **HCV** custom property and click **OK**.

   The HideCaption property is now assigned to the element reference HCV.
6. Click **OK**.
7. Place a button on the canvas and configure it with a Boolean pushbutton animation that toggles the custom property HCV.
8. Save and close the graphic.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

9. Embed the graphic in a managed InTouch application and test the data binding by clicking on the button in WindowViewer. When you do so, the visibility of the caption of the TagPicker control is toggled.

Configuring Client Control Event Scripts

You can configure a script that is executed when a client control event occurs. You do this using the **Event** animation.

To configure a script for a client control event

1. Double-click the embedded client control on the canvas. The **Edit Animations** dialog box appears.
2. In the animation list, click **Event**. The right panel shows the configuration.
3. In the **Event** list, select the event for which you want to execute a script. The **Parameters** list shows for the selected event:
   - **Type**: the data type of each parameter.
   - **Name**: the name of each parameter.
4. In the script area, write the event script.
5. If you want to insert an event parameter in your script, click the **Select Event Parameter** icon. Select the parameter. The parameter name is inserted into the script at the cursor position.
6. If you want to configure scripts for other, select the event from the **Event** list. The script area is cleared and you can write the script for the newly selected event.
7. When you are done, save and close.

Example of configuring an event script for the TagPicker control

Install and embed the TagPicker control into a graphic as described in:

- Example of installing the TagPicker control.
- Embed the TagPicker client control.

In this example, when one of the tags is picked by double-clicking on it, a message is logged in the Log Viewer.

First however, you need to:

- Import the script function library from the file aaHistClientDatabase.dll.
- Configure a connection to a valid and running Historian Server.

To import the script functions from aaHistClientDatabase.dll

1. On the Home ribbon, select **Galaxy**, then select **Import.**
2. Select **Visualization**, then **Client Controls**. The **Import Client Controls dialog box** appears.
3. Browse to the aaHistClientDatabase.dll and select it. By default, this file is in the C:\Program Files (x86)\ArchestrA\Framework\Bin\ViewAppFramework\SharedLibsfolder.
4. Click **Open**. The import starts and finishes with a message.
5. Click **OK.**

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To connect the TagPicker control to the Historian Server

1. On the canvas, place a button next to the TagPicker control.
2. Double-click the button. The **Edit Animations** dialog box appears.
3. Add an **Action Script** animation to the animation list.
4. In the script area, type the following script:

```
Dim NewServer as ArchestrA.HistClient.Database.aaServer;
Dim statusMessage as String;
NewServer  = aaTagPicker1.Servers.Add("MyHistorian");
NewServer.LoginID = "MyUserName";
NewServer.Password = "MyPassword";
NewServer.LogOn( statusMessage );
LogMessage ("Connection" + statusMessage);
```

   In the script, replace the strings MyHistorian, MyUserName and MyPassword with the Historian server name, a valid user name, and a password to connect to the server.

5. Close the **Edit Animations** dialog box.

You can now configure the client control event to log a message every time the user picks one or more tags by double-clicking on them:

1. In the Industrial Graphic Editor, double-click the embedded ActiveFactory TagPicker control.
2. In the animation list, click **Event**.
3. In the **Event** list, click the **OnTagsPicked** event.
4. In the script area, type the following:

```
LogMessage("User picked one or more tags.");
```

5. Save and close the **Edit Animations** dialog box.
6. Save and close the Industrial Graphic Editor.
7. Embed the graphic in a managed InTouch application.
8. Switch to runtime and connect to a valid IndustrialSQL Server source.
9. Double-click on one of the tags in the TagPicker control.
10. Check the SMC Log Viewer. The message "User picked one or more tags" appears.

Animating Client Controls

Every client control has these animation types:

- Data binding animations determine which attributes and element references can read and write to the client control.
- Event animations assign scripts to individual client control.

You can add the following animations that correspond to the supported client control container properties.

- Visibility
- Fill Style

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Text Style
- Location Horizontal
- Location Vertical
- Width
- Height
- Tooltip
- Disable

If you configure these animations, the resulting behavior and appearance overrides the behavior and appearance given by the native properties of the client control.

To add animation to embedded client controls

1. Double-click the embedded client control on the canvas. The **Edit Animations** dialog box appears.
2. Add animations as you would with any other element.

Export client controls

You can export client controls as ArchestrA package files (.aaPKG). You can export them:

- Directly from the Industrial Graphic Editor.
- Indirectly when you export graphics that reference them.

You can import the client controls again from exported .aaPKG files.

To export client controls directly as ArchestrA package files

1. In the **Visualization folder**, select one or more client controls that you want to export.
2. On the **Home** ribbon, in the **Export** area,select **Selected**, then select **As package**.
3. Follow the general procedure for exporting AutomationObjects. For more information, see the *Application Server User's Guide*.

Secure client controls

The client controls use the same security setting as the graphics. You can set the security for client controls and graphics in the **Security** panel of the IDE. For more information, see Configure security for graphics.

Including Dynamically Loaded Assemblies with the Client Control

When the primary client control assembly is imported into the HMI/SCADA project or application during an application's deployment, the system identifies all statically-linked dependent assemblies and imports them into the HMI/SCADA application as well. However, if the client control contains dynamically loaded assemblies, these assemblies are not automatically loaded in the HMI/SCADA application or project.

There are two methods for ensuring that the client control's dynamically loaded assemblies are included in the galaxy when the primary assembly is imported:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- By including the list of dynamically loaded assemblies in an XML manifest resource that is embedded in the primary assembly. The advantage of this method is that the required configuration information is packaged with the assembly, so no any other packing mechanism is required.
- By including the list of dynamically loaded assemblies in an external XML configuration file that is stored in the same directory as the primary assembly.

Both methods can be used simultaneously to provide redundancy, in the event that one of the dynamically loaded assembly lists is missing a required assembly.

Requirements for Both Inclusion Methods

- Store all dynamically loaded assemblies, as well as all non-system static dependencies of these dynamically loaded assemblies, in the same directory as the primary assembly.
- If a dynamically loaded assembly is loading another assembly dynamically, then include the other assembly as a dynamically loaded assembly of the primary assembly. This is a requirement because the system will not search recursively for static or dynamic dependencies.

Sample XML for a Dynamically Loaded Assembly List

A sample list of dynamically loaded assemblies in XML format is shown below. The XML list format is the same for an embedded manifest resource or an external configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
 <DependentFiles>
  <DependentFile>
   <FileName>DynDepAsm1.dll</FileName>
  </DependentFile>
  <DependentFile>
   <FileName>DynDepAsm2.dll</FileName>
  </DependentFile>
  <DependentFile>
   <FileName>DynDepAsm3.dll</FileName>
  </DependentFile>
 </DependentFiles>
</Root>
```

XML Schema for the Dynamically Loaded Assembly List

The following XML schema is applicable for the dynamically loaded assembly XML list whether the list is provided as an embedded manifest resource or an external configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="Root">
      <xs:complexType>
         <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="DependentFiles">
               <xs:complexType>
                  <xs:sequence>
                    <xs:element minOccurs="1" maxOccurs="unbounded" name="DependentFile">
                      <xs:complexType>
                        <xs:sequence>
```

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

```
                        <xs:element minOccurs="1" maxOccurs="1" name="FileName"
                type="xs:string"/>
                      </xs:sequence>
                    </xs:complexType>
                  </xs:element>
              </xs:sequence>
            </xs:complexType>
         </xs:element>
      </xs:sequence>
         </xs:complexType>
      </xs:element>
</xs:schema>
```

Embedding the XML Manifest Resource in the Primary Assembly

To embed the XML manifest resource in the primary assembly

1. In Visual Studio, add the XML list file to the client control assembly project. Any file name can be used, with the extension .aaCFG.
2. Change the **Build Action** property value to Embedded Resource.

After compilation, the XML will be available in the assembly as an embedded manifest resource file.

During the client control import operation, the system will read any embedded XML manifest resources with the extension .aaCFG. The system will then import any listed assemblies that are stored in the same location as the primary assembly.

Including the XML Manifest Resource in an External Configuration File

To include the XML manifest resource in an external configuration file

1. Create the XML list file using the same root name as the primary assembly but with the extension .aaCFG. For example, if the primary assembly name is MyControl.dll, then the configuration file name would be MyControl.aaCFG.
2. Store the file in the same directory as the primary assembly.

During the client control import operation, the system will look for a file that has the same root name as the primary assembly but with the extension .aaCFG and in the directory in which the primary assembly is stored. If this file is found and an embedded XML manifest resource exists, the system will consolidate the two lists to eliminate duplicate entries. The system will then import any listed assemblies that are stored in the same location as the primary assembly.

Preventing Dynamically Loaded Assembly Import Issues

Refer to the following guidelines to prevent issues with importing the dynamically loaded assemblies.

- Verify that the XML is valid. Invalid XML in the embedded manifest resource or the configuration file will result in the entire client control import operation for the selected primary assembly to be canceled.
- Include all assemblies on which the dynamically loaded assemblies are directly or indirectly dependent, in the same directory as the primary assembly and included in the XML list. If the system is unable to locate and load any of the direct or indirect dependencies, the entire client control import operation for the selected primary assembly will fail.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- If a dynamically loaded assembly is going to load another assembly dynamically, check that the other assembly is included in the XML list. If any such assemblies are not included in the primary assembly's manifest resource or configuration file, the import operation will succeed. However, these indirectly loaded assemblies will not be imported, which can result in the client control not behaving correctly during execution.

View additional client control information

You can view:

- Which .DLL files, or assemblies, are used for the client control.
- The class name, vendor, and version.
- Which Industrial Graphics use the client control.

This information is contained in the Client Control Properties panels.

The client control properties are different than the properties of the embedded client control. The client control properties can be viewed in the IDE directly. The properties of the embedded client control can be viewed in the Properties Editor of the Industrial Graphic Editor.

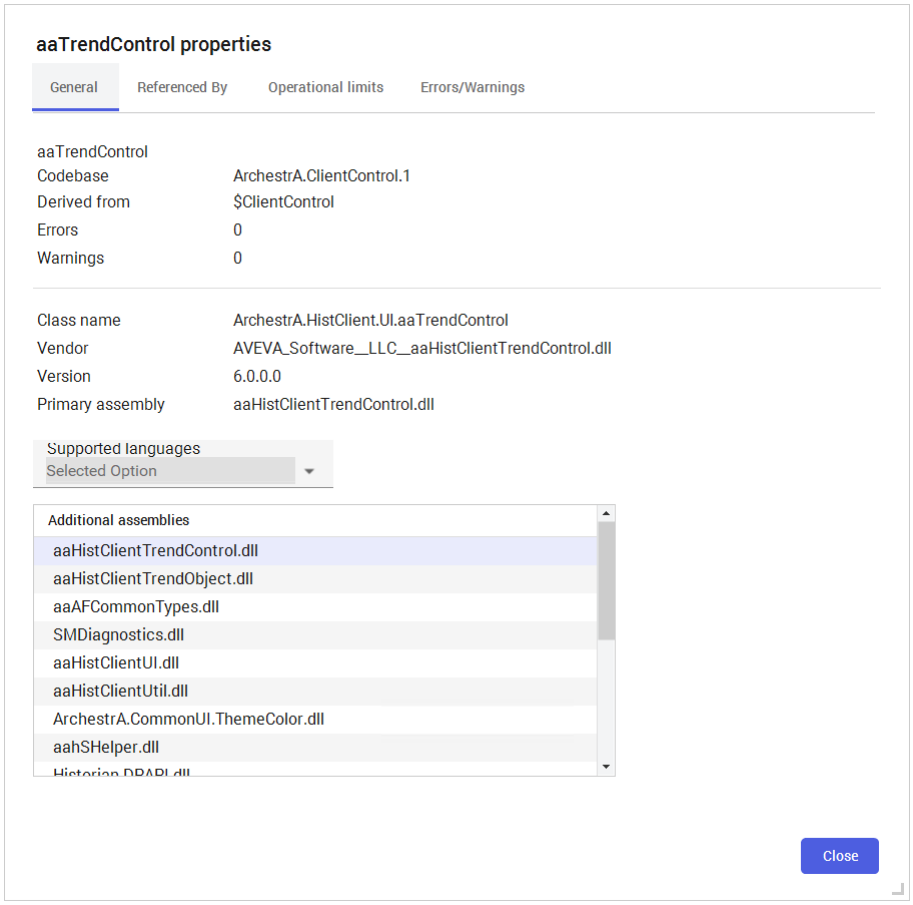
View client control assemblies

You can view which Client Control .DLL files, or assemblies, are used for the client control.

To view the client control assemblies

1. In the IDE **Visualization** folder, select the client control.
2. On the **Home** ribbon, in the **Edit** area, select **Properties**. The **Properties** dialog box appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



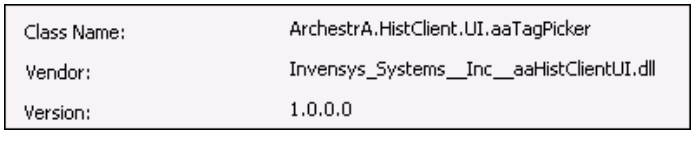
3. On the **General** tab, you can view:
   - The **Primary Assembly**, which is the main .DLL file.
   - **Additional Assemblies**, which are linked to the main .DLL file and automatically loaded.

View class name, vendor, and version of a client control

You can view the class name, vendor, and version of a client control in its **Properties** panel.

To view the class name, vendor, and version of a client control

1. In the IDE **Visualization** folder, select the client control.
2. On the **Home** ribbon, in the **Edit** area, select **Properties**. The **Properties** dialog box appears.
3. Click the **General** tab.



View objects and graphics referencing client controls

You can view which Industrial Graphics are using a given client control. This can be viewed in the **Properties** dialog box of the client control.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To view objects and graphics referencing a client control

1. In the IDE **Visualization** folder, select the client control.
2. On the **Home** ribbon, in the **Edit** area, select **Properties**. The **Properties** dialog box appears.
3. Click **Referenced By**. The list of objects and graphics using the client control is shown.

Ensure dynamically loaded assemblies for client controls are imported

When the primary client control assembly is imported into the galaxy during an application's deployment, all statically-linked dependent assemblies are identified and imported into the galaxy as well. However, if the client control contains dynamically loaded assemblies, these assemblies are not automatically loaded in the galaxy.

There are two methods for ensuring that the client control's dynamically loaded assemblies are included in the galaxy when the primary assembly is imported:

- By including the list of dynamically loaded assemblies in an XML manifest resource that is embedded in the primary assembly. The advantage of this method is that the required configuration information is packaged with the assembly, so no any other packing mechanism is required.
- By including the list of dynamically loaded assemblies in an external XML configuration file that is stored in the same directory as the primary assembly.

Note: Both methods can be used simultaneously to provide redundancy, in the event that one of the dynamically loaded assembly lists is missing a required assembly.

Dynamically loaded assemblies rules

- All dynamically loaded assemblies, as well as all non-system static dependencies of these dynamically loaded assemblies, must be stored in the same directory as the primary assembly.
- If a dynamically loaded assembly is loading another assembly dynamically, then the other assembly must be included as a dynamically loaded assembly of the primary assembly. This is a requirement because the system will not search recursively for static or dynamic dependencies.

XML sample for listing the file location of dynamically loaded libraries

A sample list of dynamically loaded assemblies in XML format is shown below. The XML list format is the same for an embedded manifest resource or an external configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
   <DependentFiles>
      <DependentFile>
         <FileName>DynDepAsm1.dll</FileName>
      </DependentFile>
      <DependentFile>
         <FileName>DynDepAsm2.dll</FileName>
      </DependentFile>
      <DependentFile>
         <FileName>DynDepAsm3.dll</FileName>
      </DependentFile>
   </DependentFiles>
</Root>
```

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

XML schema for the dynamically loaded assembly list

The following XML schema is applicable for the dynamically loaded assembly XML list whether the list is provided as an embedded manifest resource or an external configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="Root">
      <xs:complexType>
         <xs:sequence>
            <xs:element minOccurs="1" maxOccurs="1" name="DependentFiles">
               <xs:complexType>
                  <xs:sequence>
                     <xs:element minOccurs="1" maxOccurs="unbounded"
                     name="DependentFile">
                        <xs:complexType>
                           <xs:sequence>
                              <xs:element minOccurs="1" maxOccurs="1"
                              name="FileName" type="xs:string"/>
                           </xs:sequence>
                        </xs:complexType>
                     </xs:element>
                  </xs:sequence>
               </xs:complexType>
            </xs:element>
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

Embed the XML manifest resource in the primary assembly

To embed the XML manifest resource in the primary assembly

1. In Visual Studio, add the XML list file to the client control assembly project. Any file name can be used, but the name must have the extension .aaCFG.
2. Change the Build Action property value to Embedded Resource.

   After compilation, the XML will be available in the assembly as an embedded manifest resource file.

   During the client control import operation, the system will read any embedded XML manifest resources with the extension .aaCFG. The system will then import any listed assemblies that are stored in the same location as the primary assembly.

Include the XML manifest resource in an external configuration file

To include the XML manifest resource in an external configuration file

1. Create the XML list file using the same root name as the primary assembly but with the extension .aaCFG. For example, if the primary assembly name is MyControl.dll, then the configuration file name would be MyControl.aaCFG.
2. Store the file in the same directory as the primary assembly.During the client control import operation, the system will look for a file that has the same root name as the primary assembly but with the extension .aaCFG and in the directory in which the primary assembly is stored. If this file is found and an

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

embedded XML manifest resource exists, the system will consolidate the two lists to eliminate duplicate entries. The system will then import any listed assemblies that are stored in the same location as the primary assembly.

Tips on dynamically loaded assemblies

Refer to the following guidelines to prevent issues with importing the dynamically loaded assemblies.

- Make sure that the XML is valid. Invalid XML in the embedded manifest resource or the configuration file will result in the entire client control import operation for the selected primary assembly to be cancelled.
- All assemblies on which the dynamically loaded assemblies are directly or indirectly dependent must be included in the same directory as the primary assembly and included in the XML list. If the system is unable to locate and load any of the direct or indirect dependencies, the entire client control import operation for the selected primary assembly will fail.
- If a dynamically loaded assembly is going to load another assembly dynamically, make sure that the other assembly is included in the XML list. If any such assemblies are not included in the primary assembly's manifest resource or configuration file, the import operation will succeed. However, these indirectly loaded assemblies will not be imported, which can result in the client control not behaving correctly during execution.

Embedding Client Controls

You can embed an installed client control into a graphic as you would embed a graphic within another graphic.

We recommend that you not overlap client controls with other elements on the canvas. Otherwise, the client controls may not work correctly.

To embed a client control into an Industrial graphic

1. On the **Edit** menu, click **Embed Industrial Graphic**. Your HMI's attribute/tag browser appears.
2. Browse to the location that contains the client control.
3. Select a client control from the right panel and click **OK**. The pointer changes to paste mode.
4. Click the canvas where you want to embed the client control. The client control is placed onto the canvas.

Embed the TagPicker client control

To embed the TagPicker client control, follow the steps of the Example of installing the TagPicker control. Then do the following:

1. Create a new graphic in the Visualization folder.
2. Open the graphic in the Industrial Graphic Editor.
3. On the **Edit** menu, click **Embed Industrial Graphic**.
4. Select **aaTagPicker** and click **OK**.
5. Click on the canvas near the top left corner. The TagPicker control is placed on the canvas.

OMI apps and widgets

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For detailed information about placing and configuring apps and widgets in an OMI ViewApp, refer to the *Widgets, Apps, and Controls Help* (WidgetsPDFHelp.pdf).

Considerations when creating a ViewApp for the OMI web client

The OMI web client uses cross-platform technology to display ViewApps. Therefore, it supports some features differently than the OMI desktop client, and some features available in the desktop client are not available in this version of the OMI web client. When building a ViewApp that will be used with the OMI web client, and especially when beginning to use a pre-existing ViewApp with the Web Client, you need to keep these limitations and differences in mind. We strongly suggest you test existing ViewApps carefully before using them with the web client in a production environment.

The considerations and limitations are divided into these types:

- Prerequisites for using the OMI web client
- Security Considerations
- General limitations of the OMI web client
- Limitations when using AVEVA Industrial Graphics
- OMI app limitations
- Widget limitations
- Layout Script limitations
- ViewApp Namespace limitations
- Responsive Layout limitations

Note: We are still adding functionality to the OMI web client. Many of the limitations described for this version will not apply in future releases.

Prerequisites for using the OMI web client

To be able to use the OMI web client to display OMI ViewApps, AVEVA System Platform must be registered with AVEVA Identity Manager. Perform this action in the Configurator either during or after installation.

Before you can make OMI ViewApps available to users, you must define and deploy at least one instance of a WebViewEngine. Assigning ViewApp instances to a WebViewEngine makes them available from the OMI web client. See Prepare to deploy a ViewApp: assign a ViewApp to an engine for more information.

Required web browsers

The OMI web client is supported with the latest versions of the Google Chrome and Microsoft Edge web browsers.

Preparation needed on remote devices

Any remote device where you will be using the OMI web client to run ViewApps must have a security certificate installed. This security certificate ensures private communication between the remote device and the WebViewEngine, enhancing the security of your data.

A remote device is any device that is not part of the same System Platform installation as the platform that hosts the WebViewEngine. This includes:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Computers that have System Platform installed but point to a different System Management Server than the one that the platform hosting the WebViewEngine uses. This should occur only if you have two completely separate System Platform installations and want to access the OMI web client on one of those installations from a device hosting the other installation.
- Any device--such as a computer, workstation, tablet, or phone--that does not have System Platform installed.

Depending on procedures and policies at your organization, this security certificate may be installed by the IT department on all eligible remote devices, or users may need to install the certificate themselves before using the OMI web client for the first time. For instructions on installing the certificate on a Windows device, see Install a security certificate on Windows.

Note: For more information on security certificates, be sure to check the user documentation applicable to your operating system.

Install a security certificate on Windows

As described in Prerequisites for using the OMI web client, before you can start the web client on any remote device--that is, any device which does not have System Platform installed, or which has System Platform installed but uses a different System Management Server (SMS)--you must install a security certificate. The most reliable way to do this is to export the security certificate from the SMS, copy the exported certificate file to the remote device, and install the certificate. The procedures below describe how to export the certificate on the SMS and one way to install the certificate on a Windows computer.

To export the security certificate from the System Management Server

1. On the SMS, start the AVEVA Configurator.
2. Under **Common Platform**, select **System Management Server**.
3. Select **Advanced**, then, on the **Certificates** tab, select **Details**.

   This opens the **Certificate** dialog box.
4. Select the **Details** tab, then select **Copy to File**.

   This opens the **Certificate Export Wizard**.
5. Select **Next**, then follow the instructions to export the certificate. Make these selections:
   - Do not export the private key.
   - Use the default export format (**DER encoded binary X.509)**.
   - Give the file any desired name and place it in any location. Do not change the .CER extension.

Once you have exported the certificate file, you can use it for all devices which need to connect to this SMS. Just copy the file to each remote device, then import the certificate there.

To install a security certificate on Windows

After you have copied the certificate file from the SMS to the remote device, you need to install it to the Trusted Root Certificate Authorities store on the device. You do this using the standard Windows methods for installing a certificate. The steps below are an example; the procedure on your version of Windows may be different.

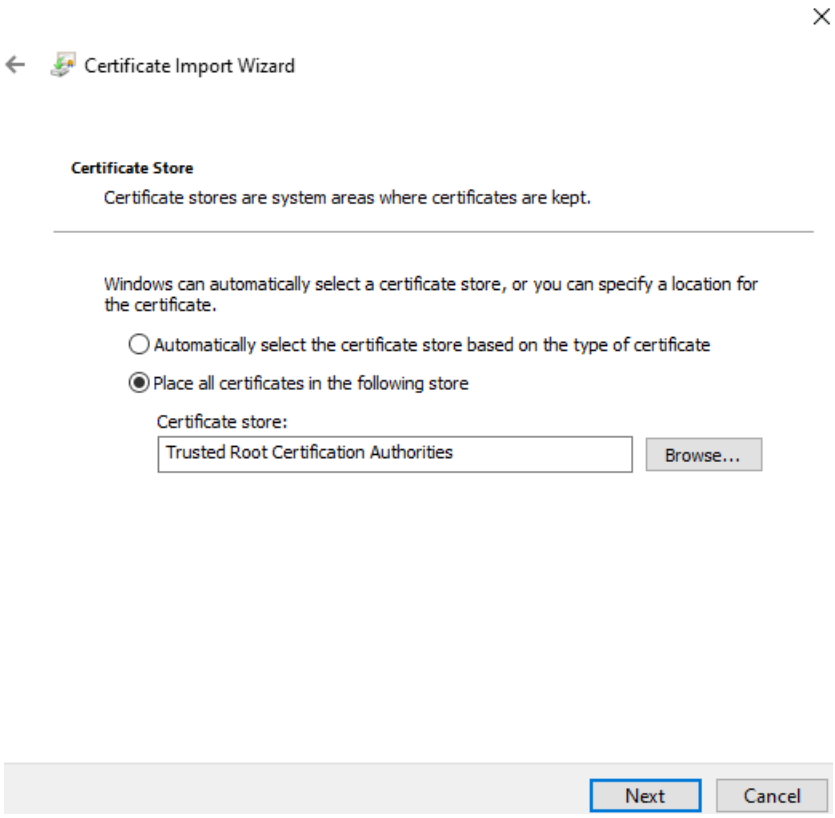
Note: If you need to install a security certificate on another type of device, such as a mobile device using the Android or iOS operating system, please see the documentation for your device.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

1. Using File Explorer, navigate to the certificate file you copied to the device and run it. The **Certificate** dialog box opens.
2. On the **Certificate** dialog box, select **Install certificate**. The **Certificate Import Wizard** opens.
3. Do not change the **Store Location**. Select **Next**.

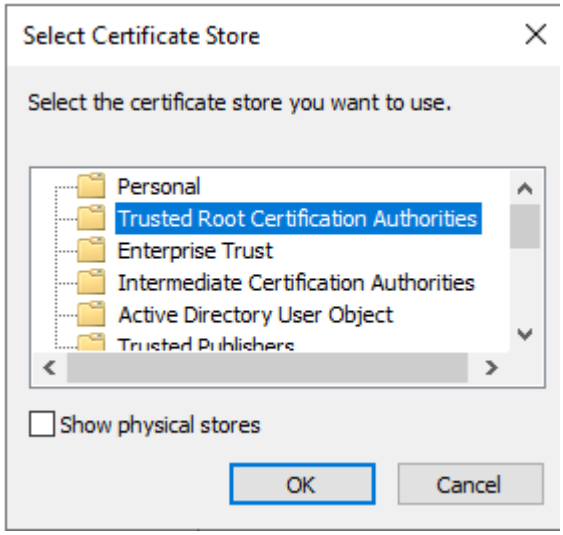
   You will now see one of two dialog boxes, depending on the version of the import wizard:

   - If you see the **Certificate Store** page, skip to step 5.
   - If you see the **File to Import** page, continue with step 4.
4. On the **File to Import** page, if the file you just ran is not shown, select **Browse**, navigate to the file, and open it. Select **Next**.
5. On the **Certificate Store** page of the wizard, select **Place all certificate in the following store**, then select **Browse**..



The **Select Certificate Store** dialog box opens.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



6. Select **Trusted Root Certification Authorities**, then select **OK**. Select **Next**. The **Completing the Certificate Import Wizard** dialog box opens.
7. Select **Finish**.
8. After the certificate is imported, close and re-open your browser if it was open. You should now be able to open the OMI web client by going to `https://<web_client_host>/omi`.

Security considerations for the OMI web client

The OMI web client allows users to open OMI ViewApps from workstations or other devices (such as a phone or tablet) where no System Platform components are installed. To support secure access from non-System Platform nodes, anonymous access is not allowed and an administrator must configure a System Management Server.

When the Galaxy uses operating system group-based security, the OMI web client does not support user role and access level-based navigation. All navigation nodes will always be displayed regardless of access levels or user roles that may be assigned to them. Any user who can open the ViewApp can access all nodes and see all objects tracked by the ViewApp

You cannot use the OMI web client if you are using Galaxy security as the authentication mode for your Galaxy. You must use one of the other authentication modes: None, OS User-based, OS Group-based, or Authentication Providers.

Note: The OMI web client only supports authentication using AIM, and the authentication happens in the AIM node.

If you have configured your Galaxy to use OS user security, and a user tries to log in as a local OS user (such as localhost/user1) to an OMI web client app running on a remote node, authentication may fail. This is because the localhost will be interpreted as the machine which AIM is running, and the localhost/user1 of the AIM machine may not have permissions set up properly in the Galaxy.

ViewApp namespace attributes related to security do not show the correct values based on the logged-in user.

When connected experience is enabled, inactivity time-out is not supported. Users must actively log out of the OMI web client to end their connection.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

General OMI Web Client Limitations

The following list summarizes the major general limitations in the current release of the OMI web client.

- Multi-Galaxy configurations are not supported with this version of the web client.
- Secured and verified writes are not supported in this version of the web client.
- You cannot customize the web client URL used to call a specific ViewApp directly.
- You cannot customize the graphics used to indicate ViewApps on the initial OMI web client page.
- Embedded .NET controls will not render correctly on non-Windows devices. Only Windows devices support rendering of .NET controls.
- History playback is not supported in this version of the web client.
- The web client cannot show multiple screens at once. If a ViewApp uses multiple screens, you can switch between screens in the web client window. See Using ViewApp in a web client vs. on the desktop for instructions.
- You cannot preview an OMI ViewApp based on how it will look in the web client. The **Preview** button on the ViewApp editor shows a desktop client preview only.
- You cannot generate a URL for a page displayed in the web client and then embed that URL in another product to access the content. This is possible in the InTouch web client, but **not** in the OMI web client.
- By default, you cannot place the web client window in an iFrame, as doing so could make the web client vulnerable to a "clickjacking" attack. If you need to put the web client in an iFrame, see Placing the Web Client in an iFrame.
- When using Operating System group-based security, nodes do not appear in a ViewApp Navigation tree based upon their specified access levels configured for a read-write user belonging to an Operating System group-based security group.
- In very complex graphics with a large number (thousands) of references, some references may take a very long time to resolve when compared to the OMI desktop client.
- If a user increases the size of a layout during runtime to exceed the defined size of the panes in the layout, the panes will expand to fill the layout, even if they are set to fixed size.

Important! Several graphics in the Visualization folder incorporate elements or settings that are not supported by the OMI web client. Before using a system graphic in a ViewApp that will be used with the web client, make sure that it will display properly on all devices where it may be accessed.

When loading a graphic in the web client, unsupported features are logged as warnings in the Logger. A generated report will include the warning path.

Place the web client in an iFrame

By default, you cannot place the OMI web client in an iFrame because doing so could make it vulnerable to a "clickjacking" attack. If you need to place the web client in an iFrame, you must whitelist the web page that will be hosting the iFrame.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To allow the OMI web client to be placed in an iFrame

1. On the node hosting the WebViewEngine, make a copy of the proxy server **AppSettings.json** file for safekeeping. The default location of this file is **C:\Program Files (x86)\Common Files\ArchestrA\Services\proxyserver**.

   Note: You need elevated privileges to save changes to the **AppSettings.json** file.

2. Open **AppSettings.json** in a text editor.
3. Find this section of the file:

```
// List down all URLs separated by space. The web sites with the urls are
allowed to embed this OMI web client inside an iframe.
  // 'self' is by default allowed. Make sure you postfix the whole value by ";"
semicolon.
  // i,e. "'self' https://www.google.com https://www.aveva.com;"
  // i,e. "'none';"
  // Proxy Service must be restarted after you update frame-ancestors value.
  "ContentSecurityPolicy": {
    "frame-ancestors": "'self'"
```

4. The comments in this section describe how to make the needed changes. Add each URL that should be allowed to host the web client in an iFrame to the `frame-ancestors` list, separating each URL with a space and ending the list with a semi-colon (;). The entire list must be enclosed in quotation marks.
5. Save your change and close the file.

Known Issues in the OMI web client

This version of the OMI web client contains the known issues listed below. Some of these issues are also described in other places in this help file.

- When opening the web client home page in Chrome, the user may receive an authentication error. This happens if System Platform was not registered with AVEVA Identity Manager in the Configurator. To correct this:
  - Open the Configurator.
  - Under **AVEVA System Platform**, select **Identity Manager Registration**.
  - Select **Configure**.

  After this procedure, there should be a green check mark next to **Identity Manager Registration** in the tree view.

Industrial Graphics limitations in the OMI web client

When including AVEVA Industrial Graphics in an OMI ViewApp for use with the OMI web client, there are some limitations and behavioral differences which you should keep in mind.

In general, be sure to test your ViewApp using each browser and type of device (for example, workstation, tablet, and/or phone) that it may be used on. This is especially true when the ViewApp was first developed for the OMI desktop client or when it includes any of the industrial graphics supplied as defaults with System Platform. Some of the supplied graphics use features or settings which are not supported in this version of the OMI web client. Also, some graphic elements may be displayed in the wrong location in the web client.

General limitations when using industrial graphics in the OMI web client include:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Custom properties with History Summary data type are not supported. All other custom properties are supported.
- Text in a graphic may not wrap properly even when word wrap is set to true.
- Graphics which extend to the very edges of a layout pane may have their edges cut off in the web client.

For information about specific limitations and differences when using industrial graphics, see these topics:

- Graphic element limitations
- Animation limitations
- Embedded widget limitations
- Script limitations
- Font limitations

Graphic element limitations

These graphic elements are not supported and will not work in the OMI web client:

- Multi Pens Trend
- Trend Control
- Alarm Client
- Trend Client

Note: While the trend-related graphic elements shown above are not supported, the Trend Pen element and the HistoricalTrendApp OMI ViewApp are supported.

While the Alarm Client graphic element is not supported, the AlarmApp OMI ViewApp is supported.

All other graphic elements are supported, with the limitations described below.

These properties are not supported for any graphic element:

- TabOrder
- TabStop

All other properties are supported.

Gradients and patterns

Because the OMI web client uses different technology to render elements on different platforms, we strongly suggest you check the appearance of gradients and patterns on each type of platform where a ViewApp may be accessed.

A gradient fill may sometimes extend beyond the element it is supposed to fill.

Gradients do no always display correctly when used with arc, pie, or chord-based elements, such as pipe connectors.

Animation limitations

The OMI web client has some limitations which apply to all animations when used with any Industrial Graphic Editor element and other limitations which apply only to a specific animation.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The following list shows the limitations for all supported animations. *Any animation not listed is not supported by the OMI web client.*

All Animations

- Keyboard shortcuts are not supported.

Action Script Animation

- Animations with a While Center Down trigger type do not work the same as in the desktop client. The script does not stop triggering at the proper time based on the mouse position.

Blink Animation

- Blinking may stop after a period of time. This has been observed with the CalendarApp and with the ComboBox element.

Push Button Animation

No limitations.

Slider Animation

No limitations.

ToolTip Animation

- Because the OMI web client uses different technology to render elements on different platforms, we suggest you check the appearance of tooltip animations on each type of platform where a ViewApp will be accessed.

User Input Animation

- You cannot encrypt string input.
- You cannot define a character to use to hide password input. The default # character is always used.
- You cannot input a multi-line string.

Embedded widget limitations

When embedding a widget into an Industrial Graphic for use with the OMI web client, keep the limitations below in mind. Other than these limitations, widgets embedded in graphics work in the OMI web client as they do in the OMI desktop client.

Map_App

Embedding the Map_App widget in an Industrial Graphic is not supported.

Displaying Industrial Graphics on an embedded widgets

Some widgets, such as the Carousel widget, support displaying Industrial Graphics in the widget. This does not work with the OMI web client.

Script limitations

- The OMI web client uses cross-platform technology to render visuals on non-Windows platform. This causes some limitation in script usage in graphics. If a client script in a graphic uses a .NET namespace that is not compatible with the .Net standard namespace, the script will not function correctly. Where possible, you

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

may need to refactor the script to use the .Net standard namespace for proper script execution. If there is no equivalent member in the standard namespace, the script is not supported in the web client.

- OMI web client does not support the EmbedContent() function.
- The OnHide script for a graphic or layout is not triggered if you close the browser tab (or the entire browser window) where the ViewApp is running. The OnHide script is triggered correctly if you close just the graphic or layout.
- Action scripts configured to trigger while the center mouse button is down do not work. All other trigger types do work.
- SQL scripts and scripts that access the device's file system are not supported. You may be able to work around this by moving the scripts' logic to object scripts and bind visual elements to object attributes.

In addition, the following script methods are not fully functional when used in an OMI web client application. Click the method name to see its limitations.

HideGraphic()

ShowContent()

ShowGraphic()

HideGraphic() limitations

The name of a graphic is used as the hide identity.

ShowContent() limitations

The ShowContent() method does not support the **Name** property.

ShowGraphic() limitations

Scripts that include the ShowGraphic() method have the following limitations when used in OMI ViewApps accessed through the OMI web client:

- Do not support Identity (only support is ShowSymbol using the graphic name as the identity).
- Do not support the **Name** property.
- Do not support the **Owning Object** property.
- Do not support the **ResizeWindow** property.
- Do not support the **HasTitleBar** property. Applications always show a title bar.
- Do not support the **HasCloseButton** property. Screens always show a close button.
- Do not support the **WindowTitle** property.
- Do not support **ScalePercentage** property.
- **Center** is the only supported value of the **Position** property.
- Do not support using the same interface element to open several graphics.

Popup limitations

When ShowGraphic() produces a popup window, the following limitations are present:

Relative to Symbol:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- The size of a popup window is always the same as the graphic's original default width and height with scaling at 100 percent.
- If a graphic's original default size is larger than the screen, the popup window appears in full screen mode as well as relative to the desktop.
- The popup window is always placed at the center of the screen.

  Relative to Desktop:

  The popup window is always shown in full screen mode.

  Custom Width and Height:

  The width and height of the popup window cannot be customized.

Font limitations

For OMI ViewApps opened in Google Chrome or Microsoft Edge on Windows, local operating system fonts are supported in most cases. However, this font support has these limitations:

- The user must accept the font use permission message from the browser. If the user does not, only one font is provided for each CJK language and Roboto is used for all other languages.
- The actual font must exist for any configured style used, such as bold or italic. For example, if Arial Italic is configured, Arial must exist on the workstation, with a defined Italic style. The OMI web client does not perform software calculations to change the regular font to italic. The characters will be shown in the regular font.
- The configured font must support the language being used. There is no fallback. For example, if you configure Chinese characters to use the Arial font, the OMI web client will not show the correct Chinese characters.

With any other browser or operating system, only one font is provided for each CJK language and Roboto is used for all other languages.

Custom fonts are supported. It is the customer's responsibility to acquire the necessary licenses for any custom font used.

Note: These font limitations apply to all fonts used in a ViewApp, including those in font style animations and those applied to any element of a graphic either directly or through an element style.

Fonts and Language Settings

If you are concerned about the possibility of missing characters or other issues because the font being used in a ViewApp may not support a user's language, you can set the ViewApp to always use the font defined for the current language in the Galaxy configuration. If you set this option, all fonts defined for any part of any ViewApp are ignored and the OMI web client uses only the font defined for the current language on the **Languages** pane.

For more information, see Set additional properties for a ViewApp.

OMI App limitations in the OMI web client

The following OMI Apps that are included in the Galaxy templates provided by AVEVA do not work in an OMI ViewApp used with the OMI web client:

- DocViewerApp

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- ImageViewerApp
- InsightApp
- PDFViewerApp
- SpreadsheetViewerApp

Limitations

All other OMI Apps that are included in the Galaxy templates provided by AVEVA are supported, but have these limitations:

- Event scripts are not supported
- Keyboard navigation is not supported.

The OMI Apps listed below have some additional limitations when used with the web client:

- AlarmApp
- ContentPresenterApp
- GraphicRepeaterApp
- HistoricalTrendApp
- MapApp
- TitlebarApp
- WebBrowserApp

AlarmApp limitations

Only the properties listed below are supported when using the AlarmApp with the OMI web client:

- Follow Current Context
- Only show Alarms on current Asset
- Show Grid
- Configuration: Alarm mode, Colors
- Client Mode supports only mode 1 (Current Alarms)

No other AlarmApp properties are supported.

The AlarmApp cannot query InTouch alarms.

ContentPresenterApp limitations

For the **Size Mode** property,only **Scaled** is supported. **Fixed** mode is not supported with the OMI web client.

When configuring the content of a ContentPresenter instance, we recommend you limit the number of graphics as much as possible. The web client loads all of the graphics into memory, so large numbers of graphics, or very complex graphics, can use a great deal of memory. Be sure to check the performance of ViewApps that use the ContentPresenter before you place them into production for use with the web client.

GraphicRepeaterApp limitations

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

These GraphicRepeaterApp properties are not supported with the OMI web client.

- Sort By
- Sort Order
- Live Sorting

HistoricalTrendApp limitations

Only these HistoricalTrendApp design-time properties are supported with the OMI web client:

- Follow Current Context
- Pens
- Grid Horizontal
- Grid Vertical
- Grid Visible
- ValueAxisLevel

In addition, during runtime, the user can set:

- Start time
- End time
- Duration
- Mode

No other HistoricalTrendApp properties are supported.

Historian system tags are not supported.

During runtime, be aware of these additional limitations:

- The label for the x-axis will not wrap.
- The pen color is randomly generated. If you refresh the browser page, it may change.

MapApp limitations

These MapApp properties are not supported with the OMI web client:

- Browser Stylesheet
- Event Handler Available Events

In addition, the Sources property supports only string constants, not reference values.

Defining a reverse proxy for the MapApp

Any map source that needs a key or credential to be used with the MapApp must use a reverse proxy to connect to the OMI web client. Configuring this for the MapApp requires updates to a settings file on the WinPlatform that hosts the WebViewEngine. You must also be sure to enter the correct information when configuring the MapApp in its layout pane.

Important! Only these map sources require a credential or key: Google, Bing, and sometimes ArcGis, Baidu, XYZ, and WMS. You do not need to perform these steps for other map sources.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The general procedure for configuring any map source to use a reverse proxy is:

1. Enter the necessary information in the source **Properties** fields when configuring the MapApp in its layout pane.
2. On the platform that hosts the WebViewEngine, edit the **C:\Program Files (x86)\Common Files\ArchestrA\Services\proxyserver\appsettings.json** file:
   a. In the `Routes` section for the map source, find and uncomment the section for your map source.
   b. In the uncommented section, set the correct `Path`, `PathRemovePrefix`, and `Append` values for the map source.
   c. In the `Clusters` section for the map source, enter the URL for the map source.
3. Clear the browser cache on the WebViewEngine after making the changes to appsettings.json.

The sections below describe the specific changes for each type of map source.

Google Maps

Properties fields: Enter the API key.

`Routes` section of appsettings.json:

- `Path` and `PathRemovePrefix`: Enter 1 for the first Google map. If there is another map configured in the editor, enter 2. For example:

  ```
  "path":"/Map/google/1/1{**catch-all}"

  "PathRemovePrefix": "/Map/google/1"
  ```

- `Append`: Enter your Google API key as the value.

Clusters section of appsettings.json: The `Address` value should already be the URL for the Google map. Correct it if it is not.

Bing Maps

Properties fields: Enter the API key.

Routes section of appsettings.json:

- `Path` and `PathRemovePrefix`: Enter 1 for the first Bing map. If there is another map configured in the editor, enter 2. For example:

  ```
  "path":"/Map/Bing/1{**catch-all}"

  "PathRemovePrefix": "/Map/Bing/1"
  ```

- `Append`: Enter your Bing API key as the value.

Clusters section of appsettings.json: The `Address` value should already be the URL for the Bing map. Correct it if it is not.

XYZ Maps

Properties fields: Enter the source URL. If there is no key parameter in the URL, this map does not need a reverse proxy, and you do not have to make any changes to appsettings.json. If there is a key parameter, make the changes below.

Routes section of appsettings.json:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- `Path` and `PathRemovePrefix`: Enter 1 for the first XYZ map. If there is another map configured in the editor, enter 2. For example:

  ```
  "path":"/Map/XYZ/1{**catch-all}"

  "PathRemovePrefix": "/Map/XYZ/1"
  ```

- `Append`: Enter your XYZ API key as the value.

Clusters section of appsettings.json: The `Address` value should already be the URL for the XYZ map. Correct it if it is not.

WMS Maps

Properties fields: Enter the source URL. If the URL does not require a credential, this map does not need a reverse proxy, and you do not have to make any changes to appsettings.json. If it does need a credential, make the changes below.

Routes section of appsettings.json:

- `Path` and `PathRemovePrefix`: Enter 1 for the first WMS map. If there is another map configured in the editor, enter 2. For example:

  ```
  "path":"/Map/WMS/1{**catch-all}"

  "PathRemovePrefix": "/Map/WMS/1"
  ```

- `Append`: Set the `Basic` value to the Base64String encoded username:password for the map URL.

Clusters section of appsettings.json: Set the `Address` value to the URL for your map server.

Baidu Maps

Properties fields: Enter the source URL. If there is no ak parameter in the URL, this map does not need a reverse proxy, and you do not have to make any changes to appsettings.json. If there is an ak parameter, Make the changes below.

Routes section of appsettings.json:

- `Path` and `PathRemovePrefix`: Enter 1 for the first Baidu map. If there is another map configured in the editor, enter 2. For example:

  ```
  "path":"/Map/Baidu/1{**catch-all}"

  "PathRemovePrefix": "/Map/Baidu/1"
  ```

- `Append`: Enter your Baidu key as the value.

Clusters section of appsettings.json: The `Address` value should already be the URL for the Baidu map. Correct it if it is not.

ArcGis Maps

Properties fields: Enter the source URL. If the URL does not require a credential, this map does not need a reverse proxy, and you do not have to make any changes to appsettings.json. If it does need a credential, make the changes below.

Routes section of appsettings.json:

- `Path` and `PathRemovePrefix`: Enter 1 for the first ArcGis map. If there is another map configured in the editor, enter 2. For example:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

```
"path":"/Map/arcgis/1{**catch-all}"

"PathRemovePrefix": "/Map/arcgis/1"
```

- Append: For ArcGis versions before 10.9, set the `Basic` value to your ArcGis token. For older versions of ArcGis, set the `Append` value to `Bearer` followed by your ArcGis token.

Clusters section of appsettings.json: Set the `Address` value to the URL for your map server.

TitleBarApp limitations

The Keyboard property is not supported with the OMI web client. You cannot show an on-screen keyboard when using the TitleBarApp in the web client.

WebBrowserApp limitations

When using the OMI web client, you cannot show a web page that has a Content Security Policy configured with frame ancestors unless the web client site is in the list. You also cannot open a page that is configured with X-Frame-Options.

Widget limitations in the OMI web client

When placing a widget into a layout pane for use with the OMI web client, keep the limitation below in mind. Other than this limitation, widgets work in the OMI web client as they do in the OMI desktop client.

Breadcrumb widget

Alarm adorners do not work in the Breadcrumb widget.

Map_App widget

Embedding the Map_App widget in an Industrial Graphic is not supported in the OMI web client.

Displaying Industrial Graphics on an embedded widgets

Some widgets, such as the Carousel widget, support displaying Industrial Graphics in the widget. This does not work with the OMI web client.

Layout script limitations in the OMI web client

Embedded .NET controls will not render correctly in the OMI web client on non-Windows devices. Only Windows devices support rendering of .NET controls.

In addition, scripts used in layouts have all of the same web client limitations as scripts used in graphics. For information on the limitations when using layout scripts in either layouts or graphics for use with the web client, see Script limitations.

ViewApp namespace attribute limitations

Attributes in the default ViewApp namespace have the following limitations when the ViewApp is opened in the OMI web client:

- Attributes which are set to retain their values do not do so. If you close and reopen the ViewApp, the attribute will be reset to its default.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- MyViewApp.Language.CurrentLanguage does not show the language's full name. For example, for American English it shows en（US）.
- The following namespace attributes are not supported:
  - MyViewApp.Graphic.ShowHelp
  - Security-related attributes

OMI web client responsive layout limitations

When using a responsive layout in a ViewApp through the OMI web client, the Column Push and Column Pull features are not supported. Do not rely on these features when designing a layout that will be used in a ViewApp through the OMI web client.

For more information about responsive layouts, see About the responsive layout mode.

About use of touch screens in ViewApp

Touch screens enable users to interact with a ViewApp similarly to applications running on touch-enabled laptops, tablets, or smart phones. On a touch screen, users can manage a running ViewApp by:

- Single tap to push a button, start action scripts, set focus within a window, select an item from a list
- Touch and swipe on vertical or horizontal scroll bars to change the viewing area within a pane
- Pan the content of a pane by swiping a screen with a single finger
- Trigger animations configured for graphic elements, ACF-apps, or NET Controls.
- Zoom a screen view in or out using two finger gestures.
- Double tap to restore the display to its original size
- Display slide out panes using three finger swipe gestures.
- Close a windows using a 5 finger scrunch gesture.
- Interact with user controls that do not natively support touch.

ViewApps can be configured with screen locks to prevent accidental screen interactions. When enabled, a screen lock must be deactivated when interacting with a touch-enabled ViewApp.

Build a ViewApp that supports touch gestures during runtime

Building a ViewApp that supports touch gestures during runtime requires configuration of the screen profile and the layouts that are associated with the ViewApp.

- Screen Profile

  When editing a screen profile, you can enable touch on a screen and set values to a Touch Lock that restricts touch interactions to ensure the ViewApp runs safely.

  For more information about screen profile properties, see About screen properties.

- Layout

  Each pane of a layout includes a set of properties. Based on values set to these properties, touch interactions can be defined at the pane level.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

For more information about layout Appearance properties, see Set pane properties 1.

Software Requirements for Touch ViewApps

Touch supports in running ViewApps requires the following system software:

- Client computer operating system: Windows 10 or higher
- Server computer operating system: Windows 2012 R2 or higher.
- Remote Desktop Protocol (RDP) connectivity between supported client and server computers

Recommendations for Touch ViewApps

Follow these recommendations to create content intended for touch interactions.

- The minimum size of on screen controls should be at least 23X23 pixels (13X13 DLUs).
- Commonly used screen controls should be larger at least 40X40 pixels (23X22 DLUs).
- Include a five pixel border around UI elements that are touch enabled to reduce the likelihood of accidentally touching other nearby screen content.
- Use the Hamburger app navigation control to save space for ViewApps designed to run on small screens.

About ViewApp touch interactions

ViewApp touch interactions are summarized in the following tables based upon the type of finger gesture and the target content shown in a running ViewApp. For more detail about touch gestures during runtime, see About touch interactions supported during runtimes.

Content Level Gestures

Content level gestures are single finger gestures applied to a graphic or control in a screen that trigger configured animations or predefined actions.

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Touch down	Touch down		• Action Scripts • User Input • Push Button • Show / Hide Graphic	• Starts action scripts • Sets ViewApp focus
Touch move	Touch down & drag		• Slider control • Pane scroll bar.	• Moves slider control • Moves panel scroll bar
Touch up	Touch up		• Action Scripts • Push button	• Stops action scripts • Releases button

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Long Press	Touch, wait and lift		Action Scripts	Functionally equivalent to a mouse right-click
Double tap	Two touches in quick succession		Pane	• Stops action scripts, • Fits display to pane

Pane Level Gestures

Pane level gestures are two finger gestures applied to a pane that change the view of content.

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Pinch to zoom	Spread fingers apart to zoom in. Pinch fingers together to zoom out.		Pane	Zoom in or out Maximum zoom in is 500 percent and the minimum zoom out is 100 percent based on pane size
Swipe	Touch + drag (+ lift)		Pane	• Pan the window • Hierarchy navigation
Flick	Touch, make a quick flick.		Pane	The inertia and acceleration of a flick gesture distinguishes it from a swipe gesture.

Layout or ViewApp Level Gestures

Layout or ViewApp level gestures require two or more fingers and affect the behavior of content shown in a screen window.

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Multi-finger swipe	Touch three fingers, drag and lift		Pane.	Three finger gesture shows a slide out pane from the edge opposite to the direction of the swipe.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Scrunch	Five finger touch and squeeze.		• Single pane • Multiple panes	On scrunch, the window's content scales until it reaches 70 percent of its original size at which point the window closes. In a multi-pane application, a scrunch gesture can be performed within a single pane or across multiple panes.

About touch interactions supported during runtime

This topic provides some detail about unique touch behavior during runtime.

Multi-Touch Gestures

- Within a layout, content and pane level gestures are supported. However, if a content level gesture is in progress, no other types of gestures function in other panes during the duration of the content level gesture
- Excluding a scrunch gesture, all other types of gestures act on the contents of the pane within which the gesture occurs. Touch interactions within one pane are independent of interactions in other panes.
- Touch gestures interact with only a single WCC\WPF at a time. However, other non-WCC process graphics can interact with touch, at the same time.
- For WPF controls implementing iSupportTouch, the WPF control is responsible for ensuring multi-touch support.

Touch Support for Apps

- The NavTreeControl and the NavBreadcrumbControl support single finger swipe and flick gestures. The hierarchical tree of the NavTreeControl can be scrolled up or down and left or right. Drop-down lists of the NavBreadcrumbControl can be shown or hidden by the supported gestures.
- Single finger swipe and flick gestures can be used to navigate the alarm hierarchy of the AlarmApp.
- Single finger swipe and flick gestures can be used to show or hide drop-down lists of the ContentPresenter and TitleBar apps.
- Native touch support on WPF controls is not supported. That is, if a user has implemented recognition of touch gestures within WPF, unless the user implements our ISupportTouch interface, it will not work.
- For WCC and WPF controls, without ISupportTouch implementation, touch will be converted to equivalent mouse gestures, where possible. These controls will respect the state of the Touch Lock if it does not implement ISupportTouch.

  Legacy ArchestrA .NET controls accessible from the **Tools** area do not implement ISupportTouch.
- Touch interactions along with mouse\keyboard actions are not supported.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Scope of Touch Interactions

- Touch interactions initiated from within a pane continue to function when the gesture continues outside the boundaries of the pane.
- In the case of overlapping graphics, the topmost element configured with input animation is the target of the touch interaction.
- The behavior of double tap and long press are customizable through the Windows Control panel. AVEVA OMI ViewApps support operating system customizations.

Runtime touch interaction limitations

Some touch limitations are apparent during runtime.

- A double tap gesture to select text in an Edit Box is difficult to achieve.
- Using mouse and touch simultaneously is not supported. Either interaction type may cancel the other, but no serious errors occur that prevents the ViewApp from continuing to run.
- Touch gestures are not supported when an app has been placed in a multi-content layout pane and hierarchy or content navigation is enabled.
- Pan and zoom gestures are not supported by apps.
- Content Tooltip animations do not appear when using touch because there is no concept of hovering directly over a screen item with touch.
- A layout supports only one long press or double tap gesture at a time. You cannot perform a double tap or long press simultaneously on a layout.

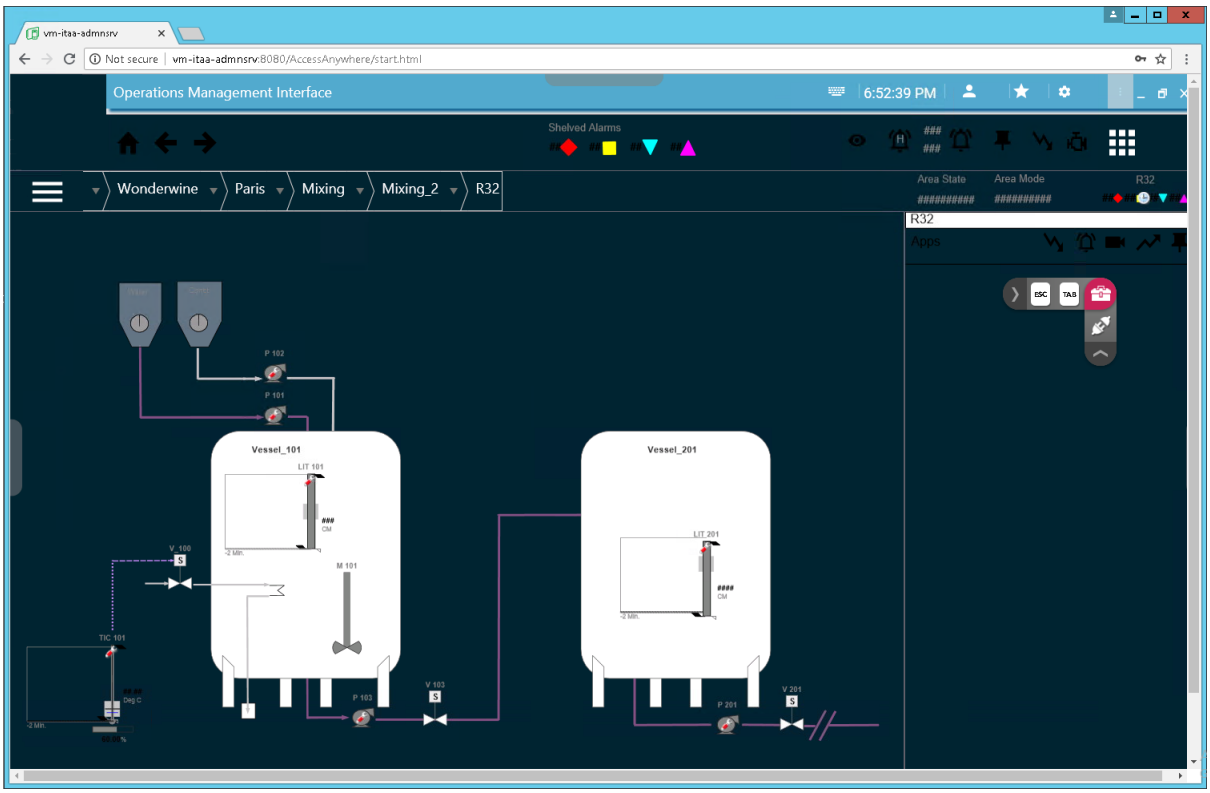
  Note: A layout can support multiple simultaneous touch down or touch up gestures.

- Multi-touch gestures do not work on WCC controls.
- WinForm controls have severe touch limitations:
  - A WinForm does not respond to a Touch Lock interaction. When Touch Lock is used with WinForm controls, touch gestures become inoperable.
  - Pan and Zoom gestures do not work on layout panes if a pane contains a WinForm control.
  - Hierarchy Navigation does not work on layout panes if a pane contains a WinForm control.
  - A scrunch gesture does not work if a finger touches any WinForm control.
  - Three finger gestures do not work when one of the fingers touches a WinForm control
- WPF control specific limitations
  - Touch Lock cannot be used to control message box dialogs thrown from a WPF application such as those created from System.The user's finger must be removed from the Touch Lock when working in Command Lock or Two Handed Operation mode to interact with these dialogs.
  - WPF touch events are not supported. Any touch interaction with a user control will be promoted to equivalent mouse events. To support touch, the control must implement the ISupportTouch interface. Refer to AVEVA OMI SDK documentation for details.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About InTouch Access Anywhere

InTouch Access Anywhere enables users to remotely view InTouch applications or AVEVA OMI ViewApps from a desktop computer or mobile devices including tablets, smartphones, or laptops. You view and control applications or ViewApps through a secure HTML5 web browser. Using InTouch Access Anywhere, a separate client application does not need to be installed on your portable device or desktop computer to show a running ViewApp.



InTouch Access Anywhere provides the following features:

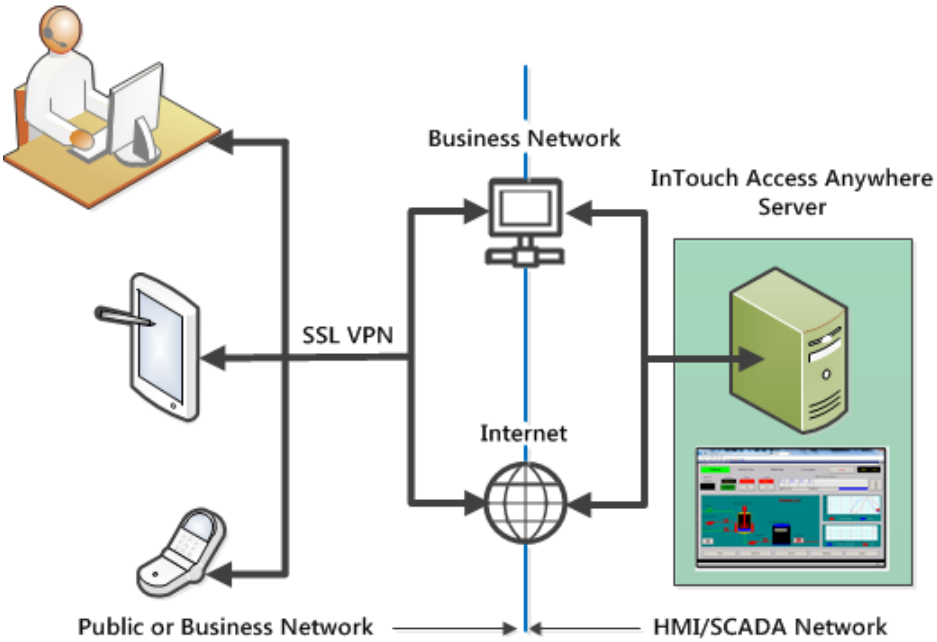
- Provides secure and remote access to InTouch applications or AVEVA OMI ViewApps
- Incorporates image compression, packet shaping, and whole frame rendering to improve network performance
- Automatically adjusts the size of an InTouch Access Anywhere session to the web browser window showing the application
- Supports touch gestures on touch screen devices
- Works on devices that only support web applications like Chromebook
- Provides an expandable session toolbar with icons to disconnect InTouch Access Anywhere sessions, access system keys, and copy application data to the Windows clipboard.

About InTouch Access Anywhere architecture

The InTouch Access Anywhere server is installed on a RDP host in your company's data center. One or more AVEVA OMI ViewApps are deployed to this host computer known as the InTouch Access Anywhere server. When you log on to the InTouch Access Anywhere server, you select the ViewApp you want to view. Typically, you will

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

use a VPN from a public or business network that supports websockets to securely connect to the InTouch Access Anywhere server.



For more information about configuring the InTouch Access Anywhere server , see Configure InTouch Access Anywhere.

Verify an InTouch Access Anywhere installation

InTouch Access Anywhere is installed as part of the overall installation of System Platform. For specific instructions to install InTouch Access Anywhere, see the *AVEVA System Platform Installation Guide*.

After installing the component parts of InTouch Access Anywhere, you need to verify the installation before starting the configuration tasks.

To verify your InTouch Access Anywhere Server installation

1. Before using InTouch Access Anywhere to connect to your Remote Desktop server, log on using a standard Remote Desktop Client, select a ViewApp from Application Manager, and launch it. As a best practice, complete this step for each user expected to display ViewApps through InTouch Access Anywhere.

   This configures the initial setup and enables InTouch Access Anywhere clients to determine the list of available ViewApps.

   The InTouch Access Anywhere Server can be used immediately after installation.

2. Open an HTML5-compliant browser and enter the URL of the InTouch Access Anywhere Server:

   http://machinename:8080/ or http://IPaddress:8080/

   This URL automatically redirects to the full URL:

   http://machinename:8080/AccessAnywhere/start.html

   The InTouch Access Anywhere Server port must be specified in the URL to tell the browser to use the web server that is built into the InTouch Access Anywhere Server service. HTTPS may also be used, but will prompt you to continue without a secured certificate.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure InTouch Access Anywhere

This set of topics provides information about the initial configuration of the InTouch Access Anywhere server after installation. For information about pre-installation requirements and the installation procedure, see the *AVEVA System Platform Installation Guide*.

The InTouch Access Anywhere Server installation attempts to create exceptions within the Windows Firewall to allow the necessary network connections.

If you experience problems connecting to InTouch Access Anywhere after installation, ensure that the InTouch Access Anywhere Server is configured to allow connections through port 8080, and that its executable is allowed to communicate by configuring the Windows Firewall as follows.

Configure InTouch Access Anywhere server

InTouch Access Anywhere Server is a server-side service that translates RDP into WebSocket communication. The InTouch Access Anywhere Server is installed on a RDS host.

The remote client running on a browser connects to the InTouch Access Anywhere Server service using WebSockets directly or through the Secure Gateway.

Configuring a Firewall Port Exception

By default, a client (browser) connects to an InTouch Access Anywhere Server using port 8080 for both encrypted and unencrypted WebSocket communication. This port number can be changed using the InTouch Access Anywhere Server Configuration utility.

To enable direct connection from the client to the InTouch Access Anywhere Server (without using the Secure Gateway), the server must be directly accessible from the client using port 8080.

You can open a port through the firewall either through the command line, or through the Windows firewall configuration. The command line option is presented for advanced users as a quicker way to configure the firewall versus using the GUI.

Using the Command Prompt

To configure the firewall through the command line, first open the Windows **Command Prompt**. Then, type the following command:

```
netsh.exe advfirewall firewall add rule name="<Description>" dir=in action-allow
protocol=TCP localport=<PortNumber>
```

Where:

<Description> = the description used to describe this firewall rule

<PortNumber> = the TCP port to open

For example:

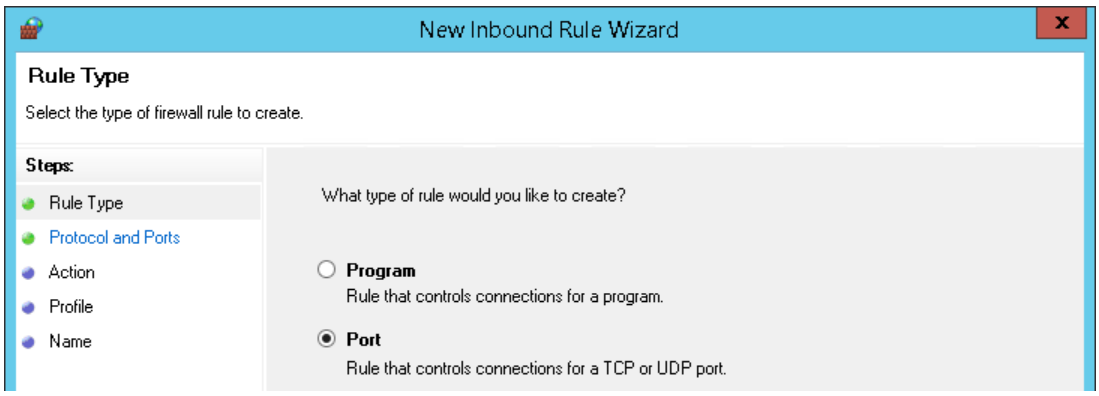
```
netsh.exe advfirewall firewall add rule name ="Open Port 8080 for InTouch Access Anywhere"
dir=in action=allow protocol=TCP localport=8080
```

Using the Windows GUI

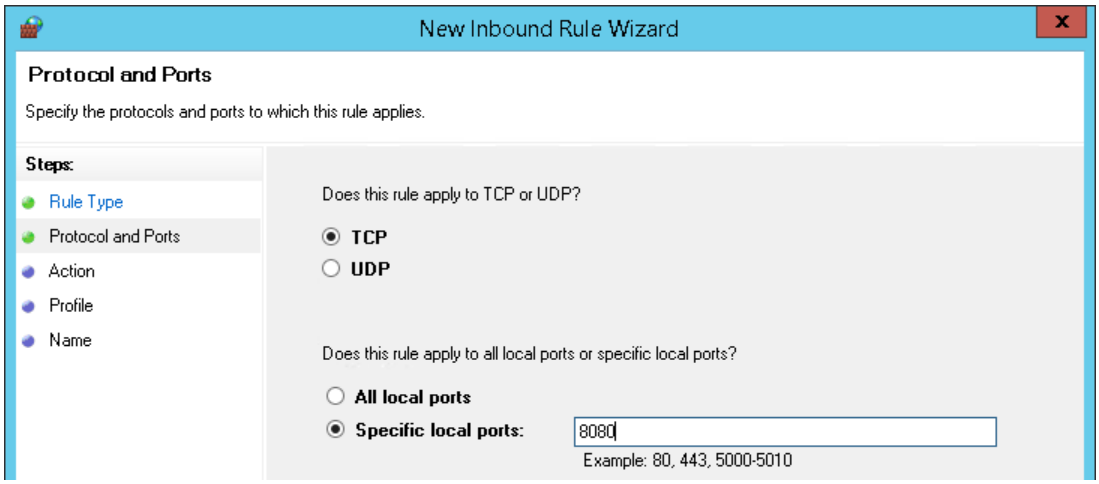
If the Windows firewall is enabled on the same computer where the InTouch Access Anywhere Server is installed, make sure to configure it to enable the InTouch Access Anywhere client connection.

1. Open the Windows **Control Panel** and then **Windows Firewall**.
2. Select **Advanced Settings** and select **Inbound Rules**.
3. Click **New Rule**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



4. Select **Port** and click **Next**.
5. Enter the specific port: 8080.



6. Click **Next** and select **Allow the connection**.
7. Click **Next** and select to apply the rule on the Domain, Private, and Public networks.
8. Click **Next**, assign a name for the rule, and click **Finish.**

Configuring a Firewall Program Exception

In addition to adding an exception for connections on Port 8080, the InTouch Access Anywhere Server program must be added to the list of programs able to communicate with the network.

Use the Server Configuration console for settings

The Server Configuration console shows a series of tabs that enable an administrator to configure various settings of the server service.

You can launch the InTouch Access Anywhere Server Configuration tool from the **Start** Menu Program Group or on the **Apps** view.

In general, changing the InTouch Access Anywhere Server configuration is not required. It is recommended to use the default settings.

Note: It is recommended to hide the Server Configuration application from end users to prevent unexpected changes to the server's settings.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The following topics describe the different configuration tabs of the InTouch Access Anywhere Server.

Use the Server Configuration console General tab

The **General** tab provides functions to start and stop the InTouch Access Anywhere Server service. For certain configuration changes, a service restart is required. This page also displays the number of active InTouch Access Anywhere Server client sessions connected to this computer.

Note: Whenever the InTouch Access Anywhere Server service is restarted, all sessions on the server are disconnected.

Performance

The **Performance** tab displays current performance statistics related to InTouch Access Anywhere connections.

Use the Server Configuration console Communication tab

The **Communication** page provides options to change the InTouch Access Anywhere Server port and the address of the host computer running RDS.

When using an InTouch Access Anywhere Server listening port other than the default (8080), the port number must be explicitly specified in the client address field (for example, http://<machine name>:5678/).

When running InTouch Access Anywhere Server on a computer with multiple network cards, change the RDP host address. Change this address from localhost to the IP or DNS address of the network card that has RDP access to the system.

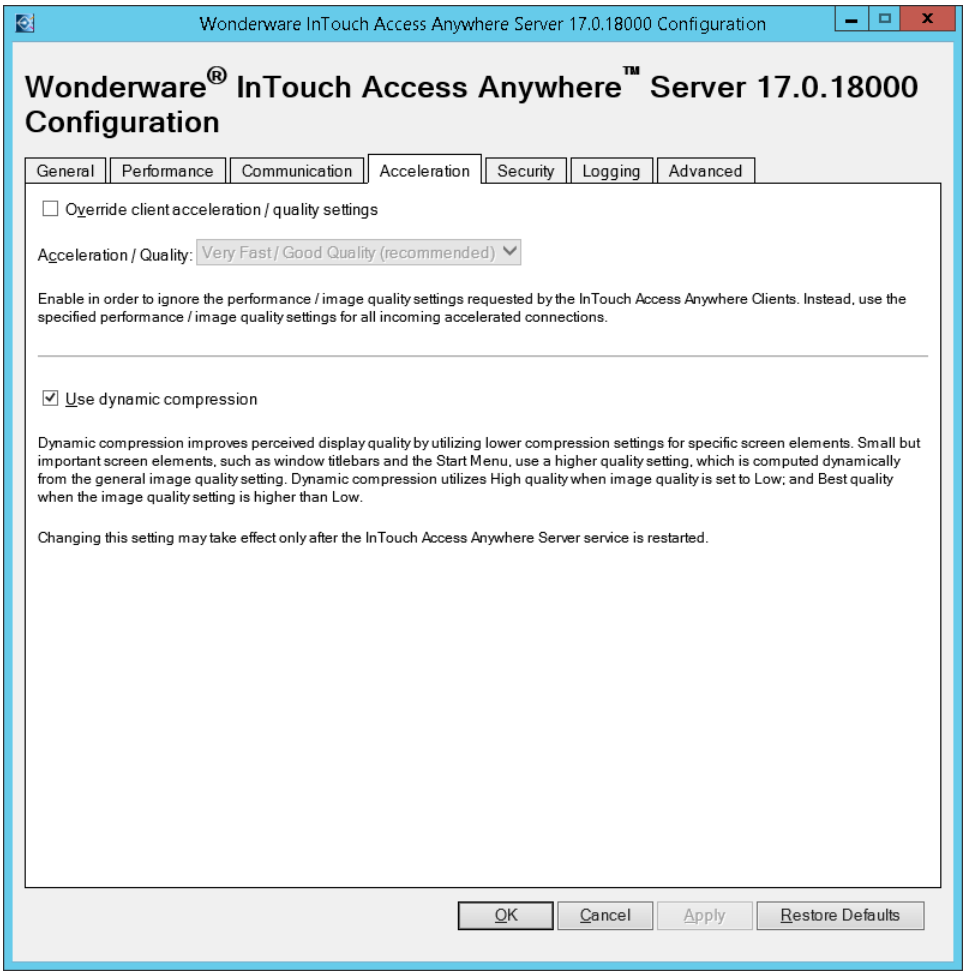
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Changes to either setting require a service restart. The **General** tab provides buttons to start or stop the service. You can also start or stop the service using the Windows Service Manager.

Note: If you change the port number, ensure that you make the corresponding changes to the config.js file's "wsport" setting. For more information, see Static Configuration of the Config.js File

Use the Server Configuration console Acceleration tab

The **Acceleration** tab provides options to change the Acceleration or Quality level and disable dynamic compression.



When the **Override client acceleration/quality settings** check box is selected, all sessions use the configured setting, and all client settings are ignored. When selecting or clearing this setting, the service must be restarted for the change to become effective. When the setting is enabled, changing the acceleration level does not require a service restart, but active users must reconnect to use the new setting.

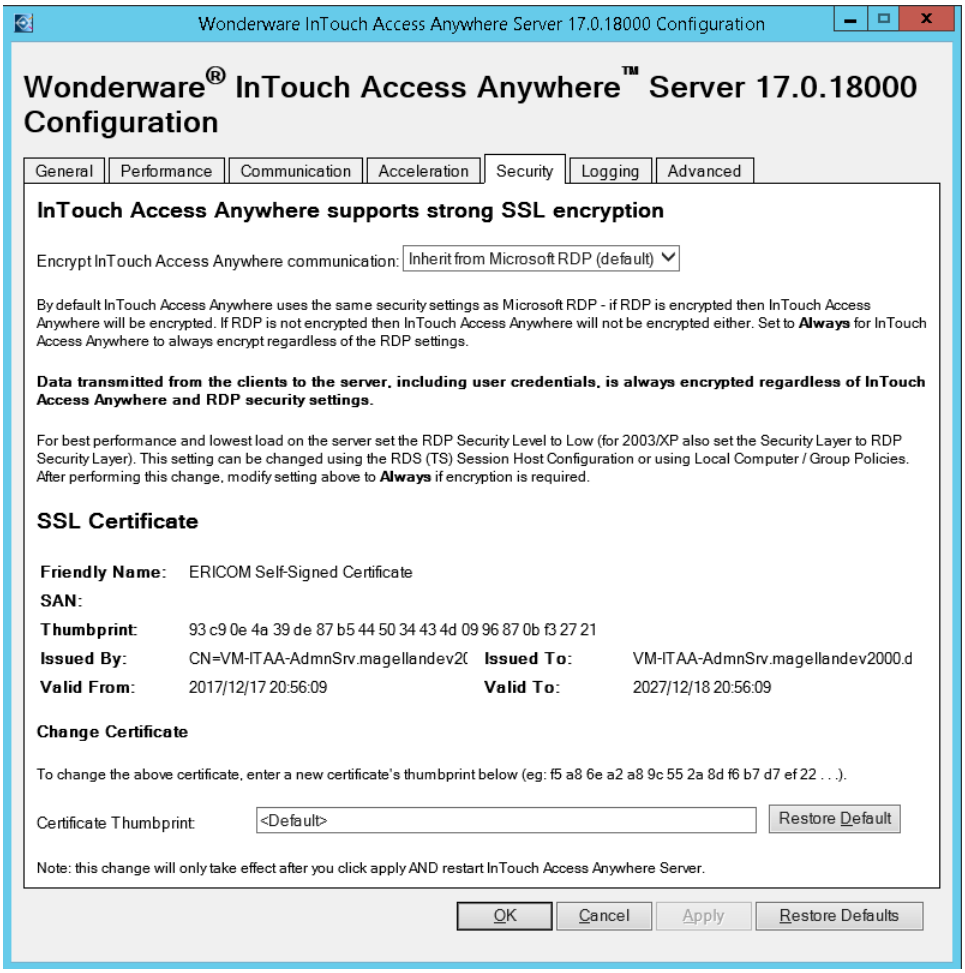
Dynamic Compression identifies small graphic elements within an application screen and compresses them during runtime. The most compression occurs when image quality is set to Low. The best quality images occur when image quality is set to higher than Low. All other graphical objects are compressed at the selected quality. This provides the visual impression of a high quality remote desktop session.

By default, this feature is enabled. To disable dynamic compression, clear the **Use dynamic compression** box.

Use the Server Configuration console Security tab

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

This **Security** page provides options to configure the InTouch Access Anywhere Server security settings.



Note: InTouch Access Anywhere provides integrated 128-bit SSL encryption. For best performance, set the host's RDP Security Encryption level to Low and change the Encrypt InTouch Access Anywhere communication to Always. Using this configuration, InTouch Access Anywhere SSL encryption will be used instead of the RDP encryption. Do not set this if users will be connecting directly to RDP regularly, as those sessions will end up using Low encryption.

To use a custom or trusted certificate, enter the thumbprint ID in the **Certificate Thumbprint** text box and click **Apply**. The certificate's properties will then appear.

Note: When installing a trusted certificate, the DNS address of the InTouch Access Anywhere Server must match the certificate name. If wildcard certificate is used, the domain must match. For example, if the certificate is for `*.acme.com`, the server name must end with `acme.com`.

Use the Server Configuration console Logging tab

The **Logging** tab provides options to enable or disable logging features. Technical Support may request a debugging log for diagnostic purposes. The debugging log is enabled here.

Advanced (For Administrator Use Only)

This page provides access to advanced Server settings that are stored in the system's registry.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Export Settings exports the InTouch Access Anywhere Server Registry key to the user's home folder (for example, My Documents).

Import Settings imports previously saved InTouch Access Anywhere Server Registry settings.

Advanced Configuration opens the Registry Editor.

About the InTouch Access Anywhere web component

The web component contains the resources used by a web browser to display an interface for the user to connect to an InTouch application. These resources include HTML pages, JavaScripts, CSS files, and graphic images. Review Configure InTouch Access Anywhere to modify the appearance and behavior of the web component interface.

About installation of the InTouch Access Anywhere Server

The InTouch Access Anywhere web components are automatically installed with InTouch Access Anywhere Server. The web components are located in the InTouch Access Anywhere Server folder, which by default is:

<drive letter>:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere

Note: Your installation may be located elsewhere depending on selections made during the installation process.

Modify the InTouch Access Anywhere interface

The InTouch Access Anywhere Server start page includes a group of images. All standard images can be edited and replaced with custom images. Keep the replacement images as close to the same dimensions as the original images. A default image is available as the logo.

The default path to the resources folder where the images are stored is:

C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere\resources

Note: Backup the resources folder before making any modifications. To roll-back to the original files, simply copy the original resources folder back to the original location.

InTouch Access Anywhere image files that are commonly customized include the following:

File Name	Description
Ericom.jpg	Logo image at the upper left-hand corner of the InTouch Access Anywhere Server landing page.
\images\Background-neuronal.jpg	Background image for the InTouch Access Anywhere Server landing page.

Note: Unless instructed by our Support group, customizations performed on the InTouch Access Anywhere page not herein described are not supported.

Modify the InTouch Access Anywhere connection name

The InTouch Access Anywhere connection name uses the RDP Host node name by default. The connection name can be modified to a custom string.

To change the connection's name:

1. Open the config.js file and add the name setting if it does not exist.
2. Set the name setting to the desired string enclosed in quotation marks.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

```
    executeTimeout: 1000,
    mMSendInterval: 100,
name: "testname",
    clipboard: true,
    clipboardTimeoutSeconds: 15,
```

Note: The name setting may also be set using the following cookie: EAN_name.

1. After setting the name parameter, the new label will appear in the connection's browser tab and in the **Establishing connection** dialog box.

About InTouch Access Anywhere secure connections

This section describes secure connection communication between WebSockets to both remote desktops and to the InTouch Access Anywhere Secure Gateway.

About secured WebSocket communication to remote desktops

The InTouch Access Anywhere Server installation includes a self-signed certificate for secure SSL connections. Some browsers, such as Google Chrome, allow self-signed certificates for SSL-encrypted WebSocket connections.

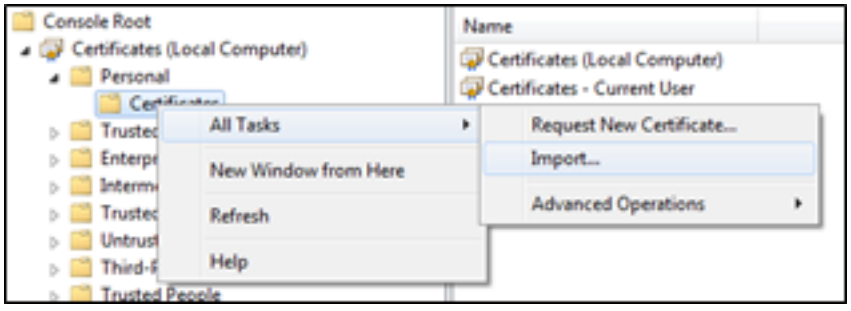
Opera browsers will notify the user that the server certificate is not signed and prompt the user to continue. Chrome OS, Safari 5.x, and Firefox do not allow secure SSL connections using a self-signed certificate.

In order to provide connectivity from these browsers, a trusted certificate must be imported into the InTouch Access Anywhere Server or into the InTouch Access Anywhere Secure Gateway if it is being used as a proxy for InTouch Access Anywhere Server. A trusted certificate must be purchased from a trusted certificate authority (for example, VeriSign).

Note: The DNS address of the InTouch Access Anywhere Server or Secure Gateway server must match the certificate name. If a wildcard certificate is being used, the domain must match. For example, if the certificate is for *.acme.com, the server name must end with acme.com.

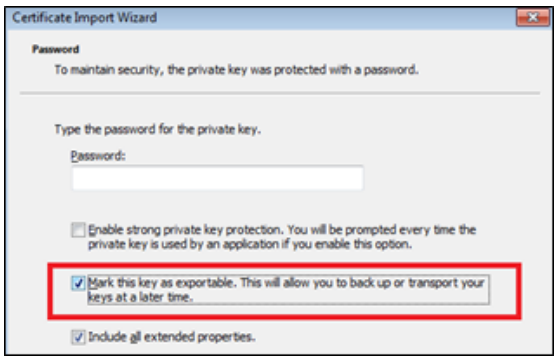
To import a trusted certificate into the InTouch Access Anywhere Server, perform the following steps using the Microsoft Certificate Manager.

1. Show the Windows Command Prompt running as an Administrator.
2. Type certmgr.msc to show the Certificate Manager.
3. Import the trusted certificate to the Computer (Personal | Certificates) store.

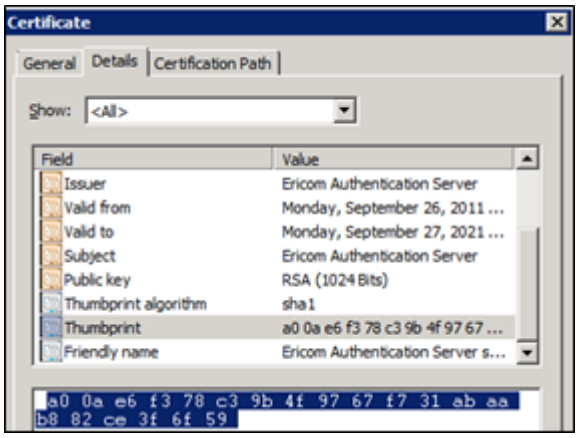


4. Mark the certificate as exportable during the import.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



5. Go to the Certificate's **Details** tab and highlight the **Thumbprint**.



6. Copy the thumbprint (Ctrl+c).
7. Stop the InTouch Access Anywhere Server service.
8. Using the Command Prompt (cmd.exe), go to the folder that contains AccessNowServer32.exe.
9. Run: AccessNowServer32.exe/genbincert <thumbprint of certificate to export enclosed in quotation marks>.

   The following is an example import command with thumbprint in quotation marks:

```
c:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server>AccessNowServer
32.exe /genbincert "18 9d f3 52 bb 35 77 12 da 87 e3 85 c6 e2 bc 45 50 50 fd 10"
```

10. After importing the thumbprint, a notification appears confirming the BIN certificate has been successfully created.
11. Start the InTouch Access Anywhere Server service and it will be ready for use.

Secured WebSocket connections using InTouch Access Anywhere secure gateway

The connection between a browser client and the InTouch Access Anywhere Secure Gateway is always secured. The InTouch Access Anywhere Secure Gateway is installed with a self-signed certificate by default, but supports trusted certificates as well. For information about using certificates with the Secure Gateway, see About InTouch Access Anywhere Secure Gateway ports and SSL certificates.

About Using a trusted certificate

Certain browsers permit HTTPS or SSL connections only when a trusted certificate is present. Install a trusted certificate in the InTouch Access Anywhere Secure Gateway or InTouch Access Anywhere Server to ensure safe

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

and reliable connections from a wide range of web browsers.A trusted certificate must be purchased from a trusted certificate authority (i.e., VeriSign).

About InTouch Access Anywhere on MobileDevices

This set of topics provides information about supported browsers and information regarding specific behavior of mobile devices.

Supported Browsers for InTouch Access Anywhere

The following Web browsers have been formally tested and verified to work with InTouch Access Anywhere.

- Microsoft Internet Explorer 11
- Microsoft Edge
- Firefox version 47
- Safari version 8
- Chrome version 51
- Opera version 38

Functionally Compatible Browsers

This list includes HTML5 browsers that should be compatible with InTouch Access Anywhere, but have not been tested.

- Microsoft Internet Explorer 10 if connected through Secure Gateway
- Firefox versions 38 and later
- Safari versions 5 and later
- Chrome versions 12 and later
- Opera versions 11 and later

Refer to the InTouch Access Anywhere Readme for more information regarding tested and supported browsers.

Older versions of Firefox and Opera require WebSocket support to be manually enabled in the browser configuration.

Multiple InTouch Access Anywhere sessions can be opened in different tabs within the web browser, or in different browser windows. When a session is not in use (its tab or window is not displayed) it will reduce its CPU and memory utilization.

Note: Each InTouch Access Anywhere session consumes an RDP session and an InTouch TSE license.

HTTPS Mode

For environments where WebSockets support is not available, InTouch Access Anywhere can work in HTTPS mode to transmit data by HTTPS only. HTTPS mode is used only if WebSockets support is not available. WebSockets will be used when available as it will provide better performance. HTTPS mode is required when using Microsoft Internet Explorer 9 with InTouch Access Anywhere Secure Gateway, or with SSL VPNs that only proxy HTTPS traffic.

To enable HTTPS mode, the InTouch Access Anywhere Secure Gateway is required. The InTouch Access Anywhere Server web pages must be delivered using the web server built into the InTouch Access Anywhere Secure Gateway (files are located under the Webserver/InTouch Access Anywhere folder).

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Complete the following procedudre to enable InTouch Access Anywhere for HTTPS support.

1. Install the InTouch Access Anywhere Server on the desired RDP Host.
2. Install the Secure Gateway on a separate computer located in a DMZ. The Secure Gateway must be installed on a server that is accessible by the target end-user group(s).
3. To connect to the InTouch Access Anywhere Server using HTTPS, enter the InTouch Access Anywhere URL of the Secure Gateway (the Secure Gateway includes the InTouch Access Anywhere web component): https://<securegatewayaddress>/InTouch Access Anywhere/start.html
4. Enter the parameters of the target InTouch Access Anywhere Server in the start.html page.
5. After connecting by HTTPS mode, a '-' character appears as a prefix of the address in the browser tab.

Note: HTTPS mode requires a browser that supports the HTML 5 Canvas. Older browsers, such as Microsoft Internet Explorer 8 (or earlier), do not support the HTML 5 Canvas.

Configure InTouch Access Anywhere

The topics in this section discuss several additional ways to configure aspects of InTouch Access Anywhere.

Modify the InTouch Access Anywhere Interface

Some images can be modified in order to customize the appearance of the interface. Graphics are stored in the "resources" sub-folder of the InTouch Access Anywhere Web Server installation (by default, "C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere\resources"), are most commonly modified:

Note: Back up the **resources** folder before making any modifications. You may then undo the changes by copying the backup to its original location.

File	Description
ericom.jpg	Logo image displayed at top left of InTouch Access Anywhere interface

Note: An experienced web developer can customize more graphics, though these modifications are not supported by AVEVA technical support.

Static Configuration of the Config.js File

An administrator can modify configuration settings of InTouch Access Anywhere by editing its config.js file that is installed as part of the InTouch Access Anywhere web component. This is a JavaScript file that can be modified using any text editor.

Important: Always create a backup before making any changes to the config.js file.

Most configuration settings in the config.js file have the following format:

name: value,

A value can be a number, a flag (true or false), or text enclosed in quotation marks. Some settings are prefixed by a double slash (//), which means they are disabled. Remove the double slash to assign a value to a setting. JavaScript rules apply in this file and certain characters need to be escaped (for example, backslash).

After the settings are configured, save the file and restart the server.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The config.js file contains the following configuration settings. Setting names are case sensitive. When settings are specified using cookies, setting names are prefixed by EAN_.

address	Address of InTouch Access Anywhere Server. This is always blank for the standard configuration.
audiomode	0 enables audio redirection (default). 1 plays audio on remote computer. 2 disables audio redirection.
blaze_acceleration	True determines if RDP acceleration is used.
blaze_image_quality	Sets the quality level using a numeric. For example: 40 (fair quality), 75, 95 (best).
dialogTimeoutMinutes	Time out period, in minutes, after which an inactive dialog is automatically closed and the session is logged off. The time out period is relevant only for dialogs that have a log off button.
disableToolbar	True (default); set to False to disable the toolbar, which contains shortcut icons and file functions, that appears within an InTouch Access Anywhere session window.
domain	The name of the domain against which the user name and password are authenticated to grant access to the Remote Desktop session.
encryption	False determines if encryption is enabled from the client to the InTouch Access Anywhere server.
endURL	URL to open to after the InTouch Access Anywhere session has ended (# value closes window). If there is a prefix with the symbol ^ then this sets the value of window.location instead of top.location. This is useful when the InTouch Access Anywhere session is embedded in a frame.
fulladdress	Address of RDP host. This is always blank for the standard configuration.
gateway_address	Defines the address and port of the Secure Gateway. For example: secure.acme.com:4343
gwport	The default gateway port that will be used if it is not explicitly specified in the address field.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

hidden	A comma or space-separated list of field names as they appear in config.js. For example, "username,password,domain". The listed fields are hidden to prevent the user from modifying them. To hide a button, such as the **Advanced** button, prefix the button text with the word `show`. For example, `"showAdvanced,showAbout"` hides both the **Advanced** and **About** buttons. All hidden variables will ignore previously saved settings.
leaveMessage	The message shown to the user after navigating away from an active session.
minDesktopWidth	Sets the minimum desktop width (in pixels) that InTouch Access Anywhere will display. The default is 800, which may not display as expected or desired on devices with a display width below 800 pixels.
minDesktopHeight	Sets the minimum desktop height (in pixels) that InTouch Access Anywhere will display. The default is 600, which may not display as expected or desired on devices with a display height below 600 pixels.
minSendInterval	Specifies the minimum duration between mouse position messages sent from the client when the mouse button is pressed. Units are in milliseconds.
name	Defines a custom string for the connection name. By default, the RDP host address is used.
noHTTPS	By default, InTouch Access Anywhere first attempts to connect using WebSockets. If the Secure Gateway is used with InTouch Access Anywhere, the connection will fall back to HTTPS when WebSockets are not available. If this setting is set to true, only WebSockets will be used and HTTPS fallback will be disabled.
onlyHTTPS	By default, InTouch Access Anywhere first attempts to connect using WebSockets. If the Secure Gateway is used with InTouch Access Anywhere, the connection will fall back to HTTPS when WebSockets are not available. If this setting is set to true, HTTPS is used immediately.
overrideSaved	False (default) settings that the user changes are preserved between sessions and override values set in config.js. Change to true for config.js to override preserved settings.
password	Password to pass into an InTouch Access Anywhere session (entered as clear text in config.js file).
reconnectOnDropped	True (default) automatically reconnects a session after recovering from a network outage. Set to False to disable this behavior.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

remember	False (default) determines whether the user's password will be saved in the InTouch Access Anywhere page for future use. Set to true to enable password saving (not recommended for kiosk usage).
resolution	Sets the resolution size of the InTouch Access Anywhere screen. The value set must be a valid option under the InTouch Access Anywhere screen resolution setting. For example: "1024,768". For Full Screen, use: screen.
sessionTimeoutMinutes	Time out period, in minutes, after which an inactive session is disconnected. The time out period resets automatically whenever the user clicks on the keyboard or a mouse button. The default value is 0, which disables this feature.
settings (URL parameter only)	Name of the Configuration Group to be used.
settingsURL	URL of the connection settings file.
use_gateway	False (default), set to true to use a Secure Gateway for remote access.
username	Username to pass into an InTouch Access Anywhere session.
wsport	The default WebSocket port that will be used by the client. The value specified in the file (8080 by default) is used for both encrypted and unencrypted WebSocket communication. The user can override this value by explicitly specifying another port address in the client user interface (UI). For backward compatibility with older versions of InTouch Access Anywhere Server, this behavior can be modified. If singlePort is set to false, then the port value specified is only for encrypted communication. The value specified in the file plus one (8081 by default) will be used for unencrypted WebSocket communication.

WARNING! Do not attempt to modify config.js settings not listed here unless directed by our Technical Support department.

Define Configuration Groups

All users share configuration settings specified in the config.js configuration file. Special settings can override global settings for certain groups of users. Multiple configuration groups are defined in the configuration file.

For example, if the Marketing group needs clipboard redirection and printing enabled, change config.js as follows:

```
var defaults = { / this already exists in the file
       …
       "Marketing": {// bold text are new additions
printing:true,
clipboard:true
```

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

```
        },
};
```

Note: The quotation marks surrounding Marketing must be identical. If necessary, delete them and re-type them if the text was copied from another source. Also, the last setting of the configuration group should not have a ',' at the end. This comma is placed after the closing bracket '}'.

In the URL to be used by the Marketing group, add the settings parameter:

http://<machine name>:8080/InTouch Access Anywhere/start.html?**settings=Marketing**

Settings Precedence

When an InTouch Access Anywhere client starts, it reads configuration information from a variety of sources. If two or more sources contain different values for the same setting, the value used by InTouch Access Anywhere is determined by the following precedence order:

Highest Precedence to Lowest Precedence

- URL parameters
- Cookies
- Saved settings from previous session
- config.js

For example, if the gateway_address is specified to be "server1" in config.js but "server2" in a cookie (EAN_gateway_address), then the value "server2" will be used.

If the setting override Saved is set to true in config.js, then any settings predefined in the config.js file will override previously used settings, and the precedence order will change slightly:

Highest Precedence to Lowest Precedence

- URL parameters
- Cookies
- config.js
- Saved settings from previous session

Note: These settings become effective only after the user starts a new session. In some cases, the local browser must be closed and reopened before changes become effective. The local browser cache may also need to be cleared.

SSO Form Post

When using a third-party authentication entity (such as an SSL VPN) that supports Form Post, the user can sign on to an InTouch Access Anywhere session using the authenticated credentials. The Secure Gateway is required for this feature.

In the authentication entity, there will be a field requesting the Post URL. Enter the SSO URL for the desired product:

AccessNow: https://sq-address/AccessAnywhere/sso

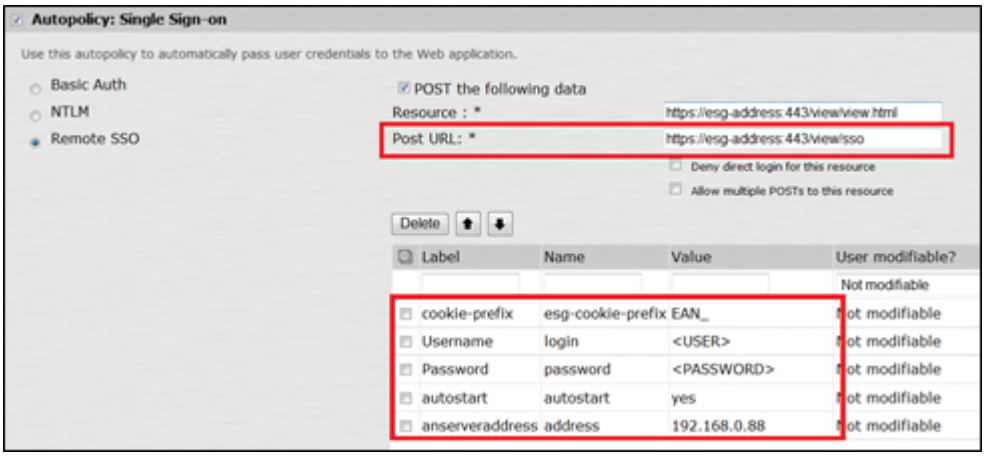
Note: The Secure Gateway will auto-redirect the request to the respective default page (start.html).

Include the following fields in the form:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- name="autostart" value="yes"
- name="esg-cookie-prefix" value="EAN_"
- name="username"
- name="password"
- name="domain"

Here is an example from a Juniper SSL VPN:



The value "esg-cookie-prefix" in the graphic above defines the Access Anywhere cookie prefix in the Single Sign-on form. For InTouch Access Anywhere connections, this is a mandatory entry.

If the target is a relative URL, it will replace the "/sso" portion in the path.

If the target is a full URL, it will completely replace the current path.

Embedding InTouch Access Anywhere in an iframe

To embed InTouch Access Anywhere within a third-party web page using the iframe mechanism, place an iframe tag within the containing page, and have the SRC attribute of the iframe reference the InTouch Access Anywhere URL.

For example:

```
<body>
    <h1>Embedded InTouch Access Anywhere</h1>
    <iframe src="http://127.0.0.1:8080/AccessAnywhere/start.html" style="width:1024px;
height:768px"></iframe>
</body>
```

When an InTouch Access Anywhere session ends, it can be configured to send the browser to a specified URL using the endURL setting.

- Specify a simple URL to redirect the iframe.
- Prefix the URL with ^ to redirect the iframe's parent (container).
- Prefix the URL with $ to redirect the top-most container.
- Specify # and the URL will close the browser tab.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

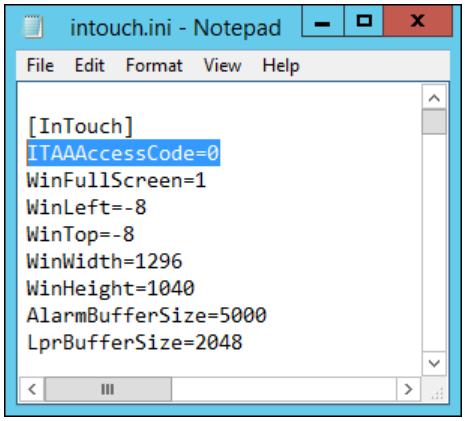
Hiding InTouch Applications from InTouch Access Anywhere

By default, all available InTouch applications are accessible by InTouch Access Anywhere.

You can hide an InTouch application from the list of applications provided by the InTouch Access Anywhere Server start page.

To hide an InTouch application

1. Browse to the location of the InTouch application in Windows Explorer.
2. Open the **intouch.ini** file in a text editor, such as Notepad.
3. Find or create the entry `ITAAAccessCode` under the `[InTouch]` section:
   - To hide the application, set `ITAAAccessCode=0` as shown below.



   - To display the application, set `ITAAAccessCode=1`. Note that this setting is unnecessary to display the application unless it exists *and* is set to 0.
4. Save the file under its original filename.

Configure gestures for touch devices

InTouch Access Anywhere automatically detects if a portable device is touch capable and automatically uses the built-in virtual keyboard for text input and gesture support for display navigation. InTouch Access Anywhere supports Windows multi-touch gesture redirection. All multi-touch gestures are redirected natively into the Windows session for use by an application running inside an Access Anywhere session.

Multi-touch gesture redirection feature is enabled using the Access Anywhere Toolbar button and by config.js settings.

Activation Criteria

Access Anywhere multi-touch gesture redirection is enabled and activated based on the following criteria:

Multi-touch gesture redirection functionality is enabled if all of the following are true:

- Touch is supported by the remote RDP host
- Touch is supported by the user's computer or mobile device
- Touch redirection is enabled by setting rdpTouchEnabled to True in the config.js file

Multi-touch gesture redirection can be activated if all of the following are true:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Touch feature is enabled
- Toggle MultiTouch icon is set to active on the Access Anywhere toolbar
- Touch is not suspended by the RDP host

Toolbar button

Users enable and disable the Windows multi-touch gesture redirection feature by toggling the Toggle MultiTouch icon in the Access Anywhere toolbar, which is enabled by default.

Multi-Touch Enabled	Multi-Touch Disabled

Multi-touch Gesture Redirection Settings in the Config.js File

The Config.js file is located in the folder path of the computer hosting the InTouch Access Anywhere server:

C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere

Config.js Settings	Description
rdpTouchEnabled	True (default) - enable remote touch. On the Server: enable the feature, create RDP dynamic virtual channel, and send RDP client touch events. On the Client – enable the feature, process the incoming server touch messages.
rdpTouchActive	False (default) – Sets the default activation state (ignored when not enabled.) On the Client this is the initial state of the toolbar button. If active, send touch events.
rdpTouchAction	0 (default) - Action to be taken if multi-touch redirection is enabled, but is not supported by the server or client device. Action codes and their assigned values: 0 - no action 1 - display an error message 2 - display an error message and disconnect from the session 3 - ask for user confirmation to continue without touch

Conflict with Local Gesture Usage

When multi-touch redirection is enabled, all gestures are redirected to the remote session. However, the user may need to use gestures locally on the device to pan and zoom around the session. When local gesture functionality is required, disable multi-touch redirection temporarily, and re-enable it when it is needed again.

HTTPS Mode

For environments where WebSockets support is not available, InTouch Access Anywhere can work in HTTPS mode to transmit data by HTTPS only. HTTPS mode is used only if WebSockets support is not available.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

WebSockets will be used when available as it will provide better performance. HTTPS mode is required when using Microsoft Internet Explorer 9 with InTouch Access Anywhere Secure Gateway, or with SSL VPNs that only proxy HTTPS traffic.

To enable HTTPS mode, the InTouch Access Anywhere Secure Gateway is required. The InTouch Access Anywhere Server web pages must be delivered using the web server built into the InTouch Access Anywhere Secure Gateway (files are located under the Webserver/InTouch Access Anywhere folder).

Complete the following procedudre to enable InTouch Access Anywhere for HTTPS support.

1. Install the InTouch Access Anywhere Server on the desired RDP Host.
2. Install the Secure Gateway on a separate computer located in a DMZ. The Secure Gateway must be installed on a server that is accessible by the target end-user group(s).
3. To connect to the InTouch Access Anywhere Server using HTTPS, enter the InTouch Access Anywhere URL of the Secure Gateway (the Secure Gateway includes the InTouch Access Anywhere web component): https://<securegatewayaddress>/InTouch Access Anywhere/start.html
4. Enter the parameters of the target InTouch Access Anywhere Server in the start.html page.
5. After connecting by HTTPS mode, a '-' character appears as a prefix of the address in the browser tab.

Note: HTTPS mode requires a browser that supports the HTML 5 Canvas. Older browsers, such as Microsoft Internet Explorer 8 (or earlier), do not support the HTML 5 Canvas.

InTouch Access Anywhere server limitations

This topic describes known behaviors and limitations of InTouch Access Anywhere when viewing an application or ViewApp on a portable device.

Networking Limitations

- Network quality

  Network quality will impact the performance of InTouch Access Anywhere running on mobile devices. Long latencies, limited bandwidth, and poor Wi-Fi coverage of the working area will impact user experience.

  We recommend that in the menu of your application you add a heartbeat or a clock that displays time, including seconds, that helps visualize good connectivity.

- InTouch Access Anywhere does not support WindowMaker

  InTouch WindowMaker is not supported in a TSE (Remote Desktop) environment. Therefore, InTouch Access Anywhere does not support InTouch WindowMaker. To prevent users from attempting to start WindowMaker from WindowViewer, do not install a license that enables WindowMaker and hide the Fast Switch menu bar in your InTouch applications.

Browser Limitations

- Browser Extension Conflicts

  Browser extensions and tool bars may inject JavaScript code into web pages, which can adversely impact the behavior of certain web pages. If InTouch Access Anywhere is not working properly, disable or uninstall any active browser extensions or tool bars. Restart the web browser after uninstalling or disabling an extension, and clear the local browser cache, to ensure that it is no longer active.

- HTTPS and SSL Encryption

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When the InTouch Access Anywhere page is delivered to the web browser using HTTPS, the SSL encryption setting will be checked by default. Modern browsers usually require WebSocket connections to be encrypted when launched from pages that are delivered using HTTPS.

- Zooming in Browsers

  Using the CTRL+ and CTRL- hotkeys to zoom only works with Internet Explorer 10.

Navigational limitations

- Mouse Events

  When designing your applications, keep in mind that certain mouse events do not have an equivalent behavior on a touch mobile device, including the following:

  - While Left Key Down
  - On Right Key Down
  - While Right Key Down
  - On Right Double Click
  - On Right Up
  - Mouse Center click
  - Pushbutton>Discrete Value>Direct

  Other mouse events are triggered with a gesture you must learn. For example, in many mobile devices a mouse over event is triggered by a tap on the screen.

- Right Click on Mac

  To perform a right-click on Mac OSX system: Command+ left-click.

- Left Click on iPad

  A single tap does not consistently toggle the state of a push button. Tap and hold to toggle a push button.

- Scroll Bars

  In some cases, moving a scroll bar in a touch environment can be difficult, particularly when the device has a small screen. As an alternative, try touching the empty area of a scroll bar in the direction you want to move.

- Dialog Boxes

  Dragging and dropping a dialog box can also be difficult on a touch device with a small screen. We recommend that you use a stylus to perform these operations for better precision, if possible.

- Using Software Keyboards

  InTouch provides the ability to invoke an InTouch keyboard or the Windows On Screen Keyboard from Input Animations. When designing applications to be accessed by InTouch Access Anywhere from mobile devices, keep in mind that these devices have their own software keyboards optimized for their specific form and size. In these cases, invoking the InTouch or the Windows keyboards from your application is not needed. In general, users have a better experience using a device software keyboard.

  Also, keep in mind that software keyboards in mobile devices in most cases do not have certain keys available in a physical keyboard, such as F1-F12, CTRL, or ALT. If you already have an application that uses Key Scripts associated with some of these keys, modify your application to use alternate available, supported keys.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Some key combinations may not be available through your mobile device, such as Shift+<letter>, CTRL+Shift, CTRL+ALT.

Network application development limitations

Distributed InTouch applications typically have a central development computer, central data storage, and client workstations that run distributed applications. You use InTouch Network Application Development (NAD) to build and maintain distributed applications. NAD enables client stations to maintain a copy of a single application without restricting the development of that application. Client stations are automatically notified when the application changes.

NAD applications do not appear on the list of applications shown on the Start.html page when the user logs on to the InTouch Access Anywhere server.

The InTouch Access Anywhere service populates the Start.html file drop-down list of available InTouch applications. This service runs under a local system account by default and cannot communicate with applications or folders on remote computers, which are typical of a NAD environment. This is the reason why NAD applications do not appear in the list of InTouch applications.

If you open Application Manager and see InTouch applications whose folders are not located on the computer's local hard drives, then those applications will also not appear on the drop down list of the Start.html page.

A workaround is to set the InTouch Access Anywhere service to run under a specific domain user account that has sufficient privileges on the Access Anywhere server and can also access the application directory on the NAD master computer.

Build an InTouch Access Anywhere Secure Gateway

InTouch Access Anywhere Secure Gateway is a complementary component of InTouch Access Anywhere server. The Secure Gateway ensures secure access to InTouch ViewApps by acting as a gateway between users in remote locations and AVEVA OMI ViewApps deployed to hosts running in a control network.

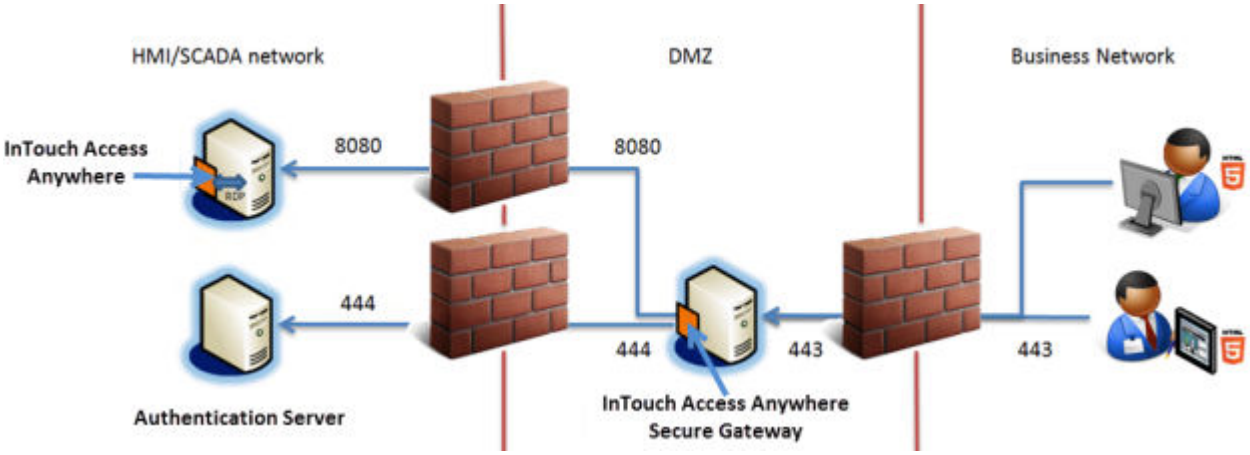
Secure Gateway provides the following benefits:

- Access to InTouch applications running on an internal network using a single secure port
- Eliminates the need to purchase, install, configure, and manage a VPN

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Located in a perimeter network, also known as a DMZ, while all other resources reside securely behind an internal firewall
- Provides the ability to install a single SSL digital certificate on the Secure Gateway node instead of requiring a certificate for every host that needs to be accessed
- Compatible with HTML5 client browsers supported by InTouch Access Anywhere

The following diagram shows the recommended architecture of the Secure Gateway in a production environment. The Secure Gateway uses a single port for secured remote access to InTouch applications. All web traffic from an external business network is tunneled through a SSL-based connection of the Secure Gateway placed in a DMZ.



The Authentication server is an optional InTouch Access Anywhere component and is disabled by default. The Authentication Server provides an additional layer of security by authenticating end-users before they can contact the Access Anywhere server. When the Authentication Server is enabled, only domain users will be able to authenticate. Local system users (such as Administrator) will not be able to log on through the Authentication Server.

02Build_ITAA_PostInstallation

This set of topics describes how to configure the Secure Gateway node to connect to an InTouch Access Anywhere Server.

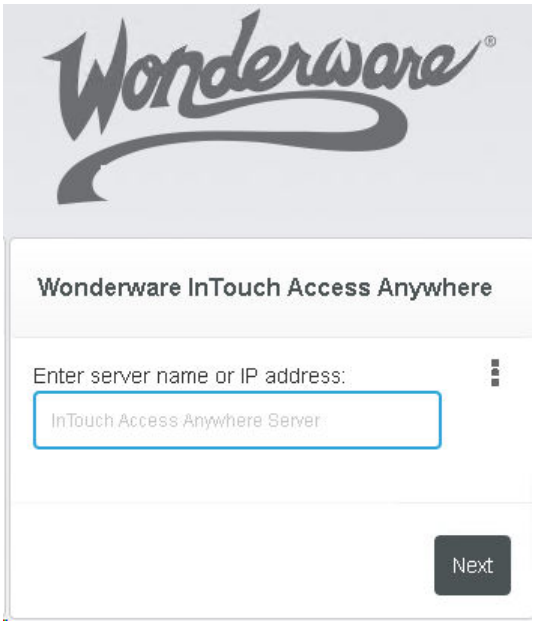
Connect to an InTouch Access Anywhere server through a secure gateway

The following logon procedures assume the InTouch Access Anywhere Server is installed on Node 1 and InTouch Access Anywhere Secure Gateway is installed on Node 2.

Important: One or more ViewApps must be deployed to the computer running the InTouch Access Anywhere server to be able to

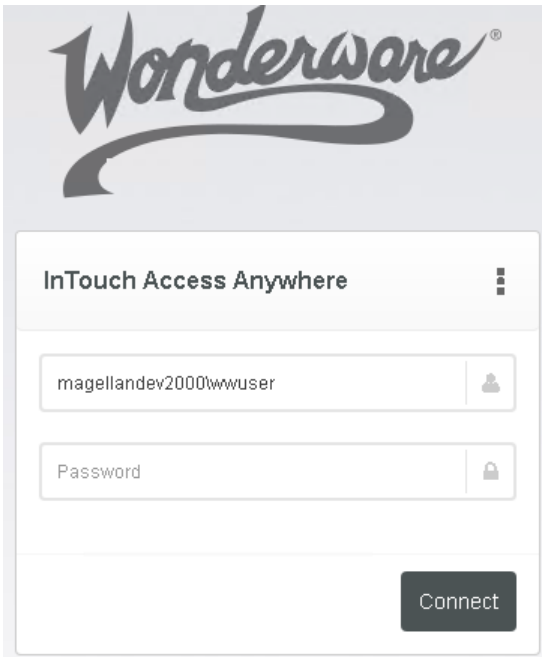
You access the InTouch Access Anywhere Server by first going through the InTouch Access Anywhere Secure Gateway node. When you navigate to https://<node2 name>/ , the **Connection Details** page appears:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



To access InTouch Access Anywhere Server on Node1, enter the computer name or IP address of Node1 in the **Enter server name or IP address** field and click **Next**.

The logon page to the InTouch Access Anywhere Server on Node1 appears with fields to enter a user name and password.



After providing your logon credentials and clicking **Connect**, there are two possible log on scenarios:

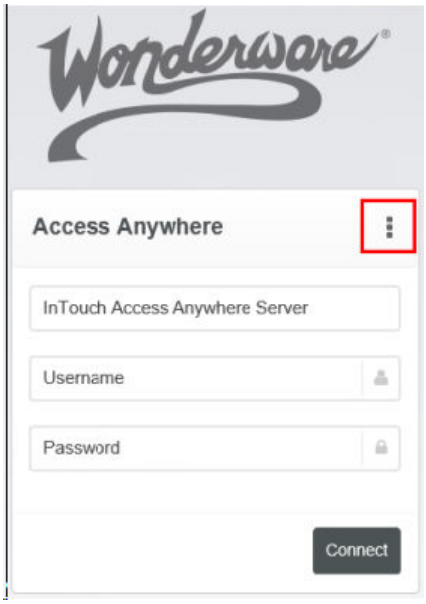
Scenario 1: InTouch Access Anywhere Secure Gateway node (Node2) does not show a list of InTouch applications

You can perform the following steps when you have a single InTouch Access Anywhere Server and will always run the same ViewApp.

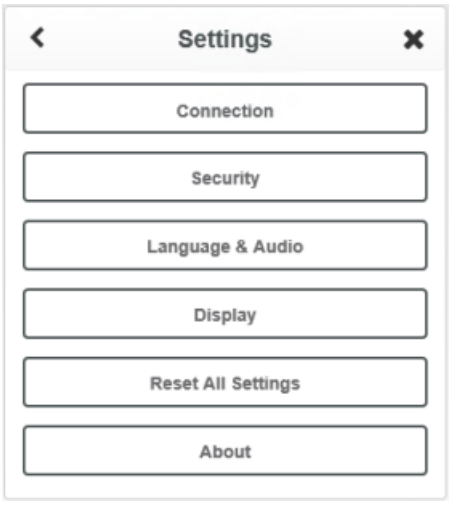
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To connect to the ViewApp you want to view:

1. Click the **Settings** icon .

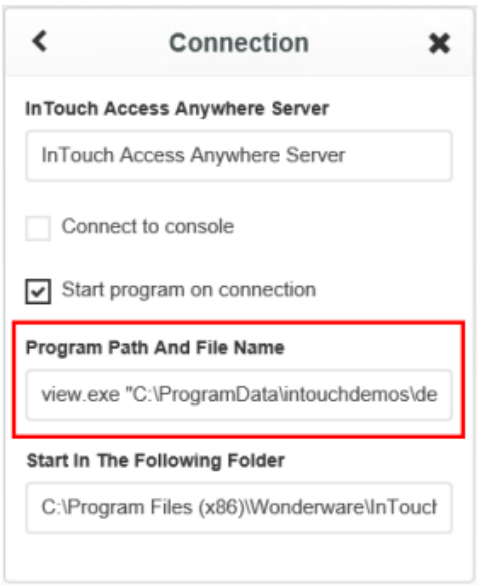


The **Advanced Settings** dialog appears.



2. Click the **Connection** option.
3. In the **Program path and filename** field, enter **view.exe** followed by the path to the InTouch application you wish to start on the server, enclosed within quotation marks (as shown below). If the path is not supplied, then WindowViewer will start with the last application it was running (as specified by the per-user win.ini file).

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The InTouch installation path is populated by default in the **Start in the following folder** field.

Scenario 2: Secure Gateway node shows a list of InTouch applications

In this scenario, you will be directed to a page that looks similar to the start page for accessing an InTouch Access Anywhere Server. In this case, select the application you want to open in WindowViewer, then click **Connect**.

Configure a secure gateway node to host a single InTouch Access Anywhere server

If your Secure Gateway points to a single InTouch Access Anywhere Server, you do not need to specify the name of the server in the start.html file. Use the following procedure to show a list of InTouch applications available from a single InTouch Access Anywhere Server.

To display a list of InTouch applications from a single InTouch Access Anywhere Server

1. On Node 2, where the Secure Gateway is installed, go to the start.html file in the following directory:

   C:\ ProgramFiles (x86)\Wonderware\InTouch Access Anywhere Secure Gateway\InTouch Access Anywhere Secure Gateway\WebServer\Access Anywhere

2. Rename the start page to StartOriginal.html.
3. On Node 1, where the InTouch Access Anywhere Server is installed, copy the Start.html file from:

   C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere\start.html

4. On Node 2, paste the Start.html from Node 1 in the following directory:

   C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Secure Gateway\InTouch Access Anywhere Secure Gateway\WebServer\Access Anywhere

You can now see the Application Name list with all InTouch applications available on the single Node 1 InTouch Access Anywhere Server.

Configuring the Secure Gateway Node to Show a List of InTouch Applications

You can display a list of your InTouch applications on the InTouch Access Anywhere Server that can be accessed through a Secure Gateway.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

1. From Node1, where the InTouch Access Anywhere Server is installed, clone (copy and paste) the Start.html page located in the following directory:

   <InTouch Access Anywhere Server installation folder>\WebServer\AccessAnywhere\

2. Rename the cloned file and go to Node2. Paste the file under <InTouch Access Anywhere Secure Gateway installation folder>\Ericom Secure Gateway\WebServer\AccessAnywhere\ folder on the Gateway node (i.e. Node 2).

Note: The start page can be renamed to any valid file name, but for better readability and compatibility, we recommend prefixing the file name with the InTouch Access Anywhere server name. For example, if the server name is Master01, the start page should be renamed to Master01_start.html.

1. Open Start.html and locate the following html element:

```
<select id="ITAAServerList" name="ITAAServerList" style="visibility:hidden">
              <!-- A sample option element
              <option ServerName="Master01" IPAddress="xx.x.xx.xx"
StartPageName="Master01_Start.html"/> -->
            </select>
```

2. Add an option element under the select element (an example is given) and update the property values as follows:
   - The ServerName property value should be set to InTouch Access Anywhere server name (Node1 in our example).
   - The IPAddress property value should be the IP Address of the server. Setting an IP value will allow the page to be accessed when you use IPAddress instead of ServerName.
   - The StartPageName property value should be set to the start page name from step 2 above.
3. Save the changes.

Now you can see the **Application Name** list with all InTouch applications available on the InTouch Access Anywhere Server node.

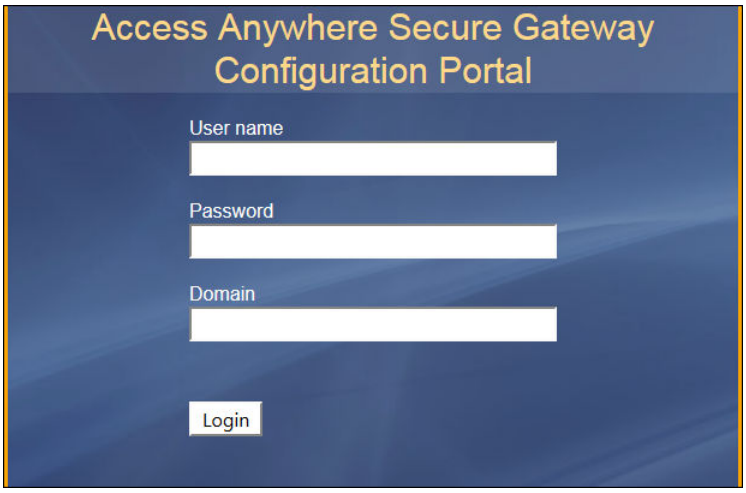
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About the InTouch Access Anywhere Secure Gateway Configuration Portal

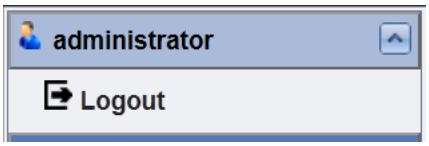
The InTouch Access Anywhere Secure Gateway includes a Configuration Portal to enable an administrator to change any related settings. To access the Configuration Portal page, use a web browser and navigate to the Secure Gateway's configuration URL:

- https://<SG-server-address>:<port-number>/admin

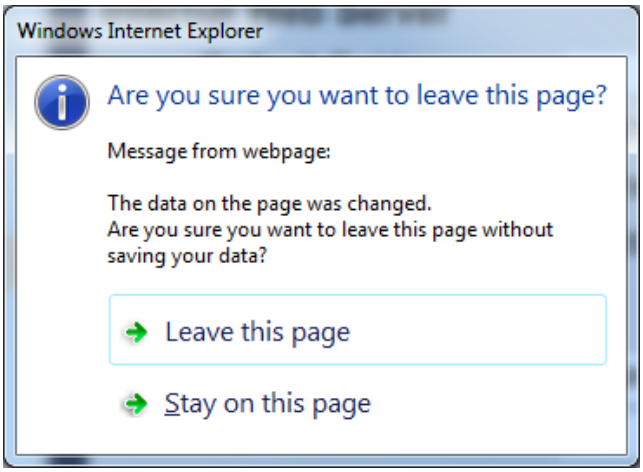
Accessing the Configuration Portal is restricted to only members of the local Administrators group of the InTouch Access Anywhere Secure Gateway server. All log ons are audited in the Secure Gateway log file. Administrators are strongly encouraged to enforce a strong password policy for Secure Gateway administrators.



To log out of the Configuration Portal, click **Logout**.



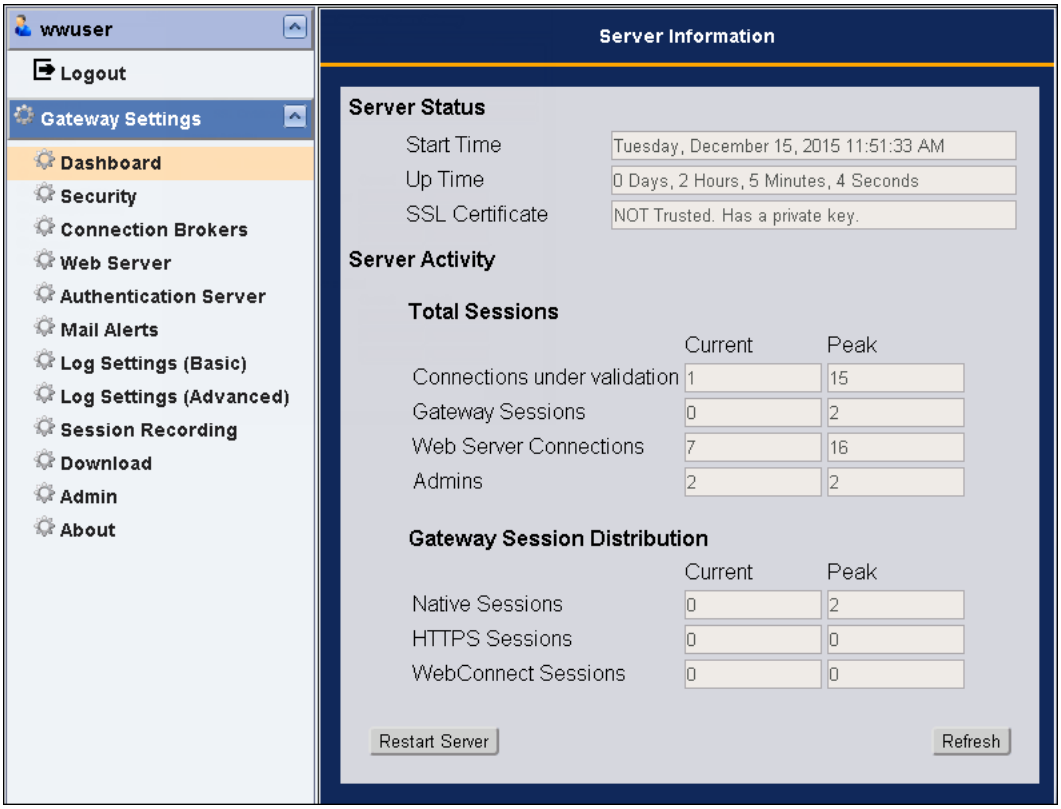
After making changes to any settings, click **Save**. If a different page is selected and the settings are not saved, a warning dialog will appear. Click **Leave this Page** to continue and cancel any changes. Click **Stay on this page** to return to the current page to save changes.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Secure Gateway configuration dashboard

Secure Gateway **Configuration Dashboard** displays useful statistics related to the Secure Gateway operation. Open this page to view server uptime, SSL certificate status, session activity, and to restart the Secure Gateway Server service.

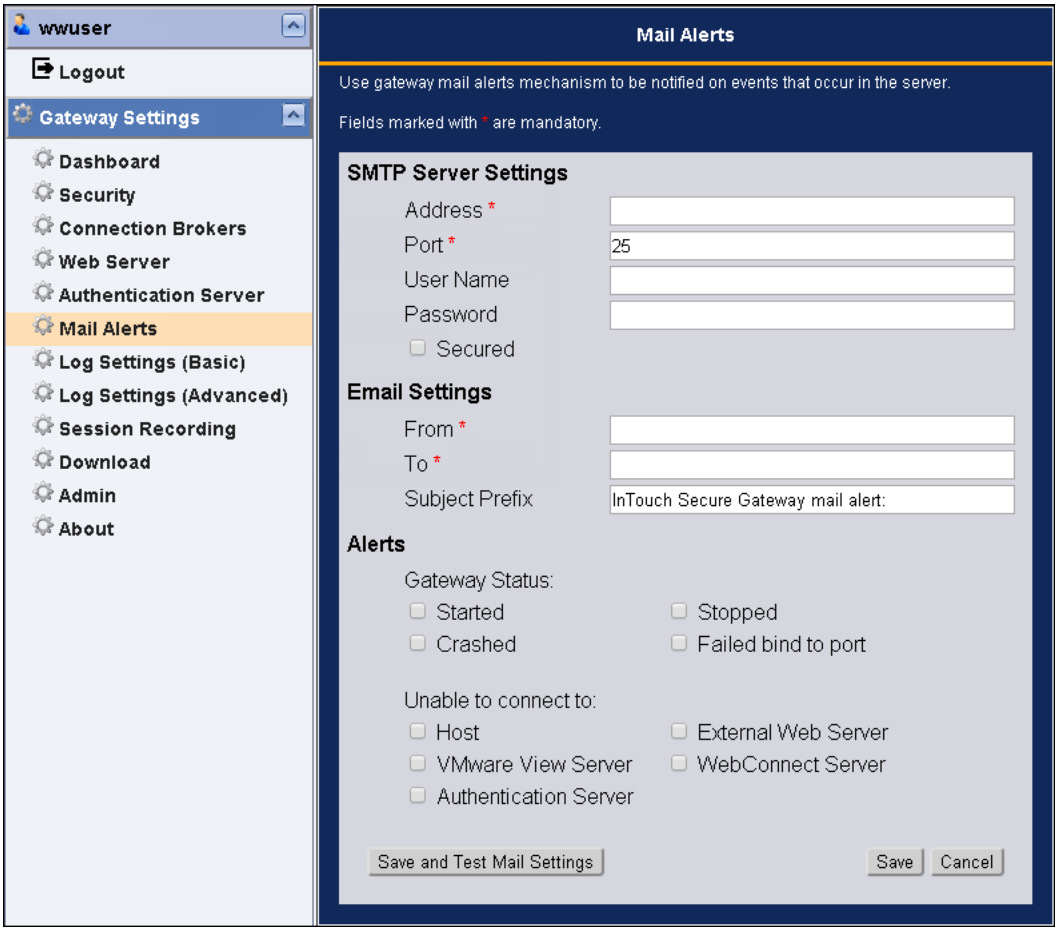


Configure Secure Gateway e-mail alerts

Secure Gateway can be configured to send e-mail alerts when specified system events occur. To configure mail alerts, enter the SMTP information of the e-mail server. Then, check the events that trigger an e-mail alert.

Click **Save** or **Save and Test Mail Settings** to apply the configuration.

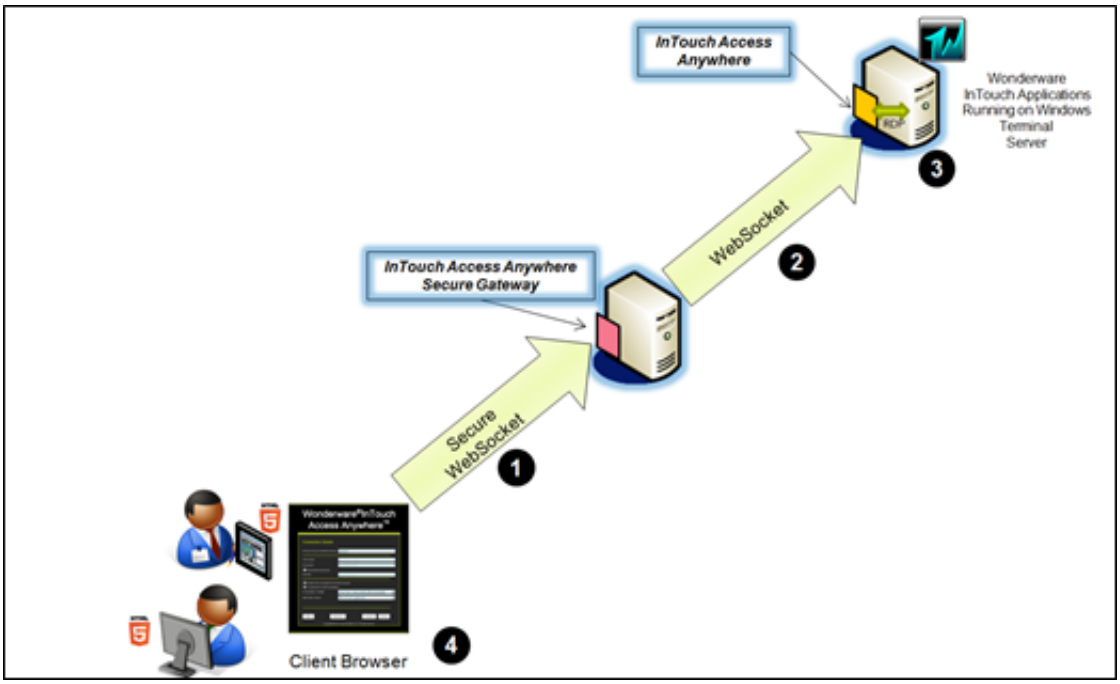
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



InTouch Access Anywhere HTML5 client configuration

InTouch Access Anywhere can use the Secure Gateway to provide secured connections between HTML5 web clients and InTouch Access Anywhere servers. The following diagram shows how these components work together.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



In this configuration, a client browser always establishes a secure WebSocket connection to the Secure Gateway. The Gateway then establishes a WebSocket connection to the InTouch Access Anywhere server.

Whether the WebSocket connection between the Gateway and the InTouch Access Anywhere server can be secured or not is based on a configuration setting in the InTouch Access Anywhere client (check the box marked **Enable SSL** for the InTouch Access Anywhere web configuration).

Configure the Access Anywhere server to work with a secure gateway

The InTouch Access Anywhere server includes a set of **Security** options that indicate a Secure Gateway should be used and the address of the computer hosting the Secure Gateway.

The Access Anywhere server always establishes a secure WebSocket connection to the Secure Gateway. The Secure Gateway then establishes a WebSocket connection to the Access Anywhere server.

To configure the Access Anywhere server to work with a Secure Gateway

1. Log on to the computer hosting the Access Anywhere server.
2. Show the Start page by entering the following URL in a web browser.

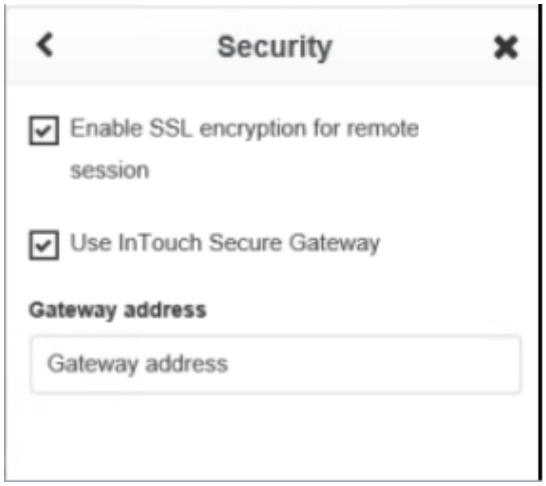
   https://localhost:8080

3. Click the Advanced Settings icon ⋮ on the Start page and select the **Security** option.

   The WebSocket connection between the Secure Gateway and the Access Anywhere server can be secured by selecting the **Enable SSL encryption for remote session** option.

4. Select **Use InTouch Secure Gateway** and enter the IP address or name of the server hosting the Secure Gateway in the **Gateway address** field.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



About security whitelists

You can configure two types of whitelists:

- End-user Address and Range
- Target Host Address and Range

The target host whitelist is enabled by default, while the client whitelist is disabled by default. When a whitelist is enabled, a list of IP addresses must be specified.

To enable a type of whitelist, change the enabled setting from "false" to "true". For example:

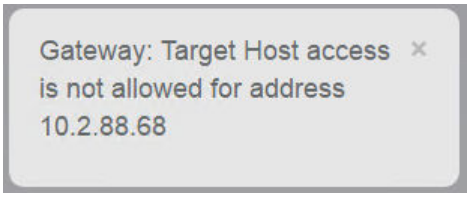
```
<add key="ClientWhitelistByIPAddressesEnabled" value="false"/>
<add key="ClientWhitelistByIPAddressesEnabled" value="true"/>
```

Note: The enabled setting is set to "true" by default for InTouch Access Anywhere.

Addresses are entered in the standard format, for example 10.2.88.1, and are separated by semicolons (;).

Address ranges are defined using a lower IP, the character "-", and the upper IP. For example: 10.2.88.1-10.2.88.5

The IP addresses of each Access Anywhere Server must be configured in the EricomSecureGateway.Config file, or you will be prompted with the below error message:



Note: This is an example IP address. This value will be associated with the InTouch Access Anywhere host you are trying to connect to.

The values below provide an example for how each type of whitelist would be configured in the EricomSecureGateway.exe.config file:

- End-user Address and Range:
  `<add key="ClientWhitelistAllowedIPv4Addresses" value="10.2.88.1-10.2.88.5;10.2.88.10" />`

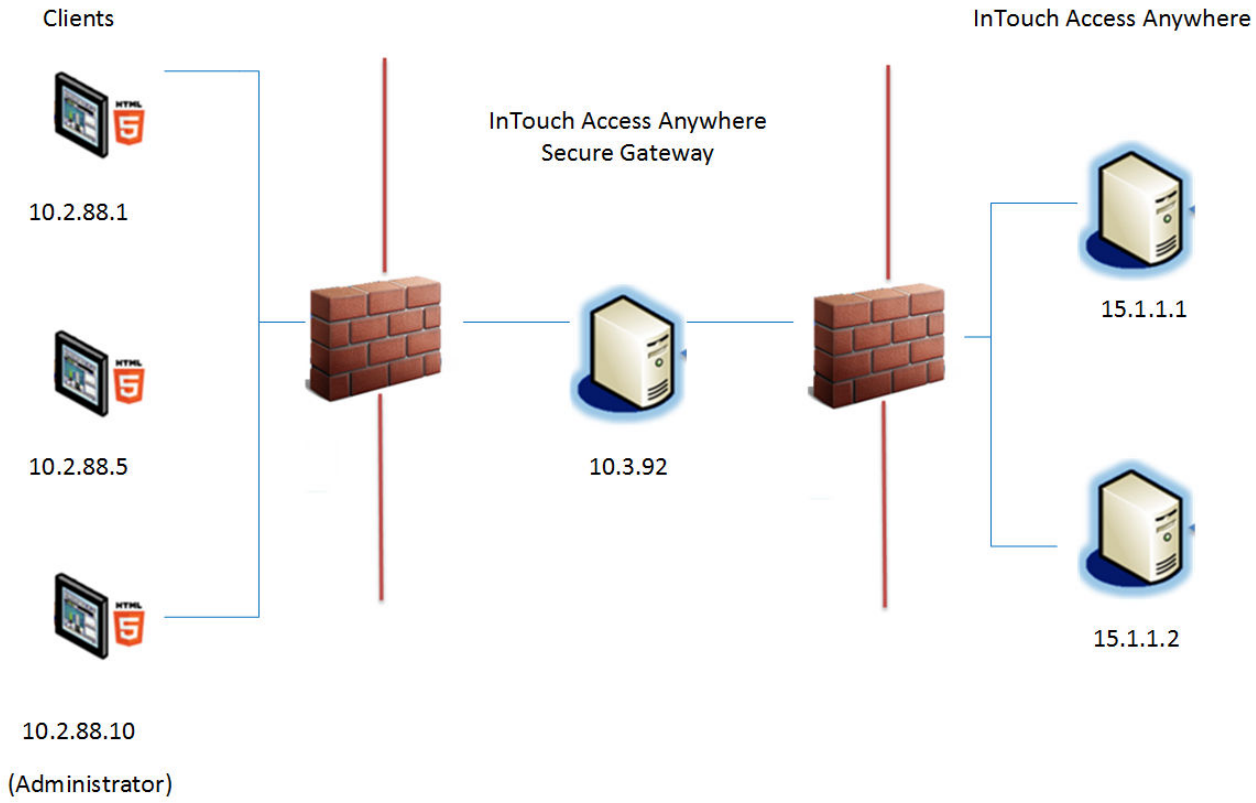
  The IP Address of each client node is included.
- Target Host Address and Range

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

```
<add key="TargetHostWhitelistAllowedIPv4Addresses" value="15.1.1.1;15.1.1.2" />
```

The IP Address of each InTouch Access Anywhere Server(s) connecting through the Gateway.

The following diagram illustrates an example whitelist security configuration for clients, the Gateway, and the Access Anywhere Server. The IP addresses of all whitelisted clients are represented.



The list of all the configuration options is:

```
<Visitor>
<add key="HandshakeTimeoutSeconds" value="60" />
<add key="ClientWhitelistByIPAddressesEnabled" value="false" />
<add key="ClientWhitelistAllowedIPv4Addresses" value="" />
<add key="ClientWhitelistAllowedIPv6Addresses" value="" />
<add key="TargetHostWhitelistByIPAddressesEnabled" value="false" />
<add key="TargetHostWhitelistAllowedIPv4Addresses" value="" />
<add key="TargetHostWhitelistAllowedIPv6Addresses" value="" />
</Visitor>
<Admin>
<add key="InactivityTimeoutMinutes" value="5" />
<add key="WhitelistByIPAddressesEnabled" value="true" />
<add key="WhitelistAllowedIPv4Addresses" value="" />
<add key="WhitelistAllowedIPv6Addresses" value="" />
</Admin>
```

Note: ClientWhitelistByIPAddressesEnabled and the Admin whitelist settings existed in previous versions as "LockdownAllowed****Addresses, if these settings are currently configured, simply copy the parameters to the new values.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure the origin header parameter for whitelist security

You can configure a parameter to whitelist the origin header field in incoming HTTP requests to the InTouch Access Anywhere Gateway. The Gateway will check for this parameter upon connection request to the host. If the "origin" HTTP header exists in the connection request, it will verify that it is in the list of trusted addresses. If there is no match, the Gateway will deny the connection request to the host.

Use the following parameter to whitelist the origin header field:

```
<add Key="OriginHttpHeaderWhitelistAddresses" value=""/>
```

where value is the Gateway IP address or node name. For example:

```
<add Key="OriginHttpHeaderWhitelistAddresses" value="http://10.3.92"/>
```

or

```
<add Key="OriginHttpHeaderWhitelistAddresses" value="http://Gatewaynodename"/>
```

If there is a load balancer or proxy server in front of the Gateway, then the value address in this parameter refers to the load balancer or proxy server. In this case, the address can be any URL. For example:

```
<add Key="OriginHttpHeaderWhitelistAddresses" value="http://URL"/>
```

For detailed information about HTTP origin header specifications, see section 7 and 8 of the Internet Engineering Task Force

Configure session cookie timeout

A session cookie is generated when browser clients connect to the InTouch Access Anywhere Secure Gateway.

Use the following parameter to configure the session cookie timeout in the EricomSecureGateway.Config file:

```
<Property name="ClientSessionCookieTimeoutMinutes" type="int" value="60" />
```

Note: The default timeout period is sixty minutes. Do not set this value to 0. A value of 0 disables the cookie timeout interval.

If you try to connect by means of a Websocket or HTTPS (if enabled) and the cookie has expired, the connection will be rejected. You will need to reload the page to re-attempt the login.

The following details apply to the session cookie lease:

- A cookie is cached in the InTouch Access Anywhere Gateway the first time an end-user's browser requests a page.
- The cookie lease duration is defined based on "ClientSessionCookieTimeoutMinutes" value.
- The lease is maintained on the Access Anywhere Server side, not in the browsers, so all browsers are treated as a single browser from your device.
- The cookie value and lease are per client (IP address), so multiple browsers on the same device will use the same cookie value and the same lease.
- The cookie lease duration is not extended each time a page is retrieved. A cookie lease expires only after the configured duration.

  This cookie lease duration requires reloading the page after each expiration to contact the Gateway.

Advanced Configuration

All configurable settings related to the Secure Gateway can be found in the EricomSecureGateway.exe.config file. This is a text file that can be modified with a text editor. The configuration settings are also defined in the section About a secure gateway built-in authentication server.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Changing parameter values marked as "Reloadable" do not require a service restart. "Not Reloadable" parameters only become effective after restarting the InTouch Access Anywhere Secure Gateway service.

High Availability

To provide high availability of the Secure Gateway, it is recommended that you install two or more Secure Gateways and use a third-party redundant load balancer to manage access.

The load balancer will provide one address for end users. As requests arrive at the load balancer, they are redirected to an available Secure Gateway based on built-in weighting criteria. A basic round-robin load balancer can be used, but it may not detect whether a Secure Gateway is active or not.

Restricting Access To and From a Secure Gateway

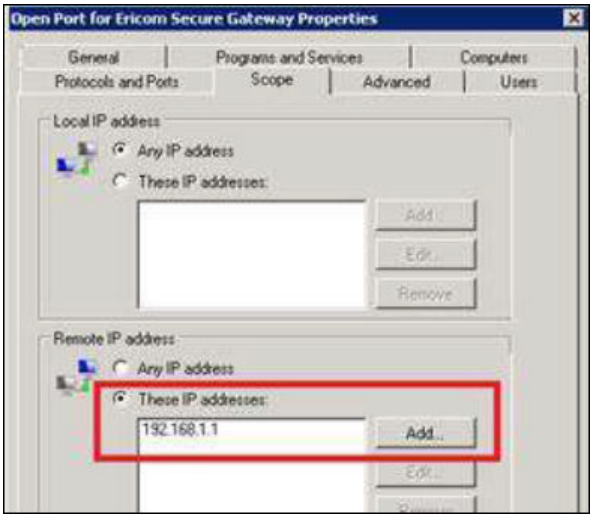
Ericom documentation mentions "Terminal Servers" here too. I assumed those should be removed.

Use the Windows Firewall Scope rules to restrict incoming connections to the Secure Gateway server.

To restrict incoming connections to the Secure Gateway

1. Access the Port rules for Secure Gateway.
2. Click the Scope tab.
3. In the Remote IP address section, click Add.
4. Enter the IP address(es) from which you wish to allow connections.

In the example below, only connections originating from 192.168.1.1 can connect to the Secure Gateway.



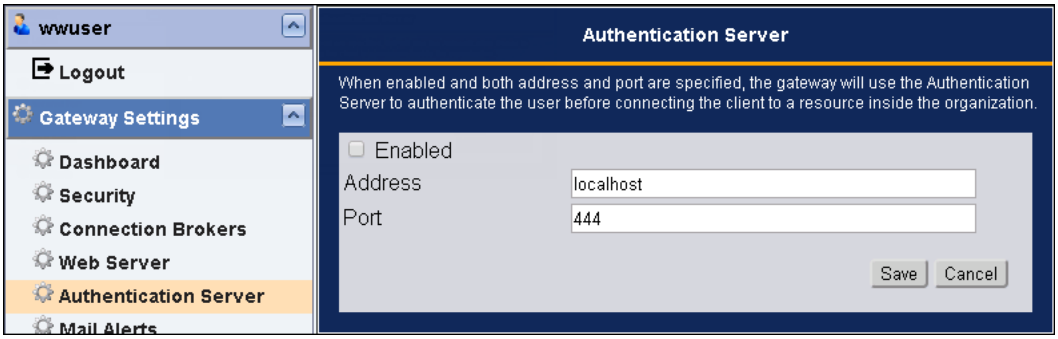
About a secure gateway built-in authentication server

The Secure Gateway includes an Authentication Server that provides a layer of security by authenticating end-users before they can access any internal resource. The Authentication Server is disabled by default and should be installed on a computer on the safe side of the firewall that is a member of the domain and which is employed to authenticate users.

Note: The Authentication Server can only be configured for one domain at a time.

Use the Secure Gateway **Configuration** page to modify some of the settings of the Authentication Server:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Other configuration settings are specified in the EricomSecureGateway.Config file, which is located at

`C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Secure Gateway\InTouch Access Anywhere Secure Gateway`

The user configurable settings are located under the **Authentication Server** section of the EricomSecureGateway.Config file and defined in the following table.

Setting	Description
Enabled	Boolean value to enable the Authentication Server or not. True enables the Authentication Server. The default is False.
Address	The IP address of the computer hosting the Authentication Server. Localhost is the default.
Port	This is the port on which the Authentication Server listens. Make sure that no other services on the system are using the same port. A port conflict will interfere with the operation of the Authentication Server. The default port is 444.
CertificateDnsIdentity	The connection between the Secure Gateway and the Authentication Server is secured. In case the Authentication Server is not using its default certificate, this parameter must be updated to include the DNS identity of the alternate certificate.
MaxClockSkewMinutes	The maximum difference in minutes between the clocks of the Secure Gateway and the Authentication Server. The default is 180.
KeepAliveFreqSeconds	The keep alive interval in seconds that maintains the connection between the Authentication Server and the Secure Gateway. The default is 30.

When an Authentication Server is enabled, only domain users will be able to authenticate. Local system users (such as Administrator) will not be able to log on through the Authentication Server.

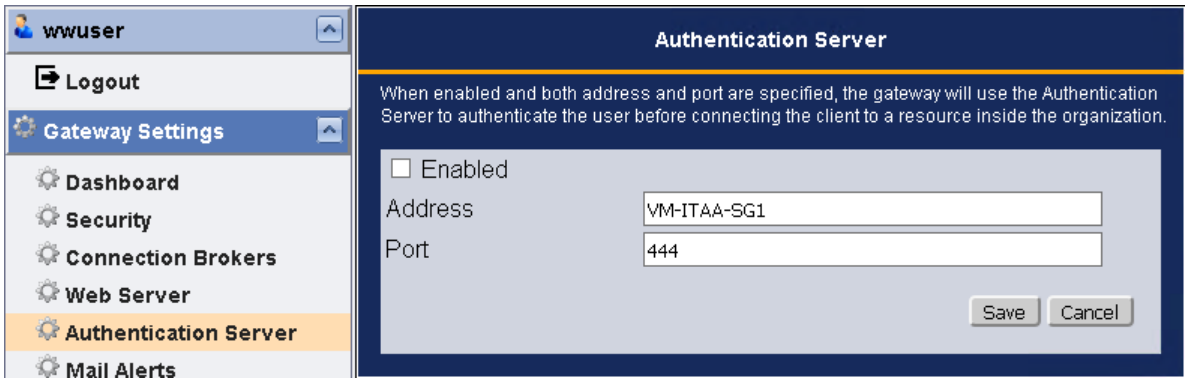
Disable authentication server when access is through a broker

When all access is through a connection broker and not from any stand-alone clients, the Authentication Server should be disabled, and the "broker only mode" enabled.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To disable the Authentication Server

1. At the **Authentication Server** page of the Secure Gateway portal, clear the **Enabled** check box to disable the Authentication Server.



2. Make the following changes to EricomSecureGateway.Config file:

   a. Under `<AuthenticationServer>`, change `<add key="Enabled" value="true"/>` to `<add key="Enabled" value="false"/>`

```
<Section name="AuthenticationServer">
<Property name="Enabled" type="bool" value="false" />
```

   b. Under `<Security>`, change `<add key="ConnectionBrokerOnlyMode" value="false"/>` to

```
<Section name="Security">
<Property name="CertificateFindBy" type="X509FindType" value="FindByExtension" />
<Property name="CertificateFindValue" type="string"
value="1.2.840.113556.1.8000.2554.57748.52896.21682.18417.45066.8514989.679433.2" />
<Property name="ConnectionBrokerOnlyMode" type="bool" value="true" />
```

Making these changes prevents any connections from stand-alone clients through the Secure Gateway. All users will log in only through a connection broker.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About InTouch Access Anywhere Secure Gateway ports and SSL certificates

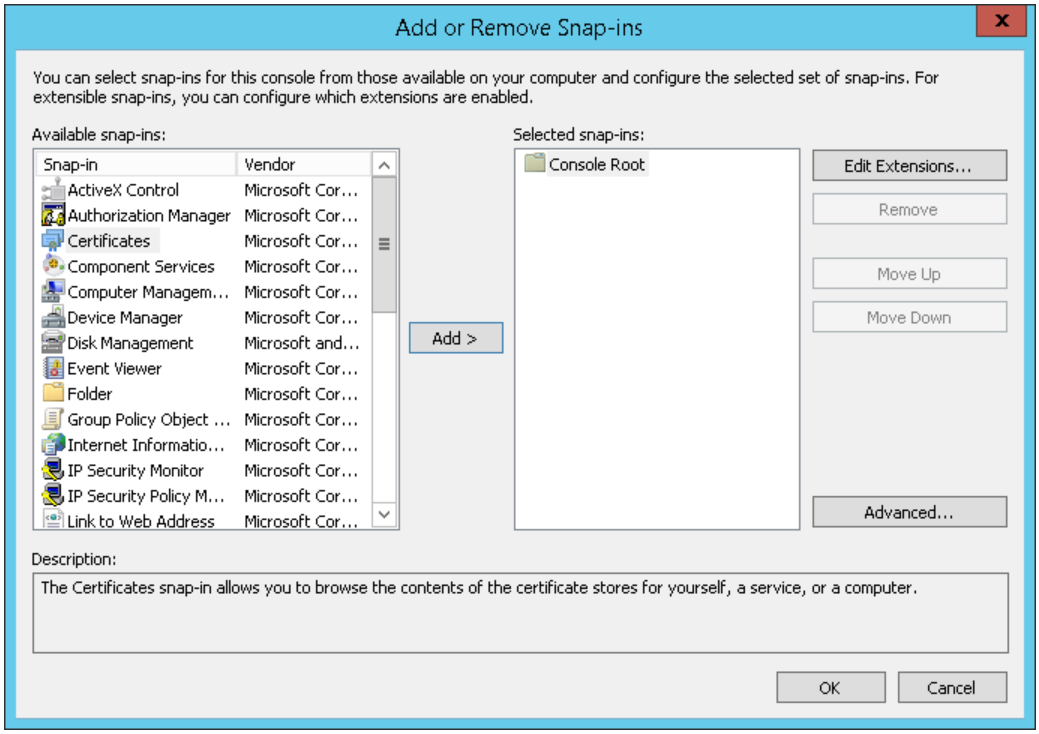
The InTouch Access Anywhere Secure Gateway includes a self-signed certificate. Some web browsers may show a security warning when a self-signed certificate is detected. To remove the warning, install a trusted certificate purchased from a trusted certificate authority (for example, VeriSign).

Important: The signed certificate must have a private key associated with it. A .CER file may not have a private key. Use a signed certificate that includes a private key, which usually has a .PFX extension.

The Secure Gateway uses the certificate in the Windows Certificate Store (Computer Account), which is accessible using the Microsoft Management Console (MMC).

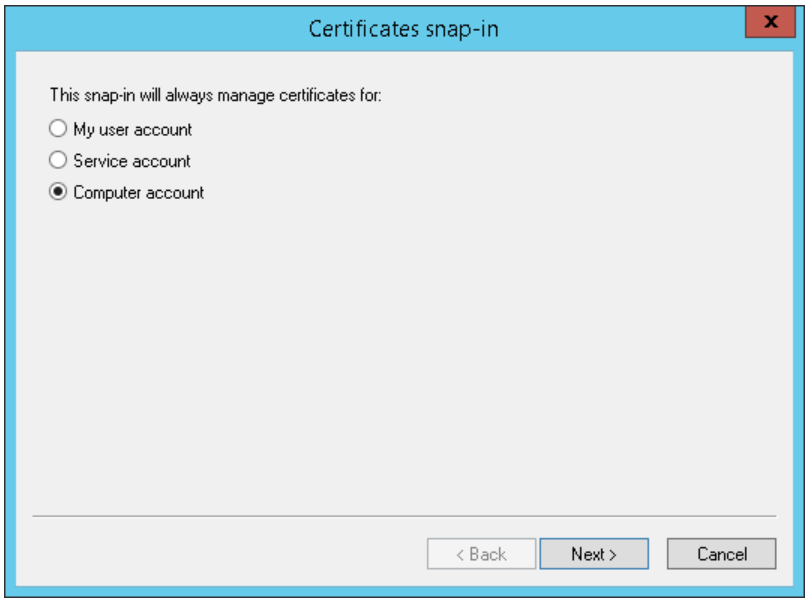
To add, view, or modify certificates

1. Log on as an administrator to the computer hosting the Secure Gateway.
1. From the Windows **Command Prompt**, run the mmc.exe command to show the MMC.
2. Select the **File** option from the menu bar and select **Add Remove Snap-in** to show the **Add or Remove Snap-ins** dialog box.



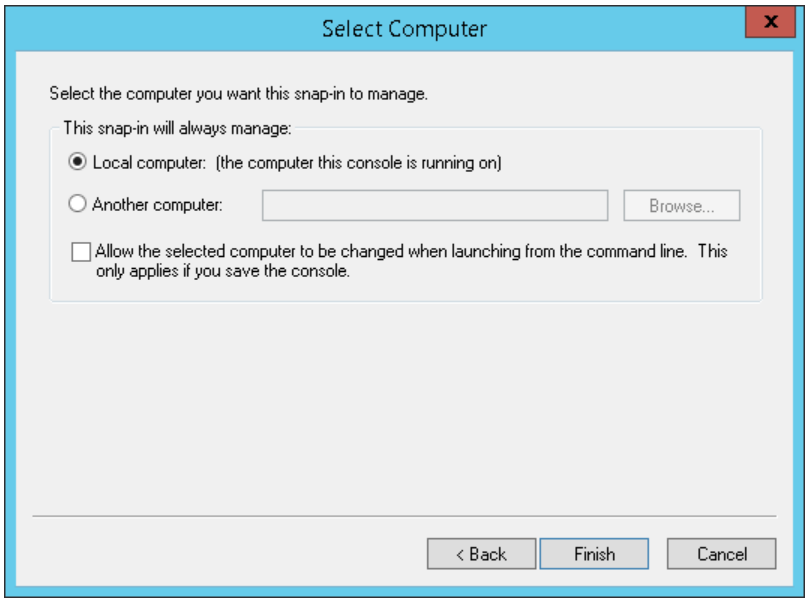
3. Select **Certificates** from the **Available snap-ins** area and select **Add**.
4. Select **Computer Account** from the **Certificates snap-in** dialog box and click **Next**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



The **Select Computer** dialog box appears with options to select a computer account.

5. Select **Local Computer**.



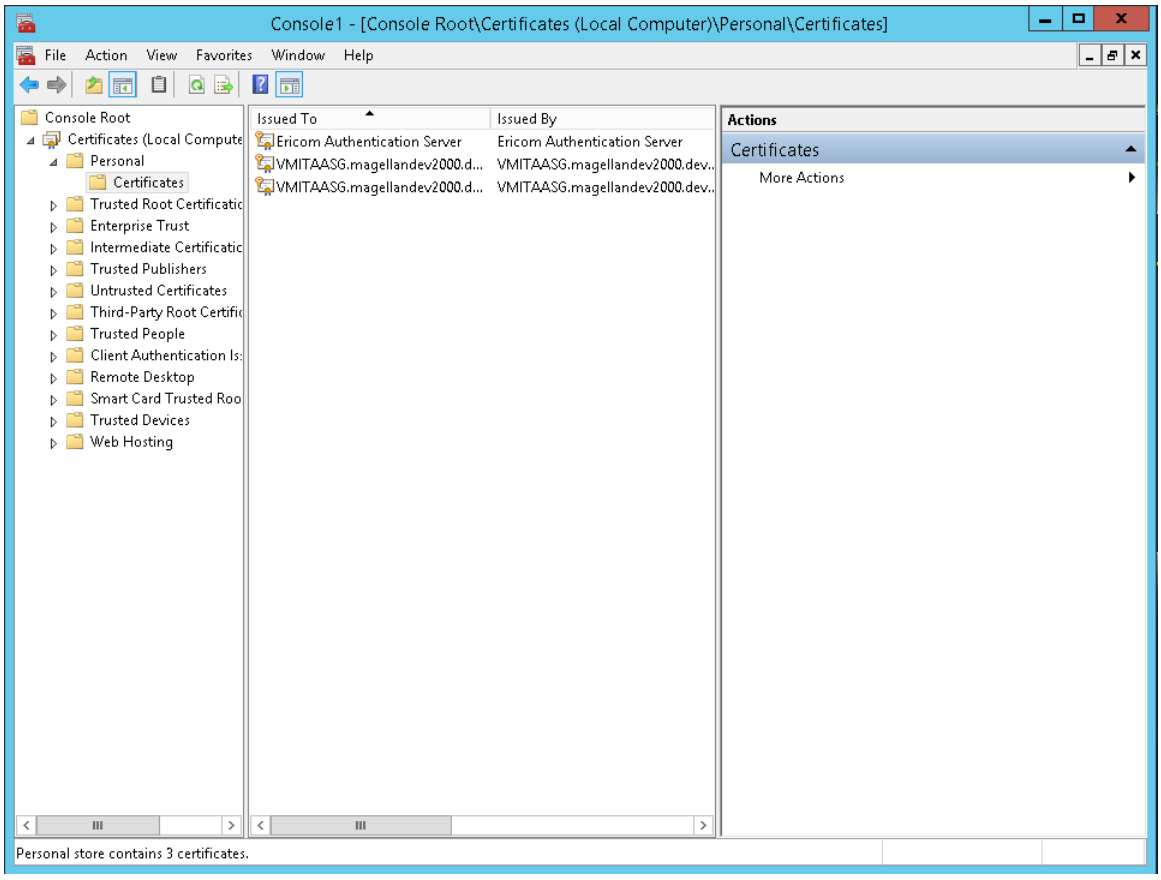
1. Click **Finish** and then **OK**.

   The Console Root shows Certificates (Local Computer) option.

1. Select the icon to the left of the **Certificates (Local Computer)** option to expand the list of sub options.

1. Browse **Certificates | Personal | Certificates** folder to view the available certificates that can be used by the Secure Gateway.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. If a trusted certificate is used with Secure Gateway, place it in the same location as the Secure Gateway **Certificates | Personal | Certificates**).

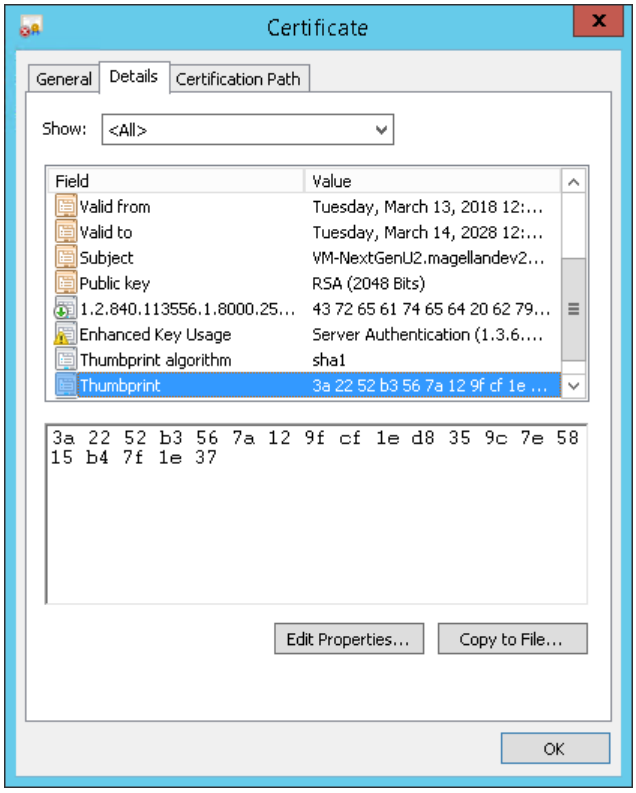
1. Browse the **Certificates | Personal | Certificates** folder of the MMC to show a list of certificates.

1. Double-click on the trusted certificate that you want to use with the Secure Gateway.

2. Select the **Details** tab and highlight **Thumbprint**.

   The Thumbprint value appears beneath the list of certificate properties.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. Select the entire thumbprint value.
4. Press CTRL+C to copy it.

   The Thumbprint can also be manually typed in.
5. Click **OK** to close the dialog.
6. Open the EricomSecureGateway.Config file, which is located in the following folder of the computer hosting Secure Gateway:

   C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Secure Gateway\InTouch Access Anywhere Secure Gateway
7. Locate the Security section of the file.

   ```
   <Section name="Security">

   <Property name="CertificateFindBy" type="X509FindType" value="FindByThumbprint" />

   <Property name="CertificateFindValue" type="string"
   value="3A2252B3567A129FCF1ED8359C7E5815B47F1E37" />
   ```
8. Ensure the value of the CertificateFindBy property value is set to FindByThumbprint.
9. Delete the existing Thumbprint from the CertificateFindValue property value field.
10. Press CTRL+V to paste the new Thumbprint in the value field of the CertificateFindValue property.

    All blank spaces in the thumbprint are removed after pasting it as the value of the CertificateFindValue property.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

11. Save the file and the new Thumbprint will be used. Restarting the Secure Gateway service will apply the new certificate immediately.

    Note: The DNS address of the Secure Gateway server must match the certificate name. If it does not, a "Connection failed" error message will appear upon attempting a connection.

Add a certificate thumbprint to the EricomSecureGateway.Config file

Another way to add a trusted certificate thumbprint is to dump the certificate values and copy the identify thumbprint to the EricomSecureGateway.Config file. When you are using the extension identity of a certificate, the `CertificateFindBy` property value of the EricomSecureGateway.Config file should be set to `"FindByExtension"`.

To add a certificate thumbprint to the EricomSecureGateway.Config file

1. Place a X.509 certificate at a known location of the computer running Secure Gateway.
2. Open a Command prompt window and enter the certutil command in the following form:

   ```
   C:\Temp>certutil -dump CertificateName.cer
   ```

   where `CertificateName` is the actual name of the certificate.
3. Find the Certificate Extensions 2 location of the output from the certutil command.

   The identify string appears immediately beneath Certificate Extensions 2

   ```
   Certificate Extensions: 2
   ```

   ```
   1.2.840.113556.1.8000.2554.57748.52896.21682.18417.45066.8514989.679433.2: Flags = 0, Length = 1a
   ```
4. Copy the identity string.
5. Edit the EricomSecureGateway.Config file and locate the Security section of the file.

   ```
   <Section name="Security">
   ```

   ```
   <Property name="CertificateFindBy" type="X509FindType" value="FindByThumbprint" />
   ```

   ```
   <Property name="CertificateFindValue" type="string" value=
   ```
6. Change `FindByThumbprint` to `FindByExtension` and copy the identity string as the value of the `CertificateFindBy property.`

   ```
   <Section name="Security">
   ```

   ```
   <Property name="CertificateFindBy" type="X509FindType" value="FindByExtension"/>
   ```

   ```
   <Property name="CertificateFindValue" type="string" value=<Certificate Identity Thumbprint Goes Here> />
   ```

   Example:

   ```
   <Section name="Security">
   ```

   ```
   <Property name="CertificateFindBy" type="X509FindType" value="FindByExtension"/>
   ```

   ```
   <Property name="CertificateFindValue" type="string" value="1.2.840.113556.1.8000.2554.57748.52896.21682.18417.45066.8514989.679433.2"/>
   ```

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

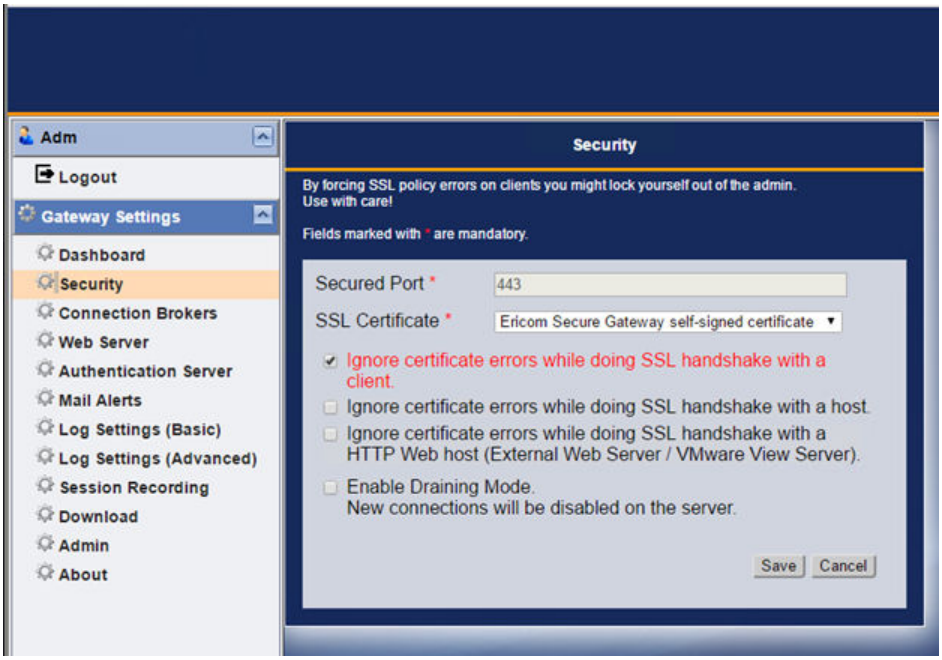
Configure a secured port and SSL certificate

In the Secure Gateway Configuration dashboard, select the **Security** page to modify the port and SSL certificate that will be used by the Secure Gateway.

Note: Before configuring the port, make sure it is not currently in use.

From the **SSL Certificate** field, select the desired SSL certificate to be used by InTouch Access Anywhere Secure Gateway. It is strongly recommended to use a trusted certificate when the InTouch Access Anywhere Secure Gateway is used in production. Verify whether the selected certificate is trusted.

Configure the desired security options. The **Ignore certificate errors while doing SSL handshake with a client** option is selected by default.



The options are as follows:

Note: The recommended deployment method is to have none of the above options checked. The most secure method is to leave all security features enabled.

- **Ignore certificate errors while doing SSL handshake with a client**: this option refers to the certificate verification between any clients connecting to the InTouch Access Anywhere Gateway.

  If this option is checked, you will not be prompted with a certificate error if a connecting client's trusted certificate is not recognized on the Access Anywhere Gateway node

- **Ignore certificate errors while doing SSL handshake with host**: this option refers to the certificate verification between the Access Anywhere Gateway and the Access Anywhere Server. If the Access Anywhere Server does not have a trusted certificate recognized on the Access Anywhere Gateway node, the connection will be rejected.

  If this option is checked, you will not be prompted with a certificate error if the Access Anywhere Server's certificate is not recognized on the Gateway node.

- **Enable Draining Mode**: this option will disable new connects being made to the Access Anywhere Server, and old connects will be closed out.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure Failover Gateways

Multiple Secure Gateways can be configured as a failover chain in the **InTouch Access Anywhere** web client. A failover chain provides improved reliability with redundant Secure Gateways. Alternate Gateways automatically become active when the primary Gateway is unavailable. If the connection to the first Secure Gateway on the list fails, the request is redirected to the next server on the list.

To specify a failover list of Secure Gateways, enter each gateway address separated by a semicolon.

The following list of servers:

Us-bl2008r2;securegateway.domainname.com;192.168.0.3:4343

- The primary gateway is Us-bl2008r2 over port 443.
- The second Secure Gateway is securegateway.domainname.com over port 443.
- The third Secure Gateway is 192.168.0.3 over port 4343 (any port value other than 443 needs to be explicitly specified).

Note: Maintain uptime for the servers at the front of the list to ensure the fastest logon time. If the primary server is unavailable, end-users will experience delays as the log on process must wait for the primary server to time out before attempting to connect to a failover server.

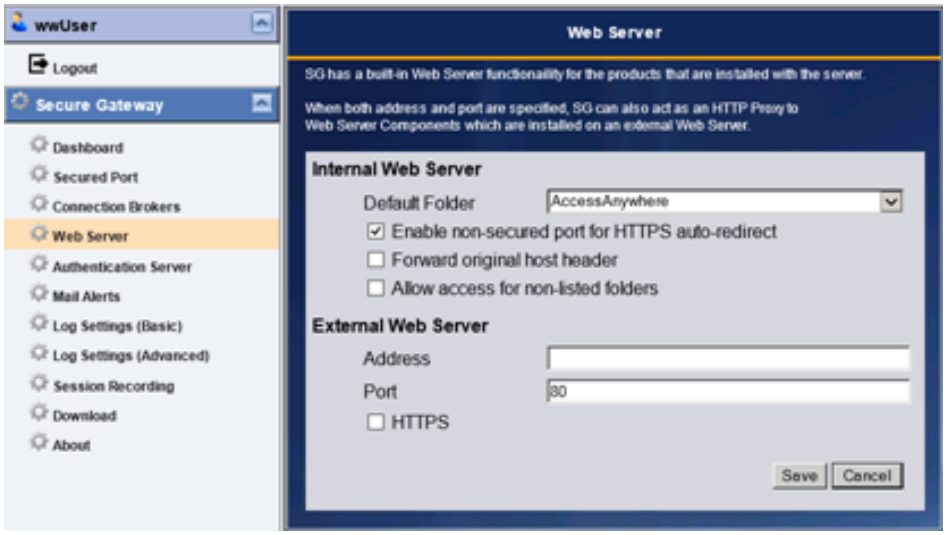
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Configure the built-In InTouch Access Anywhere web server

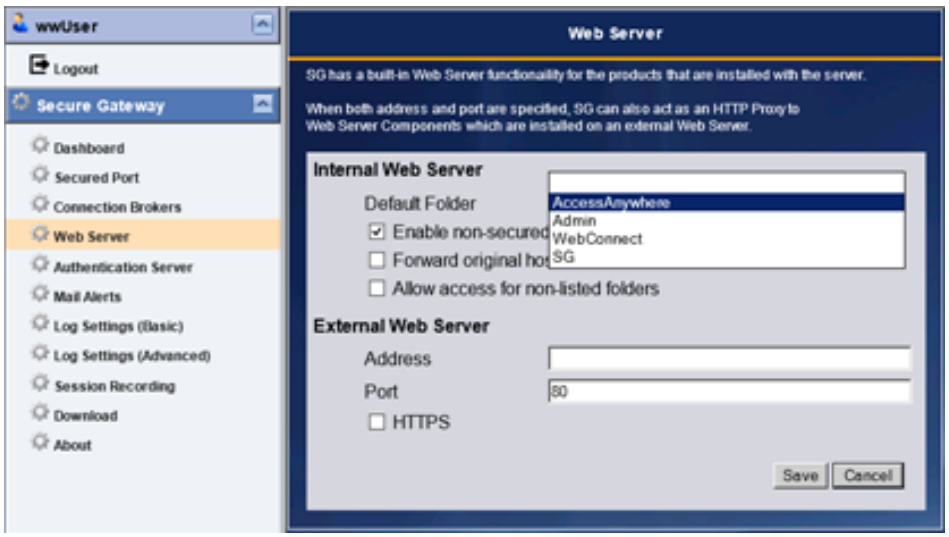
The Secure Gateway has a built-in web server to host web pages for InTouch Access Anywhere. The built-in Web server cannot be disabled and always listens on the Secure Gateway port.

To configure the Web server

1. Open the **Configuration** tool and show the **Web Server** page.



2. Click the **Default Folder** drop down list to select the default URL for the built-in web server.
3. Click **Save**.



When the user goes to the root path of the URL, the selected component will be used. For example, if InTouch Access Anywhere Server is selected, when the user navigates to https://<sg-server-address>:<port-number>/ the URL will automatically redirect to:

https://<sg-server-address>:<port-number>/AccessAnywhere/start.html

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: The Secure Gateway could technically be used to host non-related pages, but this is not officially supported. Hosted web pages should be of basic static content.

About the InTouch Access Anywhere Secure Gateway web server proxy

The InTouch Access Anywhere Secure Gateway also has a built-in Web server proxy.

Note: Using the Secure Gateway to proxy to pages other than InTouch Access Anywhere is not officially supported.

Connecting to the Web Server

To connect to an InTouch Access Anywhere server available through the Secure Gateway Web server, open a browser and navigate to the desired URL. If a port other than 443 is being used by the Secure Gateway, it must be explicitly stated in the URL. For example: https://myserver:4343/AccessAnywhere/start.html

The following URLs are available by default.

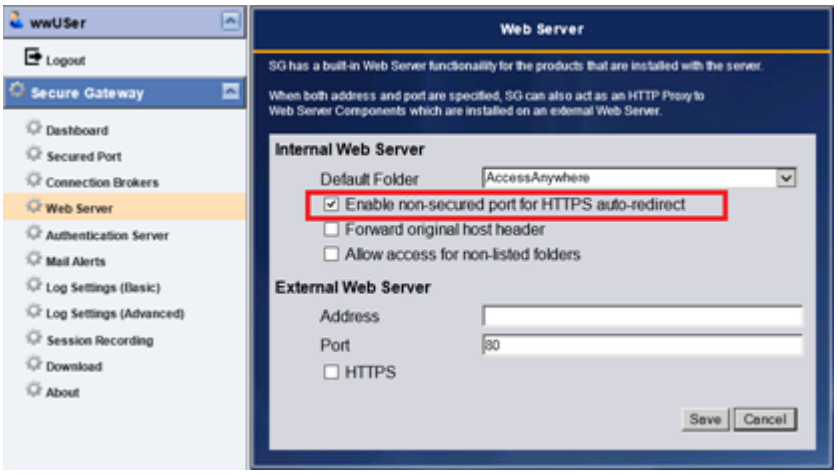
Secure Gateway Welcome Page	https://server:port/ or https://server:port/welcome.html
InTouch Access Anywhere Server	https://server:port/AccessAnywhere/start.html

Enable non-secured port for HTTPS auto-redirect

The InTouch Access Anywhere Secure Gateway Web server listens on port 80 by default. This way, HTTP references to the server will automatically redirect to the HTTPS URL.

Note: This feature only works if the Secure Gateway is listening on port 443. If it is configured to use any other port, the HTTP automatic redirect is not supported.

To enable this feature, select the option: **Enabled non-secured port for HTTPS auto-redirect** (see below).



Configure this feature in the EricomSecureGateway.exe.Config file using: <add key="EnableNonSecuredPortForHttpsAutoRedirect" value="**false**" /> in the <appSettings> section.

Disable HTTP/HTTPS filtering

Certain types of network traffic are blocked by firewalls. Port 443 on most firewalls is initially reserved for HTTP (and HTTPS) based communication. Most firewalls have a rule to filter out any non-HTTP data. Depending on what the Secure Gateway will be routing, HTTP filtering may need to be disabled on the firewall.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The Secure Gateway can proxy various types of traffic. Some are HTTP based and some are not. The only configuration where HTTP filtering does not need to be disabled is when the Web Application Portal and InTouch Access Anywhere are used together.

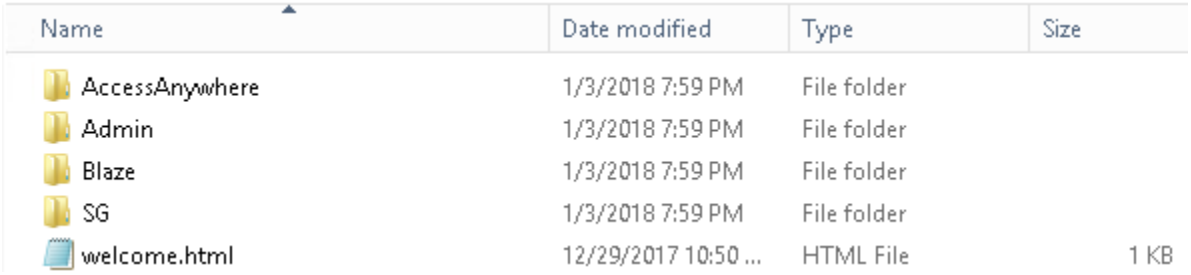
This table denotes the protocol used by a connection method:

Communication Type	Protocol Used
Web Application Portal login	HTTP/HTTPS
Application Zone login	TCP
InTouch Access Anywhere RDP session	HTTPS (Secure Gateway required)

Advanced configuration using the EricomSecureGateway.Config file

Back up the current EricomSecureGateway.Config file before making any changes.

To configure the settings of the built-in Web server, open the EricomSecureGateway.Config file using a text editor. Each folder in the WebServer directory may have a default document assigned for it, and may also be restricted so that end users cannot access it.



For example, the settings below will configure the following:

- Sets the View folder as the default folder
- Sets the view.html as the default document for the View folder
- Restricts access to any unlisted folders in the directory
- Prohibits access to the Blaze and MyCustom folders.

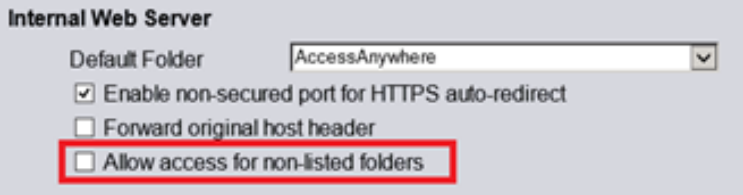
```
<<Section name="InternalWebServer">
      <Property name="Enabled" type="bool" value="true" />
      <Property name="ForwardOriginalHostHeader" type="bool" value="false" />
      <Property name="ForwardFaviconRequest" type="bool" value="false" />
      <Property name="XFrameOptions" type="string" value="" />
      <Property name="ContentSecurityPolicy" type="string" value="" />
      <Property name="AccessControlAllowOrigin" type="string" value="*" />
      <Property name="ClientSessionCookieTimeoutMinutes" type="int" value="60" />
      <Property name="AllowAccessForNonListedFolders" type="bool" value="false" />
      <Property name="DefaultFolder" type="string" value="AccessAnywhere" />
      <Property name="FolderList" type="list(WebServerFolder)">
        <Value>AccessAnywhere,start.html,True</Value>
        <Value>Blaze,blaze.zip,True</Value>
        <Value>Admin,login.html,True</Value>
        <Value>WebConnect,start.html,True</Value>
        <Value>SG,,True</Value>
```

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

```
    </Property>
  </Section>
```

Preventing Access to Non-Listed Folders

Additional subfolders can be added to the Secure Gateway WebServer folder. These folders can be accessible, even if they are not listed in the internal WebServerSettings list. To prevent access to folders that are not explicitly defined in the internalWebServerSettings list, clear the Allow access for non-listed folders (or set allow_access_for_non_listed_folders="false").



InTouch Access Anywhere Secure Gateway known limitations

This topic describes a number of known behaviors and limitations of Secure Gateway. Refer to InTouch Access Anywhere ReadMe for a more complete list of current known issues in Secure Gateway.

About InTouch Access Anywhere certificate error messages

Most web browsers require a trusted certificate to establish an encrypted network session. If you see an error message similar to the following figure, there could be a problem with the certificate on the InTouch Access Anywhere Secure Gateway server.

If this error message appears, check the address that is being used for the InTouch Access Anywhere Secure Gateway. If it is an IP address, like the image shown below, it may pose a problem.

Rather than using the IP address, use the domain name that matches a trusted certificate that has been configured in the InTouch Access Anywhere Secure Gateway.

For example, instead of using 192.168.1.111, use its domain name: sg.test.com.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

In addition, install a trusted certificate on the InTouch Access Anywhere Secure Gateway that matches sg.test.com or *.test.com

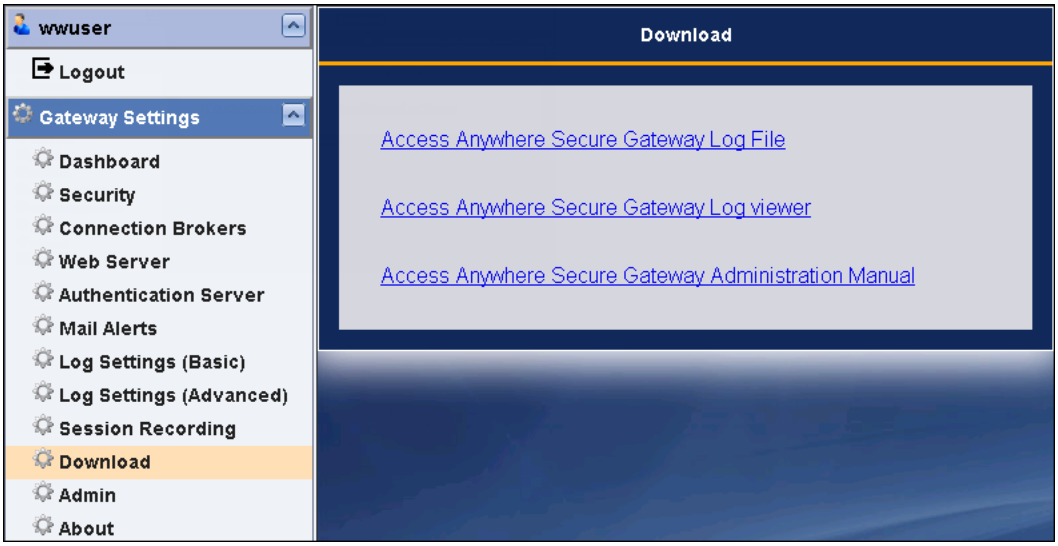
About InTouch Access Anywhere Secure Gateway log files

If you require technical support for the Secure Gateway, log files may be requested.

Note: The logs require a special viewer, which can be downloaded from the **Download** page

The current log file is accessible using the **Configuration** page under the **Download** tab. The actual diagnostic information saved in the log file can be set under the two log pages (Log Settings - Basic and Log Settings-Advanced).

Consult with a support engineer on which settings to enable.



About using InTouch Access Anywhere

This set of topics describe how to use InTouch Access Anywhere to remotely connect to your InTouch ViewApps by means of an HTML5-compatible web browser running on your desktop computer or mobile devices like tablets and cell phones. Information is provided for the different methods to log on to a Viewapp with InTouch Access Anywhere and how to use controls to manage a running Viewapp.

About InTouch Access Anywhere Supported Browsers

HTML5 websocket technology makes it possible to open an interactive communication session between your web browser and an InTouch Access Anywhere server. The following versions of HTML5 compatible web browsers have been tested and validated to work with InTouch Access Anywhere:

- Microsoft Edge
- Microsoft Internet Explorer 11
- Firefox version 38 and later
- Safari version 5 and later
- Chrome version 12 and later
- Opera version 11 and later

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Although InTouch Access Anywhere supports the listed browsers, you should review the About running InTouch Access Anywhere on different operating systems for unique behaviors and any alternative solutions for each browser.

Note: The browsers listed above have been tested only on PCs, tablets and SmartPhones. No browser testing has been done on other devices like smart TVs, eBook readers, or game consoles.

Older versions of the listed browsers below are functionally compatible with InTouch Access Anywhere. You may be able to use the following older versions of these browsers, but specific behaviors are unknown because no formal testing has been done with InTouch Access Anywhere.

- Apple Safari, versions 5, 6, and 7
- Google Chrome, versions 11 and later
- Microsoft Internet Explorer, versions 10
- Mozilla Firefox, versions 6 and later
- Opera, versions 11 and later

Important: You are strongly advised to upgrade to a recent version of a supported web browser for use with InTouch Access Anywhere.

About InTouch Access Anywhere and ViewApps

InTouch Access Anywhere can view AVEVA OMI ViewApps created by System Platform 2017 or later.

Contact your IT department to verify the AVEVA OMI ViewApp you want to view is from one of the supported versions System Platform.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

View an application with InTouch Access Anywhere

This set of topics describes how to log on to InTouch Access Anywhere from a supported device with an HTML5-compatible web browser. It includes the information you need to log on, a step-by-step log on procedure, advanced connection settings, and how to securely connect to InTouch Access Anywhere.

About information needed to sign on to InTouch Access Anywhere

Before you attempt to log on and begin viewing applications or ViewApps with InTouch Access Anywhere, contact your IT department to gather the following information:

- Has your IT department added you as a member of the Remote Desktop Users group on the InTouch Access Anywhere server? You must be a member to access the InTouch Access Anywhere server with a Remote Desktop Connection client.
- What is the domain name or IP address of your InTouch Access Anywhere server?
- Does your InTouch Access Anywhere server use default port 8080? If not, you need to specify the non-default port number when you log on to the InTouch Access Anywhere server.
- What is the name of the Windows domain specified with your user name to log on to the InTouch Access Anywhere server?
- Verify that your IT department has listed the InTouch applications you want to view on the InTouch Access Anywhere logon page.

Information Needed for an External Connection to InTouch Access Anywhere

- Does your company require a VPN to connect to the InTouch Access Anywhere server from a remote location? InTouch Access Anywhere is compatible with most VPNs that support web sockets.
- Will your data center use an InTouch Secure Gateway with InTouch Access Anywhere? If so, what is the domain name or IP address of the server hosting your Secure Gateway?
- If your data center uses an InTouch Secure Gateway, does it listen on default port 443? If not, you need to specify the non-default port number when you log on to the Secure Gateway.
- Is a trusted certificate installed on the Secure Gateway? By default, a self-signed certificate is installed on the Secure Gateway. Some web browsers show a security warning or prohibit connections to servers using self-signed certificates.

Sign on to InTouch Access Anywhere

You can log on to the InTouch Access Anywhere server by several methods based on whether your company uses a Secure Gateway or not.

Note: If you have any trouble connecting to a remote InTouch Access Anywhere node, see Troubleshoot sign on problems for help.

Sign on directly to InTouch Access Anywhere

Use the following procedure to log on directly to the InTouch Access Anywhere server. You should be using a VPN for a secure connection from an external network.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To log on directly to the InTouch Access Anywhere server

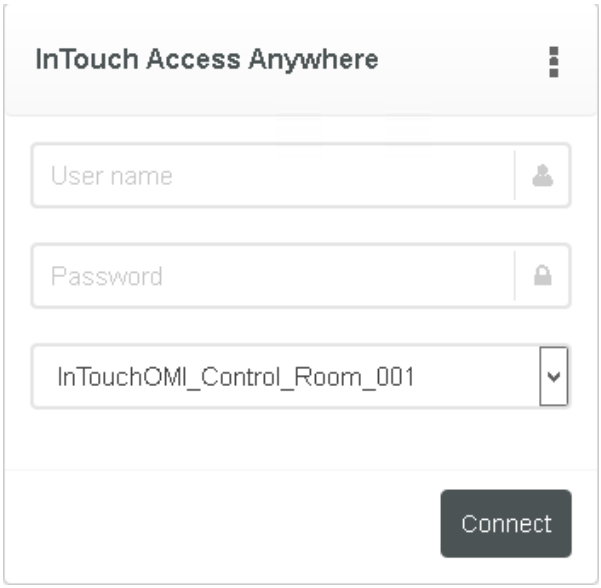
1. Using your web browser, go to

   `http://ITAA_Server_Node_Name:8080/`

   or

   `http://ITAA_Server_IP_Address:8080/`

Note: You may need to enter a different port number if your company changed the default port number assigned to the Access Anywhere server.



A logon page appears with fields to enter your user name, password, and select the InTouch application you want to view.

Connection Details	Description
User Name	Your user name to log on to the Remote Desktop Service (RDS) host. It can optionally contain a domain specification, for example, `domain\user`. If a domain name is not specified, you will be prompted for it by the RDS host.
Password	Your password to log on to the RDS host. When not specified, you will be prompted for a password by the RDS host.
Application name	A pull down list shows InTouch applications accessible from the InTouch Access Anywhere server that can be selected for viewing.

1. Enter your user name and password.
2. Select the InTouch application you want to view from the drop-down list of the **Application Name** field.
3. Tap or click **Connect** to initiate an InTouch Access Anywhere browser session.

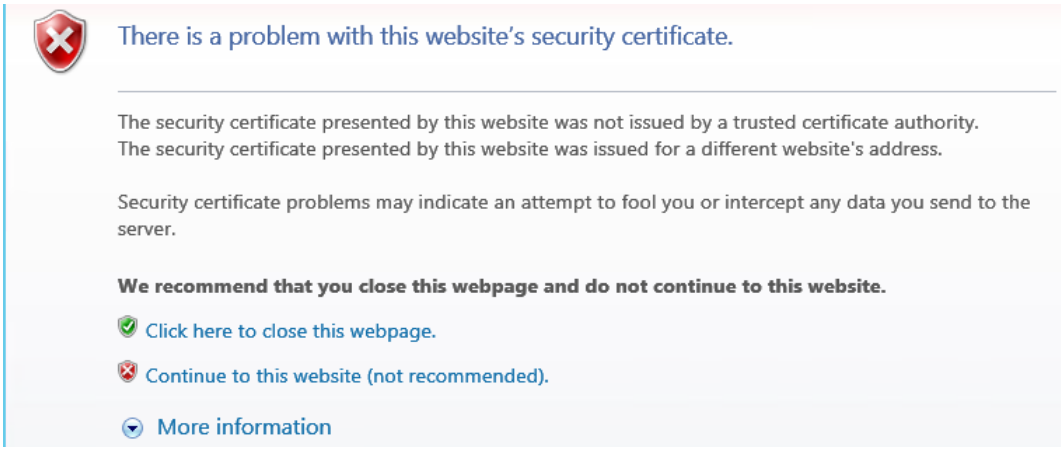
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Your initial view will show the InTouch application within the boundaries of your web browser window, which is the **Fit to browser** Advanced Settings option.

- For more information about setting your InTouch application view, see Advanced connection settings: display page.
- For more information about changing the size of an InTouch application window using the **Fit to browser** setting, see Resize an application window.

Sign on through a Secure Gateway

By default, the Secure Gateway includes a self-signed certificate. Some browsers, such as Google Chrome, allow self-signed certificates for SSL-encrypted websocket connections. Other web browsers like Internet Explorer show a warning message that the server certificate is not signed and prompt you to continue only at a significant risk.



Chrome OS and Safari web browsers prohibit connections using a self-signed certificate. In order to provide connectivity for these browsers, your IT department must import a trusted certificate into the InTouch Access Anywhere server or into the Secure Gateway.

If you encounter certificate problems trying to log on to the Secure Gateway, consider changing web browsers or ask your IT department to install a trusted certificate.

Use the following procedure to log on to the Access Anywhere server through a Secure Gateway.

To log on through a Secure Gateway

1. Using your web browser, go to

   `https://Secure_Gateway_Node_Name:443/`

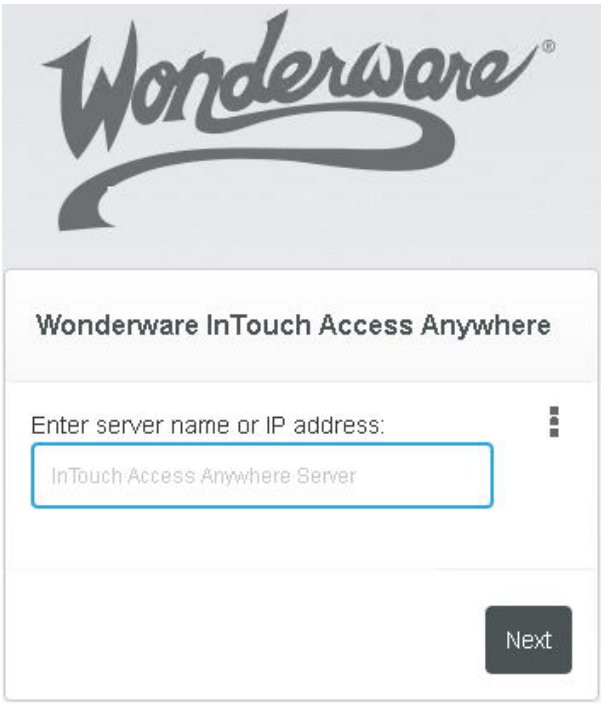
   or

   `https://Secure_Gateway_IP_Address:443/`

   Note: You may need to enter a different port number if your company changed the default port number assigned to the Secure Gateway.

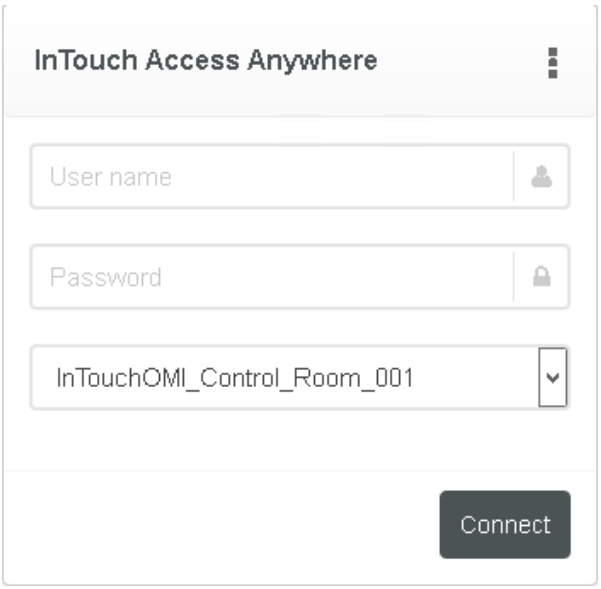
   The Secure Gateway start page appears with a field to enter the domain name or IP address of the Access Anywhere server.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. Enter the IP address or the domain name of the RDP host computer where the InTouch Access Anywhere server is installed.
3. Click or tap **Next**.

   The InTouch Access Anywhere server logon page appears with fields to enter your user name, password, and select a ViewApp you want to view.



4. Enter your user name and password.

   Enter the domain name of your RDS host account in front of your user name if not specified as an Advanced Setting.
5. Select the ViewApp you want to view from the drop-down list.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

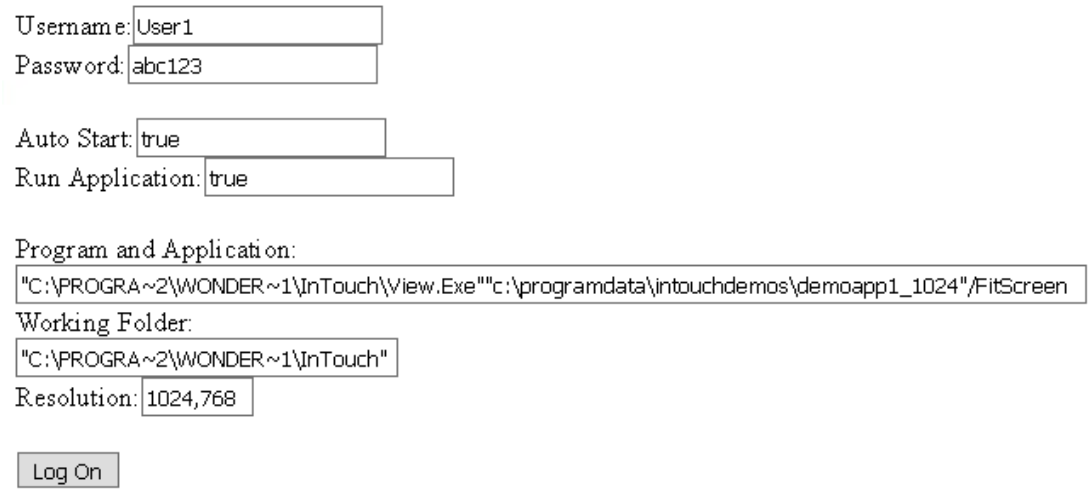
6. Tap or click **Connect** to initiate an InTouch Access Anywhere browser session.

Sign on with single sign-on

InTouch Access Anywhere supports Single Sign-On (SSO). When using a third-party authentication entity (such as an SSL VPN) that supports SSO Form Post, you only need to sign on once to an InTouch Access Anywhere session using your authentication credentials to begin viewing an InTouch application.

Typically, your IT administrator will send you an HTML file, which is an SSO form that you place on your mobile device or desktop computer. You view the file with an HTML5 web browser.

The following illustration shows a sample SSO log on form with two mandatory fields to enter your username and password. The form may contain other optional fields too based upon the viewing options that your IT administrator wants to grant users to view an application.



After clicking **Log On** to submit the SSO form, your web browser should appear with a running ViewApp.

Troubleshoot sign on problems

If you have trouble remotely connecting to your InTouch Access Anywhere server, verify network connectivity and your web browser version by viewing the InTouch Access Anywhere demonstration site:

http://intouchaccessanywhere.com

Unable to reach the InTouch demonstration site

If you cannot access the InTouch demonstration site, verify your network connectivity:

- From a desktop computer, use the Ping command to verify the connection to the Secure Gateway or Access Anywhere servers. Third party tools exist for some mobile devices to provide equivalent functionality.
- From a Windows computer, try logging on to the InTouch Access Anywhere server directly using a standard Remote Desktop Connection client. You must be able to log on using standard Remote Desktop Connection client before you can log on to an InTouch Access Anywhere server.

Unable to view the InTouch demonstration application correctly

If you can access the demonstration site but not view the InTouch application correctly, verify your web browser.

- Are you using a supported InTouch Access Anywhere web browser? For a list of supported browsers, see About InTouch Access Anywhere Supported Browsers.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Browser extensions or add-ons may inject JavaScript code into web pages. If the InTouch application is inaccessible or not working properly, disable or uninstall any active browser extensions or add-ons. Then, restart your web browser to ensure the extensions or add-ons are no longer active.

Able to view the InTouch demonstration application

If the demonstration site appears and you can successfully launch the InTouch demonstration application, you have network connectivity and your browser is compatible with InTouch Access Anywhere.

- If you cannot reach the InTouch Secure Gateway by its domain name, try using its IP address.
- If you cannot reach your InTouch Access Anywhere server by its domain name, try using its IP address.

Assistance from your IT department

If your are still experiencing logon problems, contact your IT department to get more information about the following:

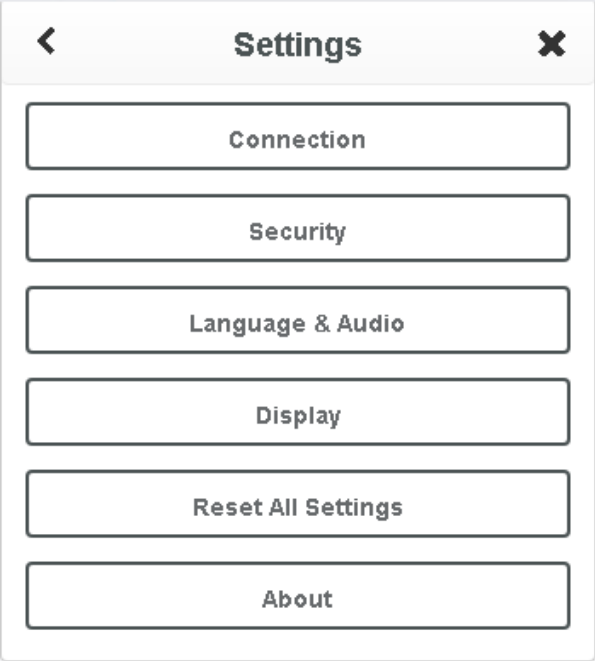
- Ask your IT department if the port numbers you entered to log on to the Secure Gateway and Access Anywhere servers are correct. The default ports are 443 for the Secure Gateway and 8080 for the Access Anywhere server.
- Ask your IT department to verify the InTouch Access Anywhere service is running.
- Ask your IT department if you are a member of the Remote Desktop Users group on the InTouch Access Anywhere server? You must be a member to access the InTouch Access Anywhere server with a Remote Desktop Connection client.
- Ask your IT department if a trusted certificate is required for the InTouch Access Anywhere Secure Gateway or the InTouch Access Anywhere Server.

Advanced connection options

Click the **Settings** icon on the logon page to set additional options for your InTouch Access Anywhere connection.

The advanced connection settings are shown on separate web pages by selecting the **Settings** buttons.

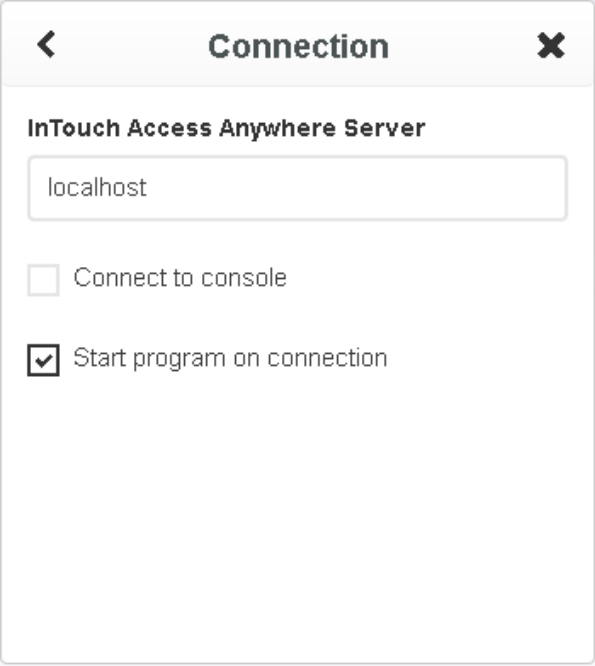
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Important: You may need information from your IT administrator to specify values for some Advanced Connection settings.

Advanced connection settings: connection page

The advanced connections settings on the **Connection** page provide options to specify the name of the InTouch Access Anywhere Server you want to connect to, as well as the option to connect to console.

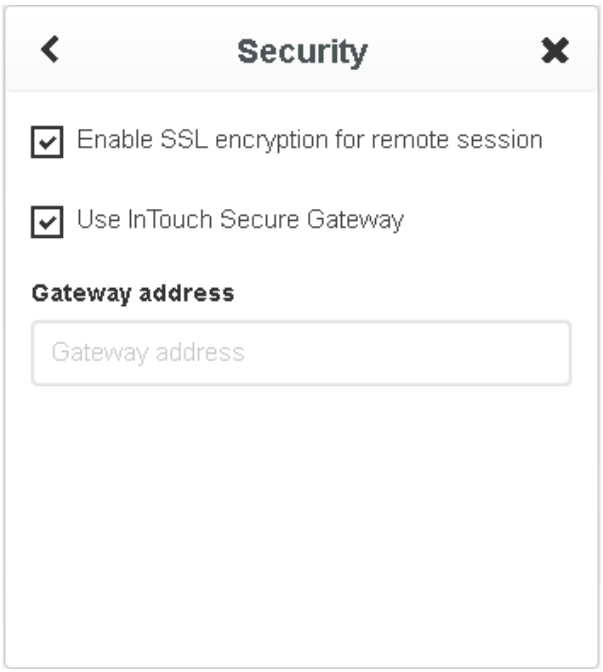


© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Settings	Description
InTouch Access Anywhere Server	The name or IP address of the InTouch Access Anywhere Server.
Connect to Console	Connects to a console session of the operating system, instead of a user session.
Start program on connection	This setting cannot be modified. The default value shows the ViewApp you selected running in a web browser.

Advanced connection settings: security page

The Advanced Connection settings on the Security page provide options to enabled SSL Encryption and specify the Gateway address.



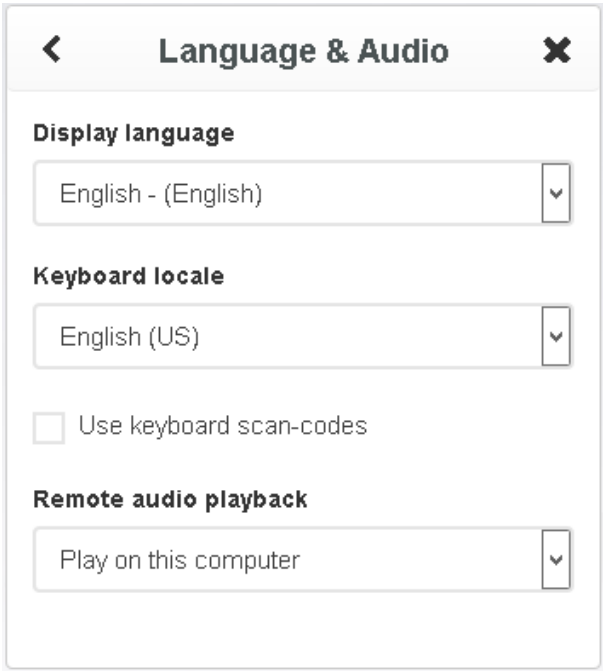
Settings	Description
Enable SSL encryption for remote session	When selected, your web browser uses SSL encrypted websocket communication to the InTouch Access Anywhere Server.
Use InTouch Secure Gateway	When selected, a Secure Gateway connects to the RDP host.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Settings	Description
Gateway address	Computer name or IP address of the server hosting the InTouch Secure Gateway. You can enter the address and port for the Secure Gateway in this field. If no port value is specified, 443 is used by default. To specify a custom port, add a colon (:) and the port number to the address. For example, gateway.com:4343. Multiple Secure Gateways can be specified as failover servers. Separate each server address with a comma (,) or semicolon (;). An asterisk (*) means the servers can be specified in any order after the asterisk. For example, if the following is specified: `aaa;*;bbb:4433;ccc:4343` The Secure Gateway aaa on port 443 is used to initially connect. If aaa is unavailable, then bbb:4433 is used followed by ccc:4343 OR ccc:4343 followed by bbb:4433.

Advanced connection settings: language and audio page

The Advanced Settings Language and Audio Page provides options to set the display language, the keyboard locale and the source of audio playback, if selected.



Settings	Description
Display Language	Changes the language used by the InTouch Access Anywhere Server log on page.

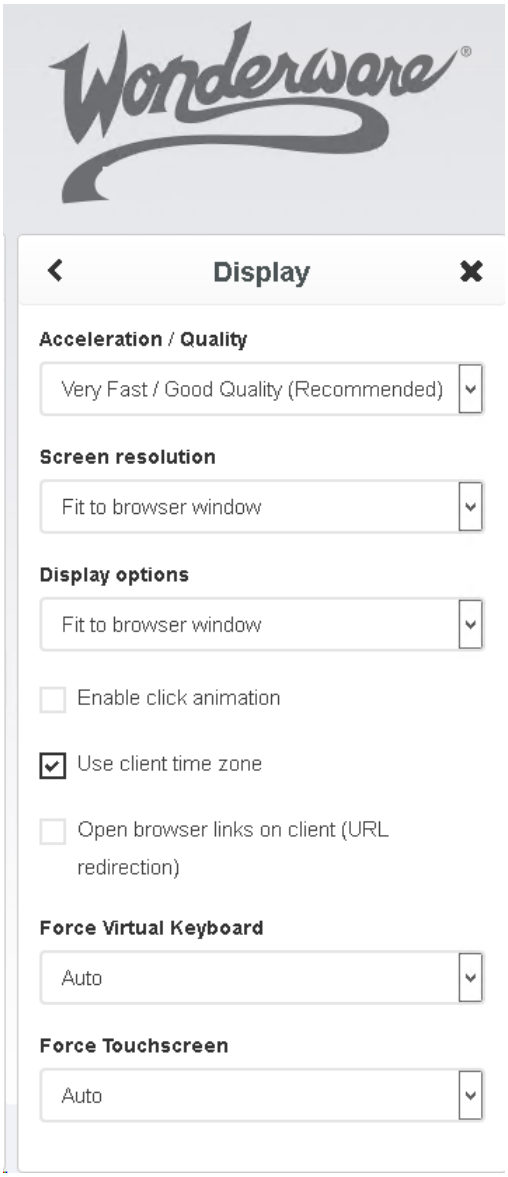
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Settings	Description
Keyboard locale	Select the keyboard region to be used in an InTouch Access Anywhere session.
Keyboard scan-codes	Enables keyboard scan codes sent by a keyboard to indicate a key has been selected and the alphanumeric character or function associated with the key. Certain applications use scan codes and require this setting to be enabled.
Remote audio playback	Select an option as the source of the session's audio: local computer, remote computer, or do not play audio.

Advanced connection settings: display page

The Advanced Connections settings on the Display Page provide options to set display quality and fit the ViewApp window your browser window or fit to your full screen.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Settings	Description
Acceleration/Quality	Controls the degree of acceleration that is enabled in the session. Faster acceleration results in lower quality application images. **Very Fast / Good Quality** is the recommended setting.
Screen resolution	The value shown by **Screen Resolution** matches the selected **Display options Fit to browse**r and **Fit to screen** values. When **Application Resolution** is the selected **Display options** value, the **Screen Resolution** option will attempt to match the actual application resolution from the drop down list. If a matching resolution is found, the resolution is selected and the **Screen Resolution** control is disabled. If there is no matching resolution, then the **Screen Resolution** option is enabled to enter a resolution value.
Display options	• **Fit to browser window (default)** Fit the WindowViewer window within the **initial** size of your web browser's window. **Note:** The application's resolution does not change from its initial size if you change the size of your web browser while viewing a ViewApp. When **Fit to browser window** is selected, the **Screen Resolution** option is disabled. • **Fit to screen (Full Screen)** Places the entire ViewApp within the current size of the window without showing your web browser's menu bars. • **Application Resolution** Displays a ViewApp in a browser at its native resolution. If the browser’s viewing area does not show the entire application, the user can use the browser's scroll bars to show the rest of application. When this option is selected, InTouch Access Anywhere attempts to match the actual application resolution to a screen resolution shown in the drop down list. If matching resolution is found, then the resolution is selected and the **Screen Resolution** option is disabled. If there is no matching resolution, then the **Screen Resolution** control is enabled.
Enable click animation	When enabled, a visual indicator appears on a touch device screen when touch gesture is performed.
Use client time Zone	When selected, this setting enables local time zone redirection. The remote session running on the InTouch Access Anywhere server uses the time of your local computer or mobile device.

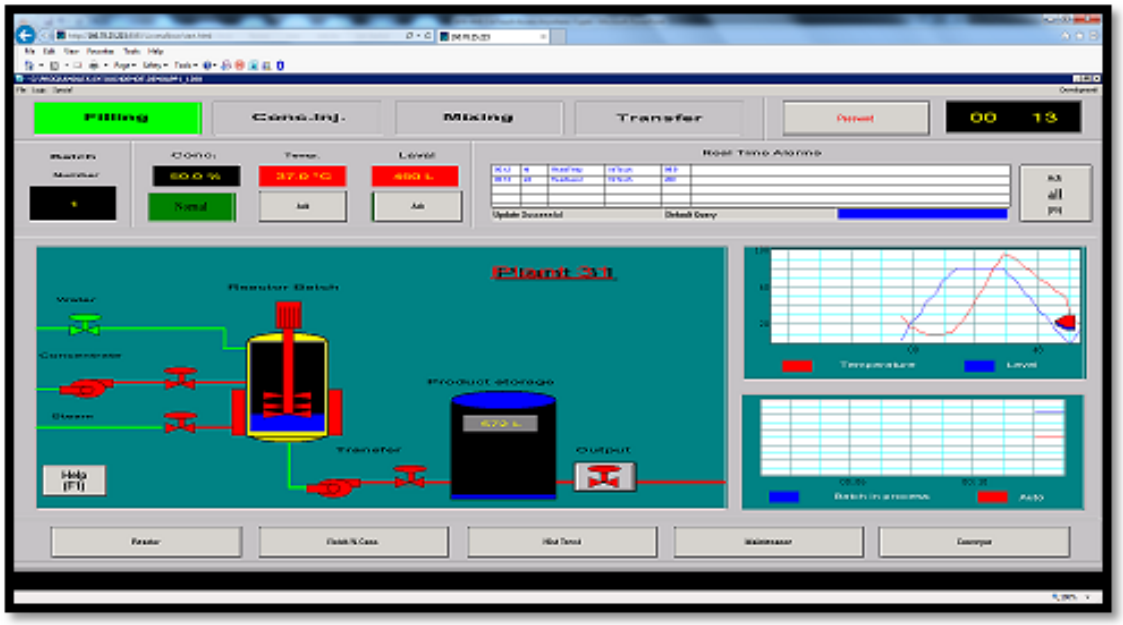
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Settings	Description
Acceleration/Quality	Controls the degree of acceleration that is enabled in the session. Faster acceleration results in lower quality application images. **Very Fast / Good Quality** is the recommended setting.
Screen resolution	The value shown by **Screen Resolution** matches the selected **Display options Fit to browse**r and **Fit to screen** values. When **Application Resolution** is the selected **Display options** value, the **Screen Resolution** option will attempt to match the actual application resolution from the drop down list. If a matching resolution is found, the resolution is selected and the **Screen Resolution** control is disabled. If there is no matching resolution, then the **Screen Resolution** option is enabled to enter a resolution value.
Open browser links on client (URL redirection)	When selected, your web browser session is redirected to the URL link launched from the RDS session.
Force Virtual Keyboard	Provides values (**Yes**, **No**, **Auto**) to select a device virtual keyboard while viewing a running InTouch application. **Auto** is the default, which forces **Yes** or **No** based on whether the device displaying the ViewApp has a virtual keyboard or not.
Force Touchscreen	Provides options (**Yes**, **No**, **Auto**) to use touch gestures while viewing a running InTouch application. **Auto** is the default, which forces **Yes** or **No** based on whether the device running the application supports touch gestures or not.

View an InTouch Access Anywhere session

After a successfully logging on, a Remote Desktop Connection (RDC) session is launched and connects you to the selected ViewApp. The ViewApp appears within your browser's window.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



While connected, InTouch Access Anywhere intercepts mouse and keyboard events and transmits them to the RDS host. As a result, various keyboard keys and mouse buttons that are usually handled by the browser may behave differently. For example, clicking the F5 button normally causes the browser to reload the current page, but when using InTouch Access Anywhere F5 does not reload the page. Instead, the current page will be transmitted to and handled by the remote InTouch WindowViewer application.

Note: In most browsers, clicking the Back, Forward, or Reload browser buttons will cause InTouch Access Anywhere Server to display a message asking if you wish to leave the current page. If you decide to proceed, the remote session will be disconnected. (Some browsers may not ask for confirmation).

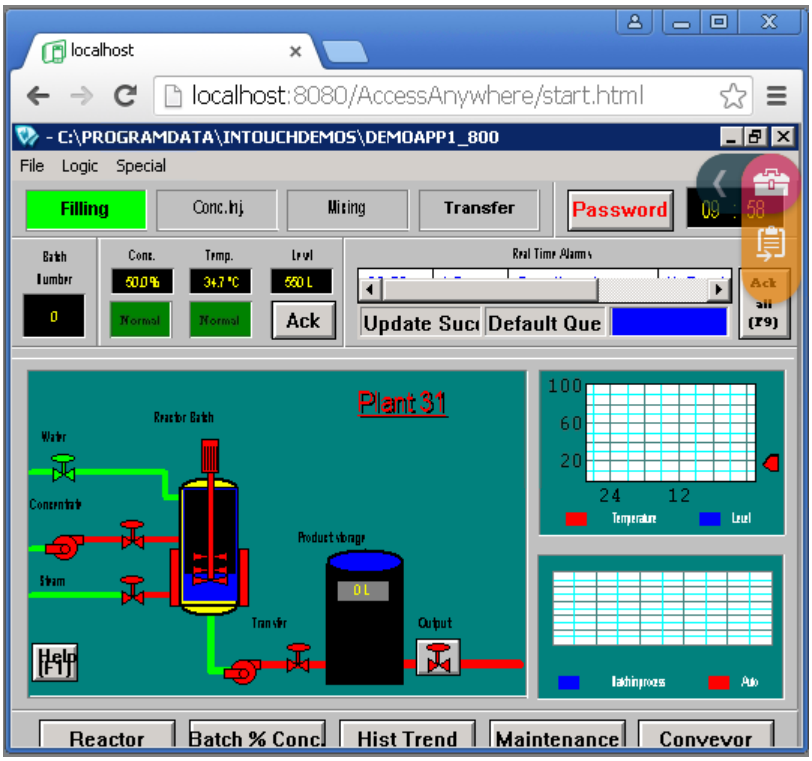
When accessed by InTouch Access Anywhere, a ViewApp cannot be minimized. If the remote InTouch application is shut down, your InTouch Access Anywhere session is automatically terminated in a configurable period set as a parameter of the InTouch Access Anywhere server.

Closing the web browser on your device leaves the RDC session running on the server with the ViewApp still running.

Resize an application window

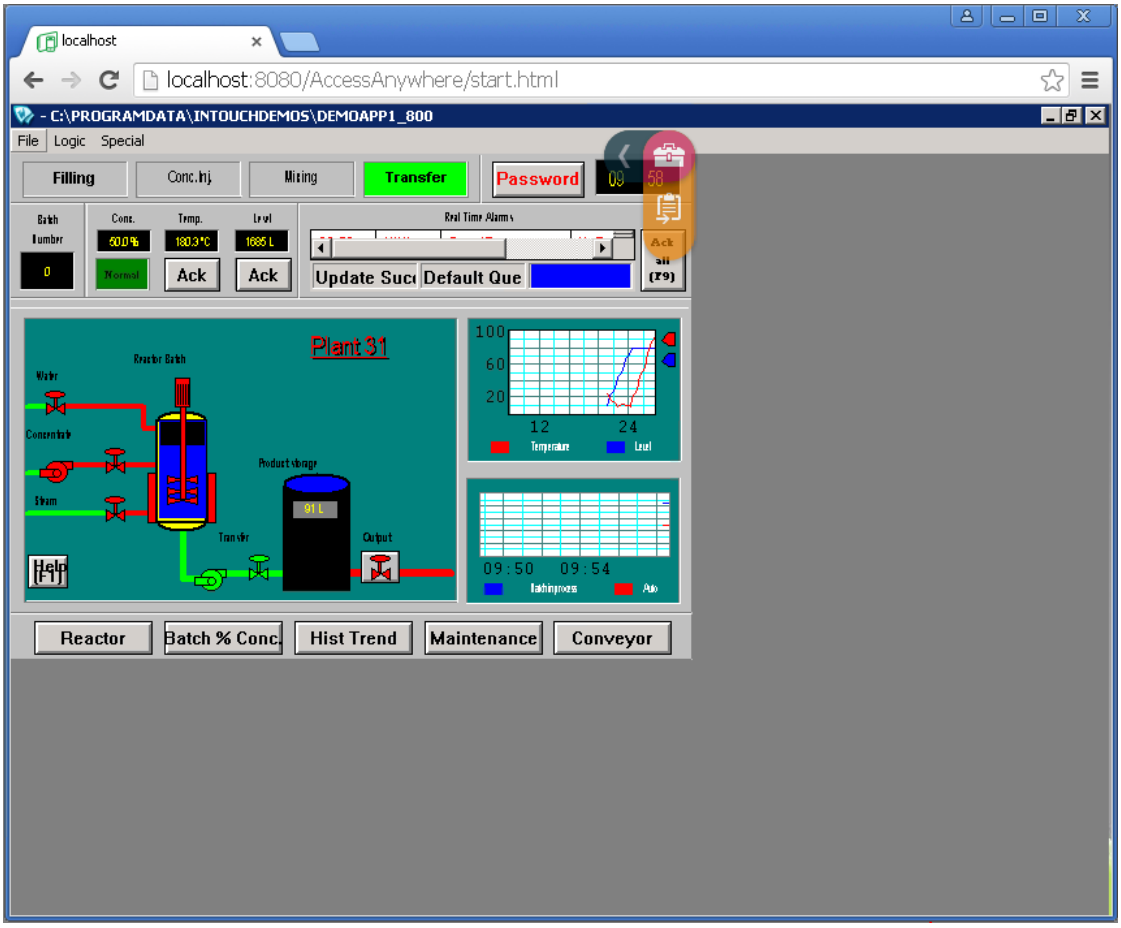
You can change the size of an InTouch application window to fit within your current web browser if you select the **Fit to browser** Display Options setting. The size of the ViewApp window is set to the initial size of your web browser's window when you log on to an InTouch Access Anywhere session.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



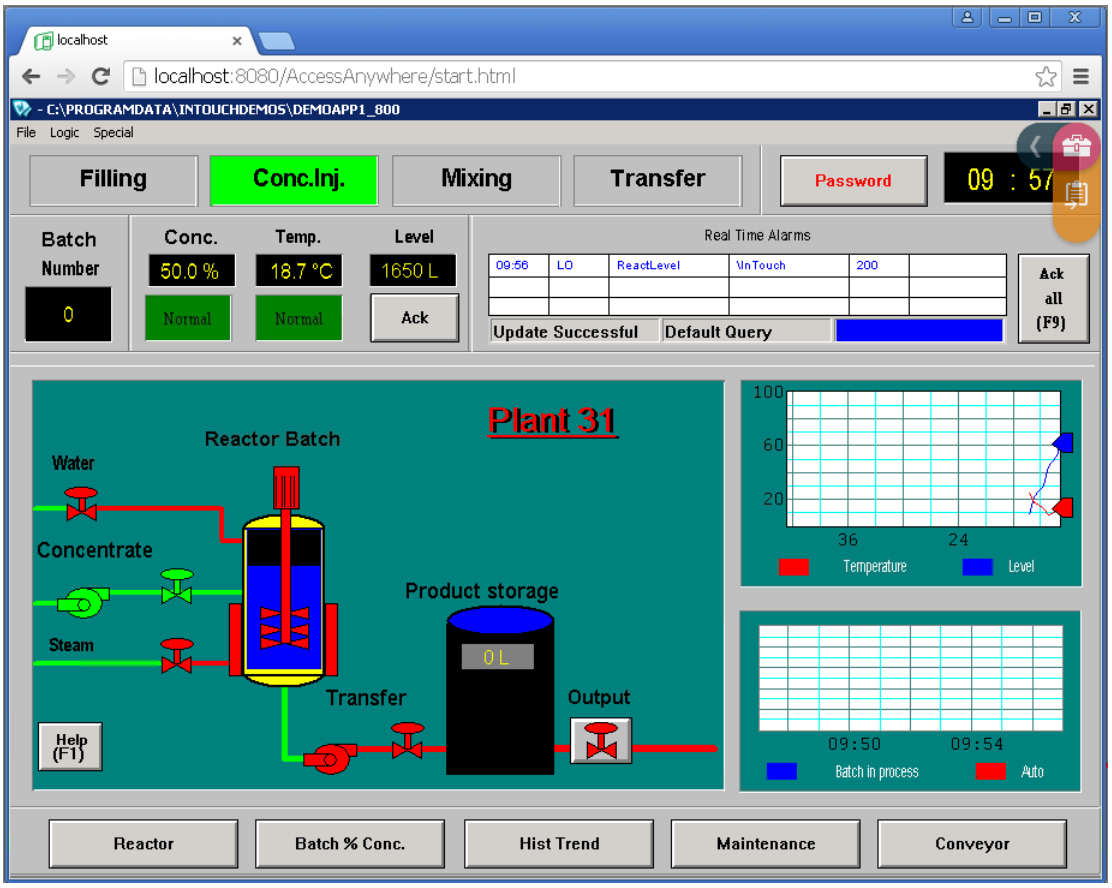
If you change the size of your web browser window during an InTouch Access Anywhere session, the size of the InTouch application window remains the same as its initial size.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



To change the size of a ViewApp window, change the size of your web browser window, close the ViewApp, and disconnect from you InTouch Access Anywhere session. Then, log back on to show your ViewApp at its new size.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Use the Access Anywhere toolbar

tInTouch Access Anywhere provides a toolbar to access a set of features and system keys while a ViewApp is running. The toolbar appears in the browser window and can be moved by dragging and dropping it to another window location.

Tap or click the toolbar icon once to show a blue arrow to expand the blue bar. The blue bar includes icons that represent system keys and additional functions.

	The gestures help icon appears only when running InTouch Access Anywhere on a touch screen device. Click or tap the help icon to see figures and short explanations of supported finger gestures.
	Click or tap the ESC icon to cancel a value entered in a data entry field.
	Click or tap the TAB icon to shift the window focus to the next data entry field of an application or the next cell of a table.
	Click or tap the Disconnect icon to end the InTouch application browser session and return to the logon page.
	Click or tap the About icon to see information about the specific version of the InTouch Access Anywhere server installed on the RDS host computer.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Tap or click once on the tool box icon to expand the orange bar to show icons that represent keyboard or clipboard functions. The functions shown in the orange bar vary based on whether you are using a computer with a keyboard and mouse or a touch screen device.

Clipboard support

The InTouch Access Anywhere Toolbar provides a feature to copy and paste text or numbers between your local desktop or laptop and a remote InTouch Access Anywhere session using a built-in clipboard.

Note: When using Internet Explorer (MSIE) 10, the Clipboard feature is integrated, so there are no AccessNow clipboard icons. Simply copy and paste text between the local device and Access Anywhere session using Windows copy (Ctrl+C) and paste (Ctrl+V) keyboard shortcuts.

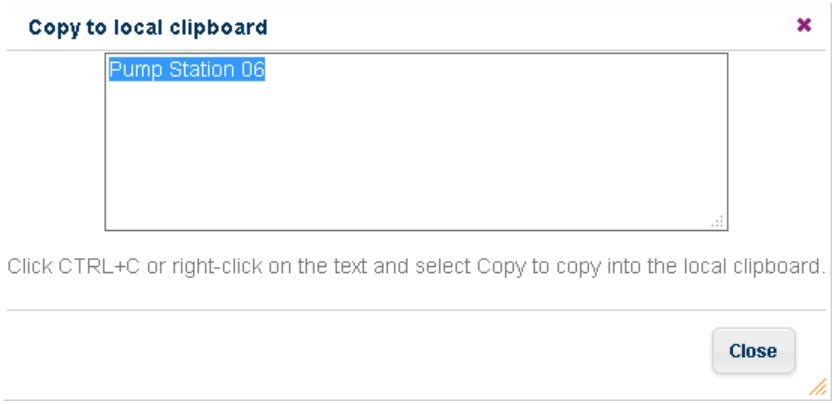
Copy Text from a Remote InTouch Application

The following procedure explains how to copy text from an InTouch application shown in an InTouch Access Anywhere browser session to your local desktop or laptop computer.

To copy text from an InTouch application

1. Select the text or number to copy from an InTouch application running in your InTouch Access Anywhere browser session.
2. Perform a copy keyboard shortcut (Ctrl +C) on the desired text or number.

   The **Copy to local clipboard** dialog box appears in your remote browser session.



3. Click on the clipboard icon to copy the text to the clipboard.
4. Click **Close**.
5. Open a document or application on your local desktop computer or laptop.
6. Press Ctrl+V to paste the text.

Copy Text to a Remote InTouch Session

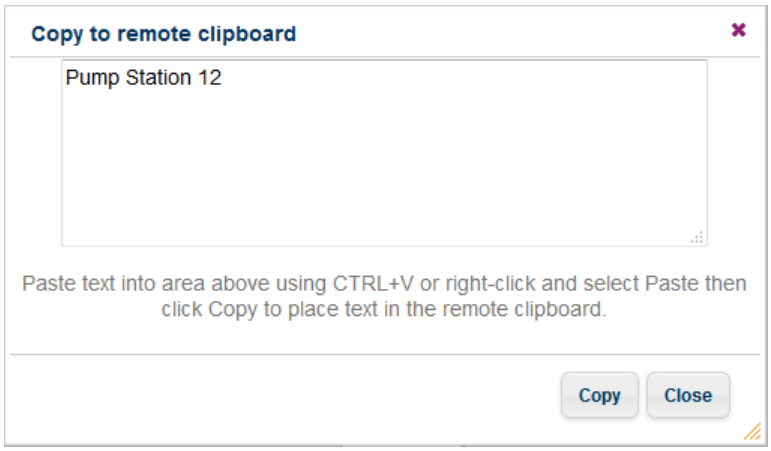
The following procedure explains how to copy text from a local desktop or laptop computer to an InTouch application running in a remote Access Anywhere browser session.

To copy text to a remote InTouch application

1. From a local application like Notepad, copy text you want to paste into a remote InTouch application running in an InTouch Access Anywhere browser session.
2. Open the InTouch Access Anywhere browser session to show a running InTouch application.
3. Click the **Open Clipboard** icon on the toolbar.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

4. Paste the copied text into the clipboard.



5. Click **Copy.**
6. Set the focus in WindowViewer to a data entry field in the running InTouch application.
7. Press Ctrl+V or right-click and select **Copy** from the shortcut menu to paste the text into the remote InTouch application's data entry field.

Use finger gestures

InTouch Access Anywhere supports control gestures when used on a touch device. The following table shows the different types of gestures you can make on a touch device to emulate mouse events or specific keyboard keys.

Gesture functionality varies across different devices, browsers, and operating systems. Tap the Gestures help icon in the Toolbar to see the full list of supported gestures. For more information about gesture limitations, see About device support for InTouch Access Anywhere.

Application Behavior or Key	Gesture	Description
Left Mouse Click		One finger tap once
Right Mouse Click		One finger tap and hold
Double Mouse Click		Tap once with two fingers

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Application Behavior or Key	Gesture	Description
Double Mouse Click		One finger double tap
Drag Object		One finger tap, hold and drag
Zoom In		Two fingers pinch out
Zoom Out		Two fingers pinch in
Page Up		Three fingers swipe down
Page Down		Three fingers swipe Up
Left Arrow		Three fingers swipe right
Right Arrow		Three fingers swipe Left

Change an InTouch application shown by WindowViewer

If a ViewApp is currently running in an InTouch Access Anywhere session, another ViewApp cannot be viewed without first logging on to the Access Anywhere server and selecting another ViewApp.

To change an InTouch application shown by WindowViewer

1. Log on to InTouch Access Anywhere that is currently running an InTouch application.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. Select the **Disconnect** icon from the Access Anywhere toolbar.
3. Click or tap **Yes** to confirm that you want to disconnect your InTouch Access Anywhere session.

   You should return to the logon page.
4. Select the ViewApp you want to view from the **Application name** field of the logon page.
5. Enter your user name and password to start another InTouch Access Anywhere session.

   After establishing a session, you should see your browser start and show the ViewApp you selected.

End an InTouch Access Anywhere session

After you click or tap the **Disconnect** icon from the Desktop Toolbar, a message appears requesting that you confirm that you want to disconnect from the Access Anywhere server.

After you click **Yes**, the browser closes and you see the InTouch Access Anywhere logon page. No trace of the InTouch Access Anywhere browser session remains on your device after ending a session. But, the RDP session is still running WindowViewer on the InTouch Access Anywhere server.

Session auto-logoff

A session is logged off when a ViewApp is closed. In some cases,a session is not closed immediately or is frozen. InTouch Access Anywhere Server includes an auto-logoff feature that automatically logs off a session if nothing is displayed on the screen for a specified period.

Your InTouch Access Anywhere administrator can set the length of the auto-logoff period with a RDP server or InTouch Access Anywhere configuration parameter.

Automatic Reconnect

Active InTouch Access Anywhere sessions automatically attempt to recover from temporary network outages by reconnecting your session. You may experience a slight delay during the reconnect attempt, but after the session is re-established, you can continue working without having to log back on to the session.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

About device support for InTouch Access Anywhere

This set of topics describes the runtime behavior of InTouch Access Anywhere on different mobile devices, web browsers, and operating systems.

InTouch Access Anywhere can operate on mobile devices with an HTML5 compliant browser to remotely access and interact with InTouch applications. Many mobile devices use touch gestures to perform the tasks normally done by a mouse using a desktop or laptop computer. Because there is no mouse, touch devices require finger gestures to emulate mouse behaviors.

Also, some mobile devices use built-in software keyboards instead of physical keyboards. Typically, mobile device software keyboards do not have the standard Windows F1-F12, Ctrl, or Alt keys.

About running InTouch Access Anywhere on different operating systems

This set of topics describes specific behaviors of InTouch Access Anywhere by specific combinations of device operating systems and browser combinations. The behaviors are grouped by operating systems. Alternative solutions to issues are provided when available.

Running InTouch Access Anywhere on Android

This table shows the behavior of different browsers on Android devices. For more common behaviors on the Android operating system see Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations and Running InTouch Access Anywhere: common behaviors.

Behavior Type	Browser(s)	Description
Drag and drop	All	You may experience difficulty dragging and dropping a ViewApp pop-up window to another location. **Alternative Solution** Use a device stylus to perform drag and drop operations.
Scroll bars	Opera and Chrome	Scroll bars cannot be scrolled by tapping or pressing and dragging the scroll button. **Alternative Solution** Tap on the gray area of the scroll bar to move a window.
Tooltip animations	Opera, Firefox, and Chrome	Tooltip animations configured on Industrial Graphics may not appear with a mouse over action. **Alternative Solution** Substitute another gesture to implement a mouse-over action for your device and operating system.
Opening an InTouch application	Firefox	Selecting a ViewApp from the Access Anywhere **Application Name** list box can immediately open the ViewApp without clicking **OK**. **Alternative Solution** There is no alternative solution.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Behavior Type	Browser(s)	Description
Keystrokes over Bluetooth	Firefox	When using a Bluetooth keyboard, a single keystroke may produce two characters. **Alternative Solution** Use the mobile device's built-in keyboard.
Text display	Chrome	When an Android mobile device runs low on memory, text in a window can appear blurred. **Alternative Solution** Close all running applications and restart your mobile device before using InTouch Access Anywhere.
Text input	Opera, Firefox, and Chrome	Typing double-byte language characters using the native Android software keypad appear as question marks (?) in a data entry field. **Alternative Solution** Configure the InTouch Access Anywhere server to use the **Windows Keyboard** option. Double-byte language characters can be entered correctly from the Android keypad or by using a paired Bluetooth hardware keyboard.

Running InTouch Access Anywhere on Google Chrome

InTouch Access Anywhere can operate on a Google Chromebook running the Chrome operating system. The following table lists tips when using InTouch Access Anywhere with a Chromebook running the Chrome operating system and web browser.

Function	How to Perform
Mouse Left-click	Click the Chromebook trackpad with one finger.
Mouse Right-click	Click the Chromebook trackpad with two fingers.
Scrolling a document or website	Drag two fingers on the Chromebook trackpad up or down to scroll.
Configure Chromebook	In the address field, enter: chrome://settings.

About Chromebook keyboard commands and InTouch Access Anywhere

The Chromebook keyboard lacks several keys that are used by Windows. ChromeOS provides standard Windows keyboard mappings that use existing keys with the Alt button to represent certain missing keys. InTouch Access Anywhere supports these key combinations:

Command	Chromebook Key Combination
Delete	Alt+Backspace

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Command	Chromebook Key Combination
Page Up	Alt+Up
Page Down	Alt+Down
Home	Ctrl+Alt+Up
End	Ctrl+Alt+Down

In addition, InTouch Access Anywhere provides special non-standard mappings for additional key combinations on a Chromebook.

Command	Chromebook Key Combination
F1	Ctrl+1
F2, ..., F12	Ctrl+2, ..., 12
Alt+Tab	Alt+"
Alt+Shift+Tab	Alt+Shift+’
Ctrl+Home	Ctrl+Alt+Left
Ctrl+End	Ctrl+Alt+Right

Running InTouch Access Anywhere on iOS

The following table shows the specific behavior of different browsers running on the Apple iOS operating system. For more common behaviors on the iOS operating system see Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations and Running InTouch Access Anywhere: common behaviors.

Behavior Type	Browser(s)	Description
Accessing InTouch Access Anywhere	Safari	You may experience difficulty accessing InTouch Access Anywhere if the Safari browser’s Private Browsing option is enabled. Private browsing is enabled if Safari bars appear black or dark instead of blue or gray. **Alternative Solution** 1. Open Safari. 2. Select the **Pages** icon that is shaped as two squares. 3. Select the **Private** setting and turn it Off.
Accessing InTouch Access Anywhere	Opera Mini on an iOS device	This browser is not a supported. You cannot reach the InTouch Access Anywhere web server. You can use a supported browser on an iOS device.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Behavior Type	Browser(s)	Description
Refresh current page	Safari and Chrome	When you click to refresh or navigate away from the current page while connected to the InTouch Access Anywhere server, you will not be prompted to confirm your action, which may result in unwanted page navigation. **Alternative Solution** There is no alternative solution.
Invoking Direct Push Button	All browsers on iPad or Android tablets	Using a single tap to toggle the state of a push button does not work consistently. A push button can be toggled consistently if the tap gesture is held for three or four seconds. **Alternative Solution** Modify the ViewApp button to change state by User Inputs>Discrete animation.
Moving between text boxes	Safari and Chrome on iPad	The **Next** and **Previous** buttons are not enabled to move between text boxes on dialog boxes with multiple text boxes. **Alternative Solution** Use a touch gesture to transfer focus to a specific window element.
Text input	Safari on iPad	Typed double-byte language characters do not appear in an InTouch data entry field when using an iPad software keypad.

Running InTouch Access Anywhere on Windows RT

This table shows the behaviors of different browsers on the Windows RT operating system. For more common behaviors on the Windows RT operating system see Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations and Running InTouch Access Anywhere: common behaviors.

Behavior Type	Browser(s)	Description
Touch gestures	Internet Explorer 10 in a touch environment	You cannot use a touch gesture to open a right-click shortcut menu on the AlarmViewer and AlarmDBView controls. **Alternative Solution** Access menu options through scripts.
Mouse pointer	Internet Explorer 10 in a touch environment	The mouse pointer may disappear after using the Surface keyboard to enter data. **Alternative Solution** Use touch gestures in place of mouse inputs.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Behavior Type	Browser(s)	Description
Pop-up dialog display	Internet Explorer 10 in a touch environment	A pop-up dialog can appear blurred after dragging it to a new screen location. **Alternative Solution** Refresh the session.
External data subscriptions	Internet Explorer 10	External data subscriptions intermittently do not refresh on starting WindowViewer without user interaction or input. A single touch on the screen will start the data subscription refreshes, except on read-only kiosks and other read-only devices. **Alternative Solution** Use a timer or clock control to enable user interaction that does not interfere with the InTouch application itself.
Delta deployment	Internet Explorer 10	Window Viewer can stop responding after a delta deploy, especially with applications comprised of a large number of windows. **Alternative Solution** Close and restart the browser session.
Shortcut keys	Internet Explorer 10	Shortcut keys Shift+ and alphabetic characters configured for scripts do not execute the script functions. **Alternative Solution** Use different combinations of keyboard shortcuts, such as Shift+Ctrl+(alphabetic character).
Unlocking a tablet device	Internet Explorer 10	The browser session may occasionally stop responding when unlocking the tablet on which InTouch Access Anywhere is running. **Alternative Solution** Close and reopen the browser.
Text input	Internet Explorer	Double-byte language characters cannot be entered into an InTouch data entry field using the Surface software keypad. **Alternative Solution** Configure WindowViewer on the InTouch Access Anywhere server to use the **Windows Keyboard** option. Also, double-byte language characters can be entered correctly from the Surface hardware keypad.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations

This table shows the behavior of several combinations of operating systems, browsers, and mobile devices.

Behavior Type	Combinations	Description
Arrow keys display	Android, iOS, and WinRT	The background of Up and down arrow keys turn black when you select them in the AlarmDBViewer control. This behavior causes difficulty in using Alarm Viewer controls. **Alternative Solution** There is no alternative solution.
Selecting an Alarm Viewer Control entry	Android, iOS, and WinRT	A black icon appears when you select any entry in the Alarm Viewer control box and right click to perform a freeze action. This behavior causes difficulty in using Alarm Viewer Controls. **Alternative Solution** There is no alternative solution.
Key script execution	Safari on iOS or Opera on Android tablets	Key scripts with Ctrl and Alt key combinations do not execute the associated key scripts. The built-in software keyboards in these devices do not have Ctrl nor Alt keys. **Alternative Solution** Modify your scripted shortcut key combinations to account for this known limitation of tablet device keyboards.
Invoking InTouch keyboard	Safari on iPad or Opera on Android tablets	User input animations do not consistently start the Windows keyboard. **Alternative Solution** Configure the InTouch application to use the InTouch keyboard if this is an existing application. For new applications, the recommended solution is to use the device's built-in keyboard.
Browser address bar	Chrome on Android or Chrome on iPad tablets	A browser address bar blocks the WindowViewer title bar and menu bar. **Alternative Solution** Upgrade to latest version of Google Chrome and use browser in Full Screen mode. Older versions of Google Chrome do not support Full Screen mode. Adjusting the browser to use a higher resolution may help to make the menu bar viewable.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Behavior Type	Combinations	Description
Function keys	Safari on iOS Opera on Android tablets	iOS and Android keyboards do not support function keys F1 to F12, or Ctrl and Alt key combinations. **Alternative Solution** Use different keyboard shortcuts.
Invoking InTouch keyboard	Safari on iPad or Opera, Firefox, and Chrome on Android tablets	The InTouch keyboard shows at the center of the InTouch window, which can cause the keyboard to appear off of the screen. **Alternative Solution** Scroll to the center of the InTouch window using the browser scroll bars to access the InTouch keyboard if this happens. For new InTouch applications, the recommended solution is to use the device's built-in keyboard.
Playing sound files	Safari on iOS or Internet Explorer version 11.0	Sound WAV files cannot be heard using the PlaySound() method in an InTouch script when the application is viewed with the Safari or Internet Explorer 11.0 web browsers. **Alternative Solution** Use a different web browser to view an InTouch application.
Slider animation	Safari on iPad or Opera, Firefox, and Chrome on Android tablets	ArchestrA slider graphic animation does not move smoothly. Slider movement is impaired but windows can be scrolled.
Slider animation	Safari on iPad or Opera on Android tablets	The slider control of ArchestrA slider graphics cannot be moved. **Alternative Solution** There is no alternative solution.
Display	Safari and Chrome on iPad or Opera on Android tablets	The InTouch Access Anywhere keyboard icon overlays the WindowViewer menu bar. **Alternative Solution** Zoom in and pan or set your browser to a different resolution.
Numeric input	Safari on iPad or Chrome and Opera on a desktop computer	Typing numbers very rapidly from the InTouch numeric pop-up keypad can result in incorrect values appearing in the display.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Behavior Type	Combinations	Description
Key combinations	All browsers on mobile devices except Internet Explorer 10 on WinRT	Ctrl+Shift key combinations do not work for script shortcuts and key scripts. **Alternative Solution** Use different key combinations for keyboard shortcuts.
Key combinations	All browsers except Internet Explorer 10 on WinRT	Certain mouse events or key combinations do not produce the expected result • Working, all browsers On Right Key Up • Not working, all browsers While Left Key Down Right Click While Right Key Down On Right Key Double Click • Behavior of a Chrome browser on an iPad While Left Key Down: The cursor bubble appears on a long press, which means an extended left click is not possible. On Right Key Double Click: The cursor bubble must appear in consecutive selections, which does not result in a double click. On Right Click: Right click functions only when unselecting; function is closer to On Right Up. • While Mouse Over: This behavior works, but if you move your stylus or finger off of the screen, the mouse-over behavior persists.

Running InTouch Access Anywhere: common behaviors

This table describes behaviors that are common to all operating systems and all browsers.

Behavior Type	Description
Lost network connection	When a network connection is lost between a ViewApp in a browser and InTouch Access Anywhere, no notification appears to indicate the loss of the network connection. **Alternative Solution** Add a timer or clock display that shows the current time to provide an indication of the status of the network connection between the application and the InTouch Access Anywhere Server.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Behavior Type	Description
TSE script functions return values	TSE script functions such as TseGetClientId() and TseGetClientNodeName() always return 'AccessAnywhere' as the Client ID and Node Name. This is because all RDP sessions are opened at the Terminal Server node by InTouch Access Anywhere on behalf of the clients. **Alternative Solution** There is no alternative solution.
Updating applications using NAD	When you update an application from a Network Application Development (NAD) master, a message appears in the NAD client computer but not on the host computer. You can make a change on the NAD master, but because notification does not work that change will not load into the client copy. **Alternative Solution** Refresh your session to see the changes. When you open a new session the latest version of the application will appear.
I/O servers	When InTouch requests to start I/O servers, if you click No, the TSE session logs off. **Alternative Solution** Click **Yes** when prompted to launch the I/O servers.
Secure access - certificates	If a trusted certificate is not installed when connecting to the Secure Gateway, you may see a warning message containing information that the site's security certificate is untrusted based on your browser's security settings. **Alternative Solution** Administrators should install a trusted certificate on the InTouch Access Anywhere server or Secure Gateway.
Mouse events	For all browsers: Tablet devices do not support mouse center click events. Script animations using the center button trigger type will not execute. **Alternative Solution** Modify your applications to use a script trigger other than a mouse center click event.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Behavior Type	Description
Dynamic Resolution Changes (DRC)	You will experience problems changing the size of an application window while running InTouch Access Anywhere with the **Fit to browser** and **Fit To Window** Advanced Options if DRC is not properly enabled and configured. **Solution** Ask your InTouch Access Anywhere administrator to complete this procedure, which enables the InTouch Dynamic Resolution Changes (DRC) feature. 1. Connect to the InTouch Access Anywhere server in a remote desktop session (RDP). 2. From InTouch Application Manager, select **Tools** from the menu bar and select **Node Properties**. 3. Select the **Resolution** tab. 4. Select the **Allow WindowViewer to dynamically change resolution** and **Convert to screen video resolution** options. **Note:** The *InTouch Access Anywhere Administration Guide* includes information to enable DRC on the host computer running WindowViewer.
Display on Devices with Resolution Below 800x600	InTouch Access Anywhere uses a minimum resolution of 800x600, which may not display as expected or desired on devices with a resolution below 800x600. **Alternative Solution** Ask your InTouch Access Anywhere administrator to modify the minDesktopWidth and minDesktopHeight settings appropriately.
InTouch Access Anywhere Reset	For all browsers: If you attempt to connect to the InTouch Access Anywhere server after clicking **Reset** on the landing page, the connection times out and fails, and a "Session disconnected" message appears. **Alternative Solution** Refresh the page after clicking **Reset.** The connection succeeds.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Chapter 4

About deployment

The deployment process instantiates objects to the runtime environment. Deployment includes installing the necessary software on a target computer to support objects. You must deploy the objects (AppObjects, Area and Platform objects, etc.) before you can run a ViewApp.

You can deploy and test your objects at any time during development. When you are ready to test or run a ViewApp in production, you deploy the Galaxy. You can see what your application looks like in a Deployment view or a Model view. Both views show you the structure of your application.

Deployment	Prepare to deploy a ViewApp	About deploying a ViewApp	Deploy an updated ViewApp	About undeploying a ViewApp

Prepare to deploy a ViewApp

Deploying a Galaxy copies objects from the computer hosting your IDE to another computer that provides an environment to run a ViewApp. Deployment makes your objects functional. Until you deploy your IDE configuration environment, changes you make to a ViewApp do not appear in the running application. This section describes a set of preparation tasks you need to complete before deploying objects to run your ViewApp.

Prepare to deploy a ViewApp: security

Galaxy security includes a set of deployment permissions that can be assigned to user roles based on the different deployment tasks they need to perform. Restricting deployment permissions to only those user roles that need them to complete their work protects your ViewApps.

Some typical user roles that interact with a ViewApp are:

User Role	ViewApp Work Responsibilities	Expected Deployment Permissions
Designer	Designs and builds the component parts of a ViewApp.	Granted all deployment permissions.
Developer	Integrates objects, graphics, and scripts to create and test a ViewApp	Granted all deployment permissions.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Maintenance	Updates a ViewApp to accurately reflect changes in physical plant processes or equipment.	Can Deploy/Undeploy Application Objects. Can Deploy/Undeploy Device Integration Objects. Can Mark an Object as Undeployed.
Operator	Interacts with a running ViewApp to monitor and manage plant processes.	None

Before assigning deployment permissions to the different user roles that will interact with a ViewApp, you should have defined users or security groups required by the security authentication mode you selected.

To set deployment security

1. Open the System Platform IDE.
2. From the ribbon, select **Galaxy**, then **Configure**, and then **Security**.
3. The **Security** page appears with the **Authentication mode** tab showing
4. Select the **Roles** tab and expand the **Deployment Permissions** item from the **General permissions** list.
5. For each user role you want to assign deployment permissions to or remove deployment permissions from:
   a. Select the role in the **Define security roles** list.
   b. Select or clear the **Deployment Permissions** to fit the role's needs.
6. Select **Save** to save your changes.

Prepare to deploy a ViewApp: galaxy status

You should view the status of your Galaxy's objects before you deploy a ViewApp to determine if there are any objects that are in warning or error state.

A red icon appears to the left of objects containing errors or warnings. To see an object's error or warning messages, right-click on it and select the **Properties** option. The **Errors/Warnings** tab lists any messages.

To determine the status of a Galaxy

1. Connect to the Galaxy hosting the Viewapp that you want to deploy.
2. From the ribbon, select **Galaxy**.
3. At the bottom left of the windoww, select **License.** The **About** page appears.
4. The **Galaxy Status** area shows the number of instances, templates, deployed instances with pending changes, undeployed instances with changes, objects in an error or warning state, objects that are checked out, and objects you have checked out.
5. Click **OK** to close the **Galaxy Status** dialog box.

Assign an engine to a ViewApp

Before deploying ViewApps, you need to assign each one to the ViewEngine (for access from the Application Manager) or WebViewEngine (for browser access) that it will run on. As you assign ViewApps, consider the I/O, memory, and CPU needs of each one. Assign your ViewApps to distribute these loads evenly across the platforms in the Galaxy environment at runtime.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

The operating system determines how engines are assigned to individual CPU cores and attempts to load balance across CPU and other computing resources on the platform. If you need to change which platforms and engines VIewApps are assigned to in order to improve load balancing, undeploy the ViewApps and reassign them to different ViewEngines and WebViewEngines.

If you are using redundant AppEngines, you can control how objects are loaded at runtime.

Prepare to deploy a ViewApp: assign a ViewApp to an engine

Once you are ready to deploy a ViewApp, you must assign the derived ViewApp template to the ViewEngine or WebViewEngine that it will run on. This creates an instance from the derived template and assigns the instance to the ViewEngine or WebViewEngine.

To make a ViewApp available to users from the desktop client, assign it to a ViewEngine. These ViewApps are started from the Application Manager.

To make a ViewApp available to users using the OMI web client in a web browser, assign it to a WebViewEngine.

If you want a VIewApp to be available from both the Application Manager and the OMI web client, assign the derived template to both a ViewEngine and a WebViewEngine. This creates two instances of the ViewApp, one running on each engine.

Assign a ViewApp to an engine

To assign a ViewApp to a ViewEngine or WebViewEngine:

1. Open the IDE and display both the **Templates** pane and the **Deployment** pane.
2. In the **Templates** pane, navigate to the folder containing the ViewApp derived template.
3. Right-click the desired ViewApp template and select **New**, then **Instance** from the context menu.

   This creates an instance of the ViewApp in the **Unassigned Host** section of the **Deployment** pane.
4. Drag the new instance to the desired ViewEngine or WebViewEngine.

   Note: You can also drag the derived ViewApp Template to the desired engine in the **Deployment** pane. This creates a new instance and places it in the **Deployment** pane.

If you want the ViewApp to be available to users from both the Application Manager and the OMI web client, repeat the procedure, assigning the ViewApp to both a ViewEngine and a WebViewEngine.

About deploying a ViewApp

You deploy object instances for three reasons:

- Testing.
- Placing the application into production to process field data.
- Updating an existing application with changes.

When you are ready to deploy, make sure the following conditions are met:

- Bootstrap software is installed on the target computers.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- The objects being deployed are not in an error state in the Galaxy database.
- You created, configured, and checked in objects to the Galaxy.
- Objects are assigned to a host.
- The object's host is already deployed. A cascade deploy operation, which deploys a hierarchy of objects, deploys all objects in the correct order. A cascade operation deploys an object’s host before the object is deployed.
- Any associated script libraries are ready for use on the target computer.

Deploy a ViewApp

To deploy a ViewApp

1. Select the object or objects that you want to deploy in an Application view.

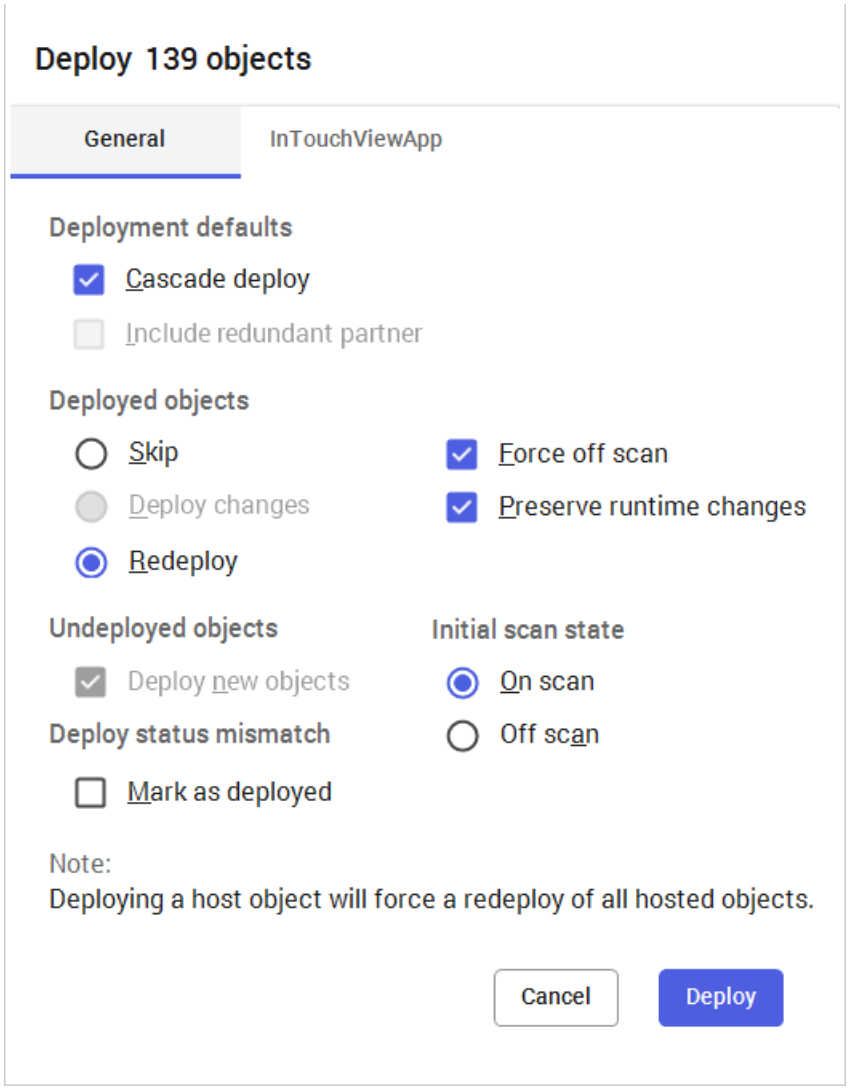
   Note: Deployment view is often best for this purpose, as it shows which objects, if any, are beneath the selected object in the deployment order.

2. On the **Home** ribbon, in the **Deploy** area, select **Deploy**.

   You can also deploy a selected object by right-clicking and selecting **Deploy** from the shortcut menu.

   The **Deploy** dialog box appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. Select one or more of the following:
   - **Cascade deploy**: Select this check box to deploy the object selected for deployment as well as any objects it hosts. This option is selected by default if the object is a host. If you are deploying an individual host object, clear the check box. Objects being deployed across multiple platforms are deployed in parallel.
   - **Include redundant partner**: Select this check box to also deploy an AppEngine's redundancy partner object. This option is selected and unavailable when the redundant engine has pending configuration changes or software updates.
4. In the **Deployed objects** area, select one or more of the following options. These options are not available if the selected object has not been deployed before.
   - **Skip**: If one of the objects you are deploying is currently deployed, selecting **Skip** makes no changes to the already-deployed object.
   - **Deploy changes**: If one of the objects you are deploying is currently deployed, this option updates the object in question with new configuration data. The runtime state from the runtime file is preserved and the state is modified with any changes.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **Redeploy**: If one of the objects you are deploying is currently deployed, this option deploys the same version as previously deployed. For example, use this option to redeploy an object that is corrupted on the target computer.
- **Force off scan**: If one of the objects you are deploying is currently deployed, this option sets the target object to off scan before deployment occurs.
- **Preserve runtime changes**: Selecting this option will produce different results, depending on whether you have chosen **Deploy changes** or **Deploy original**, if values were changed at runtime, and if configuration values were changed in the IDE.
  - If a value was changed in runtime and **Deploy changes** and this option are checked, the object is deployed with the last runtime value, even if you change the configured value in the IDE.
  - If a value was not changed during runtime and **Deploy changes** and this option are checked, the object is deployed with the last configured value from the IDE.
  - If a value was changed in runtime and **Redeploy original** and this option are checked, the object is deployed with the last runtime value, even if you change the configured value in the IDE.
  - If a value was not changed during runtime and **Redeploy original** and this option are checked, the object is deployed with the original configured value from the IDE. If you configured a new value in the IDE, the prior configured value is used.

5. In the **Undeployed objects** area, select **Deploy new objects** to start a normal deployment.
6. In the **Deploy status mismatch** area, select **Mark as deployed** to mark the object as deployed in the Galaxy. A mismatch happens when the object is previously deployed to a target node, but the Galaxy shows the object is undeployed. Clear this option to redeploy the object to the target node.
7. In the **Initial scan state** area, select one of the following:
   - **On Scan**: Sets the initial scan state to on scan for the objects you are deploying. If the host of the object you are deploying is currently off scan, this setting is ignored and the object is deployed off scan. When you are deploying multiple objects, the deploy operation deploys all of the selected objects "off-scan." After all of the objects are deployed, the system sets the scan-state to "on-scan."

     Objects can run only when both the host/engine is "on scan" and the object is "on scan." If either the host/engine or the object is "off scan," the object cannot run.

     Always deploy Areas to their host AppEngines on scan. Because Areas are the primary providers to alarm clients, deploying Areas off scan results in alarms and events not being reported until they are placed on scan.
   - **Off Scan**: Sets the initial scan state to off scan for the objects you are deploying. If you deploy objects off scan, you must use the ArchestrA System Management Console Platform Manager utility to put those objects on scan and to function properly in the runtime environment.

     Note: The System Management Console controls the state of the host/engine. The ObjectViewer controls the state of the objects.

     The default scan setting is set in the **User Default** settings in the **Configure User Information** dialog box. For more information, see Configure User Preferences.
8. Click **Deploy** to deploy the objects. The **Deploy** progress box appears. When the deploy is complete, click **Close**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Determine deployment status

You can tell if you have objects that need to be deployed by looking at the icons next to the objects listed in the right-most column of the **Model** or **Deployment** views of the IDE.

Deployment status icons for most objects, including ViewApps, are:

These additional icons may appear for redundant AppEngines:

Both redundant AppEngines are not deployed

This AppEngine is not deployed, but its redundant partner is deployed.

This AppEngine is deployed, but its redundant partner is not deployed.

Both redundant AppEngines have pending configuration updates.

This AppEngine has a pending configuration update, but its redundant partner is up to date.

This AppEngine is up to date, but its redundant partner has a pending configuration update.

Both redundant AppEngines have pending software updates.

This AppEngine has a pending software update, but its redundant partner is up to date.

This AppEngine is up to date, but its redundant partner has a pending software update.

For directions on deploying view apps, see Deploying a ViewApp.

Deploy an updated ViewApp

When deploying a ViewApp for the first time, the Galaxy Repository compiles all ViewApp configuration data, the display module, and binaries and places them in a destination folder with the server's time stamp file. When you deploy the ViewApp again, only the files changed after the time stamp from last deploy operation are deployed again.

The following list summarizes the steps to deploy an updated ViewApp.

1. The Galaxy Repository verifies if any files in the ViewApp's destination folder contain any files that have changed since the previous deployment.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. If no files have changed since the previous deployment, then the client will receive all files from CVR and deploy them to the destination folder, and also set the time stamp (server time) into the application folder.
3. If files have been changed since the previous deployment, then read the time stamp from destination folder, send it to the server, get delta changes, along with deleted files information, and stage them to the destination folder. Remove the deleted graphicl files
4. The Folder Synchronizer synchronizes the newly staged files to the computer running the ViewApp.

About undeploying a ViewApp

You may need to undeploy one of more objects. Undeploying removes the ViewApp instance and its associated data from the target node.

Before you delete or restore a Galaxy, undeploy all objects in the Galaxy.

Undeploying can fail if the target object has objects assigned to it. Make sure you select **Cascade Undeploy** in the **Undeploy** dialog box.

Undeploy a ViewApp

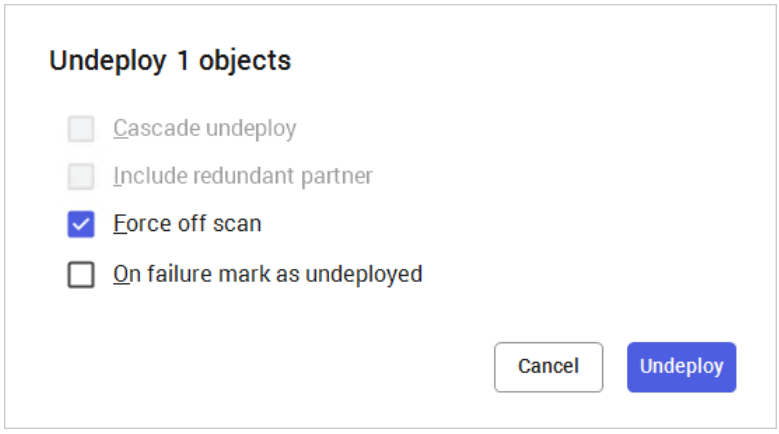
You may need to undeploy one of more objects. Undeploying removes the ViewApp's files and its associated data from the target node.

Undeploying can fail if the target object has objects assigned to it.

Before you delete or restore a Galaxy, undeploy all objects in the Galaxy. Make sure you select **Cascade undeploy** in the **Undeploy** dialog box.

To undeploy a ViewApp

1. Make sure the ViewApp you want to undeploy is not running.
2. From the application view area of the IDE, select the object or objects to undeploy.
3. From the **Home** ribbon, in the **Deploy** area, select **Undeploy**. The **Undeploy** dialog box appears.



The top of the dialog box shows the total number of objects being undeployed. This includes the objects you selected and, if you select the **Cascade undeploy** option, all objects that are beneath a selected object in the hierarchy and will be undeployed as a result.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

4. Select one or more of the following options. Some of these options may not be available, depending on the kinds of objects you selected to undeploy.
   - **Cascade undeploy**: Select to undeploy the selected object and any objects it hosts.
   - **Include redundant partner**: Select to also undeploy an AppEngine's redundancy partner object.

     Note: The Primary AppEngine in a redundant pair can be undeployed alone because objects hosted by it run on the deployed Backup AppEngine, which becomes Active.
   - **Force off scan**: If one of the objects you are undeploying is currently on scan, selecting **Force off scan** sets the target object to off scan before undeployment. If you do not select **Force off scan** and the target object is on scan, the undeployment operation fails.
   - **On failure mark as undeployed**: Marks the object as undeployed in the Galaxy if the object targeted for undeployment is not found.
5. Click **Undeploy** to undeploy the objects.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Chapter 5

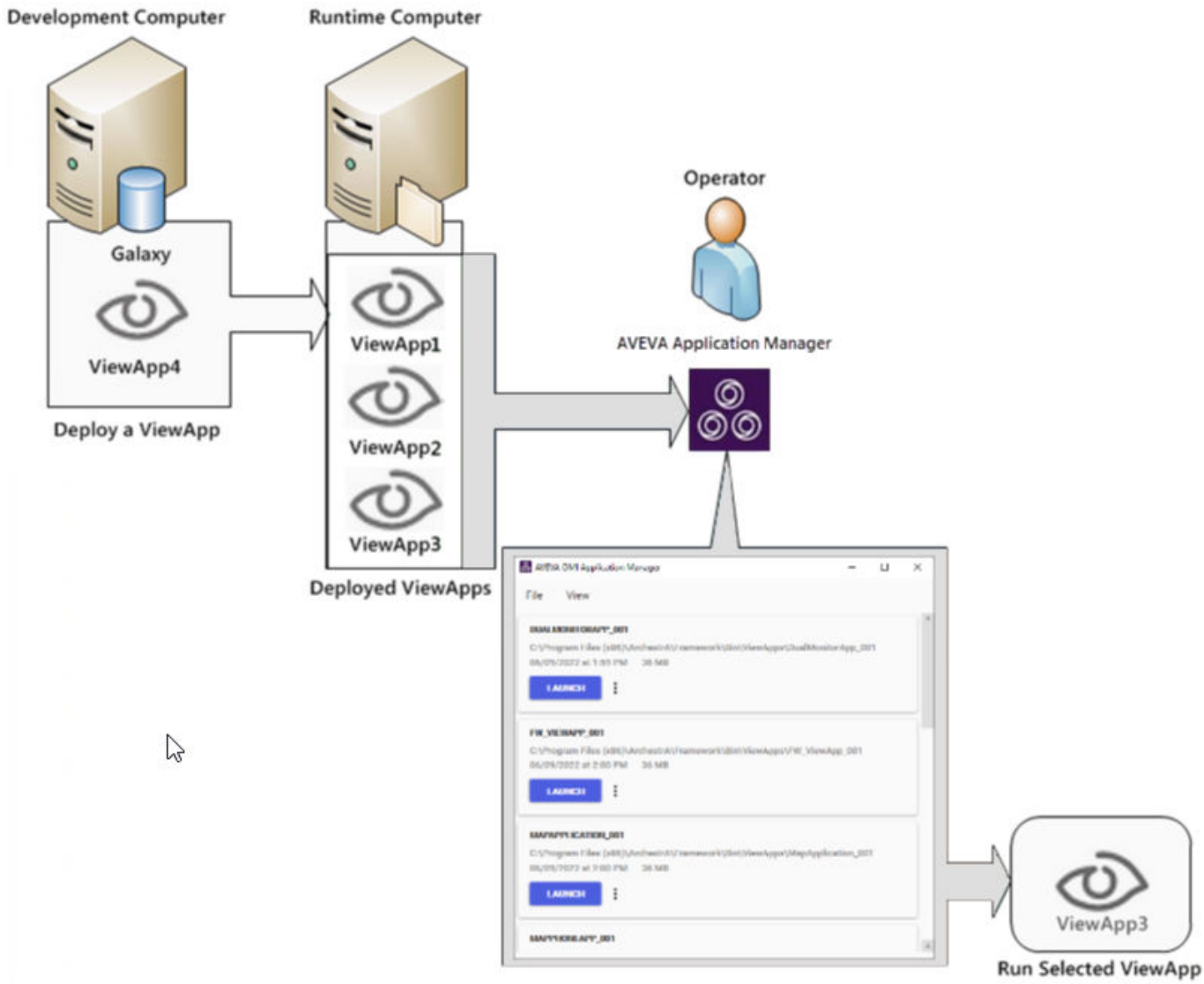
Run a ViewApp

Run a ViewApp	View deployed ViewApps	About the InTouch web client	Launch a ViewApp in a web browser
	Launch ViewApp in the OMI Web Client using connected experience, Launch ViewApp in the OMI Web Client using connected experience	Log in with AVEVA Identity Manager	Open a ViewApp
	Search a ViewApp NavTree	Process late data at runtime	Enable historical playback

View deployed ViewApps

The AVEVA OMI Application Manager lists the ViewApps deployed to a runtime computer or terminal server. When a user selects the **LAUNCH** button of a listed ViewApp, the ViewApp starts running on the computer.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Launch a ViewApp

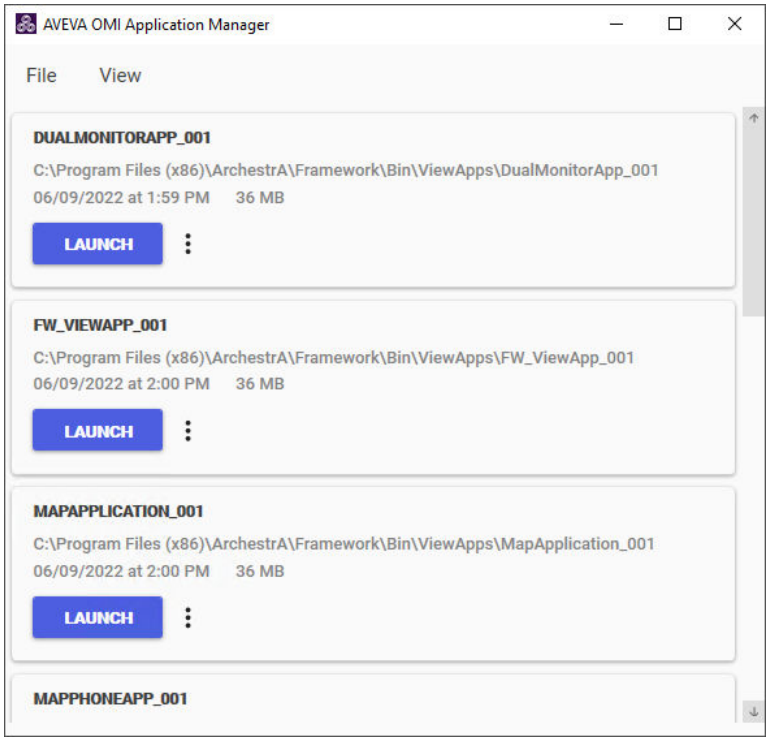
You launch a deployed ViewApp from the AVEVA OMI Application Manager. You start the Application Manager by selecting its shortcut icon on the Windows desktop or as a Start Menu entry. To see a video of this process, see Launch a ViewApp.

To launch a ViewApp

1. Show the desktop of the computer where the ViewApp has been deployed.
2. Select the AVEVA OMI Application Manager desktop shortcut.

   AVEVA OMI Application Manager appears with a list of ViewApps currently deployed on the computer. ViewApps are listed in descending alphabetic order by ViewApp name.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



For more information about the fields of the AVEVA OMI Application Manager, see About the ViewApps Application Manager.

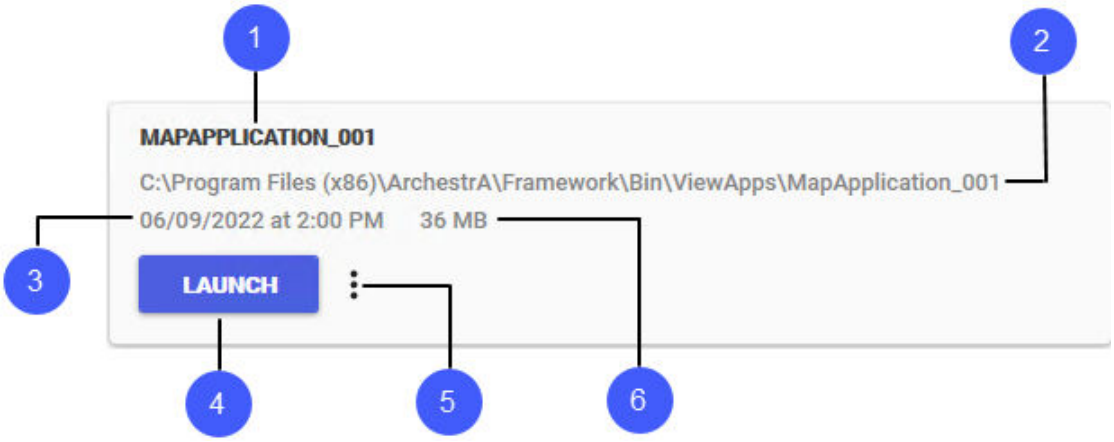
3. Find the ViewApp you want to run.
4. Select the ViewApp's **LAUNCH** button.

   The word **LAUNCHING** appears briefly followed immediately by an AVEVA OMI splash screen.

   The ViewApp appears on the screen and as an item on the computer's Windows task bar.

About the ViewApps Application Manager

The AVEVA OMI Application Manager lists all current ViewApp instances deployed to a computer and includes information about each ViewApp. Each deployed ViewApp appears as a separate item in the list.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

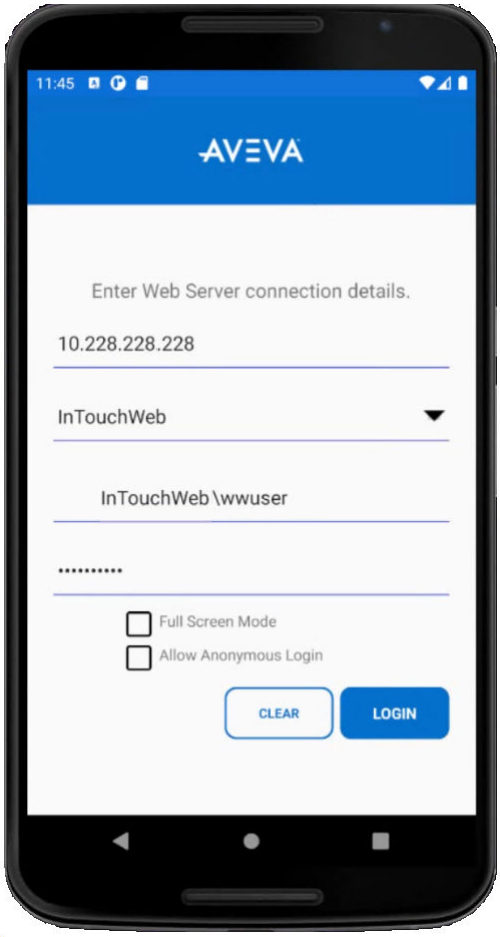
1	ViewApp name	Name of the ViewApp deployed to the computer.
2	Deployed folder path	Path to the folder of the ViewApp deployed on the computer. The Galaxy name and ViewApp name are assigned as the folder name containing the ViewApp's files in the form *Galaxy_Name-ViewApp_Name.*
3	Date and time	The most recent date and time the ViewApp's derived template was edited. The date and time is displayed in the regional format of the logged in Windows user at the time the AVEVA OMI Application Manager is launched.
4	LAUNCH button	Button to run the selected ViewApp. Click or tap the button to start the ViewApp running on the computer. The word **LAUNCHING** appears briefly and fades out after several seconds when the ViewApp starts running.
5	Operating Mode	Drop-down list to select the operating mode of the ViewApp as either Read/Write (default) or Read Only.
6	Disk space usage	Amount of disk space used to store the deployed ViewApp files on the computer.

Display a web ViewApp on a mobile device

As of the System Platform 2023 release, the AVEVA Mobile Operations application has been updated to show OMI web ViewApps on a mobile device.

From your mobile device, log in to your web server.

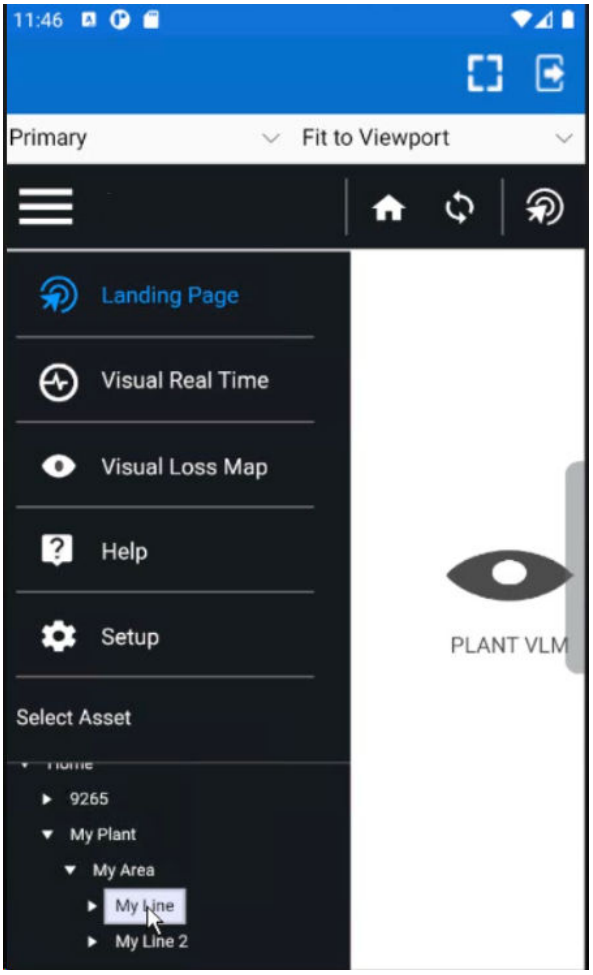
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



After logging in, the header area of screen shows the primary screen of the ViewApp and the view mode.

Selecting tabs at the left and right sides of the device screen show slide in panes. The following example shows the navigation tree of the ViewApp within the slide-in pane.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Responsive layouts work well with ViewApps intended to be displayed on mobile devices. The app also includes options to change screens and view modes.

About the InTouch web client

You can view AVEVA InTouch HMI applications by using the InTouch Web Client in a supported web browser.The Web Client is also available on mobile platforms. A built-in web server provides web browsers access to InTouch HMI applications from any Microsoft Windows client or server operating system without the use of Remote Desktop Protocol (RDP) or Internet Information Services (IIS) for Microsoft Windows® Server.

The InTouch Web Client also includes workspaces. Workspaces enable users to configure ViewApps while viewing them in the web browser. Workspaces are individual, so each user can configure their own ViewApps to meet their individual needs.

Launch a ViewApp in a web browser

If an OMI ViewApp has been deployed to a WebViewEngine, you can open the ViewApp in your web browser. To do so:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Notes: The first time you try to open the OMI web client on a remote device, you may need to install a security certificate to enable it. See Install a security certificate on Windows for more information.

1. In a supported web browser, go to the URL for the OMI web client:
   - On the machine where the WebViewEngine is deployed, the URL is `https://localhost/omi`. You can also use either of the two formats below.
   - On a machine on the same domain as the one where the WebViewEngine is deployed, the URL is `https://<node_name>/omi`, where `<node_name>` is the name of the platform where the WebViewEngine is deployed.
   - On a machine that is not on the same domain as the WebViewEngine, the URL is `https://<fqdn>/omi`, where `<fqdn>` is the fully-qualified domain name of the platform where the WebViewEngine is deployed.

   The OMI web client sign in page opens.

   Note: If your installation uses AVEVA Operations Control and the connected experience, you may or may not be asked to log in before you can open the OMI web client. see Launch using the connected experience, Launch using the connected experience for more information.

2. Enter your user name and password.

   The OMI web client home page opens, listing all of the OMI ViewApps available through the web client.

3. Select the ViewApp to run.

   When the ViewApp opens, the user name that you are logged in as shows in the top-right corner of the page.

Note: Unlike the OMI desktop client, the OMI web client does not notify you if there is an available updated version of the ViewApp that you are running. If you refresh the ViewApp web page, or return to the OMI web client home page and then re-open the ViewApp, the software checks for a newer version of the ViewApp and downloads any updates that have been made to it. We recommend you do this regularly to make sure you have the latest version.

Using ViewApp in a web client vs. on the desktop

In most ways, an OMI ViewApp that you have opened in the OMI web client behaves just as it does in the OMI desktop client. However, there is an icon at the top right of the browser page which lets you use a few features which do not exist or are different in the desktop client.

Selecting this icon lets you choose from these features:

- **Full screen**: Changes the OMI web client page to full screen mode. To leave full screen mode, press **Esc**.
- **Log out**: Closes the ViewApp and logs you out of the web client. It opens the AVEVA Identity Manager login page so you or someone else can log back in as a different user, if desired. When you log back in, you see the web client home page so you can start any VIewApp.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **Select screen**: The web client can show only one screen of a multi-screen layout at a time. This area shows the current screen name. If your ViewApp uses a multi-screen layout, you can select the screen name or the down-arrow next to it, then select the screen to view.

Panning and Zooming in the OMI web client

Panning and zooming using the mouse works differently in the OMI web client than in the OMI desktop client. It also works differently than panning and zooming in most web applications. The table below shows how various mouse actions work in the two OMI clients.

Mouse action	OMI desktop client	OMI web client
Scroll up and down	Zooms in and out	Pans the current pane up and down
SHIFT plus scroll up and down	Zooms in and out	Pans the current pane left and right
CTRL plus scroll up and down	Zooms in and out	Zooms the current pane in and out
Drag with center button down	Pans	No effect

Notes:

- The maximum zoom level is 500%.
- The current zoom level is shown at the bottom left of the layout.
- Using the keyboard to zoom (**CTRL SHIFT +** or **CTRL -**), zooms the entire browser window, not just the ViewApp display.
- You can also pan using the scroll bars that appear if the content does not fit in the pane at the current zoom level.

Troubleshoot OMI web client connection issues

If you have a problem starting or working with the OMI web client, following the steps below may help you find and resolve the issue.

1. Check Known Issues in the OMI web client. If your symptom is listed there, follow the recommended workaround.
2. On the System Management Server (SMS) host, start the Configurator and make sure the settings for the web client are correct:
   a. Under **Common Platform**, make sure there is a green checkmark next to **System Management Server**. If there isn't, select **System Management Server** and select the option to make this host the SMS.
   b. Select **System Management Server**, then select **Advanced**.
   c. Select the **Ports** tab, and make sure the **HTTPS Port** is set to 443. Select **OK**.
   d. Under **AVEVA System Platform**, there should be a green checkmark next to **Identity Manager Registration**. If there isn't:
      i. Select **Identity Manager Registration**.
      ii. Select **Configure**.

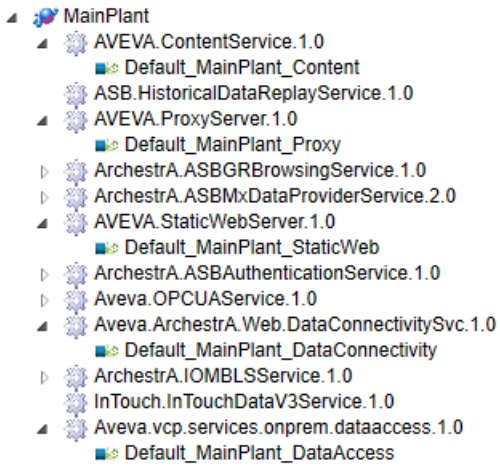
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

iii. Select **Close**.

Note: If the SMS was not already defined, after performing the steps above, go to the other nodes that have System Platform installed and use the Configurator to connect them to the SMS.

3. On the device that is having the problem, clear the browser cache (select the option to clear the cache for all time). Then close and re-open the browser and try to open the web client again.
4. On the platform that hosts the WebViewEngine, make sure that all five services that are supposed to be deployed and started when the WebViewEngine is deployed are running. These services are:
   - ArchestrA.Web.DataConnectivitySvc.exe
   - vcp.services.onprem.contentserver.exe
   - vcp.services.onprem.dataaccess.exe
   - vcp.services.onprem.frontdoor.exe
   - vcp.services.onprem..webserver.exe

   You can check for these services either on the **Details** tab of the Windows Task Manager or from the IDE by selecting **Galaxy** > **Configure** > **System** > **Services**. When you expand the tree view nodes on the **Configure Services** dialog box, each service name should show a green box to indicate that it is deployed and running. In this sample, the Galaxy name is MainPlant:



5. If the services are not all deployed, undeploy and redeploy the WebViewEngine. Then, repeat step 4 to confirm that the services have started.

If you cannot undeploy an OMI ViewApp

If you are unable to undeploy a ViewApp that is assigned to the WebViewEngine, follow these steps to fix the issue:

1. Follow the instructions in step 4, above, to see if the WebViewEngine services are deployed and running. If they are, go to the next step. If they aren't:
   a. If it isn't already open, open the **Configure Services** dialog box from the IDE by selecting **Galaxy** > **Configure** > **System** > **Services**.
   b. Expand the tree view nodes to see the service names, as shown in the image above.
   c. Undeploy and redeploy the five WebViewEngine services. (The ones that are expanded in the image.)

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. Check to see if you can now undeploy the ViewApp.
3. If you still can't undeploy the ViewApp, restart the node hosting the WebViewEngine.
4. If you still cannot undeploy a ViewApp, please contact AVEVA Global Customer Support.

Launch ViewApp in the OMI Web Client using connected experience

If your installation uses AVEVA Operations Control connected experience for user authorization and security, follow these steps to start an OMI ViewApp.

To launch a ViewApp in the OMI web client using the connected experience

1. In a supported web browser, go to the URL for the OMI web client:
   - On the machine where the WebViewEngine is deployed, the URL is `https://localhost/omi`. You can also use either of the two formats below.
   - On a machine on the same domain as the one where the WebViewEngine is deployed, the URL is `https://<node_name>/omi`, where `<node_name>` is the name of the platform where the WebViewEngine is deployed.
   - On a machine that is not on the same domain as the WebViewEngine, the URL is `https://<fqdn>/omi`, where `<fqdn>` is the fully-qualified domain name of the platform where the WebViewEngine is deployed.

   If you are already logged in to another AVEVA product that uses Operation Control connected experience, you are logged in to the OMI web client automatically using Single Sign-on. Skip to step 3.
2. If you are not already logged in to a product that uses the connected experience, a sign-in window opens for you to enter your user name and password.

   After you have entered your user name and password, the OMI web client page opens.
3. The OMI web client page lists all of the OMI ViewApps available through the web client. Select the ViewApp to run.

   When the ViewApp opens, the user name that you are logged in as shows in the top-right corner of the page.

If you have a problem logging in (authentication or entitlement failure)

- Make sure your internet connection is working.
- If user authentication fails, the ViewApp opens in read-only mode after you click OK.
- If entitlement fails, you will see a message that a valid subscription could not be found. After you click OK, the ViewApp opens in read-only mode.

Corrective action

If you receive an authentication error message, repeat the log in procedure, being sure to enter your user name and password correctly. If you still cannot sign in, contact your CONNECT administrator.

If you receive an entitlement error message, there may be a problem with your AVEVA subscription. Contact your CONNECT administrator.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

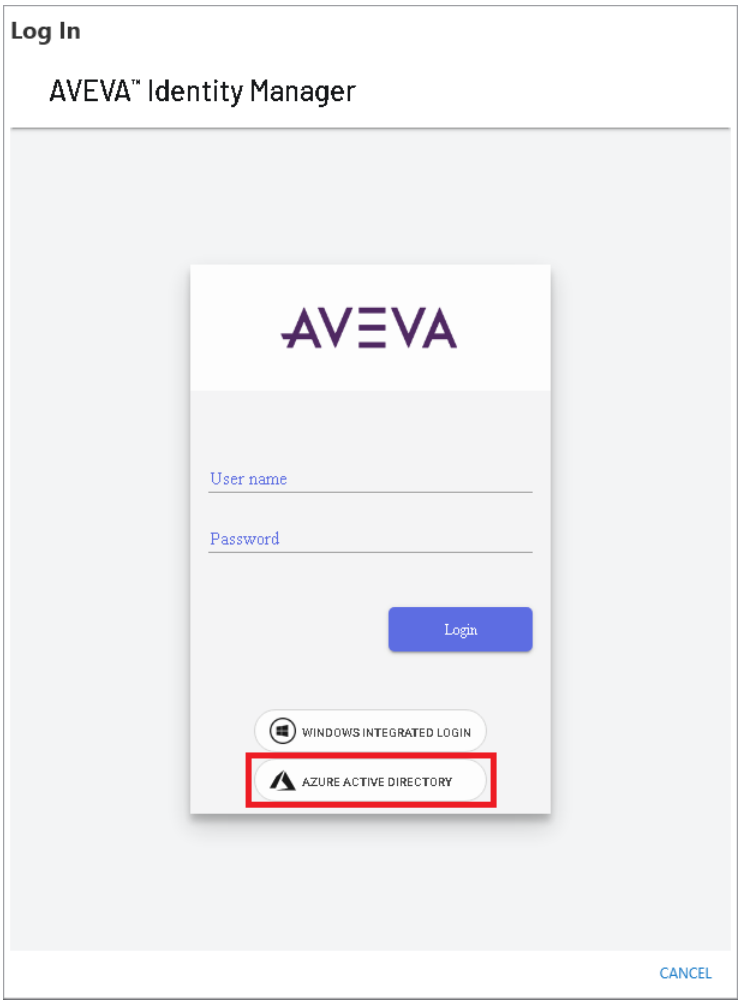
Log in with AVEVA Identity Manager

When Azure AD is configured as the identity provider, all required user log-ins will use the AVEVA Identity Manager. This applies to all System Platform components and secured operations, including:

- System Platform IDE
- Operation Control Management Console
- AVEVA OMI ViewApps
- Secured Writes
- Verified Writes

Use Azure Active Directory credentials to sign on

A logon dialog opens when you click **Login** button in the Title Bar, or when the ShowLoginDialog() script function is invoked.



To use Azure Active Directory credentials for log in

1. Select the **Azure Active Directory** button. This launches the Azure AD log-in screen.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Note: Windows Integrated Login is not supported in Controlled Release 2.

2. Enter your credentials and follow the on-screen instructions.

   You are logged into the ViewApp when your credentials are validated. A successful log in requires that:

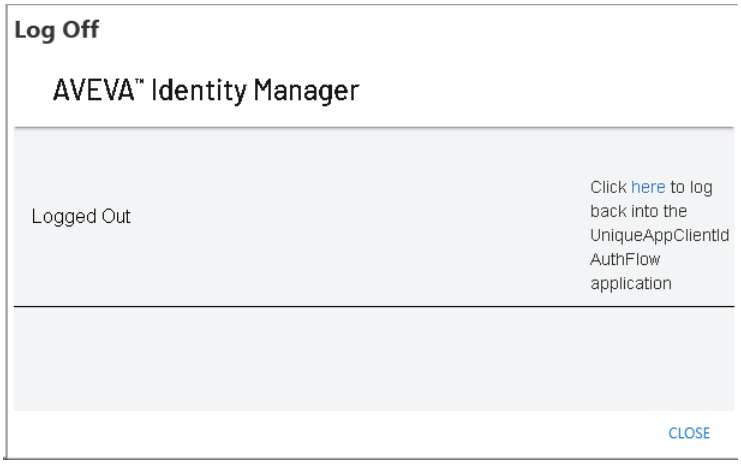
   - At least one security role is assigned to you.
   - Group membership for Azure AD is valid.

   Authentication fails if the group was not added for your user role, and the following error is displayed:

```
The login attempt failed.
The user must have at least one role assigned to it to be able to log in.
```

Log Off via Azure AD

To log out of the ViewApp after signing with Azure AD credentials, click the Logoff button in the Title Bar to display the Logoff dialog. This signifies that the user has ended their session. The dialog is also displayed when the Logoff() script function is invoked.

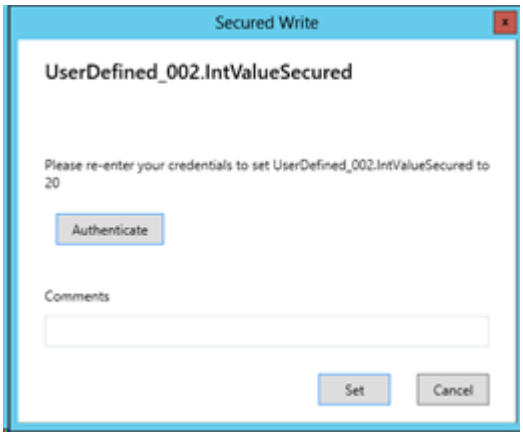


Use Azure Active Directory credentials for secured write authentication

When security is configured with Azure AD as an authentication provider, users can use their Azure AD credentials to authenticate secured write operations. For general information about secured writes, see Working with the SignedWrite() function for secured and verified writes.

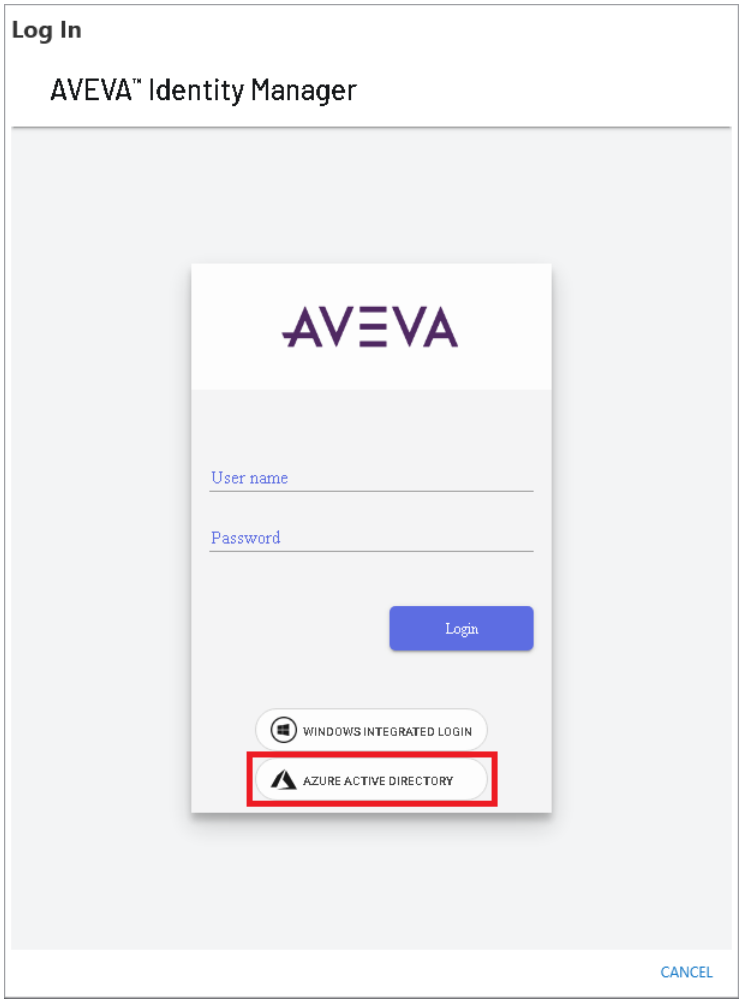
An authentication dialog opens when you attempt to write to a Secured-Write attribute, or the SignedWrite() attribute is invoked.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



To use Azure Active Directory credentials for Secured Write authentication

1. Click the **Authenticate** button. The AVEVA Identity Manager window opens.



2. Select the **Azure Active Directory** button. This launches Azure AD log-in screen.

   Note: Windows Integrated Login is not supported in Controlled Release 2.

3. Enter your credentials and follow the on-screen instructions.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

4. To complete the secured write:
   - At least one security role assigned must be assigned to you.
   - Group memberships for Azure AD must be valid.
   - You must have at least one role assigned that allows you to write to attributes.

   Authentication fails if the group was not added for your user role, and the following error is displayed:

   ```
   The login attempt failed.
   The user must have at least one role assigned to it to be able to log in.
   ```

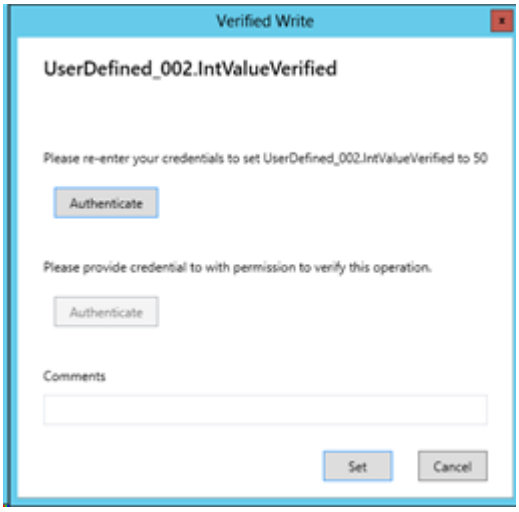
5. In the authentication dialog, add comments and select **Set** to verify the attribute changes.

Use Azure Active Directory credentials for verified write authentication

When security is configured with Azure AD as an authentication provider, users can use their Azure AD credentials to authenticate verified write operations. For general information about verified writes, see Working with the SignedWrite() function for secured and verified writes.

Two users with valid Azure AD permissions are required to write to a verified-write attribute.

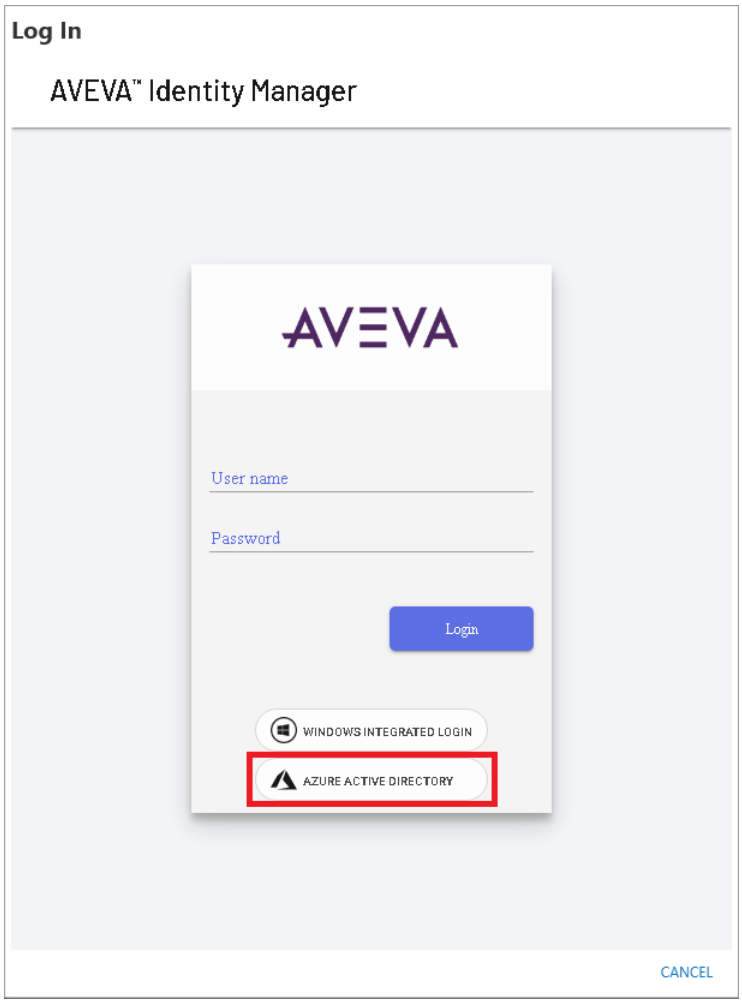
An authentication dialog opens when you attempt to write to a Verified-Write attribute.



To use Azure Active Directory credentials for Verified Write authentication

1. User 1 clicks the **Authenticate** button in the authentication dialog. The AVEVA Identity Manager window opens.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. User 1 selects the **Azure Active Directory** button. This launches Azure AD log-in screen.

   Note: Windows Integrated Login is not supported in Controlled Release 2.

3. User 1 enters their credentials and follows the on-screen instructions.
4. The verified write operation can proceed to the next phase if:
   - User 1 has at least one security role assigned.
   - User 1 group memberships for Azure AD must be valid.
   - User 1 has at least one role assigned that allows them to write to attributes.
5. When User 1 is successfully authenticated, the second **Authenticate** button in the Verified Write dialog is enabled.
6. User 2 clicks the enabled **Authenticate** button. The AVEVA Identity Manager window opens.
7. User 2 selects the **Azure Active Directory** button. This launches Azure AD log-in screen.

   Note: Windows Integrated Login is not supported in Controlled Release 2.

8. User 2 enters their credentials and follows the on-screen instructions.
9. The verified write operation can proceed to the verification phase if:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- User 2 has at least one security role assigned.
- User 2 group memberships for Azure AD must be valid.
- User 2 has at least one role assigned that allows them to verify attributes.

When User 2 is successfully authenticated, they can add comments in the authentication dialog. User 2 selects **Set** to verify the attribute changes.

If verification is attempted using the same credentials as user 1, the following error is displayed:

```
Operator and Verifier have to be two different users
```

Authentication fails if the group was not added for the user's role, and the following error is displayed:

```
The login attempt failed.
The user must have at least one role assigned to it to be able to log in.
```

Open a ViewApp

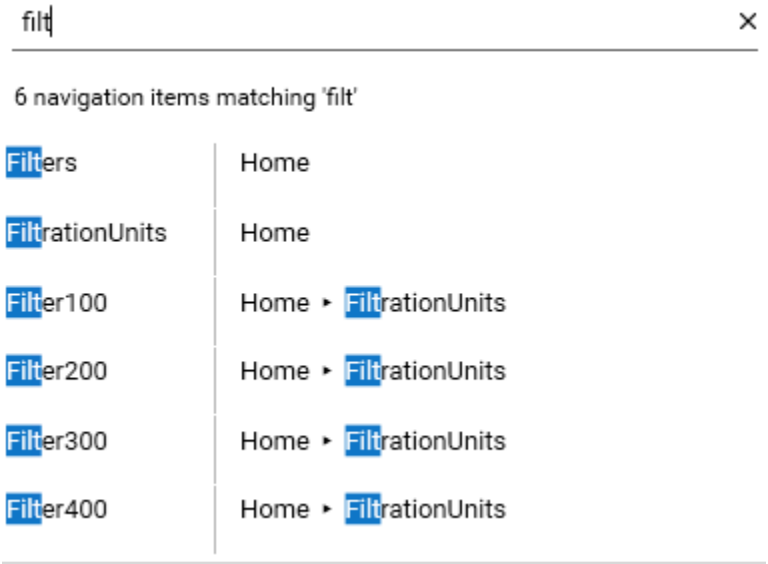
To open a ViewApp, go to the Windows Start Menu, then select either:

- **AVEVA OMI Application Manager**, located in the AVEVA System Platform folder.
- **Application Manager**, located in the AVEVA InTouch HMI folder.

Search a ViewApp NavTree

During runtime, the content of a ViewApp can be searched by entering a search string to find navigation items by their names, aliases, or paths that appear in the navigation model. A search entry field appears above the NavTree control.

While entering a search string, any matching navigation item names, aliases, or paths appear in a search results list.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

After selecting an item from the search results list, the ViewApp shows the content of the navigation item you selected.

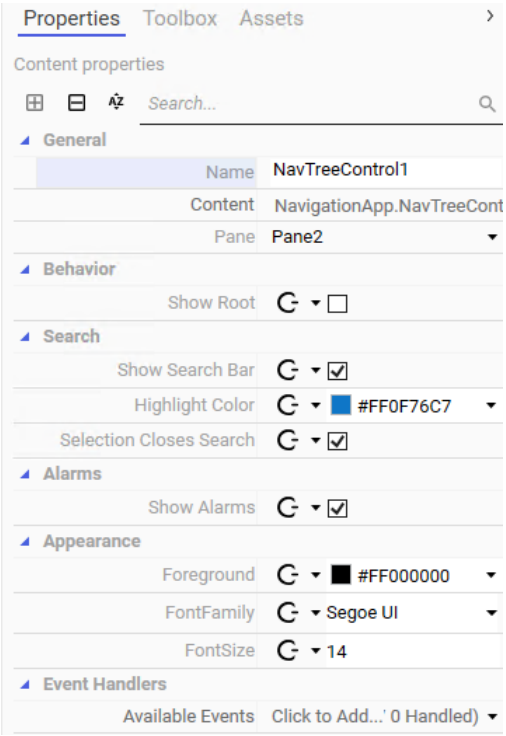
Enable runtime navigation model searches

The search field above a NavTree control can be displayed or hidden during runtime based on the value set to the **Show Search Bar** property. A search field appears when **Show Search Bar** is selected, which is the default value.

The NavTree control has other properties that affect runtime searches. See About search results for instructions to show the root of the navigation model.

To enable runtime navigation model searches

1. Open a layout or ViewApp in their respective editors.
2. Drag and drop a NavTree control onto a layout pane.
3. Select the NavTree control placed on the pane.
4. Select the **Properties** grid to show the properties of the NavTree control.

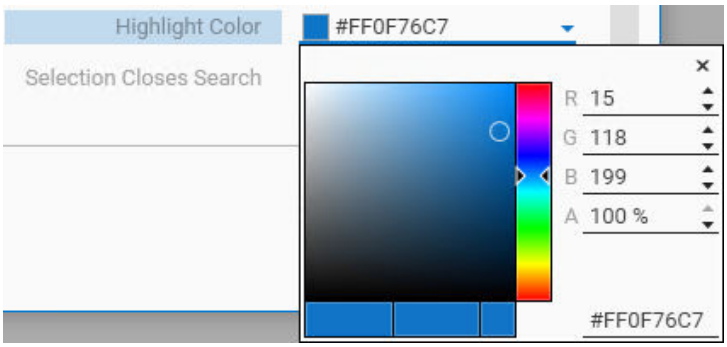


5. Select the **Show Search Bar** property to show the search bar above the navigation tree.
6. Select the **Show Alarms** property if you want to see the alarm status of navigation items that appear in the search results list.
7. Select the **Selection Closes Search** property if you want to hide the search list immediately after selecting an item from the search list.
8. If you want to change the highlight color that appears in the search results list, select the **Highlight Color** property.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

a. Place the mouse cursor over the data entry field to show a triangle icon at the right.

b. Select the triangle icon to show a color picker control.



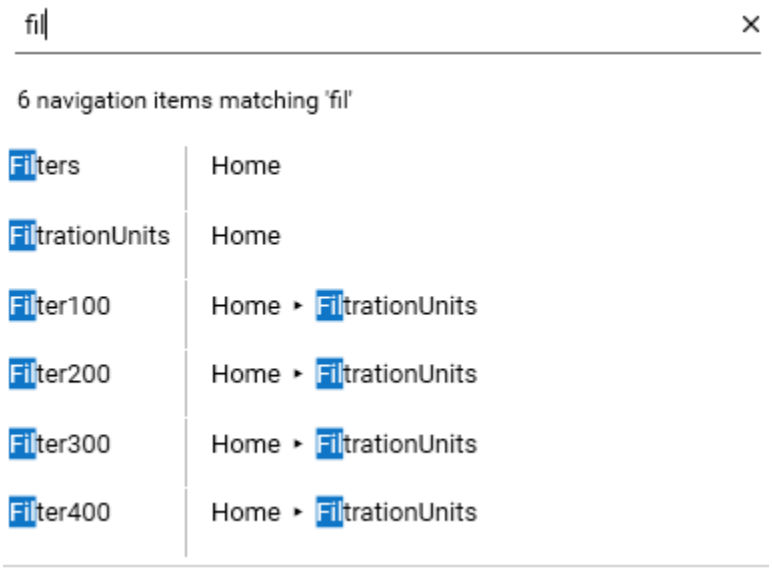
c. Select a highlight color from the picker.

9. Save your changes to the NavTree control.

About search results

After you enter a search string, a search is conducted over the entire navigation model of the running ViewApp. The search results appear in a two column list containing all navigation items that match the search string. The first column displays the navigation item name and Title as <Name (Title)>). The second column displays the path of the item in the navigation model. Highlighting identifies the portion of the item name or path that matches the search string.

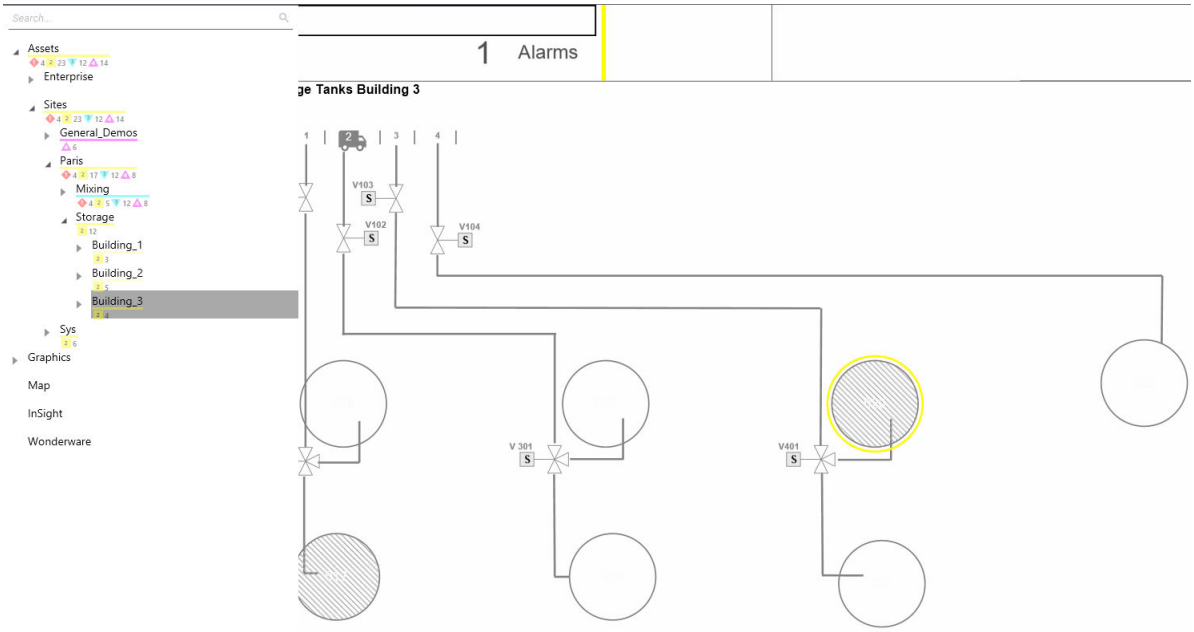
If the NavTree control is configured to show alarm propagation states, the aggregate alarm count by priority appears in the search results beneath each navigation item containing active alarms.



You can change the column widths by draqging the vertical line that separates the columns. Placing the mouse cursor directly over the path shown in the search results list shows the complete path in a tooltip.

Selecting an item from the search results shows its content in the running ViewApp. The NavTree control shows the item you selected.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Alarm Status in the Search Results List

The NavTree control includes the **Show Alarms** property to show or hide aggregated alarm counts by priority for all items of a ViewApp navigation model. The default selected value of the **Show Alarms** property shows an *adorner* or alarm border beneath a navigation item in the search results list whose path has active alarms.

The physical appearance of an alarm adorner in the search results list provides more information about the alarm

- If the color of the alarm border is opaque (solid), the alarm is on the selected (local) object. If the color of the alarm indicator has some transparency, the alarm is on a contained object.
- If enabled, the severity (S1 to S4) and status of the alarm (UNACK_ALM, UNACK_RTN, ACK_ALM) is shown graphically, along with a count for each alarm severity that has been enabled.
- The color of the blinking line beneath the navigation item name indicates the highest-severity (most urgent) alarm.

For more information about the meaning of the physical appearance of an alarm border, see Aggregated Alarm Display in AVEVA OMI Navigation.

Clear Search Results

You can hide the search results list during runtime by clearing the search string from the search entry field.

- Backspace over the search string

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Press the Esc key
- Select the Clear button at the right of the search entry field.

After clearing the search string from the entry field, the search results list disappears and the navigation model appears again.

Process late data at runtime

Data received from the Telemetry Server or other sources may arrive late, especially when using publish/subscribe models such as MQTT, with or without Sparkplug B. "Late data" refers to process data that is received after the time of the actual event. For example, an OMI ViewApp may receive data along with its VTQ timestamp some number of minutes or even hours after the triggering event has occurred.

Application Server allows late data received through a DI Object to backfill historical data in the Historian while presenting live data to OMI ViewApps. Note that value historization is done only for attributes that have the History extension enabled.

When connection to the data source is lost, historical data and alarms are backfilled once the connection is restored. References to ScanGroup items directly in scripts allow late data to be processed. For example, "me.Int1 = DiObject.Scangroup.Item1" assigns late data values to user-defined attribute (UDA) Int1.

Late data received from DI objects referenced through the I/O extension is not backfilled. In the meantime, OMI ViewApps show the last known good value. The live data feed resumes once the data connection is restored. Historical data is sent directly to the Historian and does not affect the runtime values shown in the ViewApp.

While the Historian is processing the historical data, Application Server uses a Quality and Status animation, "Processing Historical Data," to notify the ViewApp that this is occurring, while the live data feed continues. At the end of the historical data processing, the current data/states/values are determined based on either the live data or, in the absence live data, by using the last-known good values from the historical data. Trend pens, the Trend Client, and any other visualized data that relies on historical data are updated when historization is finished.

The historical data is historized appropriately and can be later queried as if it had been received in order.

Enable historical playback

AVEVA OMI provides the ability for users to review KPI-level information by playing back historical data from an Historian server. Users can switch to playback mode and visualize historical data from historized attributes. Attributes that are not historized, typically static data such as tag names and descriptions, are also shown via a connection to the live data source.

Any non-historized attribute values that are displayed are shown as live data. The appearance of the live data is differentiated from historical data by applying the style associated with the **Live Value Playback** Quality and Status indicator. This “Live Value Playback” Quality and Status indicator is applied to all live values while Historical Playback is active.

Note: Only the highest priority Quality and Style (QS) indicator is shown, and Live Data Playback has the lowest priority of all QS indicators. Therefore, if during playback any other QS indicator is active, live data will not be shown and any non-historized values will be displayed as NaN.

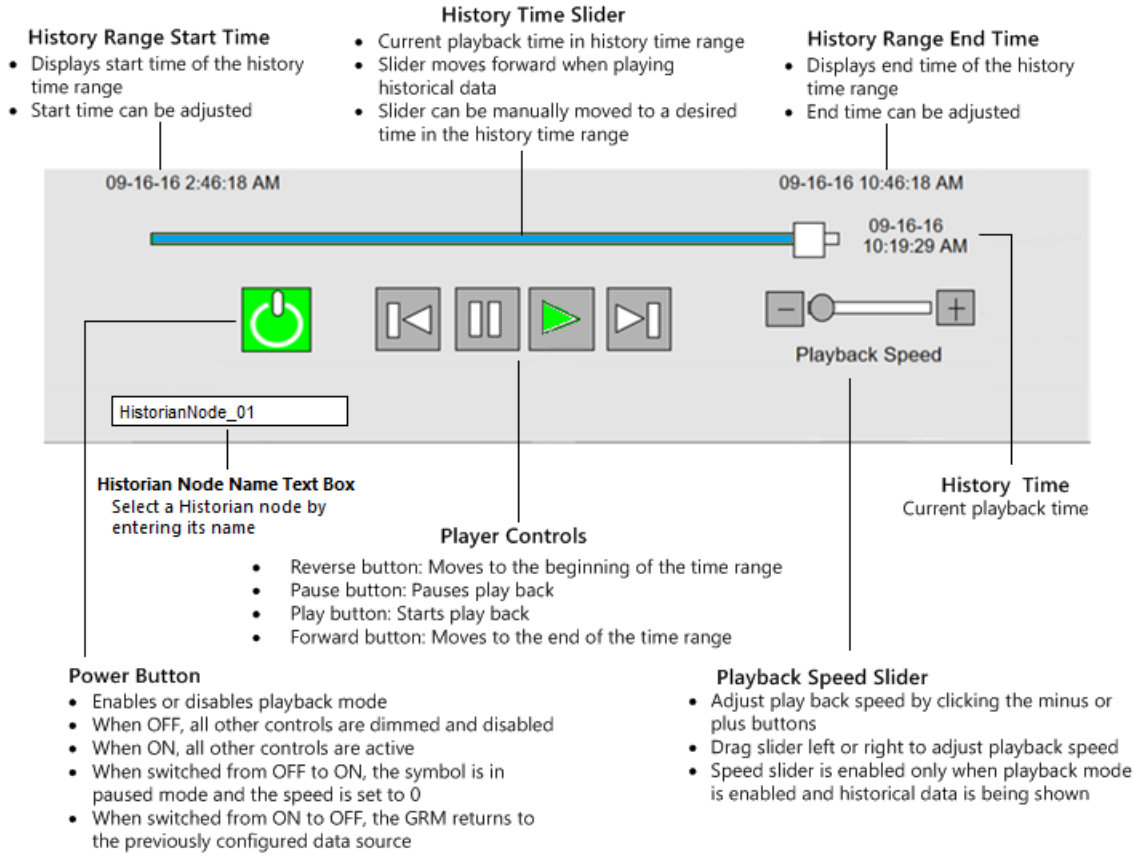
ViewApp Playback attributes can be used with graphic animations or action scripts to provide the functionality to show selected historical data. The graphic's graphic elements incorporate animations or scripts with Playback

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

attributes set as reference values to provide the equivalent functionality of media player controls. For example, you can include a switch graphic element associated with the Boolean **Enabled** attribute that powers on the media player graphic.

To enable historical playback

1. Create a media player graphicin which you associate playback attributes with the various elements of the media player. For example, associate the Boolean **MyViewApp.Playback.Enabled** attribute with the media player power button, and the **MyViewApp.Playback.Playing** attribute with the play graphic. A typical media player graphic looks something like the following illustration, and has similar capabilities, enabled through the attributes provided in the Playback namespace:



2. Add a text box to the media player graphic and link it to the **MyViewApp.Playback.HistorianServer** attribute. Users can enter a node name in the text box to connect to different Historian servers at runtime.
3. Manually deploy the playback service that will play the historical data. See Manually deploy the playback service for instructions on deploying the playback service.

To configure the Live Playback indicator

1. Select **Galaxy** from the ribbon bar, then select **Configure.**
2. From the list of **Configure** options, select **Galaxy**.
3. Select **Styles** and then select **Quality & Status**.
4. Select the **Live Playback** indicator to configure it. For more information, see the following topics:
   - Set Graphics Properties with a Galaxy Style Library

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Use Quality and Status Indicators to Provide Run-Time Feedback

About playback attributes

Playback attributes are Read/Write and typically used as graphic element animation reference values. When the user interacts with a graphic element's animation or script, Playback attribute values are modified to enable some aspect of showing real time or historical data.

Tags and other static attribute values are generally not historized due to their unchanging nature. When these and any other non-historized static attributes are shown during playback, there are no values that can be pulled from the historical data. Prior to System Platform 2023, non-historized attribute values were displayed as empty or NaN (not a number) in place of the missing data. With System Platform 2023, a live data connection is opened to provide the these missing values. You can configure the appearance of the live data during Playback in the IDE.

If any other Quality and Status indicator is displayed (Communication Error, Security Error, Warning, etc.), live data is not shown for the non-historized values and the missing numerical values are shown as NaN. Other data types (string, date, Bool, etc.) may be shown as empty.

Important: Playback attributes are predefined and available only for AVEVA OMI ViewApps. Playback attributes cannot be included in an InTouch HMI application.

All Playback attributes operate in the MyViewApp.Playback namespace. The names of Playback attributes are specified with the MyViewApp.Playback prefix in the form MyViewApp.Playback.*attribute_name*. To access the Playback namespace, you can reference the SDK assembly ArchestrA.ClientRuntimeData.dll.

Attribute Name	Data Type	Attribute Type	Initial Value	Description
CurrentTime	DateTime	Read/Write	Current playback time	**CurrentTime** is set to the current playback time when in playback mode. Setting this attribute adjusts the current playback time. The time assigned to **CurrentTime** can be only within the historical playback period. When the current time is being set and the speed is 0 (pause mode), the client will receive a data update based on the current time of the local system.
Enabled	Boolean	Read/Write	False	When **Enabled** is set to True, other attributes are initialized to their default values and playback is enabled. When set to False, historical playback is disabled and play real time data.
EndTime	DateTime	Read/Write	Current time	**EndTime** is initialized to the current time when the **Enabled** attribute is set to True. Setting this attribute adjusts the ending time of the historical playback range. **EndTime** can be set to a value regardless of the value set to the **Enabled** attribute.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Attribute Name	Data Type	Attribute Type	Initial Value	Description
HistorianServer	String	Read/Write	Empty string	Node name of the computer hosting the Historian and its historical data. The value assigned to **HistorianServer** is retentive. If the initial value of **HistorianServer** is an empty string, the Historian is expected to be installed on the GR node.
Playing	Boolean	Read/Write	False	When **Playing** is set to True, historical playback becomes active and playback speed is set to 1. The **Enabled** attribute must be set to True to set **Playing** to True.
Speed	Integer	Read/Write	0	Historical data playback speed as an integer value. 0 represents playback is stopped or paused. 1 represents normal playback speed. **Speed** values greater than 1 are functionally equivalent to the fast forward function of a media player that plays data faster than it was recorded.
StartTime	DateTime	Read/Write	8 hours prior to the current time	**StartTime** is initialized to 8 hrs prior to the current time when the **Enabled** attribute is set to True. Setting this attribute adjusts the starting time of the historical playback range and the current playback time adjusts accordingly. **StartTime** can be set to a value regardless of the value set to the **Enabled** attribute.

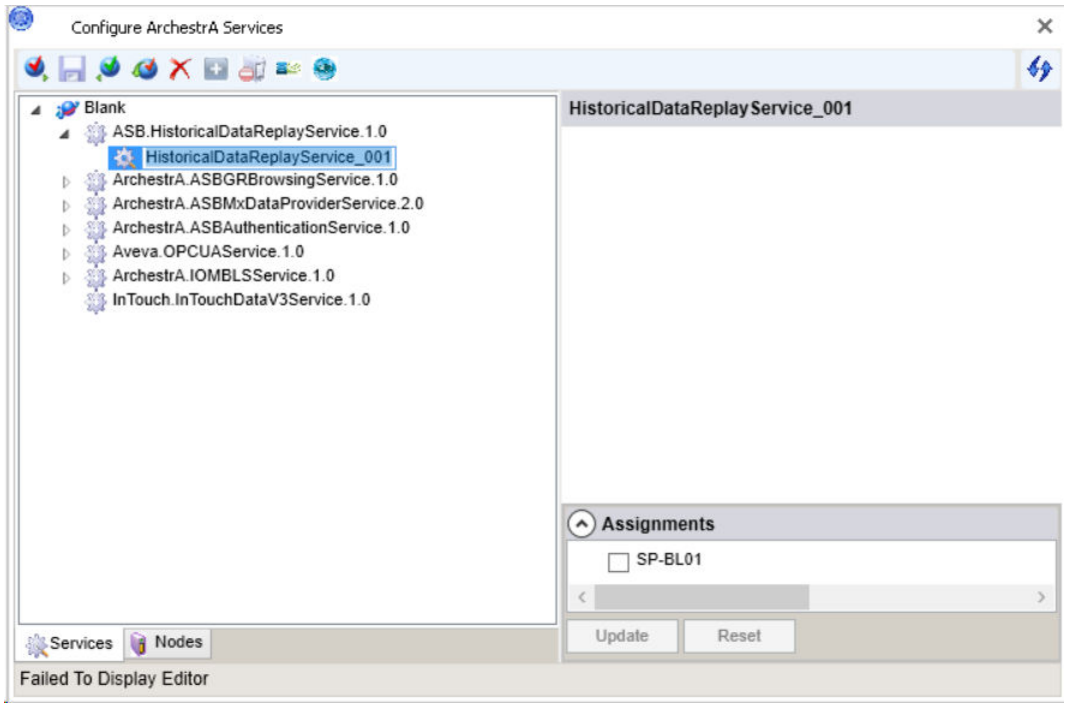
Manually deploy the playback service

In AVEVA OMI Version 1.0, you must manually deploy the playback service to play historical data from a graphic that incorporates Playback attributes.

To manually deploy the playback service

1. Open the System Platform IDE.
2. Select **Galaxy** from the ribbon bar, then select **Configure**.
3. From the **Configure** options, select **System** and then select **Services**.
4. When the **Services** window opens, select the ASB.HistoricalDataReplayService.1.0 and press **Ctrl + N** to create an instance.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



5. Right-click the instance name and select **Rename Node** (or press F2) to rename the instance to the computer name of your Historian Server node.
6. Right-click the instance and select **Deploy** (or press Ctrl + D) to deploy the instance.

Manage OMI applications

This section describes how to manage OMI applications including licensing and entitlements, user authentication and authorization, security, and managing graphics.

This section also describes the behavior of OMI ViewApps and Application Server when running under Operations Control and connected experience. If you do not use Operations Control and connected experience, ViewApps, Application Server, and the remainder of System Platform products will continue to function as they have in previous releases.

Manage product licensing and user authorization

System Platform can run under different types of licensing and security. Originally, System Platform was sold with a perpetual license that allowed customers to use the product for an unlimited length of time.This release of System Platform introduces Operations Control mode, and under Operations Control, additional capabilities for managing user access called the connected experience.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Operations Control uses AVEVA Identity Manager to provide identity service for AVEVA products. The AVEVA Identity Manager helps administrators to implement authentication and allows access tokens to be reused across different AVEVA products.

Connected experience extends Operations Control to enable single sign-on of all your users, and allows centralized user management in CONNECT

The section that follows describes the behavior of System Platform when running under Operations Control and connected experience. If you do not use Operations Control and connected experience, System Platform will continue to function as it has in previous releases.

Connected experience and operations control

Connected experience requires an CONNECT account with a valid Operations Control subscription and user management. Connected experience allows single sign-on to the AVEVA products supported by Operations Control that you are entitled to use. Note that licenses must be installed and activated.

- Unified User Management: Selecting the Connected Experience option enables all Operations Control products on the node to require log in authentication with CONNECT. CONNECT-based authorization is the only security mode available under connected experience.
- Compatibility across nodes: Connected experience must be enabled on all nodes in your system. Applications previously build on nodes not enabled for connected experience must be reconfigured to function in the connected experience environment.

You can disable the connected experience at any time, but if you do, connected experience must be disabled on all nodes in your system. If connected experience is disabled, any applications that were built under connected experience must be reconfigured to function without connected experience, including both authentication methods and product licensing.

Configure operations control and connected experience

To operate Application Server and other System Platform products under AVEVA Operations Control, either as a product subscription only, or with connected experience, you must configure your installed and licensed products.

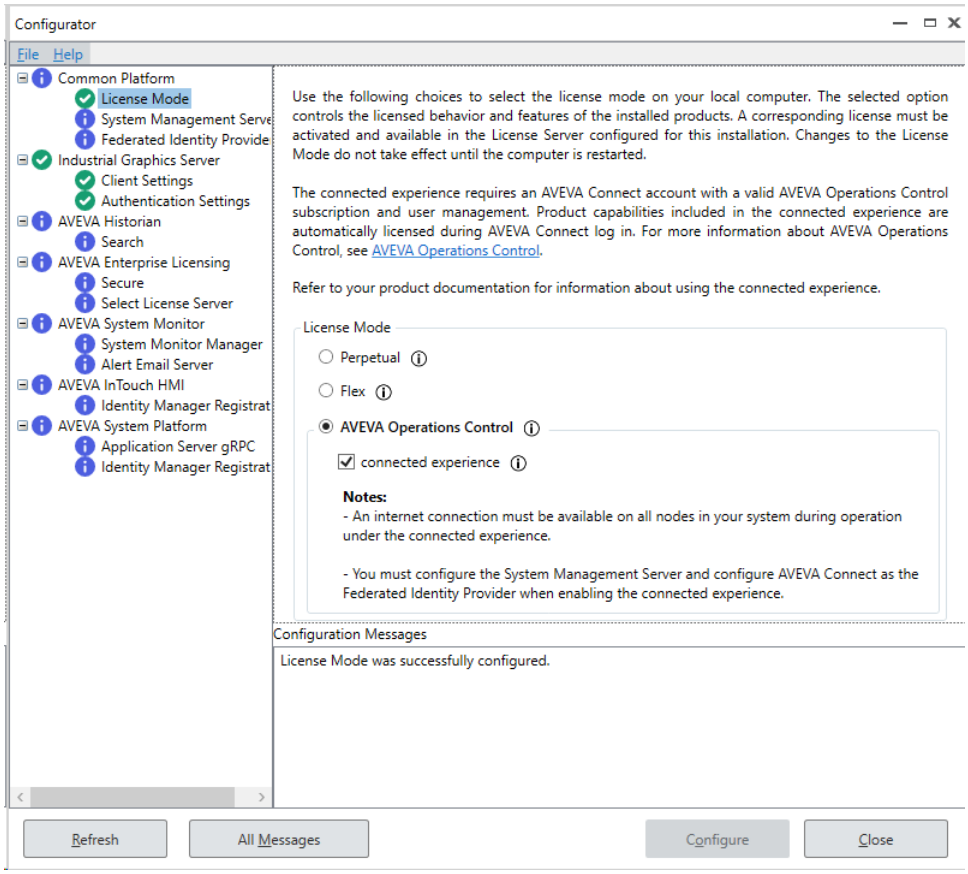
For more information refer to:

- Operations Control Configurator
- AVEVA Identity Manager Help

To configure Operations Control connected experience mode:

1. Open the Configurator from **Start** > **AVEVA** > **Configurator**.
2. Select **License Mode** in the left pane to configure license mode.
    a. Select the **AVEVA Operations Control** radio button to enable a subscription license for one of the AVEVA Operations Control packages (Edge or Supervisory); that includes unlimited use of all products in the product package for your defined set of users.
    b. Select the **connected experience** check box to enable a Single Sign-on (SSO) experience across all Operations Control products on this node with CONNECT cloud capabilities, available in the products by default.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

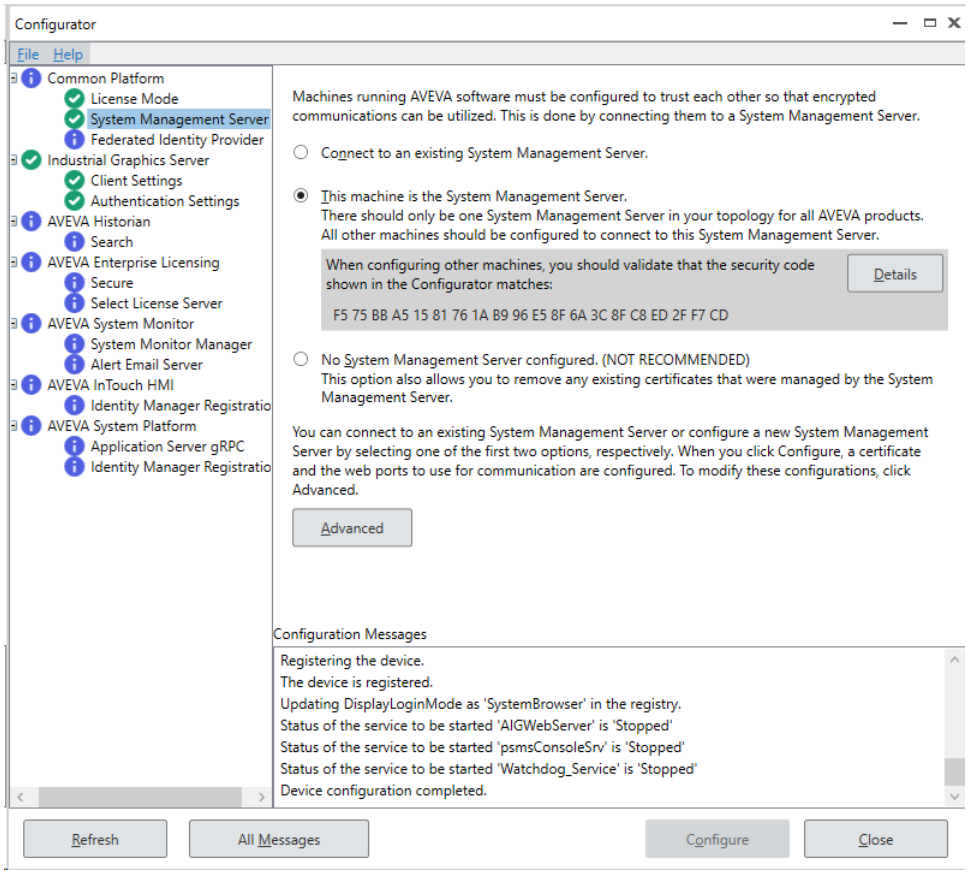


3. To configure the System Management Server, in the left pane, under **Common Platform**, select **System Management Server**.

Note: If you are prompted for user credentials for the System Management Server, use the following format to enter the user name: DomainName\UserName. The prompt for user credentials may be displayed if you have domain admin privileges but are not an admin on the local machine. You must be a member of the Administrators or aaAdministrators OS group to configure the System Management Server.

- Connect to a remote server or configure the local machine as the SMS.
- No System Management Server option is **not recommended**. You must configure a System Management Server to use Operations Control and connected experience.
- Only one of the machines in the network is identified and configured as a System Management Server. Machines running AVEVA products can then connect to that single System Management Server to establish trust and configure encrypted communications.
- Configuring a Redundant SSO server does not configure a separate System Management Server. A Redundant SSO server is not required, but provides a backup in case the System Management Server loses access to the external authentication provider.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

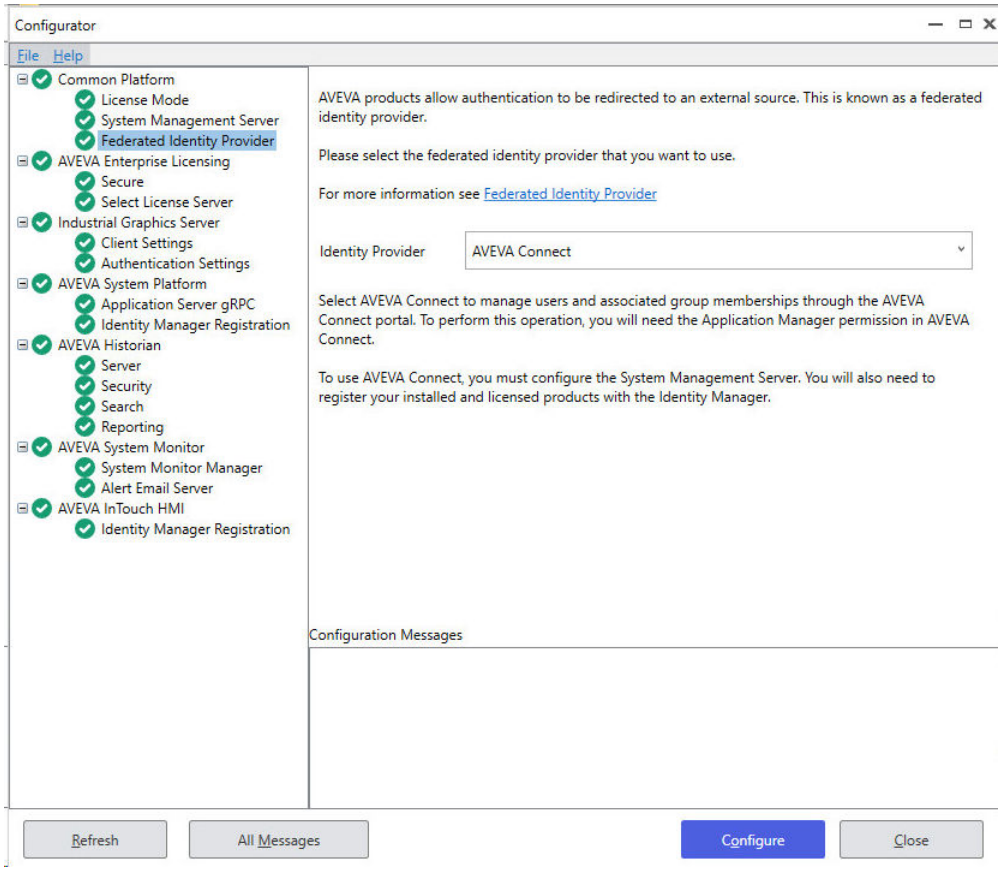


4. Select **Federated Identity Provider** under Common Platform in the left pane to configure a Federated Identity Provider.

   The Identity Manager component in the PCS Framework, available on the System Management Server (SMS) and Redundant SSO (RSSO) nodes, can be configured to enable "federated" login with CONNECT. This allows a user to enter the email address associated with their CONNECT account into the Identity Manager login form. Then, they are redirected to CONNECT to complete the log in process.

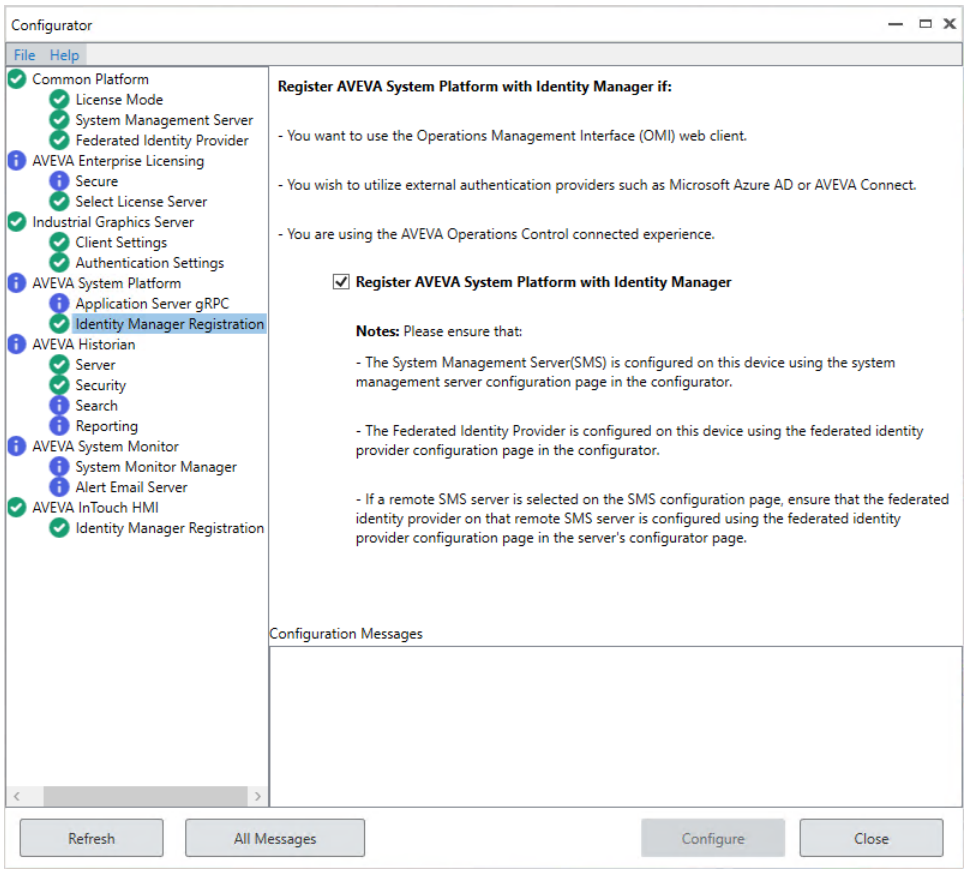
   Note: You must have a valid CONNECT account and be an administrator on that account to configure the Federated Identity Provider.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



5. Under **AVEVA System Platform**, select **Identity Manager Registration**. Check that the required configuration requirements are complete. Then, Enable "Register AVEVA System Platform with Identity Manager" and click the **Configure** button. The configuration requirements that must be met before you can register with Identity Manager are:
    - The System Management Server (SMS) is configured.
    - The Federated Identity Provider is configured.
    - If you are connecting to a remote SMS, make sure that the Federated Identity Provider on the remote SMS is configured correctly.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Other Components

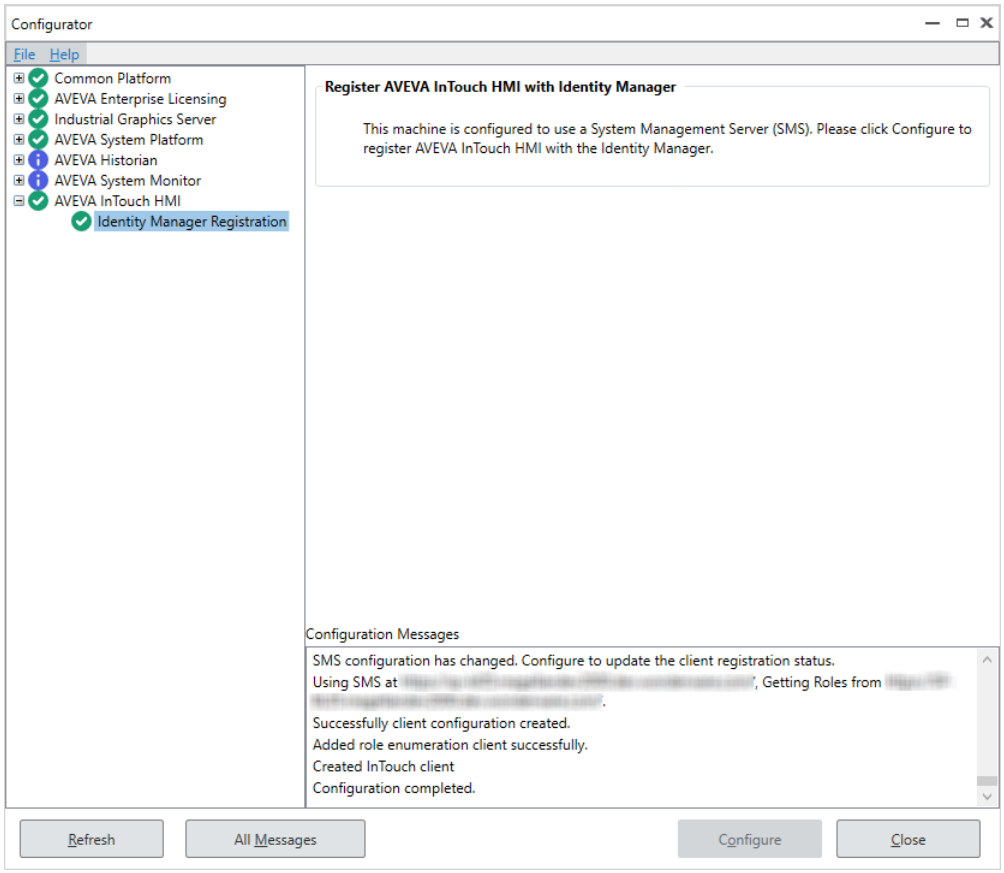
1. To register InTouch with AVEVA Identity Manager, in the left pane, under **AVEVA InTouch HMI**, select **Identity Manager Registration**.

   Register InTouch HMI with Identity Manager only when you have completed the preceding steps:

   - You have enabled AVEVA Operations Control connected experience as your license mode.
   - You have configured the System Management Server.
   - You have selected CONNECT in the Federated Identity Provider.

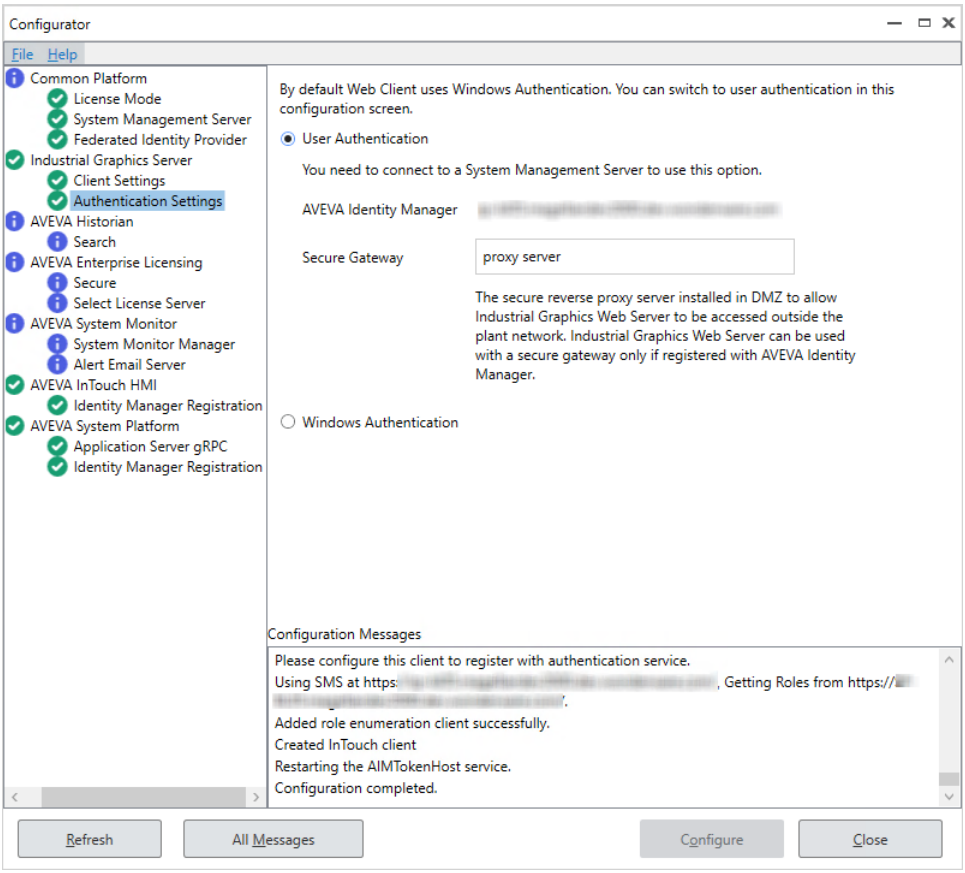
2. Click the **Configure** to register with Identity Manager.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. To Configure the InTouch Web Client to use User Authentication, in the left pane, under **Industrial Graphics Server**, select **Authentication Settings**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Authentication and entitlement

Authentication and entitlement, along with authorization, are security elements that serve to control user access to the AVEVA products that your organization uses.

- *Authentication* refers to verifying the identity of a user, through the use of a set of unique credentials (user name and password).
- *Entitlement* refers to the products that your organization has subscriptions for, and to which the authenticated user has been granted access. A user, even though they have been authenticated, may not be entitled to use a particular product, if the user does not have a valid Operations Control subscription.
- *Authorization* refers to the level of privileges associated with an authenticated user. This is generally done through the assignment of a user to a user group or through federated identity management.

  Federated identity is managed through a *Federated Identity Provider* (FIDP), and allows unified user management and single sign-on for users across the complete portfolio of AVEVA products. This is enabled through the Configurator, and is generally completed at the time of installation but can be reconfigured at any time.

The license mode you select plays an important role in how authentication, entitlement, and authorization are handled within your System Platform environment. See License mode for more information.

When Operations Control connected experience is enabled, each time a user logs in to a System Platform component, such as the IDE, the Historian, an OMI ViewApp, and so on, the log in is saved as an event in the CONNECT audit log.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

View the Data Log

Audit logs let authorized users from both your organization and AVEVA view the log of events or operations that occur in your account from the CONNECT portal. The logs display all operations that have taken place in your organization's account during the selected time period. You must be an Account Administrator, or must have the Report Viewer role to view the audit logs.

Note: Data in the audit logs is available from the time when the Audit functionality is available for your account.

To view the audit logs, go to the CONNECT web site. On the site navigation menu, select Audit. The Audit log page is displayed.

The audit logs provide the following information:

- Timestamp (date and time) when the operation was performed.
- Details of the operation, such as username, machine name, and the application name (for example, AVEVA OMI or System Platform IDE. The information that is logged includes which product or products were accessed by which authenticated user, as well as their available entitlement access.
- The audit log data can be filtered by using the **Entitlement Check** function in CONNECT. This function provides all data related to product usage through connected experience.

Note: Local time is used while displaying the data on the user interface. However, the UTC time is shown in the operation details and when the data is exported.

See the CONNECT help for more information about audit logs.

Authenticate with the AVEVA Identity Manager

When Operations Control connected experience is enabled in the Configurator, you are prompted to authenticate with the AVEVA Identity Manager when when you start your first AVEVA application. Once you are authenticated, single sign-on (SSO) is enabled. With SSO enabled, you can run additional AVEVA applications without having to re-enter your credentials each time.

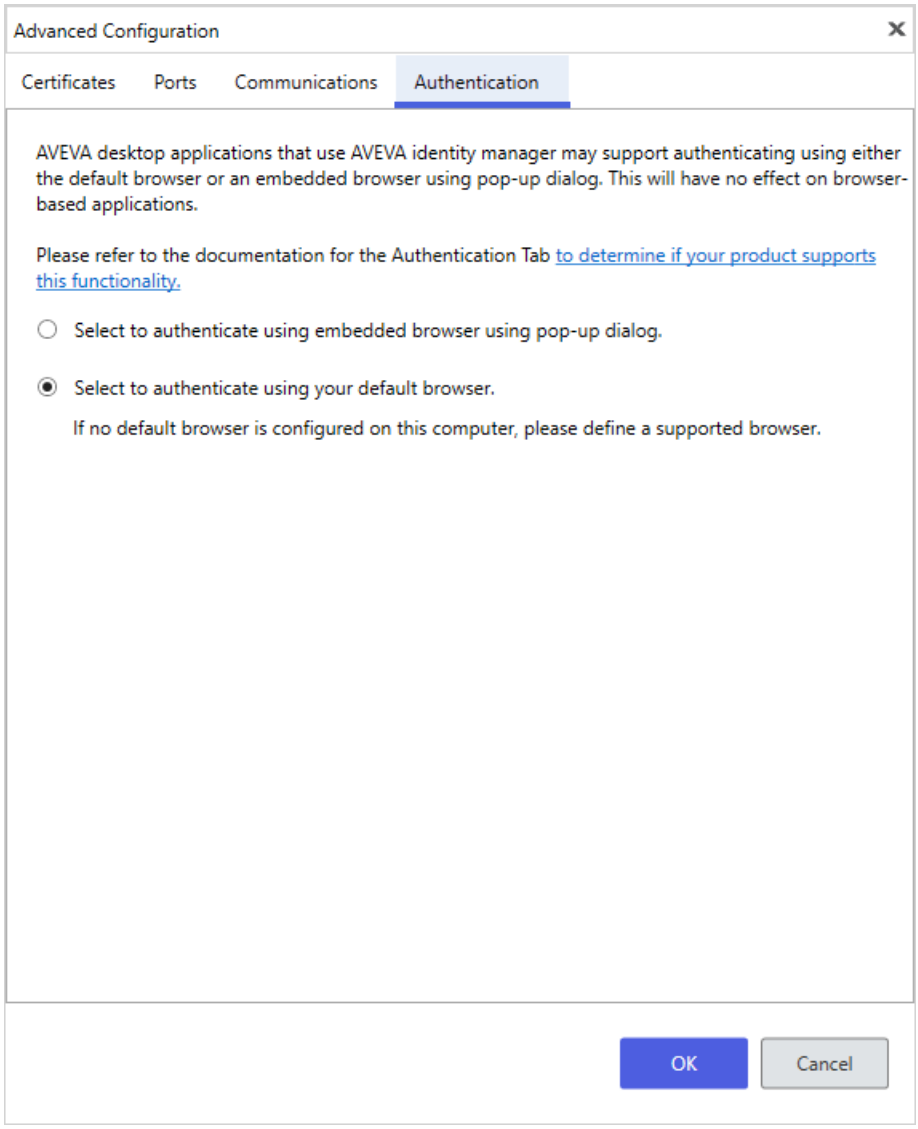
AVEVA Identity Manager allows two options for signing in, which are set in the Configurator, usually during System Platform installation. This is listed in the Configurator under **Common Platform** > **System Management Server** > **Advanced** > **Authentication**.

- To use your default browser, choose the **Select to authenticate using your default browser** option.

To authenticate using AVEVA Identity Manager on a web browser:

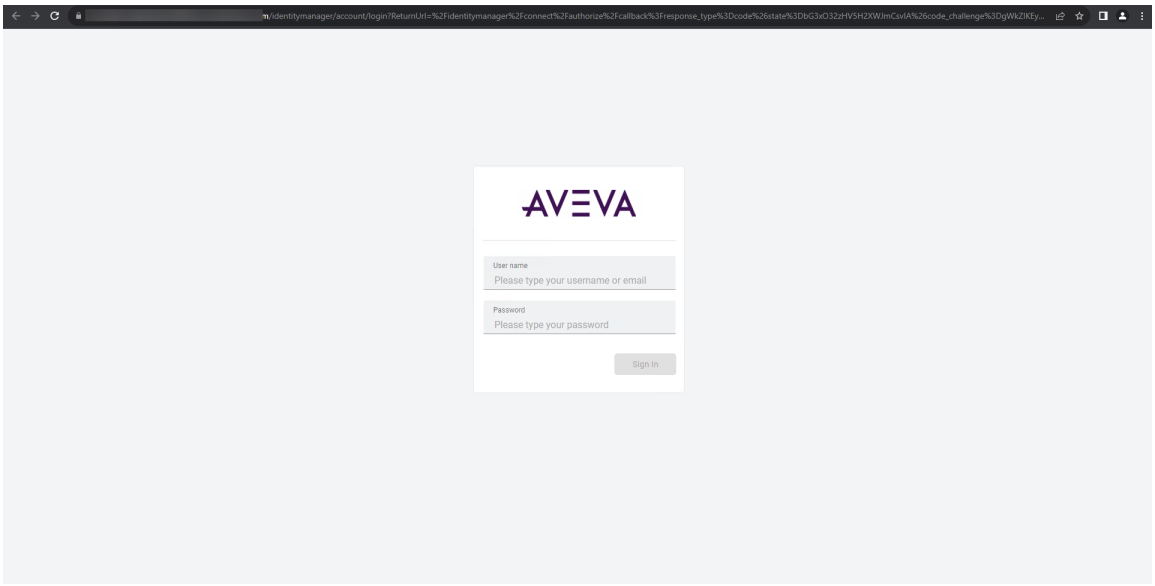
1. Set the option to use the default browser in the Configurator.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. When you open an AVEVA Operation Control application, the AVEVA Identity Manager login page opens in the default browser set for your system.
3. In the AVEVA Identity Manager login page, your **User name** is your email address that is registered with an CONNECT account. Enter your email address to login to AVEVA Identity Manager.
4. Next, enter your password In the **Password** field and click **Sign In**.

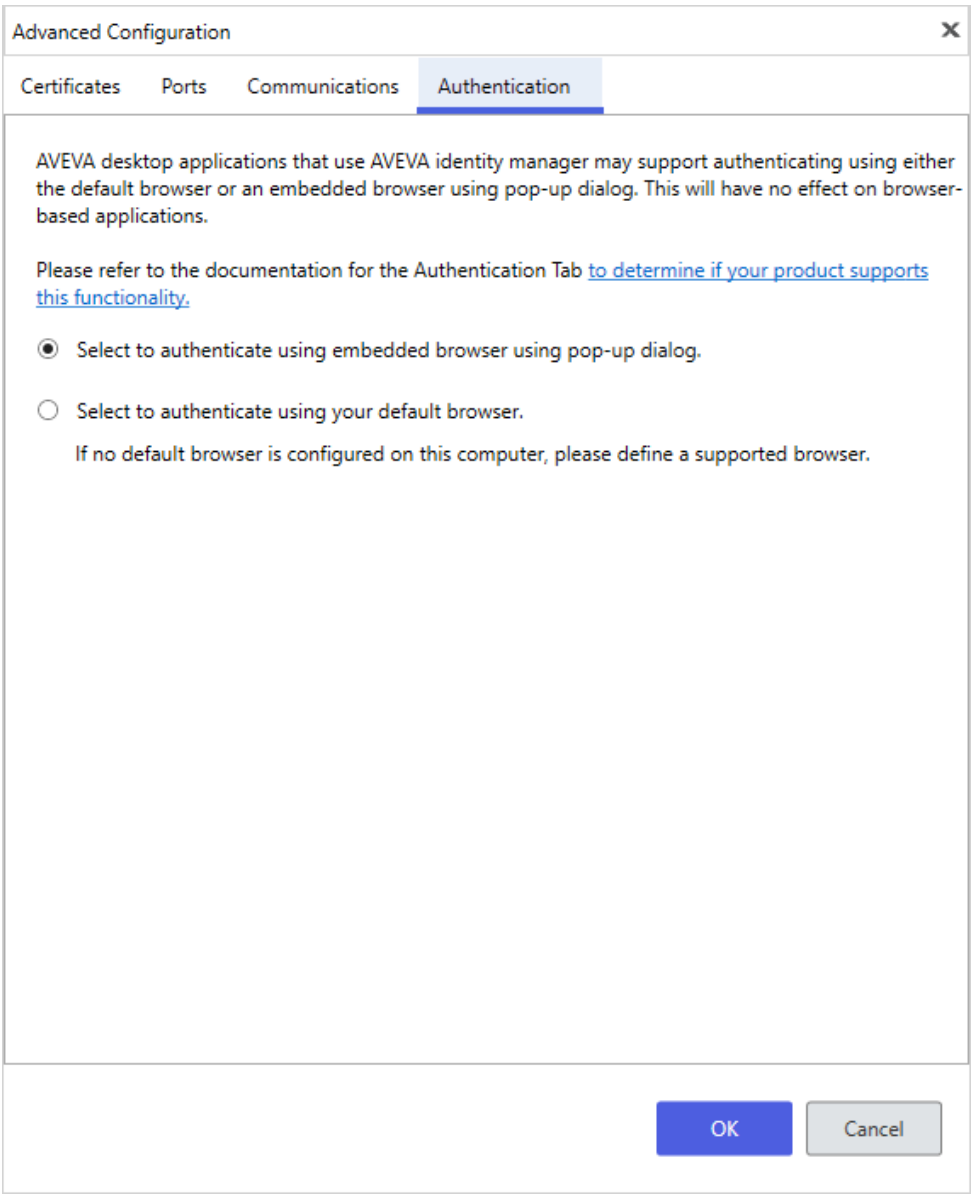
© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



To authenticate using AVEVA Identity Manager on an embedded browser:

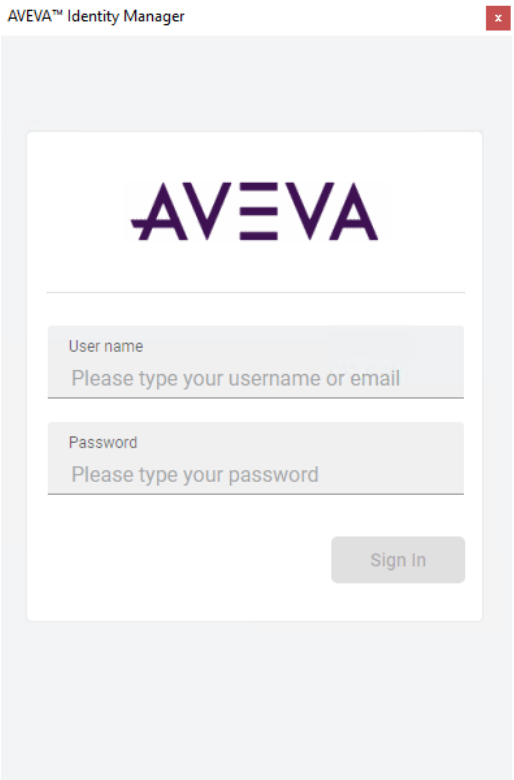
1. Set the option to use the embedded browser in the Configurator.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. When you open an AVEVA Operation Control application, the AVEVA Identity Manager login page opens in the embedded browser.
3. In the AVEVA Identity Manager login page, your **User name** is your email address that is registered with an CONNECT account. Enter your email address to login to AVEVA Identity Manager.
4. Next, enter your password in the **Password** field and click **Sign In**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Behavior if connection is lost

When you successfully sign in, your user profile which shows your user name and account number, appears in the title bar.

Two different types of connection loss may occur. These are:

- The connection between the application and the AVEVA Identity Manager.
- The connection between the AVEVA Identity Manager and CONNECT.

Lost connection between AVEVA Identity Manager and CONNECT

If, while your application is running, the connection between AVEVA Identity Manager and CONNECT is lost and prevents you from signing in again:

- On-premise capabilities will continue to function.
- Connected experience capabilities will stop working.
- If you try to start another connected experience application on the same node, it may not successfully run until the connection is restored. This applies to another instance of same product or a different Operations Control product.
- You have to sign in again using the Sign in option once the connection is restored.

Lost connection between the application and AVEVA Identity Manager

If, while your application is running, the connection between the AVEVA Identity Manager and the IDE or an OMI ViewApp is lost:

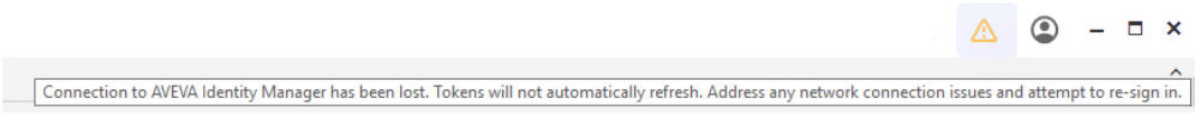
- A warning icon is displayed and provides a notification about the connection state.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

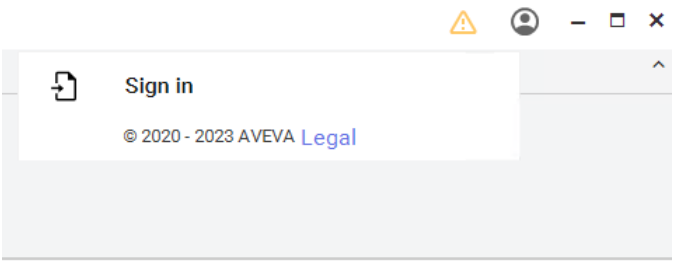
- Any open OMI ViewApps will continue to run.
- If you try to open a ViewApp, it will run in read-only mode.
- You will not be able to perform any writes.
- The IDE will continue to function, but you will lose access to any shared Cloud folders.

To address connection loss issues

- If there is loss of connectivity to CONNECT or AVEVA Identity Manager, a warning icon is displayed and provides a notification about the connection state. When you hover the cursor over the warning icon, a tooltip with the connection loss message appears.



- Address any network connection issues. In the IDE, click on the profile icon and sign in again. For OMI ViewApps, you must sign out before you can sign in again.



- When the connection is restored, the warning icon will no longer be visible.

Single Sign-On (SSO)

The connected experience option enables Single Sign-on (SSO) experience across all Operations Control products with CONNECT cloud capabilities.

The connected experience requires an CONNECT account with a valid Operations Control subscription and user management.

Selecting the connected experience option enables all Operations Control products to require log in authentication with CONNECT when starting the first product. Products that are subsequently opened will authenticate using SSO. CONNECT-based authorization is the only security mode available under the connected experience.

Disable connected experience

You can disable the connected experience at any time, but the connected experience must be disabled on all nodes in your system. Galaxies and ViewApps built under the connected experience must be reconfigured to function in an environment that does not have connected experience capabilities, including both authentication methods and product licensing.

License mode

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

License Mode is set in the Configurator under the **Common Platform** tab. The Configurator lets you select from the following License Mode options:

- **Perpetual:** A specific AVEVA product license purchased for use in perpetuity.
- **AVEVA Operations Control:** A subscription license for at least one of two AVEVA Operations Control packages (Edge, Supervisory); includes unlimited use of all products in the product package for your defined set of users.
  - **Connected experience:** Enables Single Sign-on (SSO) experience across all Operations Control products with CONNECT cloud capabilities. See About connected experience configuration for more information.

Note that **Operations Control** mode is required in order to enable **connected experience**.

Start and run the IDE

The System Platform IDE in connected experience mode works according to the following steps. These steps assume that the IDE is the first Operations Control application that you are opening. Subsequent applications that you open will not require a separate sign-on (SSO).

1. Open the IDE.
2. You are prompted to authenticate with the AVEVA Identity Manager. For more information, see Authentication and entitlement.
3. Enter your username (email address) and password to sign in to AVEVA Identity Manager. The email address must be registered to an CONNECT account.
   - If the credentials are not recognized, an error message appears. If you made a mistake entering your username and/or password, re-enter your credentials. If you are unable to log in after correctly entering your credentials, contact your administrator for assistance.
   - When you sign successfully, and if you have a current (unexpired) Operations Control subscription available for your account, the IDE opens and your username is shown as the logged-in user.
   - If a valid subscription is not found, the IDE does not open.
4. IDE capabilities are enabled as per the permissions associated with the groups and roles assigned to you as the signed-in user.
5. To connect to a galaxy, see the next section.

Connect to a galaxy

Once you have started the IDE, you can connect to a galaxy. The process for connecting to a galaxy in connected experience mode is similar to the process for connecting to a galaxy where connected experience is not enabled. See Connect to a Galaxy for more information.

Galaxy migration to support connected experience

When you open an existing galaxy that was configured with the security mode set to Authentication providers (using Azure AD), you must set the security mode to “None” (when in non-connected mode), prior to opening the galaxy in connected experience mode. Connected experience mode is enabled and disabled under **License Mode**, in the System Platform Configurator. See License mode for details.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Failure to connect to the galaxy

However, the internal system checks are different when connected experience is enabled. You will not be able to connect to the galaxy if:

- Operations Control mode is not enabled on the GR node.
- The galaxy has a security mode that is not compatible with Operations Control connected experience mode. The only compatible security modes are **None** and **Authentication Provider**.
- The credentials you used to log into the IDE do not give you authorization to open the galaxy. In this case, you must restart the IDE and enter the correct credentials.

Successfully connect to the galaxy

The galaxy will open if the above system checks allow you to connect to an existing galaxy or create a new galaxy.

- Your user information will be shown in the IDE.
- CONNECT records that you are using the IDE.

For details about creating a galaxy, see Create a Galaxy.

Connect to a galaxy with no security configured

For compatibility with connected experience, you must set the security mode of the galaxy to either None or Authentication Providers. This applies to both new and existing galaxies.

This section describes what happens after you started the IDE and have connected to a new or existing galaxy. For more information, see:

- Start and run the IDE
- Connect to a galaxy

Security mode configuration

1. Log into the IDE as described previously. See Start and run the IDE for details.
2. Create a new galaxy, or open a galaxy that does not have security configured.
3. Go to security configuration. See Working with Security in the *Application Server User Guide* for details.
4. Select either **None** or **Authentication Providers**. If connected experience is enabled for System Platform, you are logged in once you start the IDE. You are not prompted to log in again when you connect to a galaxy.
    - If the galaxy security mode is **None**, users will still need to sign in to the IDE before they can connect to the galaxy. Sign in can be explicit or through SSO, if the user has already signed in to another connected experience application. Signing in to IDE is required, regardless of whether or security has been enabled for the galaxy.

      This completes security mode configuration for the galaxy; you can disregard steps 5 through 7.
    - If the galaxy security mode is **Authentication Providers**, select the **Roles** tab, and then click the **+** (add new group) button. The list of CONNECT user groups will be shown. Continue to step 5.
5. From the list of groups that are shown, associate IDE permissions to at least one of the groups. These permissions must include:
    - Can Start the IDE

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- User Configuration > Can Create/Modify/Delete Users
- User Configuration > Can Modify Own User Information

6. Save the security information.
7. If prompted, log in again with your username and password.

Connect to a galaxy not using Operations Control connected experience (configuration mismatch)

This workflow describes how to connect to an existing galaxy that is not configured for Operations Control connected experience, when connected experience has been enabled for System Platform.

1. Open the IDE and select a galaxy.

   You are notified that the galaxy security mode is not compatible with Operations Control connected experience mode, and that you need to change the security mode to one that is compatible.
2. If you proceed, you are allowed only to change the galaxy security setting to **None** or **Authentication Providers**, as needed.
3. Configure the galaxy security as needed.
4. Save your changes. After the security changes are saved, the galaxy restarts.
5. If prompted, log in again with your username and password.

For additional details, see Galaxy migration to support connected experience.

Sign out of the IDE

The following information applies if you are signed into the IDE in a Operations Control connected experience environment.

- Once you sign out of the IDE, the IDE closes.
- However, you remain signed into any other connected experience applications, such as OMI ViewApps, that you have signed into.
- You can sign back into the IDE with SSO, provided that you are still signed into other connected experience applications.
- If another user signs into the IDE and you are still signed into other connected experience applications on the same node, you remain as the active user on the other applications.

Start Application Manager

Using Operations Control mode and connected experience has no effect on starting and running Application Manager.

Launch a ViewApp using the connected experience

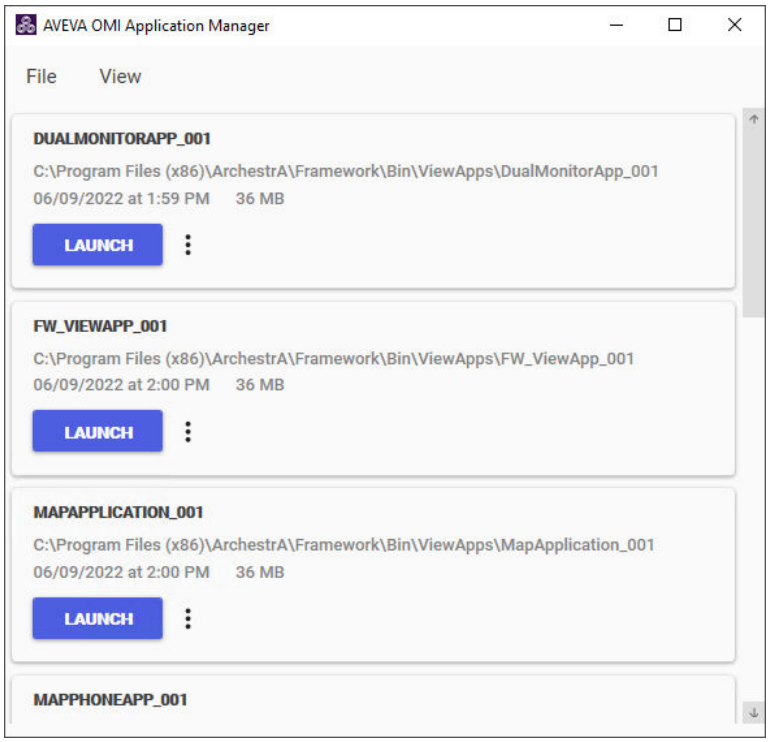
If your installation uses AVEVA Operations Control and the connected experience for user authorization and security, follow these steps to start an OMI ViewApp.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To launch a ViewApp using the connected experience

1. Show the desktop of the computer where the ViewApp is deployed.
2. Select the AVEVA OMI Application Manager desktop shortcut.

   AVEVA OMI Application Manager appears with an alphabetic list of ViewApps deployed on the computer.



   For more information about the fields of the AVEVA OMI Application Manager, see About the ViewApps Application Manager.

3. Select **LAUNCH** for the ViewApp you want to run.
   - **Already signed in (Single Sign On):** If you are already logged in to another ViewApp or other AVEVA product that uses Operation Control connected experience, you are logged in to the ViewApp automatically using Single Sign-on.
   - **Not yet signed in:** If you are not already logged in to a product that uses Operation Control connected experience, a browser window opens for you to enter your user name and password. The browser window is either the System Platform embedded browser or your default browser, depending on what which authentication option has been configured for the System Management Server.

After you successfully log in, the ViewApp opens. Your user name is shown near the right side of the titlebar in the ViewApp (but only if the ViewApp includes the OMI TitleBarApp).

If you have a problem logging in (authentication or entitlement failure)

- Make sure your internet connection is working.
- If user authentication fails, the ViewApp opens in read-only mode after you click OK. In read-only mode,

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- If entitlement fails, you will see a message that a valid subscription could not be found. After you click OK, the ViewApp opens in read-only mode. In this mode, you cannot interact with the ViewApp (for example, to change a setpoint or to acknowledge an alarm); you can only monitor what the ViewApp is showing.

Corrective action

If you receive an authentication error message, repeat the log in procedure, being sure to enter your user name and password correctly. If you still cannot sign in, contact your CONNECT administrator.

If you receive an entitlement error message, there may be a problem with your AVEVA subscription. Contact your CONNECT administrator.

Implementing SSO in your ViewApp

OMI ViewApps leverage the Security.LoggedInUserSsoToken property. OMI Apps consume the user access token via the SDK using security attributes. These attributes are updated whenever the token is refreshed by a user signing in or out.

Configure user permissions for connected experience

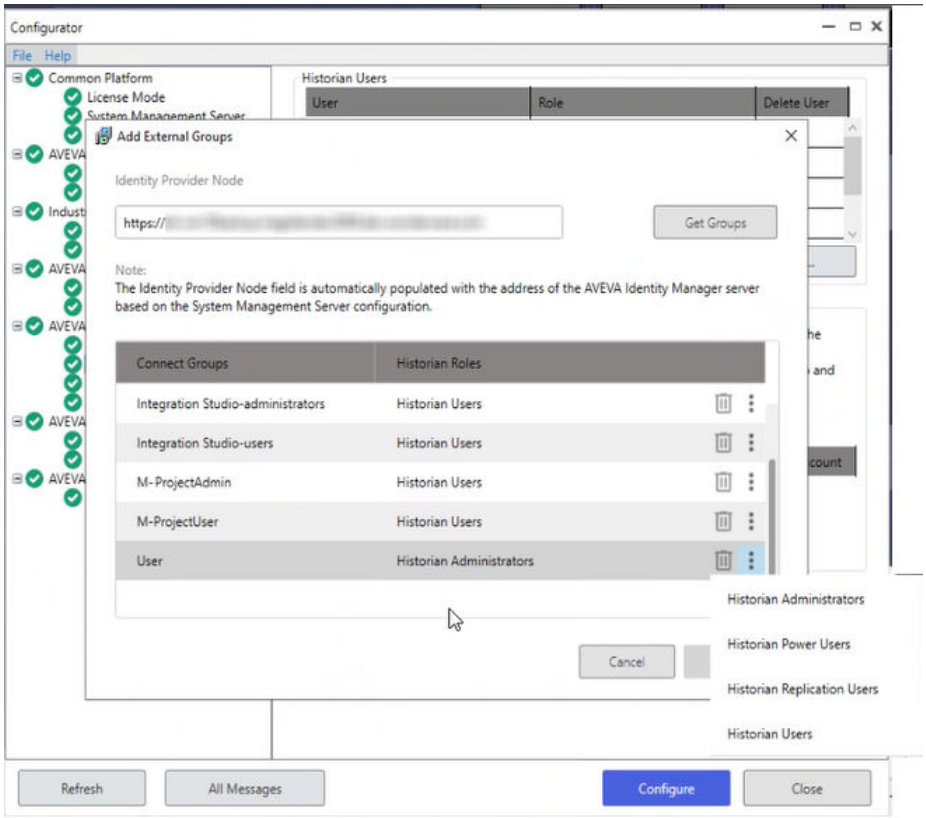
When connected experience is enabled, ViewApp users will need to be granted additional permissions by granting them either the "Historian Power User" or "Historian Administrator" role.

You can add users to these roles on a Historian node by using the System Platform Configurator.

1. Open the Configurator and select the **Historian Security** plugin.
2. Click the **Add External Groups** button. The **Add External Groups** window opens.

   Note: Add External Groups is only active if the System Management Server is configured with a federated identity provider, as is the case when .

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



3. Click the **Get Groups** button in the **Add External Groups** window.
4. Select the group or groups that require additional permissions (roles).
5. Add the appropriate role or roles to the group(s).

Launch ViewApp in the OMI Web Client using connected experience

If your installation uses AVEVA Operations Control connected experience for user authorization and security, follow these steps to start an OMI ViewApp.

To launch a ViewApp in the OMI web client using the connected experience

1. In a supported web browser, go to the URL for the OMI web client:
   - On the machine where the WebViewEngine is deployed, the URL is `https://localhost/omi`. You can also use either of the two formats below.
   - On a machine on the same domain as the one where the WebViewEngine is deployed, the URL is `https://<node_name>/omi`, where `<node_name>` is the name of the platform where the WebViewEngine is deployed.
   - On a machine that is not on the same domain as the WebViewEngine, the URL is `https://<fqdn>/omi`, where `<fqdn>` is the fully-qualified domain name of the platform where the WebViewEngine is deployed.

   If you are already logged in to another AVEVA product that uses Operation Control connected experience, you are logged in to the OMI web client automatically using Single Sign-on. Skip to step 3.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

2. If you are not already logged in to a product that uses the connected experience, a sign-in window opens for you to enter your user name and password.

   After you have entered your user name and password, the OMI web client page opens.

3. The OMI web client page lists all of the OMI ViewApps available through the web client. Select the ViewApp to run.

   When the ViewApp opens, the user name that you are logged in as shows in the top-right corner of the page.

If you have a problem logging in (authentication or entitlement failure)

- Make sure your internet connection is working.
- If user authentication fails, the ViewApp opens in read-only mode after you click OK.
- If entitlement fails, you will see a message that a valid subscription could not be found. After you click OK, the ViewApp opens in read-only mode.

Corrective action

If you receive an authentication error message, repeat the log in procedure, being sure to enter your user name and password correctly. If you still cannot sign in, contact your CONNECT administrator.

If you receive an entitlement error message, there may be a problem with your AVEVA subscription. Contact your CONNECT administrator.

Sign out from an OMI ViewApp

You can sign out from an OMI ViewApp by clicking the **Sign Out** option under the **Profile** icon. When you sign out of an ViewApp:

- Operations Control connected experience supports running multiple applications with with different users signed in at the same time.
- Sign out closes the user session from CONNECT.
- Inactivity time-out is not supported when connected experience is enabled. Users must actively log out of the OMI ViewApp to end their connection.
- When you sign out of a ViewApp, you are only signed out of that specific ViewApp. Other ViewApps and connected experience applications will continue to run with their current signed in user.
  - The ViewApp from which you signed out continues to run, but in read-only mode. No user information is shown in the title bar.
  - If another user signs into that ViewApp, the CONNECT login dialog is shown. Once the user is authenticated, the ViewApp is loaded again, based on the security mode of the associated galaxy. The information for the new user is shown in the title bar. The information for the new user is now used for SSO.
- When you sign out of a desktop application using the embedded browser, web applications will continue to run.
- When you sign out of a web application, all web applications will close.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Closing an application does not do anything with the session. Only that instance of the application gets closed. The session is still on the CONNECT server. When you reopen that application, your session is available and you will be authenticated with Single Sign On.
- When you sign out of the IDE, the session is closed and the IDE exits.

Switch OMI ViewApp users

To change the user signed into an OMI ViewApp:

1. The current user signs out from the ViewApp.

   This action logs the user out of the ViewApp. However, the ViewApp reloads under the context of the logged-in Windows user, and is now running in read-only mode. The TitleBarApp in the ViewApp does not show any user information.

2. A new user now signs into the ViewApp. The user uses their AVEVA login credentials to sign in.

   The new user is authenticated, and a message is logged in CONNECT for this user.

3. The OMI ViewApp reloads, based on the security mode in the associated galaxy.

   The TitleBarApp in the ViewApp now shows the new user as the signed-in user.

Secured and verified writes

Operations that require you to log in every time you use them include secured writes, verified writes, and acknowledging alarm states. The behavior of these features does not change, whether or not you are using the connected experience.

To perform a secured write in Operations Control mode

1. Change a runtime value. The secured write dialog is displayed.
2. Enter your AVEVA username and password. The AVEVA Identity Manager authenticates you and the value change is allowed.

To perform a verified write in Operations Control mode

1. Change a runtime value. The verified write dialog is displayed.
2. Enter your AVEVA username and password. The AVEVA Identity Manager authenticates you.
3. The verifier clicks on the authentication button and enters their username and password.
4. Both identities are verified and the value change is allowed.

Deploy scenarios

Deployment in an Operations Control connected experience environment

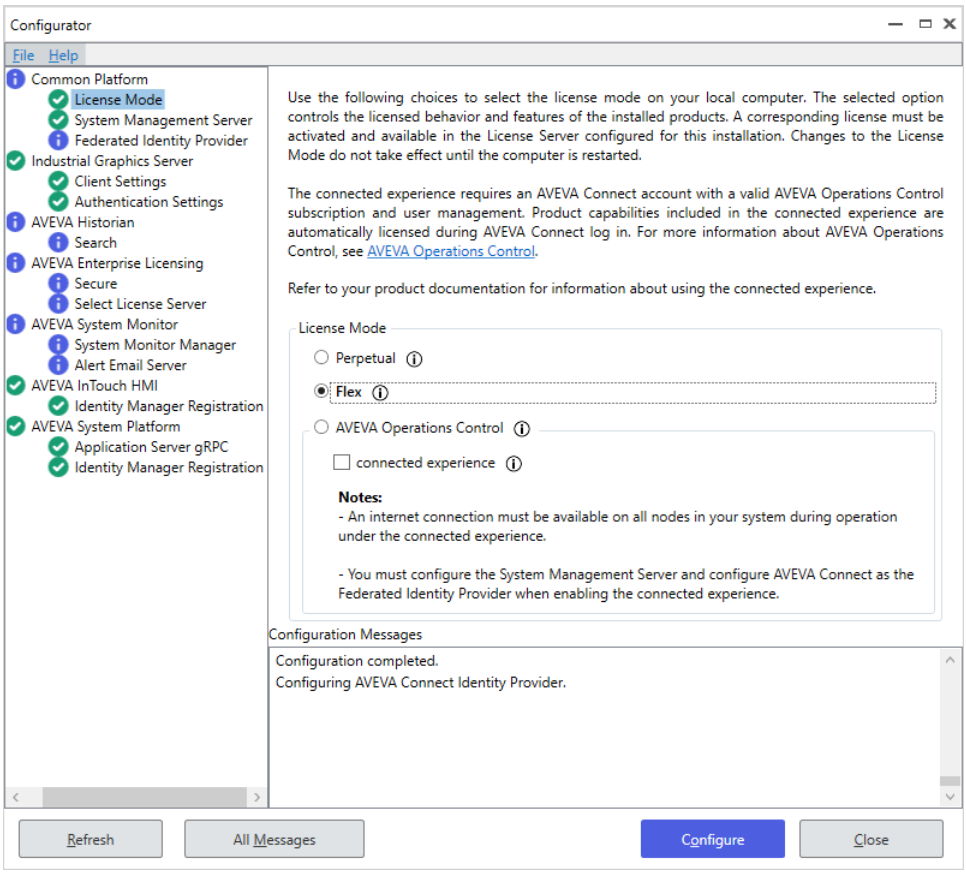
- As long as all nodes are configured for Operations Control connected experience, you can deploy a galaxy that has been enabled for Operations Control connected experience from the GR node to a remote Application Object Server node, provided that the license types match.
- If your GR node is not configured for Operations Control connected experience, but the remote node is, you cannot deploy objects to that node. The opposite case is also true: you cannot deploy from a GR node configured for Operations Control connected experience to a remote node that is not similarly configured.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Change Operations Control connected experience mode

If System Platform is using Operations Control connected experience mode, you can switch out of this mode if necessary. You will also need to change the security settings for any galaxies that use Operations Control connected experience. However, keep in mind that you will have to turn off Operations Control connected experience on all nodes, not just the GR or IDE node.

The license mode change is done through the Configurator. To change the license mode, you can simply uncheck the connected experience option, if you will still be using Operations Control mode. Or, you can change the license mode from AVEVA Operations Control to either **Perpetual** or **Flex**, depending on the type of license you have. Note that a system restart is required before any change to License Mode will take effect.



1. Now that you have changed the license mode, restart the GR.
2. Start the IDE. If the IDE is on a different node, you must change the license mode on this node as well. The IDE is no longer under Operations Control connected experience.
3. Connect to a galaxy. Note that the galaxy is still configured for Operations Control connected experience (using either **None** or the **Authentication providers** security setting).
4. Log into the galaxy as administrator.
5. Optional: Change the galaxy security model to the type of security you want to use.
6. Configure user credentials as needed to ensure that you and other legitimate users can still access the galaxy.
7. Save your changes and redeploy the galaxy.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Using the OCMC with connected experience

When using the Operations Control Management Console (OCMC) with AVEVA Operations Control and the connected experience, you are not required to log in when you start the OCMC itself. Instead, you are asked to log in with your user name and password the first time you open a node under Platform Manager in the tree pane of the OCMC.

If you have previously logged in to another Operations Control application, or to the OCMC, you will be authenticated using Single Sign-on and will not have to log in again.

Galaxy migration to support connected experience

If the galaxy security mode is Authentication providers (using Azure AD), the security mode has to be set to “None” (when in non-connected mode), prior to opening the galaxy in connected experience mode. Use the following procedure to migrate an galaxy to use connected experience if:

- You have a galaxy created or modified under System Platform 2023.
- The galaxy is configured with the **Authentication providers** security mode using Azure AD.
- You want to use Operations Control - connected experience.

This procedure is applicable to both fresh installs of System Platform 2023 R2 and upgrades from System Platform 2023 to System Platform 2023 R2.

To migrate an existing galaxy that uses Authentication providers security

1. Install System Platform 2023 R2 (fresh install) or upgrade your installed System Platform components to System Platform 2023 R2, as applicable.
2. Set the **License Mode** (under Common Platform) to **AVEVA Operations Control**.
3. Leave the **connected experience** option disabled.
4. Use the IDE **Import** function to migrate the existing galaxy to System Platform 2023 R2. The galaxy **Security mode** remains set to **Authentication providers**.
5. Connect to the galaxy.
6. Change the **Security mode** to **None**.
7. Open the Configurator and enable **connected experience**.
8. Connect to the galaxy.
9. Change the **Security mode** to **Authentication providers**.

Note: If you enable connected experience initially (see step 3, above), you will not be able to connect to the galaxy. This is because connected experience uses CONNECT as the federated identity provider, while the galaxy is using Azure AD as the identity provider. In this scenario, when you attempt to connect to the galaxy, you will not be permitted to sign in.

Licensing and the OMI web client

AVEVA System Platform licensing controls which products your installation has access to and the number of users who can access them. This topic describes how licensing works with the OMI web client.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Each deployed WebViewEngine consumes one license. This allows an unlimited number of users to run OMI ViewApps deployed on that WebViewEngine. Each user has read/write access in each ViewApp they use. The process of acquiring and renewing the license is as follows:

1. When a WebViewEngine is deployed, it tries to connect to the License Server to acquire a license.
2. Whether it acquired a license or not, it continues to try to connect to the License Server periodically, either to acquire a license or to refresh the license that it has.
3. As long as the WebViewEngine succeeds in acquiring and refreshing its license, all ViewApps deployed to it continue to function normally.
4. If the WebViewEngine fails to acquire or renew a license--for example, if it cannot connect to the License Server--a two-hour countdown begins. During this countdown period:
    - Users can continue to use any open ViewApp deployed on the WebViewEngine. ViewApps will continue to function as normal, with no change in functionality.
    - Users will not see any message or other indication anywhere in the web client that the ViewApp does not have a license.
    - The WebViewEngine continues to try to acquire a license periodically.
    - A warning message is sent to the logger every fifteen minutes about the failure to acquire a license and saying that the OMI web client is in demo mode.

If the WebViewEngine acquires a license before the two-hour period is over, ViewApps return to normal operation. Users will not have seen any difference in operation.

If the WebViewEngine does not acquire a license within the two-hour period, it stops processing new requests for either data or content. This means:

- Any open ViewApp will not receive any new data from external references (User-Defined Attributes).
- Users will not be able to write to any external references. Writes to custom properties and ViewApp namespaces will continue to work.
- Users will see an error message and the action will fail if they try to access the web client home page, open any other ViewApp, or refresh the page of an open ViewApp. The exact error message varies.
- The warning messages sent to the logger change to say that demo mode has expired.

The WebViewEngine continues to try to acquire a license after demo mode has expired. If it is able to contact the license server and acquire one, it resumes normal operation.

Note: If you redeploy the WebViewEngine during or after the demo period, and it cannot acquire a license, it restarts the two-hour demo mode period. For uninterrupted service, you should ensure that the License Server always has enough valid licenses for all deployed WebViewEngines.

About ViewApp security

Security is the responsibility of almost everybody in an organization who works with ViewApps. Designers must incorporate security standards in objects and graphics. Builders set security requirements for each instance of a ViewApp. Control room operations staff must maintain security for ViewApps running in a production environment. LIne operators interact with ViewApp security as part of their daily tasks.

Different roles within an organization perform different security workflows.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

ViewApp Designer	ViewApp Builder
Manage Security Certificate Verification	Manage Security Groups
Configure Security	Manage Security Roles
Implement Log in Security	Manage Security Users
Log on with Security Enabled	Security in Graphic Scripts
Client Access Rules and Galaxy Security	Configure Security Attributes
Set Object Security	Configure Navigation Security
Configure Security for Graphics	Set Up a Map Server
Group Locking Security	Configure a Web Control
OPC Communication Driver Security	Configure Custom Navigation Properties
Locking and Security	Working with the SignedWrite() Function for Secured and Verified Writes
	Securing Client Controls
Control Room Operators	**Line Operator**
Set Deployment Security	Signature Security for Acknowledging Alarms
Configure User Information	User Access and Security During Run Time
Create User Credentials	Gather the Information You Need to Log On to InTouch Access Anywhere
Create Named Credentials	Shelve and Unshelve Alarms at Runtime
Manage Credentials	Acknowledging Alarms with Signature Required
	Gather the Information You Need to Log On to InTouch Access Anywhere

About Galaxy security

Galaxies are created without security. After creating a Galaxy, you can assign security to manage access to:

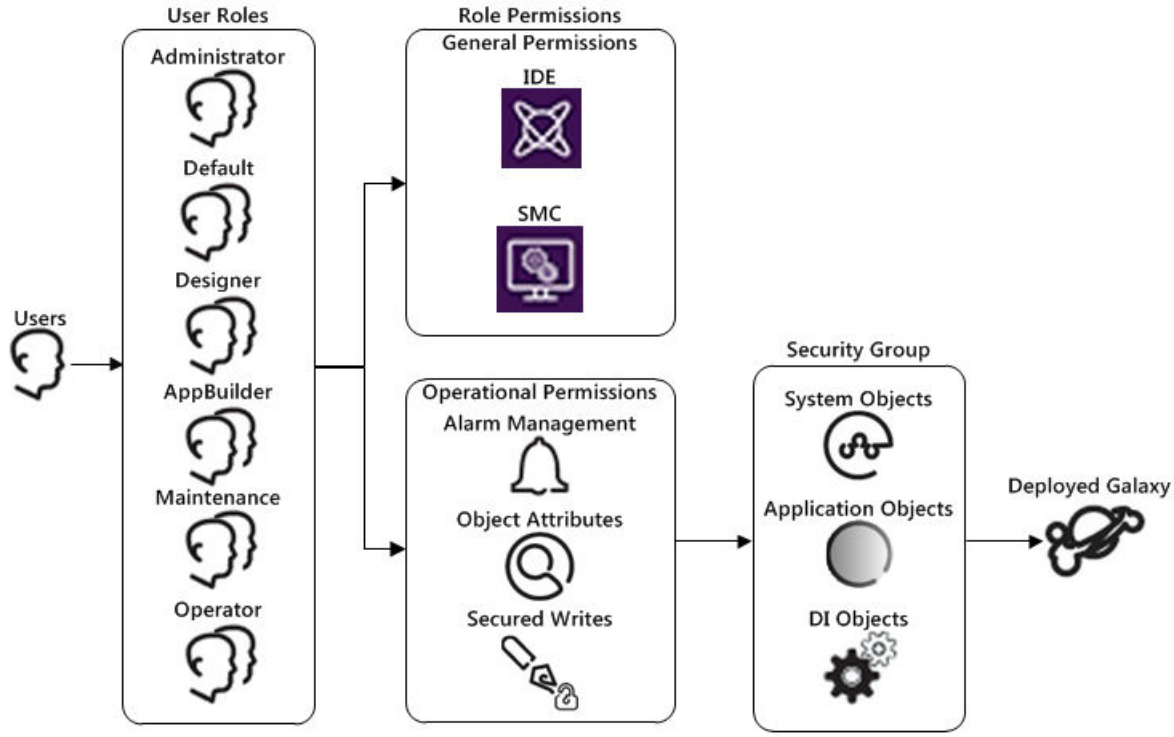
- IDE for configuring and managing objects and graphics
- Operations Control Management Console for performing maintenance and system administration functions
- Alarms, object data, and secured writes for a running ViewApp

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Credentials, which are linked to specific user groups and allow AVEVA OMI apps to connect to external resources that do not recognize Windows credentials. The security authentication mode must be set to OS Group based or OS User based in order to use credentials.

Each Galaxy in the Galaxy Repository manages its own security model. The security schema managed in a Galaxy is a three-level configuration model to create and maintain the following:

- Users assigned to roles
- Security roles granted system administration, configuration, and runtime (operational) permissions
- Security groups containing specific objects in a Galaxy



The security matrix defines a cascading model of users associated with specific roles that are associated with specific security permissions that are associated with specific objects. User runtime permissions can vary from object to object, action to action, and process to process. The security icons associated with object attributes map directly to control points in the security model.

Configure Galaxy security

Galaxy security uses the local Galaxy configuration to authenticate users. All security for the Galaxy is specified and contained at the specific Galaxy level. When the user logs on, security credentials are checked and access to areas and activities is granted at the Galaxy level.

Regardless of the security system, a single user can have multiple accounts. For example, a user can have an account that provides permissions for working with instances but not templates. The same person can have another supervisory account for working with templates and managing users in the ArchestrA environment.

Each account requires a different user name and password. For example:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

User Name	Password	Access
bsmith	password	Instances, not templates
bobsmith	super	Instances, templates, not managing users
Robertsmith	admin	Instances, templates, managing users

After you change security for a Galaxy, you see the following behaviors and conditions:

- When you change the authentication mode security, the IDE restarts.
- To switch users, the person must log on as the new user by clicking **Change User** on the **Galaxy** menu.
- If you previously configured security under one authentication mode and then switch authentication modes, only those users created while configuring the new mode are available. Other users are not deleted, just unavailable in the new mode.
- Objects that are reassigned to different security groups are marked as "pending update" and require redeployment for the change in security group to take effect.
- If security was previously configured for an OS-based authentication mode, reconfiguring security synchronizes the user's full name and OS groups if some data in the OS has changed.

About Galaxy security groups

Every object in a Galaxy belongs to only one security group. You create a security group to contain a set of objects in the Galaxy whose attributes and alarms can be managed by users.

By default, all currently used objects are assigned to the Default security group. A user who is a member of the Default security role are granted all Operational permissions.

A user who is a member of a role assigned to the Default security role has permission to:

- Acknowledge alarms
- Change attribute values with "configure" security mode
- Change attribute values with "operate" security mode, including "secured write" and "verified write"
- Change attribute values with "tune" security mode
- Verify writes to Attributes with "verified write" security mode

Manage security roles

You can create and manage user roles that apply to your organization's processes and work-based authorities. Two roles are defined by default: Administrator and Default.

You can specify General and Operational Permissions for each role.

- General permissions relate to application configuration and administration tasks.

  Note: You cannot modify the General permissions of the Administrator role.

- Operational permissions relate to the security groups listed on the Security Groups page. By default, the Administrator has all permissions.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

There are five basic Operational permissions that can be granted to a security role

- Manage alarms
  - **Can Acknowledge Alarms**: Enables users to manually acknowledge an alarm while a ViewApp is running.
  - **Can Shelve Alarms**: Enables users to shelve and unshelve alarms.
  - **Can Modify Alarm Modes**: Enables users to modify the mode of an alarm.
  - **Can Modify Plant States**: Enables users to modify plant states for state-based alarming.
- **Can Modify "Configure" Attributes**: Enables users to configure the attribute's value. Requires that the user first put the object Off scan. Writing to these attributes is considered a significant configuration change, for example, a PLC register that defines a Discrete Device input.
- **Can Modify "Operate" Attributes**: Enables users with operational permissions to do certain normal day-to-day tasks like changing setpoint, output and control mode for a PID object, or commanding a Discrete Device object.
- **Can Modify "Tune" Attributes**: Allows users to tune an attribute in the runtime environment. Examples of tuning are modifying attribute values that specify alarm setpoints and PID sensitivity.
- **Can Verify Writes**: Enables users to provide an authentication signature for attributes configured with Verified Writes security classification. Only users with this permission can verify a task performed by users with the **Can Modify "Operate" Attributes** permission.

Create a security role

Creating a security role consists of two major steps of adding a role to the Galaxy and then assigning permissions to the role.

Notes About General Permissions:

- In the **SMC** permissions list, clearing the **Can Start SMC** check box still enables a user assigned to this role to start the SMC, but not to connect to Platform Manager.
- In the **General permissions** area, clearing the **Can Start/Stop the Engine/Platform** permission still permits the user assigned to this role to set the engine or platform objects On Scan or Off Scan.
- If a role is granted the **Can Modify Deployed Instances** IDE permission, make sure the **Can Create/Modify/Delete...** permissions in the **System Configuration**, **Device Integration Objects**, and **Application Configuration** groups are also selected. This provides the role with the ability to check in and check out objects.

To create a security role

1. On the **Galaxy** menu, click **Configure** and then click **Security**. The Configure Security dialog box appears.
2. On the **Authentication Mode** tab select **Galaxy**.
3. Click the **Roles** tab.
4. Click the + button to add a security role.

   A role is added to the **Roles available** list named **NewRole**.
5. Change the name of the new security role.
6. Select the new security role in **Roles available** list and assign **General** and **Operational** permissions to the role by selecting the checkbox next to each permission you want to add to the role.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Similar IDE permissions are grouped into lists that can be expanded or hidden. Click the small gray box next to the permission groups to expand or hide the permissions.

7. Click **OK** to save your changes and close the **Configure Security** dialog box.

About the default security role

The Administrator and Default users are defined by default when a new Galaxy is created. These users cannot be deleted in an open security setting and are associated with the default Administrator and Default roles.

The Default security role has permission to:

- Acknowledge alarms
- Change attribute values with “configure” security mode
- Change attribute values with “operate” security mode, including “secured write” and “verified write”
- Change attribute values with “tune” security mode
- Verify writes to Attributes with “verified write” security mode

When you create a new user name, it is assigned to the Default role. For more information about creating users and assigning roles, see the IDE help.

About the Credentials tab

The **Credentials** tab lets you add login credentials that some AVEVA OMI ViewApps may need for access to third-party data and applications that do not support standard authentication methods, such as Windows OS credentials, Active Directory, or OpenID Connect.

Initially, a galaxy does not include any credentials. Credentials are created on a galaxy-wide basis, not for individual ViewApps. Although you enter credentials as plain text, the credentials are encrypted when saved, and the credentials are sent to runtime nodes as encrypted data.

Each credential you create is associated with one and only one OS user group. In order to create credentials, OS group-based or OS user-based security must be configured. The credentials you create here are associated with an OS group name. When a user logs into a ViewApp, the user obtains access to credentials that are associated with the same OS group of which the user is a member. For example, if you create credentials that are associated with the OS group "Operators," a user that is also a member of "Operators" has access to those credentials.

If an app requires credentials to access data, it typically will include a dropdown menu. At runtime, your can select the appropriate credential, that is, a credential assigned to your user group.

Note: Credentials are supported only when **OS User based** or **OS Group based** authentication mode is enabled. If authentication mode is set to **None** or **Galaxy**, you cannot add or delete credentials, or use them at runtime, even if the credentials were created while the authentication mode was set to OS User or OS Group.

Create Named Credentials

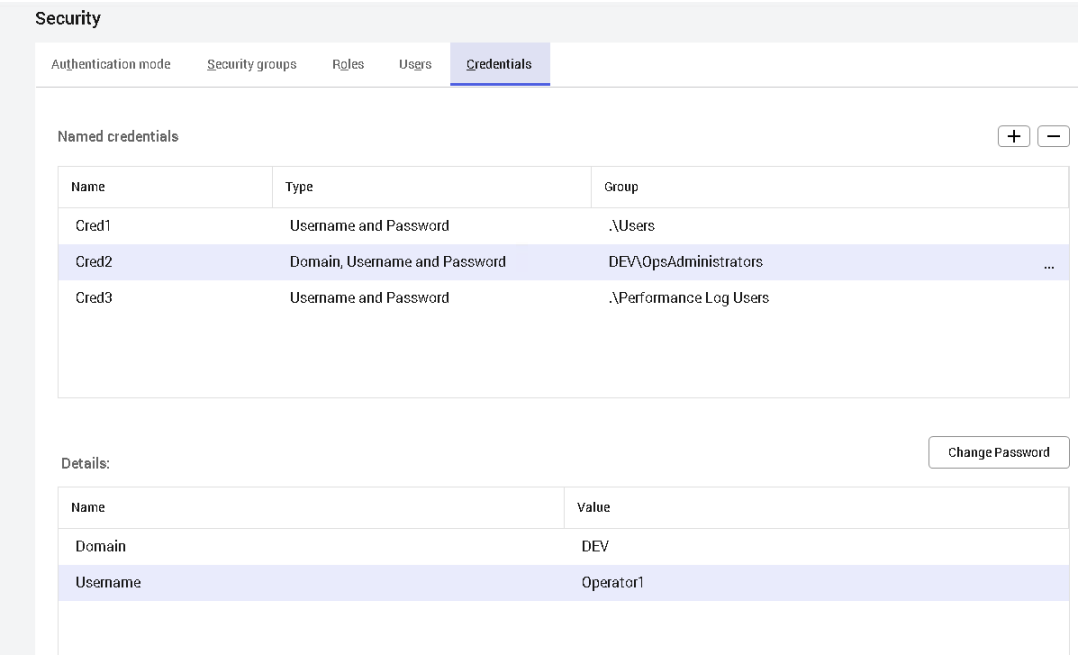
From the **Configure Security** page, select the **Credentials** tab. The **Credentials** tab is divided in three columns:

- Name
- Type
- Group

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To create a new credential

1. Click the **Add Credential** button. A new credential is added to the **Available Credentials** window.
2. Rename the credential by selecting the default name from the **Name** column and then type the new name.
3. Set the credential **Type** by selecting the down arrow at the right edge of the **Type** column. Type can be one of the following:
   - Username and Password
   - Domain, Username and Password
4. To select the **Group** to associate with the credential:
   a. Click the ellipses button (...) at the right of **Group** column. The **Select Groups** dialog opens. See Associate an OS Group with a Credential for more information.
   b. Choose the applicable OS group name from the list of available OS groups, or type the name of an existing OS group.
5. Configure the credential after you have created a credential by entering values for Domain, Username, and Password. See Configure Credentials for additional information.



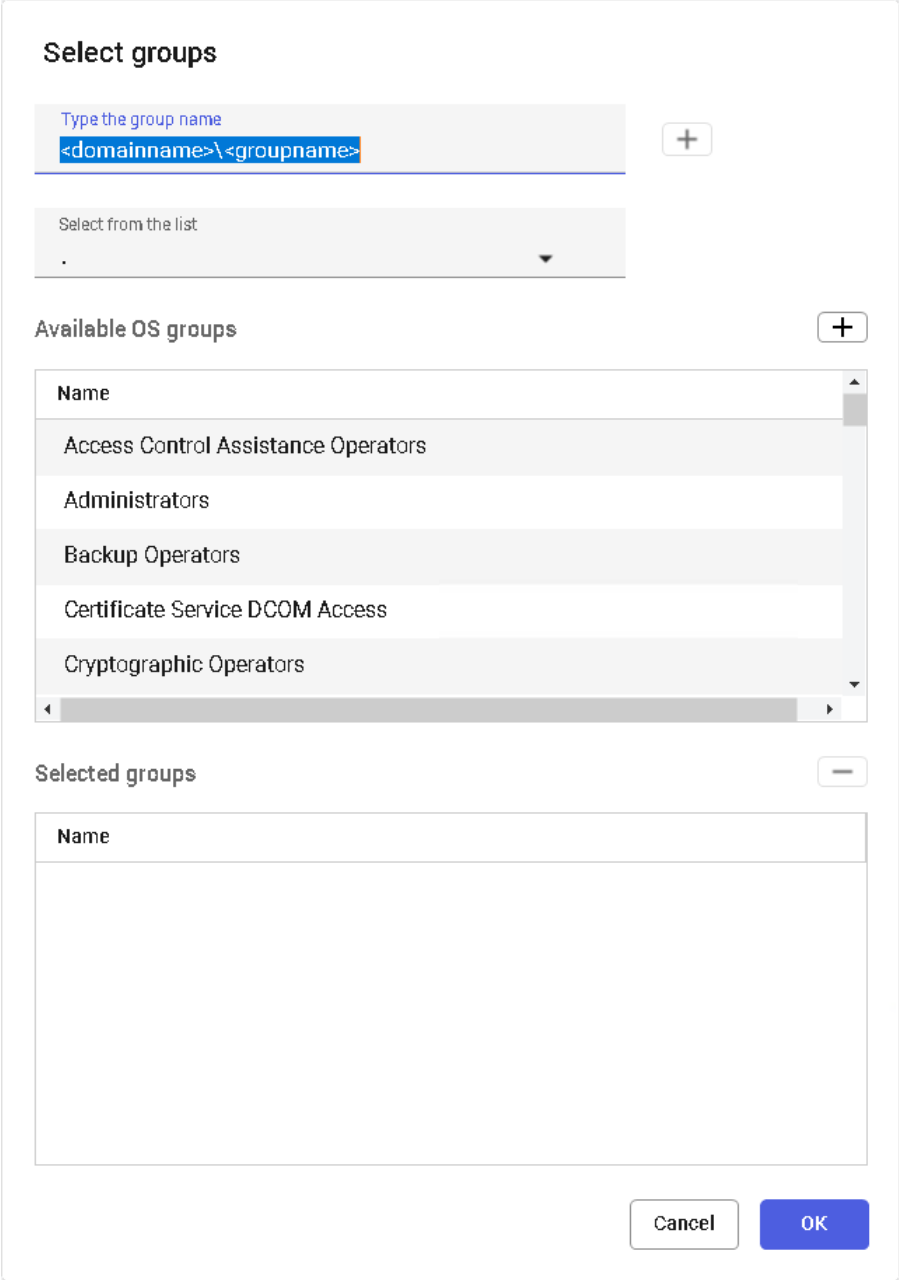
Associate an OS Group with a Credential

To associate an OS group with a credential

1. Select the the ellipses button (...) next to the group name in the **Credentials** tab.

   The **Select groups** dialog opens with the local machine selected, and the list of OS groups that exist on the local machine will populate the list of **Available OS groups**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



2. Choose an existing OS group from the **Available OS groups** list. Or,type the domain and OS group name in the **Type the group name** text box.
3. Add the OS group by clicking the **+** button above the list, on the right side. The OS group is added in the **Selected groups** window.
   - If you are creating a credential that includes domain, select the applicable domain from the **Select from the list** dropdown list, then select the OS group name from the **Available OS groups** window. If the credential type you created does not include domain, you will not be able to select another domain.
   - If you manually enter the domain and/or the OS group name, use a backslash to separate the domain from the group name.
   - If the OS group is on the local machine, you must include ".\" before the group name.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

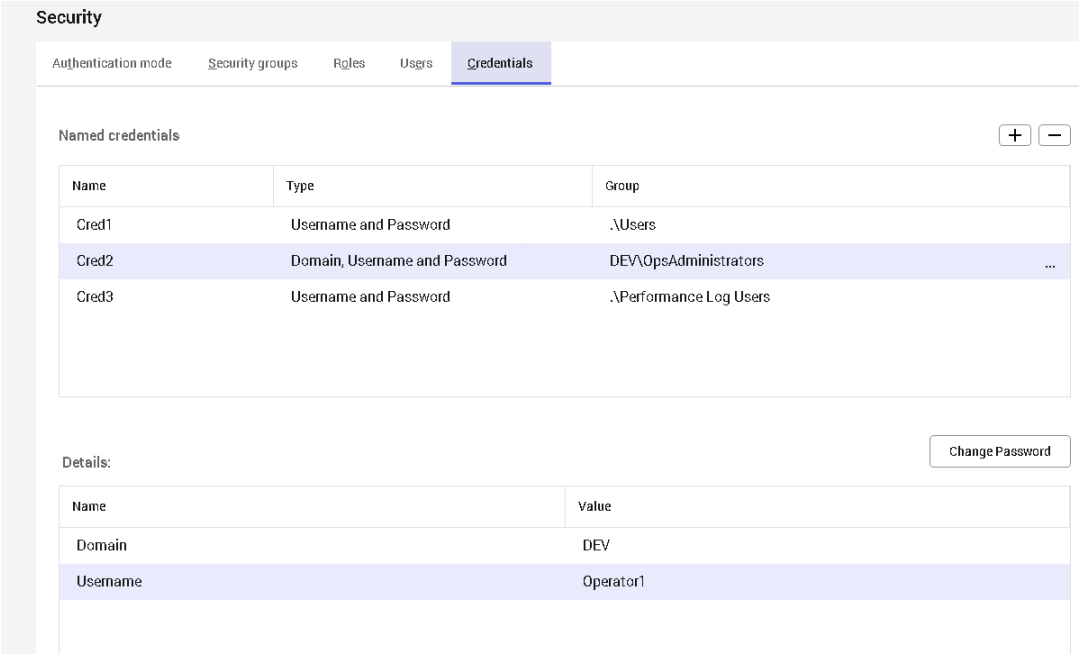
Note: Only one OS group can be added per credential.

4. Click **OK** to close the **Select groups** dialog and return the **Credentials** tab.

Configure Credentials

To configure a credential

1. Select the credential to be configured in the **Named credentials** lsit.



2. In the **Details** window, enter the **Username** for the credential. If the credential type is Domain, Username, and Password, also enter the **Domain** name.
3. To enter the password for the user, select **Change password**.
4. On the **Change password** dialog box, enter and confirm the password for the user name you entered. Select **Save**.

   You can enter null values for the username and password. If a domain value is required, enter the domain name that was used to create the credential.

5. Select **Save** to save the credentials.

Credential Definitions

Name: This is the name of the credential. The default credential name is "Credential," and subsequent names are appended by a number that automatically is incremented by one. To rename a credential, just highlight the name and enter new one. Credential names must contain at least one character and must be unique.

Configured credentials are typically shown in a dropdown menu in a ViewApp that has been configured to use credentials. The user can then select the applicable credential. In some cases, an app developer may choose to hard-code a credential name. In this case, no user action is required to access the credential at runtime.

Type: Two types of credentials are supported:

- **Username and Password.**

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- **Domain, Username and Password.**

Note: Null values are valid for all fields, except the credential name.

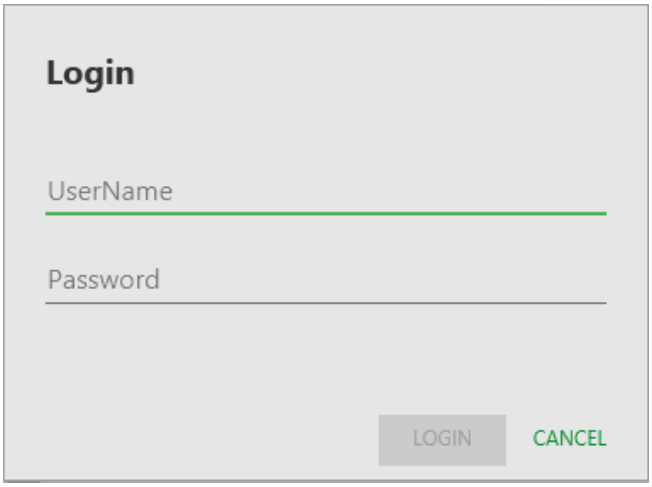
Group: When you add credentials, you must associate each credential with an OS Group name (local or domain) that has been defined previously.

To associate a credential with an OS group, simply select the group name when you add the credential. You can change the group at any time in the **Credentials** tab, but doing so may impact AVEVA OMI apps that have been configure to use the credential. Associations are saved when you click **OK**. See Associate an OS Group with a Credential for more information.

Apply log in security

Log in security can be applied to restrict access to a ViewApp to only authorized users. With security, operators or other users must authenticate themselves with their username and password. Based on how the ViewApp is configured, operators may be able to interact with a ViewApp even though they have not logged in. When a operator logs in they can perform Operate Level writes. Also, operators can perform any actions granted to them on the basis of logging in to a ViewApp..

You configure a ViewApp to show an initial **Login** dialog box when a ViewApp is deployed and started. Typically, an operator will be presented with a button to select and show the **Login** dialog box.



Apply log in security example

This topic describes a simple example of implementing login security interface to permit only authenticated users access to a ViewApp.

Add Users to a Security Authentication System

To implement ViewApp login security, users must be an authenticated member of an AVEVA OMI security system. System Platform AVEVA OMI provides three security modes:

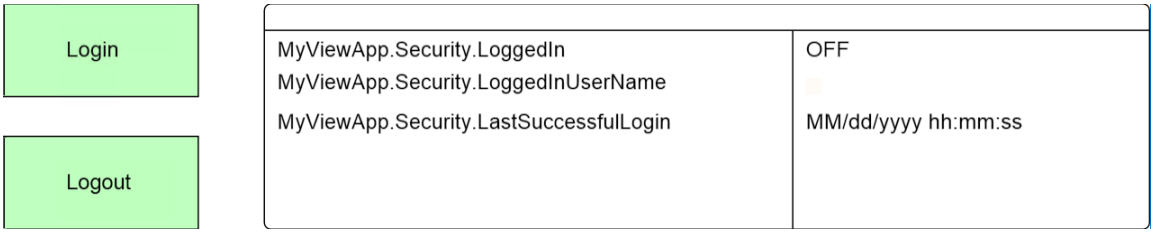
- Galaxy security
- Windows User-Based security
- Windows Group-Based security

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Based on the authentication mode you select, ViewApp users must be added to the security system with a user name and password. The user should be assigned a role in the security authentication system with a defined Access level.

Build a Login Interface

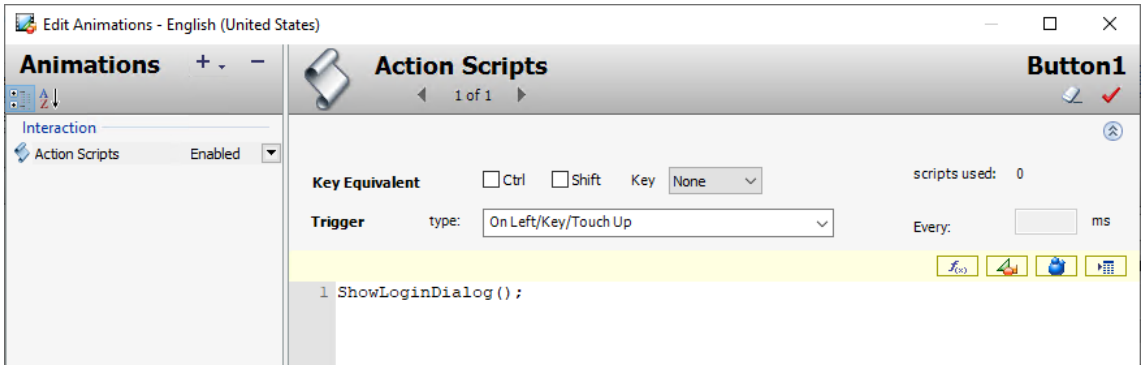
A typical login interface includes a login button that is selected by the user to show the **Login** dialog box with fields to enter a username and password. Also, a log off button can be added for the user to log off from the ViewApp.



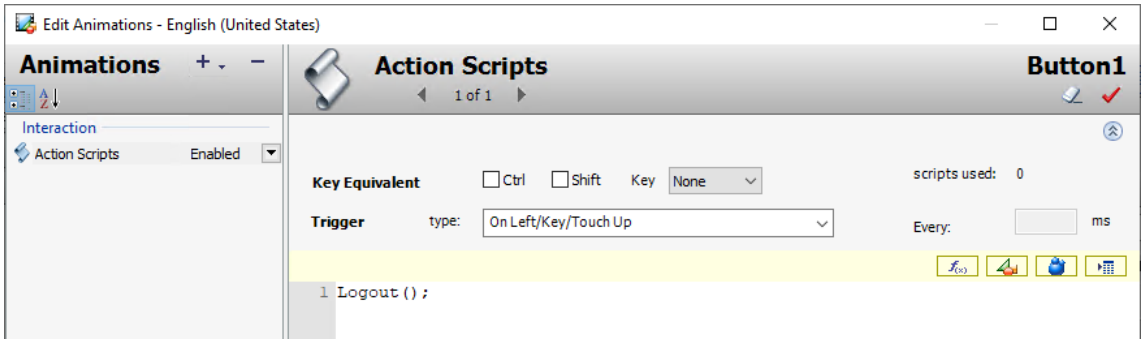
This example of a login interface includes a table that shows the current value of the login security attributes. The attribute status column includes text elements with value display animation mapped to the login security attributes shown in the left column of the table.

Add Action Scripts to the Login and Logout Buttons

- The Login button should include an action script calling the ShowLoginDialog() method, which shows the **Login** dialog box.



- The Logout button should include an action script calling the Logout() method, which logs the user out of the ViewApp.



© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

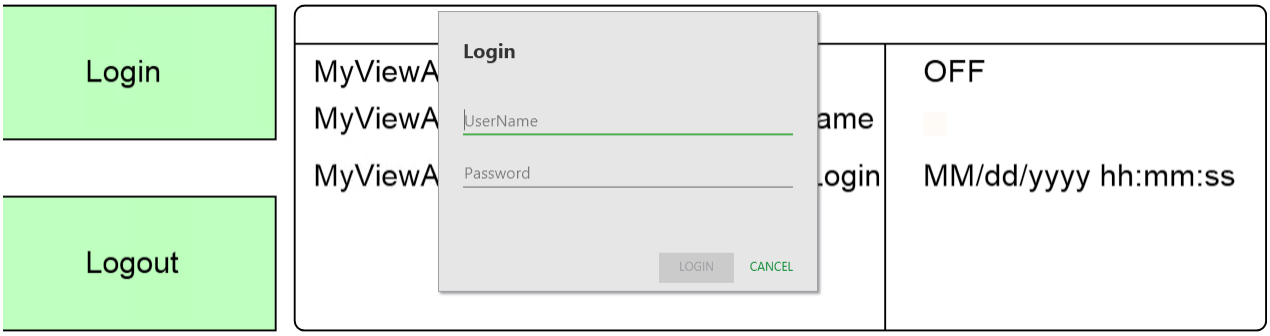
Use Security Attributes with Value Display Animations

Add Value Display animation to a graphic element that you want to use to show security information. Configure the animation to use the Security attributes as reference values.

For information about the animation data types to assign Security attributes, see About security attribute data types.

Test the Interface

Run the ViewApp that includes a login security interface. After selecting the Login button, you should see the **Login** dialog box.



After entering a valid user name and password, the **Login** dialog box should disappear and the security attributes should show that a user is logged on, the user name of the logged on user, and the time when the successful log on attempt was made.

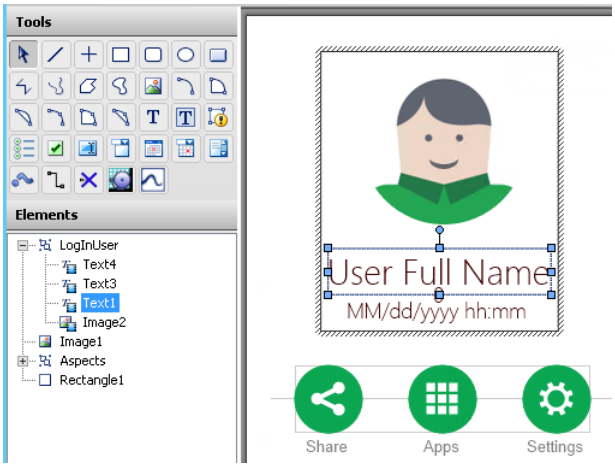
About security attributes

Security attributes are read only attributes that are typically set as reference values of Value Display animation.After a user logs on to a ViewApp, the graphic elements configured with Security attributes show the authorization characteristics of the logged on user.

Security attributes can show the logged on user's user name, and Access Level. Also, the LastSuccessfulLogin and LoggedIn attributes show the date and time of the last successful logon to a ViewApp and if a user is currently logged on to a ViewApp or not.

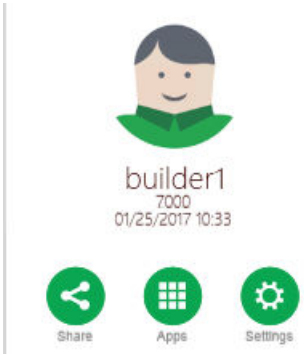
Graphic element to show user name:

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



Value display animation:

ViewApp (runtime):



Configure security attributes

This topic explains the general sequence of steps to implement a Security attribute in a graphic.

Note: Security must be enabled and users defined in the security system to be able to show user security characteristics as reference values of Security attributes.

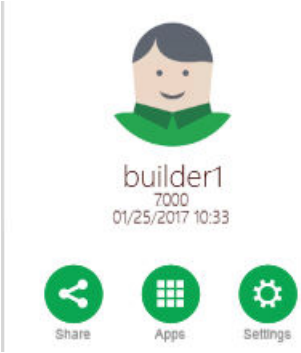
To configure Security attributes

1. Place a text box within a graphic.
2. Using a mouse or touch, select the text element to show the **Animations** dialog box.
3. Select the **Add Animation** button to show the list of possible animations.
4. Select **Value Display** animation as the type of animation to show the current time in the text box added to the graphic.
5. Select the data type and enter the fully qualified Security attribute name.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

In this example, **String** is the data type for the **LoggedInUserName** attribute. For more information about the data types associated with each Security attribute, see More Information About Security Attributes (see About security attribute data types on page 726).

6. Save your changes to the graphic that contains the text element configured with the Security attribute.
7. Deploy the ViewApp containing the graphic.
8. When the ViewApp is running, you should see the user's log in name in the graphic's text field configured with the security attribute.



About security attribute data types

All security attributes operate in the MyViewApp.Security namespace. The names of security attributes are specified with the prefix MyViewApp.Security in the form MyViewApp.Security.*attribute_name*.

Attribute Name	Data Type	Attribute Type	Initial Value	Retentive	Description
AutoLogon CurrentUser	Boolean	Read/ Write	True	True	Returns true or false to indicate if users can log on automatically to a ViewApp with their Windows credentials.
AutoLogOff	Boolean	Read/ Write	False	True	Returns true or false to indicate if a user will be logged off after a period of inactivity. When set to true, the user is logged off from the ViewApp when the inactivity period is exceeded.
AutoLogOff TimeSpan	Integer	Read / Write	600	True	Specifies the length of the inactivity period in seconds a logged in user is allowed.
AutoLogOff Remaining Time	Integer	Read Only	0	False	Returns the remaining length of a user's inactivity period in seconds.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Attribute Name	Data Type	Attribute Type	Initial Value	Retentive	Description
Last Successful Login	DateTime	Read Only	1/1/1601 12:01:01.000 AM	True	UTC timestamp of the most recent successful log in to a ViewApp. When no user is currently logged on to a running ViewApp, **LastSuccessfulLogin** shows 01/01/0001 00:00:00 as the timestamp.
LoggedIn	Boolean	Read Only	False	False	Returns True or False to indicate if a user is currently logged into a ViewApp or not.
LoggedInAccessLevel	Integer	Read Only	No initial value	False	Integer value of the Access Level of the security role assigned to the user logged in to a ViewApp. When no user is currently logged in to a running ViewApp, such as after an AutoLogOff, **LoggedInAccessLevel** returns **-1**.
LoggedInUserName	String	Read Only	No initial value	False	Returns the user name specified by the user to log in to a Viewapp.
LoggedInUserRoles	String	Read Only	No initial value	False	Returns the security roles assigned to the logged-in user in CSV format.

Manage industrial graphics

This section describes Industrial Graphics, how they are stored in the ArchestrA environment, and how they are managed from the IDE.

Industrial Graphics are graphics you use to visualize data in an AVEVA OMI ViewApp.

You manage Industrial Graphics from the IDE to:

- Create a new graphic.
- Edit a graphic with the Industrial Graphic Editor.
- Organize graphics within the Visualization folder.
- Duplicate graphics.
- Import and export graphics.
- Delete a graphics.
- Configure security for a graphic's operations.
- Open the graphic in read-only mode with the Industrial Graphic Editor.

Import and export graphics

You can import graphics to and export them from the Visualization folder.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

Import graphics from .aaPKG files

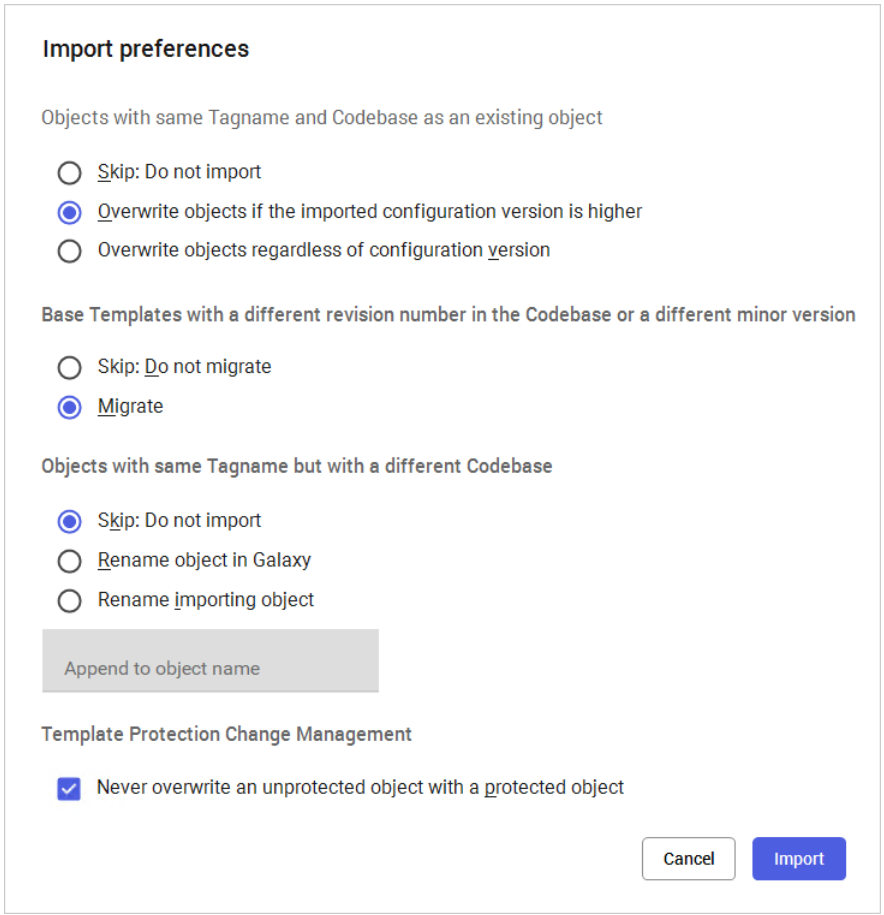
You can import graphics and folders from a graphic .aaPKG file.

When you import templates or instances that contain graphics, the graphics are imported with the template or instance.

When you import all AutomationObjects, the contained graphics and the graphics in the Visualization folder are also imported..

To import graphics from a graphic .aaPKG file

1. On the ribbon, select **Galaxy**, then **Import**.
2. Select **Objects**, then **From package**. The **Import Objects from package** dialog box opens.
3. Select one or more graphic .aaPKG files you want to import and click **Open**. The **Import Preferences** dialog box appears.



4. Select the appropriate options for the import and click **Import**. The graphics and folders are imported.
5. Click **Close**.

Export graphics

You can export graphics to a graphic.aaPKG file.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

When you export templates or instances that contain graphics, the graphics are exported with the template or instance.

When you export all AutomationObjects, the contained graphics and the graphics in the Visualization folder are also exported.

When you export an AutomationObject that contains graphics, if these graphics contain embedded Industrial Graphics from the Visualization folder, they are exported along with the graphics associated with the AutomationObject.

If the graphics contain other AutomationObjects, the graphics, including any embedded graphics from the Visualization folder contained in them, and their parent AutomationObject are not exported with the AutomationObject. They remain as references. Upon import, the system tries to reestablish the connection with those exact references in the target Galaxy.

To export graphics to a graphic .aaPKG file

1. In the Visualization folder, select the graphics to export.
2. On the **Home** ribbon, in the **Export** area, select **Selected**, then **As package**.

   If you are exporting a graphic containing embedded graphics associated with an object, you see the **Export Selected objects** dialog box with an option to include the associated automation objects hosting the graphics to be exported.
3. Select whether to include the hosting objects or just the selected graphics, then click **Export**.

   The **Export Selected objects** dialog box appears.
4. Browse to the save location and type a name for the graphic .aaPKG file.
5. Click **Save**. The graphics and their folders are exported.
6. Click **Close**.

Programmatically import and export industrial graphics

You can use the Industrial GraphicAccess application programming interface (API) to programmatically export a graphic from the Visualization folder to an XML file. You can use the same API to import a graphic from an XML file to create Industrial Graphics in another galaxy or overwrite an existing graphics. You can also import a graphic XML file created by another application to an ArchestrA Galaxy.

The programmatic API exports or imports an extensive set of properties of an Industrial graphic. For standard Industrial Graphics, an exported or imported graphic contains the following:

- Custom properties
- Graphic elements
- Connector lines
- Graphic groups
- Graphic animations
- Element styles
- Named scripts
- Predefined scripts

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

- Overridden text strings
- Numeric format styles
- DataStatus elements
- Trend Pen
- Alarm Client
- Trend Client

In addition to the properties of a standard Industrial graphic, an exported or imported Symbol Wizard contains the following:

- Wizard Options
- Choice groups
- Choices
- Layers
- Rules

Implement and use the GraphicAccess API

The set of GraphicAccess application programming interfaces (API) allow users to export graphics to an XML file programmatically. Use the same set of APIs to create an Industrial Graphic in another galaxy or to overwrite an existing graphic(s) by importing the graphic definition from an XML file. The APIs can also be used to export the configured references or GPI for a graphic and importing multiple XML files from a folder location.

Before using the set of GraphicAccess APIs, you must use the GRAccess Toolkit to log in and connect to an Galaxy repository.

The APIs are available through the following interfaces:

IGraphicAccess – Export and import an Industrial Graphic

IGraphicAccess2 – Export the substitutable strings and configured references for an Industrial Graphic

IGraphicAccess3 – Export the GPI of an Industrial Graphic

IGraphicAccess4 – Import Industrial Graphics from a folder location

For details about how to use these APIs, see IGraphicAccess in the System Platform help file.

Exporting and Importing Overridden Text Strings

The ArchestrA programmatic API supports overridden text strings in graphics. Typically, a string substitution is performed on a text string from the Industrial Graphic Editor before exporting the graphic with the programmatic API. The exported XML file shows the content of overridden and substituted text strings as elements and attributes of the SubstitutedStrings element.

When a graphic is imported using the programmatic API, the graphic shows the text string assigned to the New attribute of the XML file's String element. Users can substitute text strings before importing a graphic by editing the contents of the String element in the exported XML file.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

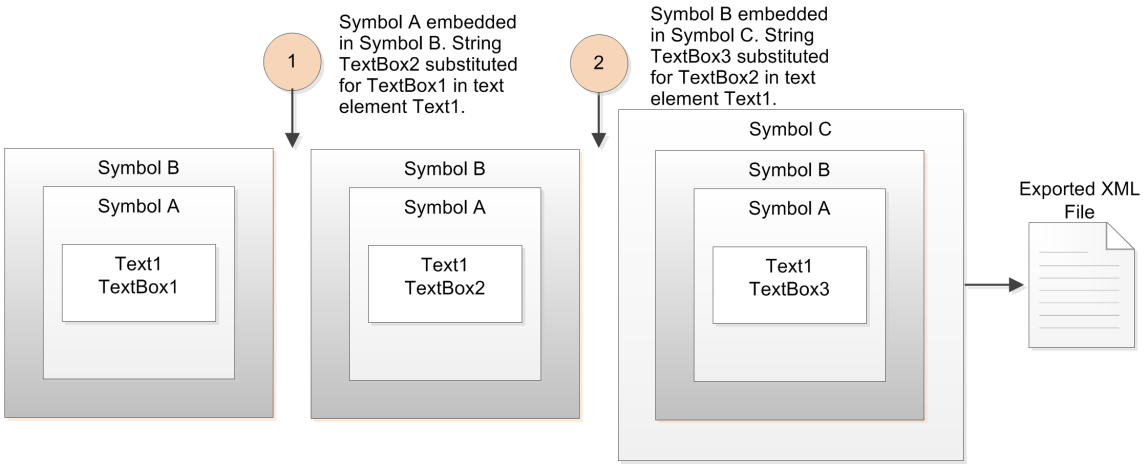
Exporting Overridden Text Strings

After exporting a graphic, the XML file's String element specifies the original string and the new substituted string with the Old and New attributes. The ElementID attribute specifies the specific text element to be overridden.

```
<SubstituteStrings>
   <String Old="TextBox1" New="TextBox2" ElementID="A.Text1"/>
<SubstituteStrings>
```

In this example, the TextBox1 text string was overridden by the new substituted text string TextBox2. The string substitution occurred in Symbol B that substituted the text of the text element in Symbol A.

Text strings can be overridden within graphics that are embedded within other graphics. Consider an example that shows Symbol A embedded in Symbol B, which is embedded in Symbol C. Two string substitutions are made to a single text element.



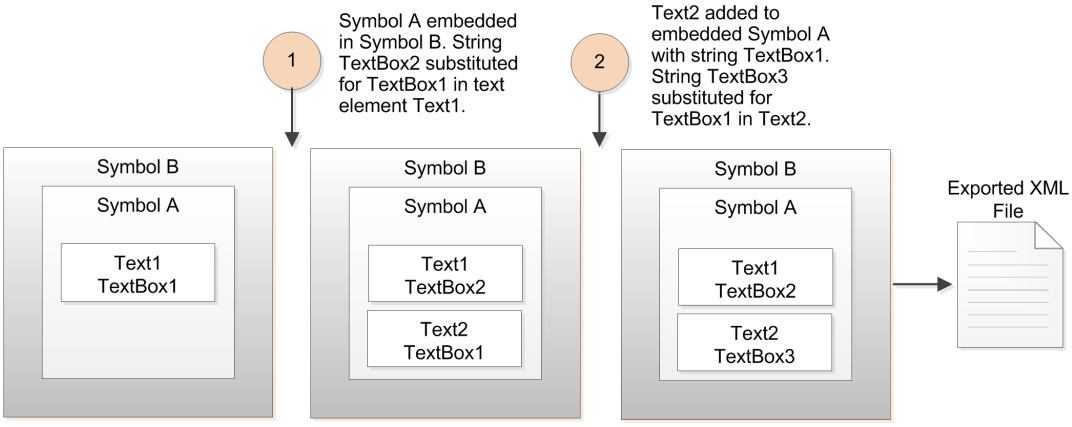
The export XML file's SubstituteStrings element shows the text string override information.

```
<SubstituteStrings>
   <String Old="TextBox2" New="TextBox3" ElementID="B.A.Text1"/>
<SubstituteStrings>
```

Notice the Old and New attributes show the text strings of the most recent string substitution before Symbol C was exported. Also, the ElementID attribute indicates Symbol A containing text element Text1 is embedded in Symbol B in the form ElementID="B.A.Text1".

The programmatic API can export an embedded graphic containing multiple text elements with the same text string that is overridden by different string substitutions. Consider an example that shows Symbol A embedded in Symbol B. Two string substitutions are made to a single text string in two text elements of embedded Symbol A.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.



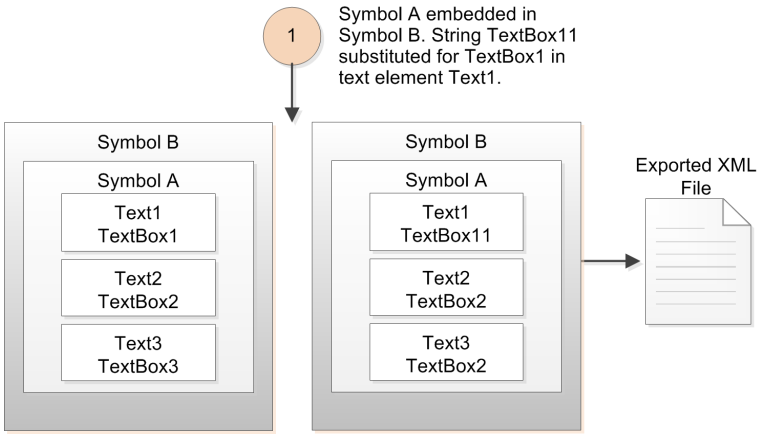
The export XML file's SubstituteStrings element shows the text string override information with a String element for each text override.

```
<SubstituteStrings>
   <String Old="TextBox1" New="TextBox2" ElementID="B.A.Text1"/>
   <String Old="TextBox1" New="TextBox3" ElementID="B.A.Text2"/>
<SubstituteStrings>
```

Importing Overridden Text Strings

An exported graphic's text strings can be overridden by editing the SubstituteString elements of the export XML file before importing the graphic.

The following illustration shows Symbol A embedded in Symbol B. Symbol A contains three text elements with text strings. The TextBox1 text string of the Text1 element was overridden to Textbox11. Then, Symbol B was exported using the programmatic API.



The expected SubstituteStrings element in the export XML file should be similar to the following:

```
<SubstituteStrings>
   <String Old="TextBox1" New="TextBox11" ElementID="A.Text1"/>
</SubstituteStrings>
```

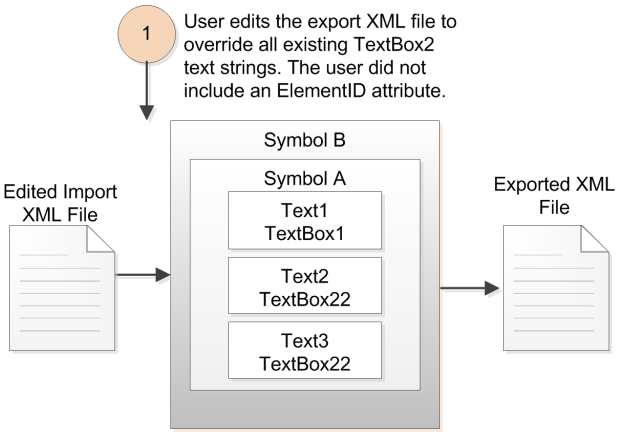
The user edits the export XML file and adds an entry to override a text string without using an ElementID attribute to identify the text element.

```
<SubstituteStrings>
   <String Old="TextBox1" New="TextBox11" ElementID="A.Text1"/>
   <String Old="TextBox2" New="TextBox22">
```

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

```
</SubstituteStrings>
```

After importing the graphic, the text strings for the Text2 and Text3 elements are overridden to TextBox22. Without an ElementID to identify the text element, the string override replaces all text strings that match the text string specified by the Old attribute.



If the graphic was exported again using the programmatic API, the exported XML file shows that ElementID attributes have been added to each String element.

```
<SubstituteStrings>
        <String Old="TextBox1" New="TextBox11" ElementID="A.Text1"/>
        <String Old="TextBox2" New="TextBox22" ElementID="A.Text2"/>
        <String Old="TextBox2" New="TextBox22" ElementID="A.Text3"/>
</SubstituteStrings>
```

Configure security for graphics

You can set IDE security permissions so that at design time, the user cannot:

- Import or export graphics.
- Create, modify, or delete graphics in the Visualization folder.
- Create, modify, or delete graphics in any AutomationObject template.
- Create, modify, or delete graphics in any AutomationObject instance.
- Create, modify, or delete View Applications, such as InTouchView Applications.
- Deploy or undeploy View Applications, such as InTouchView Applications.
- Edit the configuration of the quality and status display.

To restrict a user, the user must be assigned to a role and the permissions must be assigned to that role, and security must be enabled. For more information on how to configure security, users, and roles, see the Application Server User's Guide.

If the user attempts to export a graphic without appropriate permissions, the message "User doesn't have permission to export graphics object" appears.

If the user attempts to import a graphic without appropriate permissions, the message "User doesn't have permission to import graphics object" appears.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.

To configure security for graphics

1. On the ribbon, select **Galaxy**, then **Configure**, and then **Security**. The **Security** page appears.
2. Click the **Roles** tab.
3. In the **Roles available** list, click the role you want to assign permissions to.
4. In the **General Permissions** list, expand the **Graphic management permissions**.
5. Select the permissions that you want uses assigned this role to have. Clear the check box for each action that the users *should not* be able to perform.
6. Click **Save**.

© 2015-2024 AVEVA Group Limited or its subsidiaries. All rights reserved.