AVEVA InTouchHMI

aveva.com

© 2015-2026 AVEVA Group Limited 或其子公司版权所有。保留所有权利。

未经 AVEVA Group Limited 事先书面明确同意，不得以任何形式或通过任何手段（机械、影印、录制或其它方式）复制、传输本出版物中的任何部分，或是将其存储到检索系统。AVEVA 对用户使用本文档中所含信息产生的任何问题，不承担任何责任。

虽然在准备本文档时已采取预防措施，但 AVEVA 对文本中可能出现的错误或遗漏不承担任何责任。本文档中的信息如有更改，恕不另行通知，并且不代表 AVEVA 做出任何承诺。本文档中所述的软件根据许可协议提供。只能根据此类许可协议的条款使用或复制此软件。AVEVA、AVEVA 徽标和标识、OSIsoft、OSIsoft 徽标和标识、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 和 Wonderware 是 AVEVA 和/或其子公司的商标。所有其他品牌可能是其各自所有者的商标。

美国政府权利

美国政府使用、复制或公开本软件内容受 AVEVA Group Limited 或子公司许可协议以及 DFARS 227.7202、DFARS 252.227-7013、FAR 12-212、FAR 52.227-19 或其后续协议（如适用）中相关条款限制。

出版日期：Thursday, September 10, 2026

出版物 ID：1632349

联系信息

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

如需销售和客户培训人员的联络信息，请参阅 。

如需技术支持人员的联络信息，请参阅 。

要访问 AVEVA 知识与支持中心，请访问 https://softwaresupport.aveva.com。

目录

Standard. 21

集成开发环境. 22

关于 InTouch 应用程序管理器. 22

使用应用程序浏览器. 22

在应用程序浏览器中浏览. 24

将应用程序添加到应用程序浏览器. 25

管理工具栏. 26

设置 WindowMaker 首选项. 26

鼠标快捷方式. 28

使用全屏模式. 29

从文件菜单管理窗口. 29

设置字体缺省值. 30

使用方向键移动对象. 30

使用图形画布和图形定位点. 31

使用调色板. 32

打开调色板. 32

创建自定义颜色. 33

导入与导出自定义颜色. 33

平移与缩放. 34

使用缩略图窗口进行平移和缩放. 34

使用鼠标滚轮进行缩放与平移. 35

平移与缩放限制. 35

使用屏幕网格与标尺. 36

将对象对齐到网格. 36

使用标尺. 36

关于 WindowViewer. 37

自定义运行时环境. 37

配置远程会话中运行的应用程序的用户访问权限. 41

关于管理 WindowViewer 的内存. 42

使用 WindowViewer 窗口. 50

工业图形. 51

创建工业图形. 52

工业图形编辑器. 52

应用程序. 53

独立的 InTouch 应用程序. 53

托管的 InTouch 应用程序. 54

发布的 InTouch 应用程序. 55

InTouchView 应用程序. 56

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

应用程序服务器架构. 57
Galaxy 内的通讯. 58
比较独立、托管及发布的 InTouch 应用程序. 59
构建应用程序. 60
运行应用程序. 61
标记. 62
使用 InTouch 标记. 62
InTouch 标记的类型. 63
内存标记. 63
I/O 标记. 64
间接标记. 65
其它标记. 66
系统标记. 66
系统标记引用. 67
标记属性. 71
内存标记属性. 72
I/O 标记属性. 73
远程标记引用. 75
定义间接标记. 75
在脚本中使用间接标记. 76
为本地标记使用间接标记. 76
对远程引用使用间接标记. 77
定义可复用的标记结构. 78
管理 SuperTag 模板与成员标记. 80
SuperTag 实例. 81
SuperTag 属性. 86
引用 SuperTag 成员. 87
使用批量导入实用程序导入 SuperTag. 87
在 InTouch 应用程序迁移期间导入资产. 87
从其它应用程序访问历史标记值. 88
使用 DDE 项目显示历史数据. 88
使用 DDE 访问记录数据. 90
从其它应用程序中访问历史数据. 98
排解 HistData 错误. 99
IEEE 十进制单位. 101
在 Historian HMI 中显示浮点数. 101
配置和使用 InTouch OPC UA 服务器. 102
OPC UA 配置检查清单. 102
配置 InTouch OPC UA 服务器. 102
配置 OPC UA 服务的防火墙. 104
为第三方 OPC UA 客户端应用程序配置服务器和客户端证书. 108
使用 OI Gateway 配置客户端安全证书. 111
报警. 116
关于 InTouch 报警. 116
报警优先级. 117
报警子状态. 117
报警确认. 117
报警组. 117

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

关于 InTouch 事件. 118
InTouch 报警的类型. 118
离散报警. 119
模拟报警. 119
InTouch 分布式报警系统. 120
报警供应器与接收器. 122
分布式报警组列表. 122
摘要报警与历史报警. 123
报警禁用、约束及抑制. 123
终端服务报警支持. 124
分布式报警系统数据储存. 124

Build. 125

管理 InTouch HMI 应用程序. 126
启动应用程序管理器. 126
使用应用程序管理器. 127
创建 InTouch 应用程序. 128
创建新的 InTouchView 应用程序. 130
将 InTouch 应用程序更改为 InTouchView 应用程序. 130
在 WindowMaker 与 WindowViewer 中打开应用程序. 131
自定义应用程序管理器窗口. 132
使用应用程序磁贴. 132
锁定与解锁 InTouch 应用程序. 133
修改 InTouch 应用程序. 134
从应用程序管理器中删除 InTouch 应用程序. 136
查找 InTouch 应用程序. 137
使用应用程序模板. 137
查看 InTouch 应用程序文件夹. 139
导出 InTouch 应用程序以用作模板. 139
使用应用程序管理器更新 InTouch Web 客户端设置. 140
启动 InTouch OMI ViewApp. 141
向 AVEVA Identity Manager 进行注册. 141
使用凭据管理器. 142
管理命名凭据. 142
导出和导入命名凭据. 143

使用 IDE 管理 InTouch 应用程序. 146
创建托管的 InTouch 应用程序. 146
从应用程序模板创建托管的应用程序. 150
从 IDE 启动 WindowMaker. 152
托管的 InTouch 应用程序的语言切换. 153
提交 InTouch 应用程序的更改. 153
导入 InTouch 应用程序. 154
导入与导出 InTouchViewApp 对象. 155
导出与导入标记数据. 155
保留标记值与参数. 156
删除托管的 InTouch 应用程序。. 157
在 InTouchViewApp 中关联所有 Galaxy 图形. 157

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

窗口 159
创建应用程序窗口 159
将应用程序窗口设置为模板窗口 161
从模板窗口创建应用程序窗口 162
使用框架窗口 163
使用属性面板 164
使用框架窗口中嵌入的图形 166
使用框架窗口上的功能区 167
修改应用程序窗口 168
打开、保存和关闭窗口 168
查看窗口的缩略图预览 169
创建窗口的副本 170
删除窗口 170
使用快速访问工具栏 171
打印有关 InTouch 窗口的信息 172
从命令提示符中打印窗口信息 174
用于打印窗口的 .CSV 格式 174
从命令提示符中进行打印的语法 175
配置上下文菜单的深度 175
图形元素 177
WindowMaker 对象 177
关于 WindowMaker 对象 177
简单对象 178
复杂对象 180
常见操作 181
所有对象的特殊操作 188
特殊对象的特殊操作 189
动画对象 194
使用工业图形编辑器 (IGE) 228
管理工业图形 228
用于 HMI/SCADA 应用程序的工业图形编辑器工作流程 228
符号更改传播 232
符号动态大小传播 232
比较 WindowMaker 与工业图形编辑器 233
连接动画与 InTouch 标记 241
在 WindowMaker 中使用工业图形 244
关于在 WindowMaker 中使用工业图形 244
将工业图形嵌入 InTouch 窗口 245
调整嵌入的工业图形的大小 250
在 WindowMaker 中配置工业图形 250
在“工业图形编辑器”中编辑工业图形 260
使用 InTouch 标记创建图形元素和工业图形 262
从工业图形编辑器的工具箱选项卡嵌入图形 263
在 WindowViewer 中测试工业图形 264
评估图形性能 266
创建新的自动化实例 266
自动创建新的 Application Server 对象实例 266
将 InTouch 窗口转换为工业图形 267

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

准备转换窗口 267
窗口转换规则和例外 267
转换动画脚本 267
窗口转换的已知限制 268
转换窗口之后 269
窗口转换过程 269
诊断窗口转换错误 270
使用符号向导编辑器创建符号向导 271
创建符号向导 272
符号向导设计者工作流程 272
符号向导使用者工作流程 272
了解趋势笔历史数据检索 273
在运行时期间更改趋势笔属性 274
MinValue 属性 275
MaxValue 属性 275
StartTime 属性 276
EndTime 属性 276
PlotType 属性 276
TimeMode 属性 276
FillTrend 属性 277
配置多笔趋势 277
配置笔名和引用 278
配置笔详细信息 279
配置笔选项 280
配置源 280
自定义多笔趋势 280
在运行时使用多笔趋势 282
关于 SmartSymbol 290
SmartSymbol 管理器与库 291
InTouch SmartSymbol 与工业图形 SmartSymbol 291
SmartSymbol 的限制 294
创建 SmartSymbol 模板与实例 294
管理 SmartSymbol 301
编辑 SmartSymbol 306
迁移 InTouch SmartSymbol 311
标记 318
使用标记名字典管理标记 318
规划标记的使用 319
创建新标记 320
配置标记属性 321
修改标记 326
从 OPC UA 服务器创建 InTouch 标记 327
删除标记 331
打印标记列表与使用信息 332
使用标记点域来查看或更改标记属性 332
标记类型可用的点域 332
更改标记的值极限 341
使用 I/O 进行数据访问 348

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用 Gateway Communication Driver. 349
支持的 InTouch 通讯协议. 351
设置访问名. 359
使用 I/O 标记访问 I/O 数据. 361
将标记转换为远程引用. 368
通过远程引用访问 I/O 数据. 372
从 InTouch 访问 Application Server 数据. 375
查看 I/O 标记的时间标签与质量信息. 384
在运行时初始化与重置 I/O 连接. 394
使用访问名的故障转移功能. 397
监视 I/O 连接的状态. 403
从其它应用程序中访问 InTouch 标记数据. 406
记录标记值. 407
配置历史记录. 407
在运行时启动与停止历史记录. 414
绘制标记数据的趋势. 414
InTouch 趋势的类型. 415
在历史趋势中显示保存的标记值. 416
使用历史趋势对象. 416
使用历史趋势向导. 431
使用脚本控制历史趋势向导. 435
使用实时趋势对象. 446
在运行时打印趋势. 449
显示来自其它 InTouch 节点或 Historian 服务器的历史标记值. 450
用户定义数据类型. 453
关于 UDT. 453
UDT 规格. 454
手动创建用户定义数据类型. 455
创建新数据类型 455
创建新成员. 456
创建新衍生数据类型. 457
创建新实例. 458
将成员嵌套在另一种数据类型下. 459
导入数据类型. 461
导出数据类型. 464
管理用户定义数据类型. 465
批量编辑用户数据定义类型. 466
查看用户定义数据类型. 467
“用户定义类型”视图. 468
“模型 - 标记名”视图. 468
在“属性”网格中编辑用户定义数据类型. 469
更改传播. 472
UDT 对现有功能的支持. 473
智能感知. 474
动画. 475
脚本. 475
替换引用. 476
浏览 UDT 成员. 477

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将实例的 UDT 成员拖放到工业图形编辑器画布中. 478
动画和脚本. 479
报警与报警客户端控件. 479
历史和趋势客户端. 481
将 UDT 成员配置为 I/O 标记. 482
将 UDT 实例用作父对象. 484
Web 客户端. 485
MapApp. 486
交叉引用. 487
删除未使用标记. 488
与 SuperTag 共存. 489
许可. 490
运行时标记查看器. 490
UDT 限制. 492
报警. 493
配置报警. 493
定义报警层次结构. 493
为标记配置报警条件. 497
为单独的标记设置事件属性. 501
配置报警与事件的全局设置. 502
创建报警组列表文件. 504
报警查询. 506
示例报警查询. 507
获取更多有关 InTouch 查询的信息. 508
通过冗余报警配置增强工厂安全性. 508
理解热备份. 509
配置热备份对. 511
热备份对示例. 519
有关热备份对的备注. 523
创建报警审核跟踪. 524
确认需要身份验证的报警. 525
使用分布式报警显示对象. 527
关于使用分布式报警显示对象. 527
关于分布式报警显示对象. 527
在设计时配置分布式报警显示对象. 527
在运行时使用分布式报警显示对象. 537
使用函数与点域控制分布式报警显示对象. 541
查看报警层次结构. 581
配置 Alarm Tree Viewer 控件. 581
在运行时使用 Alarm Tree Viewer 控件. 587
使用 Alarm Tree Viewer 控件 ActiveX 属性. 589
使用 Alarm Tree Viewer 控件 ActiveX 方法. 590
使用方法与属性时的错误处理. 596
使用 Alarm Tree Viewer 控件 ActiveX 事件触发脚本. 596
打印报警. 596
配置报警打印与记录. 597
使用报警打印机打印报警. 607
将报警记录到文件. 607

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用特定配置启动 Alarm Printer. 608
使用脚本控制 Alarm Printer. 608
将报警记录到报警数据库. 624
Alarm DB Logger Manager 的 SQL Server 帐户. 625
使用 Alarm DB Logger Manager. 625
报警数据库视图. 632
报警数据库存储过程. 637
查看记录的报警. 639
配置 Alarm DB View 控件. 639
在运行时使用 Alarm DB View 控件. 659
使用 Alarm DB View ActiveX 属性. 659
使用 Alarm DB View ActiveX 方法. 683
处理与使用方法和属性有关的错误. 688
使用 Alarm DB View ActiveX 事件触发脚本. 688
分析标记间的报警分布. 688
配置 Alarm Pareto ActiveX 控件. 689
在运行时使用 Alarm Pareto 控件. 699
使用 Alarm Pareto ActiveX 属性. 699
使用 Alarm Pareto ActiveX 方法. 701
使用方法与属性时的错误处理. 704
使用 Alarm Pareto ActiveX 事件触发脚本. 704
脚本. 706
基本脚本概念. 706
脚本类型. 706
高级脚本概念. 707
OLE 对象. 707
使用 ActiveX 控件编写脚本. 707
创建与编辑脚本. 707
使用 InTouch 脚本编辑器. 708
打开脚本进行编辑. 712
保存或放弃对脚本的更改. 712
复制、剪切和粘贴文本. 713
在脚本内搜索. 713
插入代码元素. 718
访问脚本函数的帮助. 718
验证脚本的语法是否正确. 719
打印脚本. 719
删除脚本. 720
调整缩放选项以查看脚本. 720
脚本触发器. 721
脚本触发器的类型. 721
使用多个触发器. 721
定期执行脚本. 722
配置应用程序脚本. 722
配置窗口脚本. 724
配置键脚本. 726
配置条件脚本. 728
配置数据改变脚本. 730

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置动作脚本. 731
配置 ActiveX 事件脚本. 735
在运行时暂停脚本执行. 737
脚本语言. 737
基本语法规则. 738
调用标准函数. 740
调用自定义函数 (QuickFunction). 740
赋值语句与运算符. 741
使用条件程序分支结构. 749
使用程序循环. 750
使用局部变量. 752
自定义脚本函数. 753
关于 QuickFunction. 753
配置 QuickFunction. 753
调用 QuickFunction. 755
创建异步 QuickFunction. 755
内置函数. 756
在动画显示链接中强制更新. 756
数学计算. 757
字符串运算. 763
转换数据类型. 772
在运行时处理 InTouch 窗口. 776
处理日期与时间信息. 787
与其它应用程序交互. 795
处理文件. 801
检索系统相关信息. 806
检索 InTouch 相关信息. 809
安全性相关脚本. 810
其它脚本. 812
使用 OLE 对象编写脚本. 816
创建、验证及释放 OLE 对象. 816
使用 OLE 对象的属性与方法. 817
将多个指针指定给相同的 OLE 对象. 819
排解 OLE 错误. 819
使用 OLE 的好处. 821
编写 ActiveX 控件脚本. 824
调用 ActiveX 控件方法. 824
从 InTouch HMI 访问 ActiveX 控件属性. 825
创建与复用 ActiveX 事件脚本. 826
导入 ActiveX 事件脚本. 829
ActiveX 控件. 830
使用 ActiveX 控件. 830
配置 ActiveX 控件. 831
为 ActiveX 控件命名. 832
ActiveX 控件上的标准操作. 832
安装与删除 ActiveX 控件. 833
向导. 834
使用向导. 834

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

向导类型. 834
将向导添加到工具栏. 835
粘贴向导实例. 836
配置向导. 836
对向导执行标准操作. 836
安装或删除向导. 836
趋势对象. 837
Windows 控件向导. 838
创建与配置 Windows 控件. 838
创建文本框. 839
创建列表框. 840
创建组合框. 840
创建复选框. 841
创建单选按钮组. 842
编写 Windows 控件脚本. 843
获取或设置控件的值. 843
启用或禁用用户输入控件. 844
动态隐藏 Windows 控件. 845
对组合框中的项目进行添加与删除. 846
从文件中加载列表项或将列表项保存到文件. 849
在组合框或列表中查找项目. 851
在组合框或列表中使用项目索引. 852
统计列表框或组合框项目数. 854
获取或设置列表项的值. 855
获取列表项的名称. 857
加载文本框的内容. 857
检查文本框是否为只读. 859
获取或设置复选框的标签 860
理解 Windows 控件错误消息. 860
小组件. 862
轮播小组件. 862
Web 浏览器小组件. 863
QR 码扫描仪. 863
Map_App 小组件. 864
配置 MapApp：概述. 865
地图设置. 866
在图形中配置 MapApp. 876
示例. 877
DataGrid 小组件. 879
配置 Data Grid 小组件. 882
为 Data Grid 小组件准备源数据. 885
Teamwork 小组件. 885
使用 XML 文件导入窗口. 887
预备 XML 文件. 887
预备命令文件. 888
创建最小命令文件. 888
将打印信息发送到文件. 888
将交叉引用发送到文件. 889

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建日志文件. 889
命令语法. 889
创建一个应用程序。. 891
将标记添加到新生成的应用程序. 891
删除窗口. 891
重命名窗口. 892
导入窗口. 892
处理错误. 893
缺少 SmartSymbol. 893
缺少工业图形. 893
表达式、标记名和脚本. 893
从命令提示符运行 WindowMaker. 894
System Platform IDE 扩展. 894
从托管的 InTouch 应用程序运行 WindowMaker. 894
从命令提示符运行 DBDump. 895
为托管的 InTouch 应用程序运行 DBDump. 896
从命令提示符运行 DBLoad. 896
为托管的 InTouch 应用程序运行 DBLoad. 896
XML 格式. 897
常规 XML 文件格式. 897
通用元素定义. 898
窗口元素. 904
不支持的 InTouch 功能. 965
在 CONNECT 中使用工业图形. 967
登录到 CONNECT. 967
在共享驱动器之间导航. 968
将图形上传/下载到云. 968
将图形上传到 CONNECT. 968
将图形下载到本地可视化文件夹. 969
对各种云内容的上传/下载支持. 969
管理 CONNECT 中的图形. 970
多个用户管理图形. 970
关于云图形版本. 970
覆盖云中的图形内容. 970
云图形版本控制 - 情形和示例. 971
键盘快捷方式. 972

Deploy. 978
部署和使用终端服务与远程桌面服务. 979
规划终端服务器应用程序. 979
在终端服务环境中部署 InTouch 应用程序. 979
终端服务环境中的报警. 980
终端服务环境中的安全性. 980
终端服务环境中的 I/O. 980
终端服务环境中的脚本执行. 981
正确登录到终端会话以运行 InTouch. 981
终端服务环境中的报警查询语法. 981

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

终端服务环境中的其它限制. 981
使用脚本检索关于 InTouch 客户端会话的信息. 982
TseGetClientId() 函数. 982
TseGetClientNodeName() 函数. 982
TseQueryRunningOnConsole() 函数. 983
TseQueryRunningOnClient() 函数. 983
远程桌面服务概述. 983
“远程桌面服务”角色服务. 983
确保远程桌面服务 (RDS) 连接的安全. 984
Windows Server 2016 远程桌面服务最佳实践. 985
分发应用程序. 986
支持的 InTouch 架构. 986
单机架构. 986
基于客户端的架构. 986
基于服务器的架构. 987
网络应用程序开发 (NAD). 987
规划网络化应用程序. 988
网络化应用程序的 I/O 数据访问. 988
访问共享的文件. 989
在分布式环境中记录数据. 991
特殊网络的注意事项. 995
针对 NAD 配置 InTouch 应用程序. 995
执行 NAD 自动更新. 997
执行 NAD 手工更新. 997
应用程序编辑锁定. 999
在 NAD 更新期间对应用程序更改. 999
在运行时调整应用程序分辨率. 1000
锁定应用程序分辨率. 1002
将应用程序发布到远程节点. 1003
发布的文件的内容. 1004
发布独立的 InTouch 应用程序. 1005
发布托管的 InTouch 应用程序. 1008
发布托管的 InTouch 应用程序. 1008
将应用程序发布到 Insight. 1009
运行时使用托管的 InTouch 应用程序. 1011
部署托管的 InTouch 应用程序. 1011
第一次部署 InTouchViewApp 对象. 1012
部署对托管的 InTouch 应用程序所作的更改. 1012
启动托管的 InTouch 应用程序. 1012
在部署托管应用程序时控制 WindowViewer 重新启动等待时段. 1013
在操作员节点上接受新的应用程序版本. 1013
运行嵌入的工业图形中的 ArchestrA 脚本. 1015
在终端服务环境中部署 InTouchViewApp 对象. 1016
在 AVEVA Edge 中使用 InTouch HMI 应用程序. 1018
下载 AVEVA Edge Zip 文件. 1018
上传 AVEVA Edge Zip 文件. 1018
为边缘设备配置用户. 1019

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Operate 1021
在运行时查看应用程序 1022
关于在运行时查看应用程序 1022
在运行时以不同的目标分辨率大小查看应用程序 1022
关于 InTouch Web 客户端 1023
原始应用程序分辨率 1023
在运行时使用键盘、鼠标和触摸手势进行平移和缩放 1024
在运行时缩放 1024
在运行时平移 1025
对触摸手势的动画支持 1026
对框架窗口使用 ShowGraphic() 函数 1027
在 Internet 上运行 InTouch 窗口 1027
将窗口设置为在运行时显示 1028
在 Web 客户端中查看应用程序图形 1030
使用 Web 客户端 1030
设计 InTouch Web 客户端应用程序 1031
保护 InTouch Web 客户端访问安全 1032
在 InTouch Web 客户端中配置用户访问权限 1036
使用 InTouch Web 客户端虚拟帐户 1036
启用 InTouch Web 客户端功能 1037
自定义 InTouch Web 客户端 1037
配置图形，以在 InTouch Web 客户端中查看 1038
设置 InTouch Web 客户端主页符号 1039
了解 InTouch Web 客户端主页图标的行为 1039
在 Web 浏览器中查看应用程序 1039
InTouch Web 客户端快速切换 1040
了解 InTouch Web 客户端页面 1041
调整应用程序图形和浏览器的大小 1043
平移与缩放支持 1043
将 InTouch 应用程序图形托管在外部网站上 1045
生成 iFrame 代码块 1045
支持的图形元素和已知限制 1046
已知限制 1046
所有受支持的图形元素支持的属性 1047
所有受支持的图形元素均支持的属性以及一些限制 1048
支持的图形元素及其它属性 1049
支持的动画 1053
Web 客户端移动应用程序 1055
操作系统要求 1055
登录到移动应用程序 1055
使用 InTouch Web 客户端移动应用程序 1056
在运行时切换语言 1057
关于运行时语言切换 1057
为运行时语言切换配置语言 1057
为配置的语言更改字体设置 1058
添加运行时语言切换功能 1059
SwitchDisplayLanguage() 函数 1061
$Language 系统标记 1061

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

导出应用程序文本进行脱机翻译. 1062
将文本导出到现有的字典文件. 1063
翻译导出的字典文件. 1063
导入翻译的字典文件. 1065
运行时语言切换进行报警. 1065
导出报警注释进行翻译. 1066
导出到现有的报警注释文件. 1067
导入翻译的报警注释. 1069
导出报警组名称进行翻译. 1070
导入翻译的报警组名称. 1071
在运行时测试语言切换功能. 1071
将本地化版的文件分发到网络应用程序开发客户端. 1072
标记. 1073
标记查看器入门. 1073
启用 Tag Viewer. 1073
启动标记查看器. 1074
在标记查看器中导航. 1076
关闭标记查看器. 1076
使用标记查看器. 1077
搜索标记. 1077
执行快速搜索. 1078
管理标记. 1078
管理观察窗口. 1080
减少标记使用. 1083
InTouch 与 Historian. 1083
确定标记使用. 1084
删除未使用标记. 1091
报警. 1094
查看当前报警. 1094
配置 Alarm Viewer 控件. 1095
在运行时使用 Alarm Viewer 控件. 1106
使用 Alarm Viewer 控件 ActiveX 属性. 1109
使用 Alarm Viewer 控件 ActiveX 方法. 1115
使用方法与属性时处理错误. 1129
使用 ActiveX 事件触发脚本. 1129
实时确认报警. 1130
理解报警确认模型. 1130
使用点域确认报警. 1132
使用脚本函数确认报警. 1137
标记值恢复到正常时使用自动确认. 1138
使用报警客户端确认报警. 1138
使用报警与确认注释. 1139
在运行时解除报警. 1139
使用点域解除报警. 1140
在运行时控制标记与组的报警属性. 1141
确定标记或报警组是否处在报警条件中. 1145
将报警状态处理转换为 InTouch 7.1 行为. 1154
确定是否为标记设置了报警限. 1155

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

启用与禁用标记或报警组的报警. 1159
更改标记的报警限. 1172
更改标记的报警死区. 1177
更改与标记关联的报警注释. 1179
将用户自定义信息关联到报警实例. 1179
确定标记或报警组的约束标记. 1181
统计活动的或未确认的报警数. 1187

Maintain. 1194

迁移和升级应用程序. 1195
从传统应用程序移至新的独立应用程序. 1195
迁移和升级旧版应用程序. 1195
将较早版本的 InTouch 应用程序迁移到当前版本. 1195
转换旧的报警显示. 1196
管理应用程序设置. 1196
导入 InTouch 应用程序. 1197

使用 System Platform IDE 管理 InTouch 应用程序. 1198
InTouchViewApp 对象. 1199
关联 InTouchViewApp 模板与 InTouch 应用程序. 1199
编辑托管的 InTouch 应用程序. 1199
测试托管的 InTouch 应用程序. 1200
部署 InTouchViewApp 对象. 1200
导出与导入 InTouchViewApp 对象. 1200
InTouchViewApp 对象的属性. 1201
InTouchViewApp 对象与其它 AutomationObject 的区别. 1201
ViewEngine 对象. 1202

导出与导入 InTouch 组件. 1203
导出与导入同托管的 InTouch 应用程序关联的标记数据. 1203
导出标记定义. 1203
查看导出的标记定义. 1204
导入标记定义. 1204
标记名字典导入文件格式. 1205
创建导入文件模板. 1206
设置字典导入文件的操作模式. 1206
设置访问名与报警组. 1208
定义标记类型关键字与属性. 1212
在导入文件中使用空字符串. 1233
使用字段的缺省值. 1233
创建 SuperTag 实例. 1234
使用 DBLoad 导入标记定义. 1234
在 InTouch 应用程序之间导出与导入 InTouch 窗口. 1236
导入窗口. 1236
转换导入窗口的占位符标记. 1237
导出窗口. 1238
导入脚本. 1239
转换导入的脚本中的占位符标记. 1240
导入的窗口与脚本中的标记占位符. 1241

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从应用程序中导出工业图形. 1243
将工业图形导入到应用程序. 1244
从工业图形工具箱导出所选符号. 1245
导入与嵌入自定义客户端控件. 1245
解决当导入重复客户端控件时产生的冲突. 1246
在工业图形中嵌入客户端控件. 1247
导入 HTML5 小组件. 1247
将脚本函数库导入 InTouch 应用程序. 1248
解决 .NET 脚本库中冲突方法的导入问题. 1248
配置应用程序的应用程序样式库. 1249
导出和导入应用程序样式库. 1250
配置应用程序的报警优先级映射. 1251
从应用程序中导出工业图形文本字符串. 1252
将工业图形的文本字符串导入到应用程序. 1253
从符号导出本地化字符串. 1254
导入工业图形库. 1255
设置多监视器系统. 1256
多监视器配置. 1256
单显卡配置. 1256
多显卡配置. 1257
规划多监视器应用程序. 1258
选择多监视器显卡. 1258
确定应用程序屏幕分辨率. 1258
确定用于显示应用程序的监视器数. 1259
确定应用程序窗口的位置. 1260
开发多监视器 InTouch 应用程序. 1260
配置多监视器参数. 1260
配置屏幕分辨率转换. 1261
部署应用程序与验证多监视器设置. 1261
在运行时验证多监视器支持. 1262
在 Tablet PC 上使用 InTouch. 1263
注解可视化屏幕与将它当作电子邮件消息发送. 1263
给窗口添加注解. 1263
选择、复制及删除窗口注解. 1264
保存、打印及使用电子邮件发送注解的窗口. 1264
AnnotateLayout() 函数. 1265
更改屏幕方向. 1265
管理 InTouch 服务. 1266
将 WindowViewer 作为服务运行. 1266
将 WindowViewer 配置成作为服务启动. 1267
编辑 WIN.INI 以在 WindowViewer 中将应用程序作为服务运行. 1268
手动启动服务. 1268
停止服务. 1268
配置 InTouch 服务的用户帐户. 1269
InTouch 服务疑难排解. 1269
查看服务的错误消息. 1269
服务用户帐户问题的疑难排解. 1270
停用提示的 I/O 项目. 1270

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 服务的注册表项 . . . 1270
报警 . . . 1271
从旧有的报警系统迁移 . . . 1271
从标准报警系统迁移到分布式报警系统 . . . 1271
维护报警数据库 . . . 1272
关于维护报警数据库 . . . 1272
配置清除或归档设置 . . . 1272
恢复报警数据库 . . . 1282
从 INTOUCH.ini 文件中自定义应用程序设置 . . . 1287
自定义的 INTOUCH.ini 参数 . . . 1287
设置自定义的记录属性 . . . 1288
禁用 WindowMaker 快捷菜单 . . . 1289
设置自定义的 WindowViewer 属性 . . . 1289

Diagnose . . . 1292

QuickScript 疑难排解 . . . 1293
将消息记录到 Log Viewer . . . 1293
LogMessage() 函数 . . . 1293
查看 Log Viewer 消息 . . . 1294
理解错误消息 . . . 1295
主窗口 . . . 1295
添加标记引用 . . . 1295
修改标记值 . . . 1296
重命名观察窗口 . . . 1296
管理 InTouch Web 客户端 . . . 1298
InTouch Web 客户端错误处理 . . . 1298
浏览器和移动支持 . . . 1298
在 Web 浏览器中查看图形的疑难排解 . . . 1299
无效证书错误 . . . 1299

Manage . . . 1301

AVEVA Operations Control 连接体验 . . . 1302
关于 Operations Control 连接体验 . . . 1302
配置 Operations Control 连接体验 . . . 1302
身份验证和权利 . . . 1308
使用 AVEVA Identity Manager 进行身份验证 . . . 1309
连接丢失期间的行为 . . . 1311
许可和权利 . . . 1312
查看订阅信息 . . . 1313
启动并运行 WindowMaker . . . 1314
启动并运行 WindowViewer . . . 1315
在 Operations Control 连接体验中配置应用程序安全性 . . . 1317
在托管应用程序中配置安全性 . . . 1318
在独立应用程序中配置安全性 . . . 1320
在混合模式下操作 . . . 1321
启动并运行应用程序管理器 . . . 1322

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从应用程序管理器中选择并运行应用程序. 1323
启动并运行 InTouch Access Anywhere. 1324
启动网络应用程序开发 (NAD) 应用程序. 1324
执行安全和验证写入. 1325
启动并运行 Web 客户端（内部部署）. 1326
启动并运行趋势控件. 1328
从 AVEVA 应用程序注销. 1331
支持的 InTouch HMI 功能. 1331
不支持的 InTouch HMI 功能. 1333
在非 Operation Control 模式下使用 InTouch. 1334
InTouch HMI 中的许可证. 1334
理解许可证标记计数. 1334
可用于 InTouch HMI 的许可证. 1336
RDS 和非 RDS 环境中的 InTouch 许可证. 1336
关于 InTouchView 应用程序许可证. 1337
WindowMaker 许可. 1338
WindowViewer 许可. 1338
在服务器上管理 InTouch 许可证的最佳做法. 1338
查看许可证信息. 1340
管理启动后不同许可证的使用. 1342
以自由模式和演示模式运行. 1342
利用宽限期. 1343
远程标记计数函数. 1345
授权 InTouch Web 客户端. 1350
获得 InTouch 应用程序图形的许可证. 1350
许可证功能. 1351
单一会话模式. 1352
宽限期. 1353
定期续订. 1353
许可证释放. 1353
关于辅助组件. 1353
使用 Recipe Manager. 1354
从 InTouch 中使用 SQL 数据库. 1373
使用 16 笔趋势向导. 1417
Symbol Factory. 1431
保护 InTouch 安全. 1442
InTouch 安全性功能. 1442
基于身份验证与授权的安全性. 1448
管理用户并设置其授权级别. 1456
登录与注销. 1464
基于操作员或访问级别启用与禁用功能. 1470
检索当前登录的操作员的有关信息. 1472
安全系统标记与函数的摘要. 1475
允许非管理员用户使用的应用程序管理器操作. 1476
将基于角色的安全性应用于应用程序文件夹. 1478
托管和 NAD InTouch 应用程序的本地工作目录完整性检查. 1482
管理 InTouch HMI 的安全性. 1484
InTouch HMI 的安全配置. 1484

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Standard

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 1 集成开发环境

WindowMaker 是用于创建 InTouch HMI 应用程序的开发环境。您可以使用 WindowMaker 来创建由操作员用于查看与管理生产过程的可视化界面。InTouch HMI 界面显示来自生产环境的数据，并可以将数据写回生产环境。

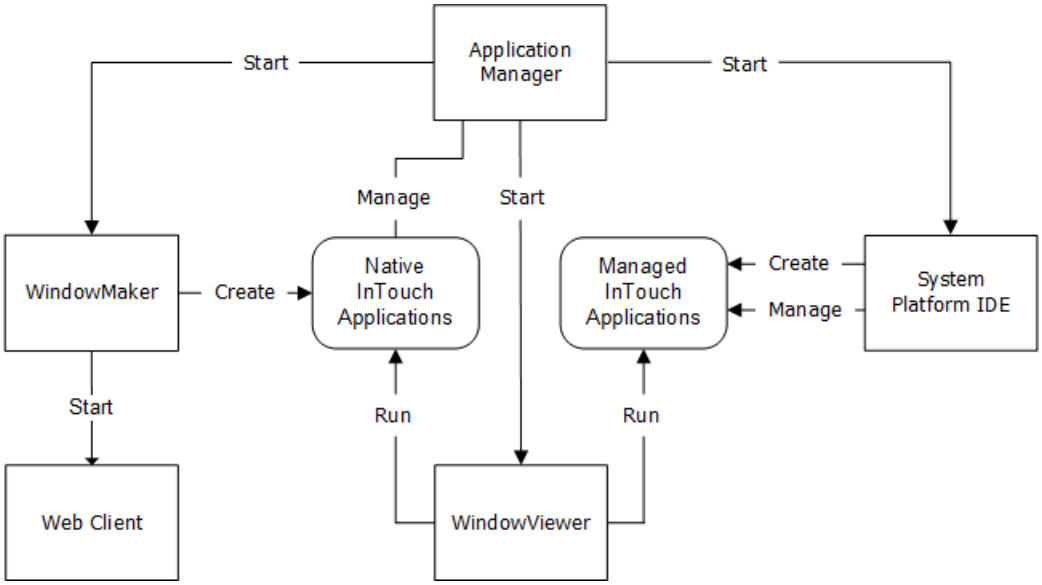
您可以使用 WindowMaker 来配置 InTouch 应用程序的以下可视化界面元素：

- 窗口包含工厂操作员管理生产过程所使用的交互式可视化元素。
- 基本对象是简单的图形元素（如长方形、圆、线条）及文本。
- 用户自定义的复杂对象由一个或多个基本对象组成，代表生产环境中的元素，如阀门和贮料罐。
- 预定义的复杂对象执行特定的功能，如报警、历史趋势或符号向导。
- 动画链接是简单与复杂对象的属性，可使其外观呈现动画效果，将用户输入传递到任务及促使生产数据的变化。
- 向导是执行特定功能或具有特定外观的预定义复杂对象，如游标和仪表。
- ActiveX 控件是可以放入 InTouch 窗口中以执行特定功能（如显示当前报警）的控件。

如需了解更多信息，请观看标题为“The WindowMaker Interface（WindowMaker 界面）”的视频。

关于 InTouch 应用程序管理器

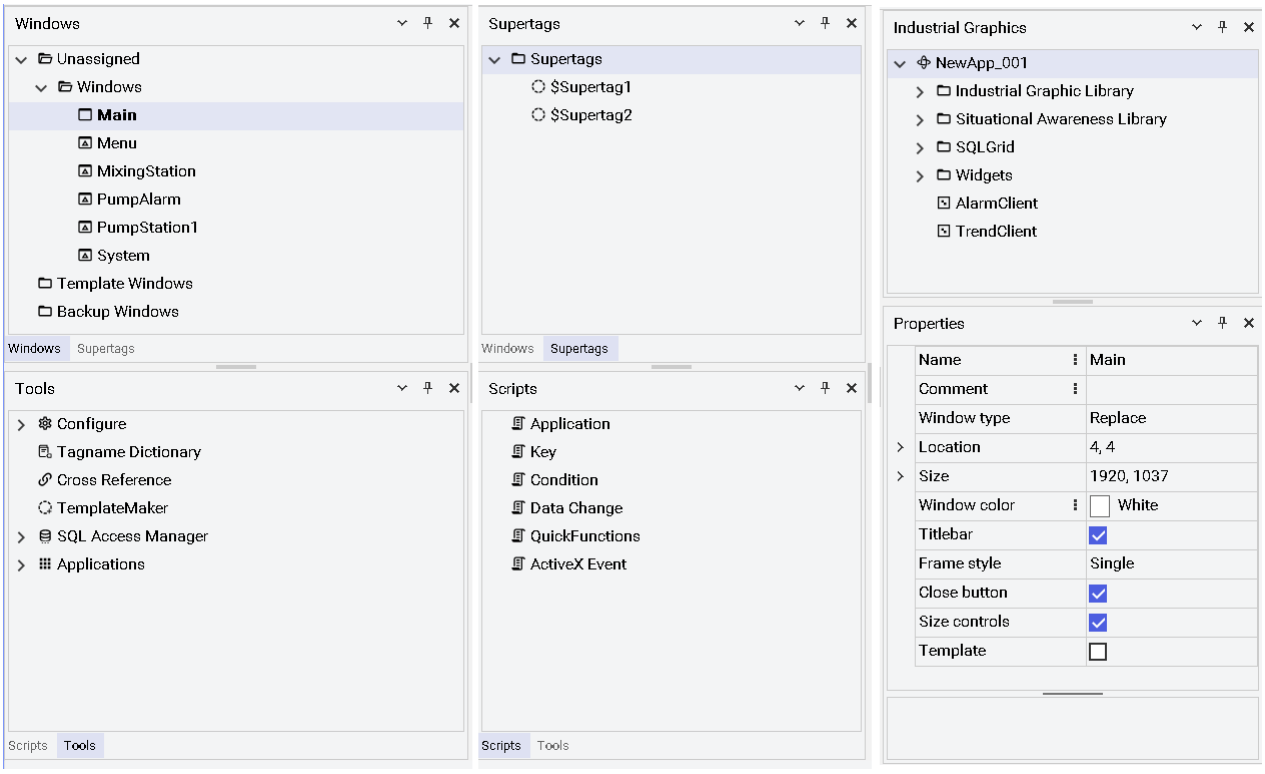
您可以使用“InTouch 应用程序管理器”管理大多数全局性任务，如创建、删除以及修改 InTouch 应用程序。“应用程序管理器”显示当前的 InTouch 应用程序列表。您可以从该列表中选择要在 WindowMaker 或 WindowViewer 中打开的应用程序。



使用应用程序浏览器

“应用程序浏览器”包含“窗口/SuperTag”、“脚本/工具”和“工业图形工具箱”选项卡部分。和其它工具栏一样，这些选项卡也可以打开或关闭、固定或停靠。缺省条件下，窗格显示在下图所示的位置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



在这些视图中，可以访问所有应用程序窗口、脚本、配置菜单、“标记名字典”和向导。

固定窗格

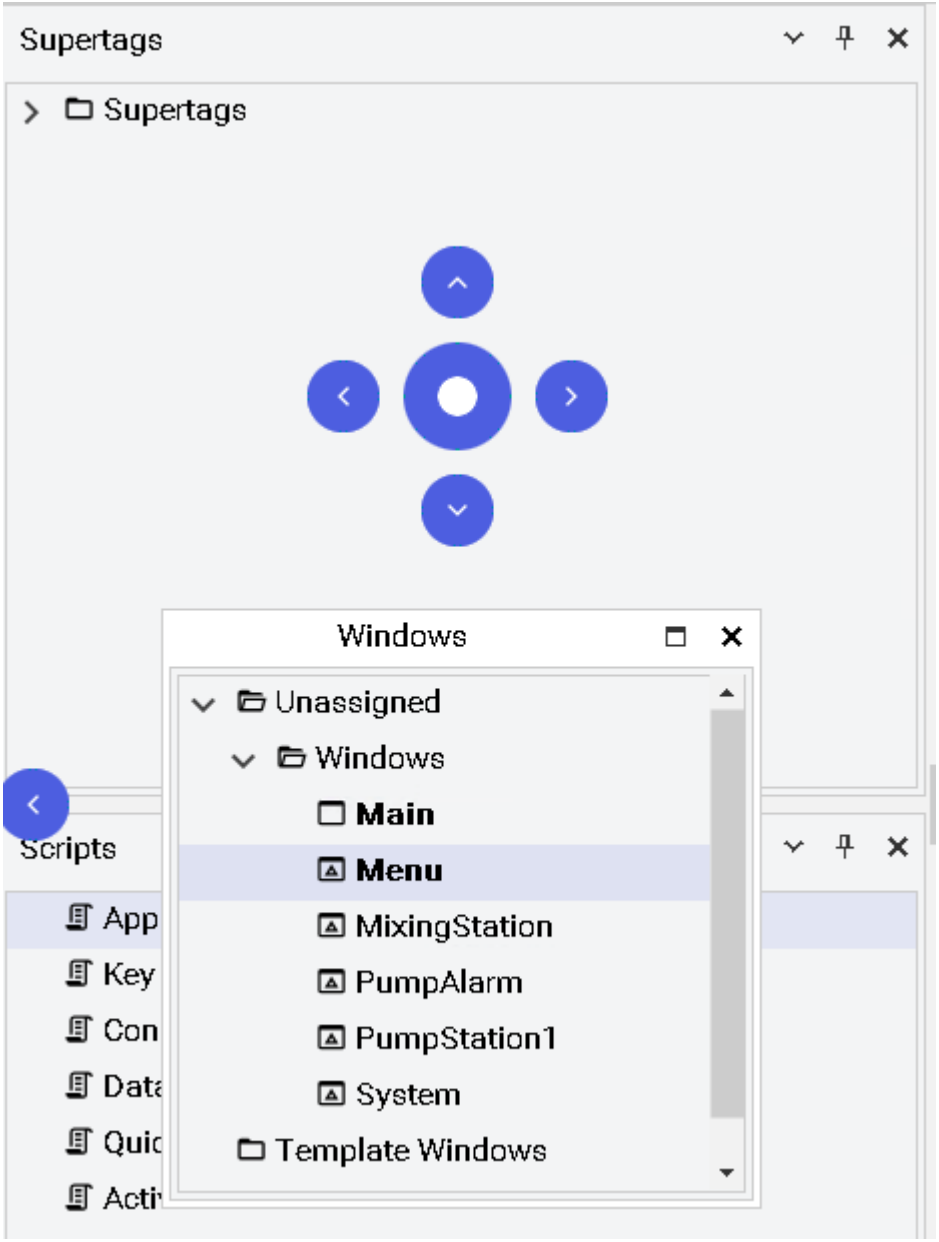
这些窗格中的每一个都可以四处移动，并固定在屏幕的左侧、右侧、顶部和底部位置。要固定窗格，请选择窗格的标题。此时光标变成双向箭头。将窗格拖放到屏幕上的任意位置。

您也可以使用导航辅助功能来固定浮动窗格。在屏幕上的特定位置：

- 选择向上箭头可将窗格固定到屏幕顶部。
- 选择向下箭头可将窗格固定到屏幕底部。
- 选择向右箭头可将窗格固定到屏幕右侧。
- 选择向左箭头可将窗格固定到屏幕左侧。

窗格可以以垂直样式固定，也可以以水平样式固定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



当窗格的位置发生更改时，即使在关闭并重新打开 WindowMaker 之后，也会保留自定义。

对窗格位置和固定的更改存储在 DockSettings.xml 中。如果在更改固定设置后遇到任何导航问题，请执行以下操作：

1. 关闭 WindowMaker。
2. 导航到 **C:\Users\<用户名>\AppData\Local\Wonderware**。
3. 找到 DockSettings.xml 文件并将其删除。
4. 重新打开 WindowMaker。

在应用程序浏览器中浏览

您可以在两个“应用程序浏览器”工具栏的任何一个中展开或折叠文件夹。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

应用程序视图显示其它已安装的应用程序。

展开或折叠“应用程序浏览器”中的文件夹

1. 双击文件夹或图标以展开并显示组成员。
2. 双击成员可打开该成员。

将应用程序添加到应用程序浏览器

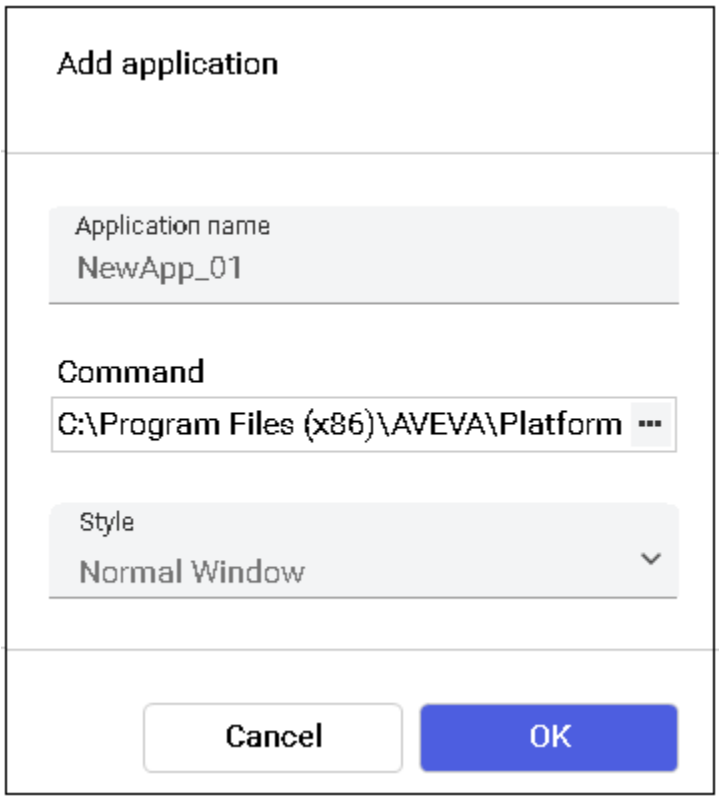
WindowMaker 的“应用程序浏览器”可以从 WindowMaker 中启动其它应用程序。例如，您可以一边开发应用程序，一边运行“I/O 服务器”并配置它。您也可以启动第三方程序，如 Windows 的“记事本”、“写字板”、Microsoft Excel、Microsoft Word、Microsoft 的“画图”等等。

还可以配置“应用程序浏览器”以打开特定的文件，如文档或电子表格。

将新的应用程序添加到应用程序浏览器

1. 在工具窗格中，使用鼠标右键单击应用程序，然后选择新建。

   此时出现添加应用程序窗口。



2. 在应用程序名称框中，输入应用程序的名称。
3. 在命令框中，输入应用程序的完整路径。选择省略号按钮可浏览到应用程序。

   在命令行框中，可以给应用程序添加命令行参数。
4. 从样式列表中选择应用程序启动时的显示方式。可用的选项包括“普通窗口”、“最小化”和“最大化”。
5. 选择确定。

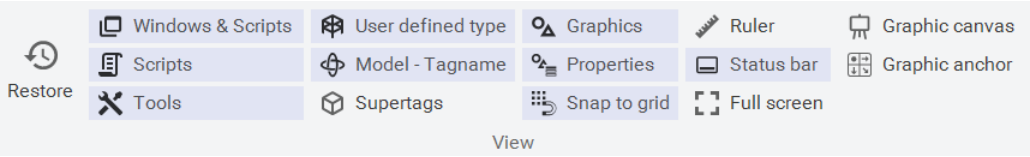
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

该应用程序会添加到应用程序浏览器中的“应用程序”下。现在您随时可以从 WindowMaker 中运行该应用程序。

注意：请勿将 WindowViewer (view.exe) 添加到**应用程序浏览器**。启动 WindowViewer 的正确方法是：**选择文件**菜单上的 **WindowViewer**，或是**选择**工具栏中的**运行时进**行快速切换。

管理工具栏

WindowMaker 中的工具栏允许快速访问常用命令。启动 WindowMaker 时，缺省条件下会显示所有工具栏。使用“查看”菜单来管理工具栏以及显示或隐藏任何工具栏。



您可以通过拖动工具栏来浮动或固定工具栏。显示原本隐藏的固定工具栏时，它会重新出现在上次它在窗口中所固定的位置上。任何工具栏都可以从它的缺省固定位置移到**开发**窗口中的任何其它位置。浮动工具栏有标题栏，而且大小也可以更改。

显示或隐藏工具栏

- 在**查看**菜单上，**选择**工具栏图标。

更改浮动工具栏的大小

1. 将光标移到工具栏的任何一个边缘。此时光标变成一个双向箭头。
2. 拖动工具栏的边缘可以移动或调整工具栏的大小。

   在移动光标的过程中会出现一个框，指出释放鼠标按钮时工具栏的大小。

设置 WindowMaker 首选项

通过使用 WindowMaker 配置屏幕，可以配置会影响 WindowMaker 行为的首选项与选项。您可以：

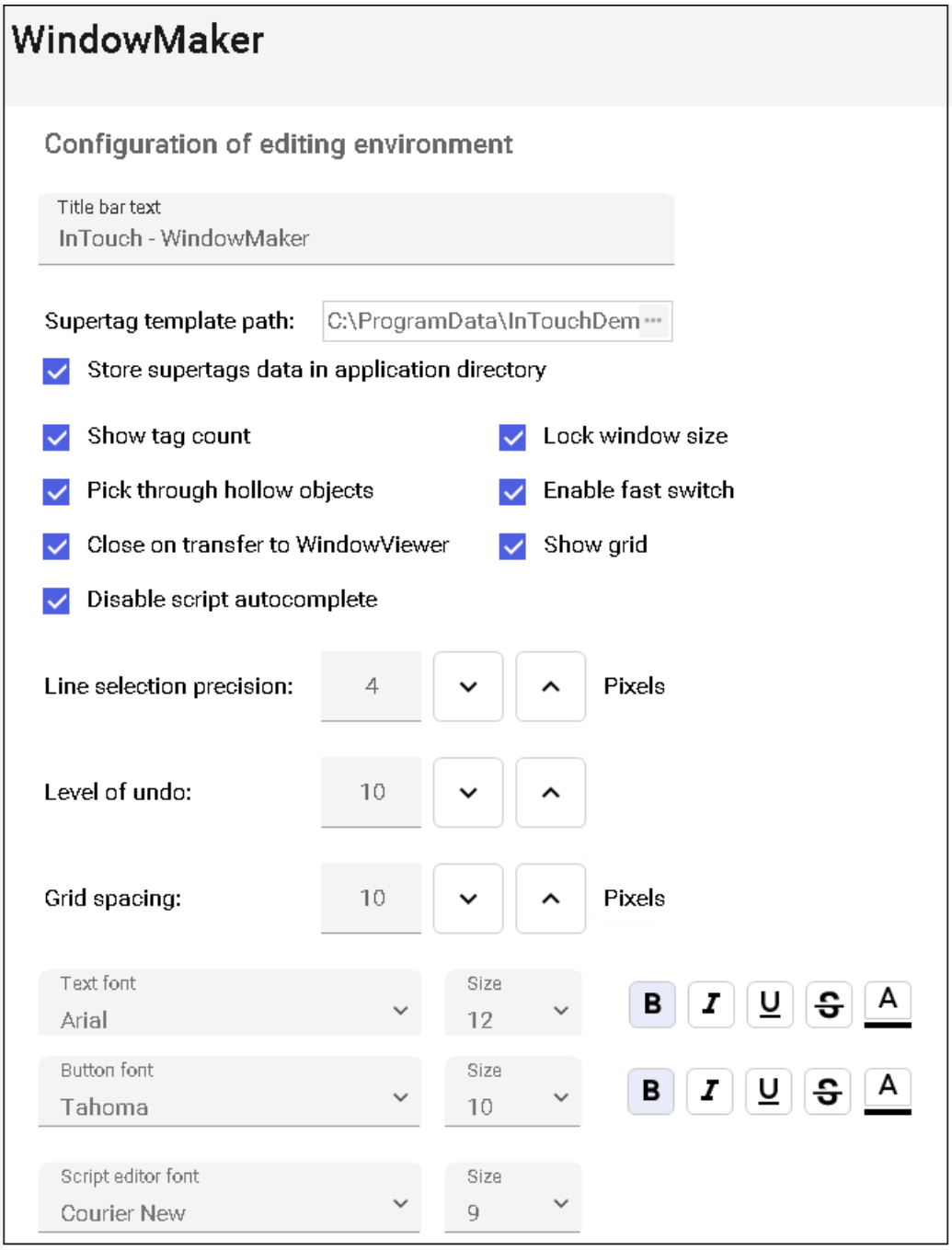
- 更改标题栏文本。
- 显示网格或关闭网格。
- 更改网格上像素之间的间距。
- 显示标记计数。
- 更改文本与按钮的缺省字体。
- 设置行选择的精度。
- 设置选项以便在切换到 WindowViewer 时关闭 WindowMaker。
- 设置选项以透过中空对象进行选取。
- 启用从 WindowMaker 快速切换到 WindowViewer 的功能。
- 设置撤消级数。

设置 WindowMaker 的属性

1. 在**文件**菜单上，**选择配置**，然后**选择 WindowMaker**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现 WindowMaker 配置屏幕。



2. 在 WindowMaker 编辑环境配置区域中，配置标题栏的外观。执行以下任意操作：
   - 在**标题栏**文本框中，输入要在设计期间显示在标题栏中的文本。
   - 在 **SuperTag 模板路径**中，指定存储 SuperTag 模板的路径。
3. 选中**在应用程序目录中存储 SuperTag 数据**复选框，以将 SuperTag 数据存储在应用程序目录中。这将覆盖在 **SuperTag 模板路径**字段中指定的路径。
4. 配置窗口的其它属性执行以下任意操作：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 选中显示标记计数复选框，以在菜单栏中显示“标记名字典”中的标记名数量。如果有许多标记，则显示标记计数时，可能会影响“标记名字典”的性能。

  如果使用大小受限的“标记名字典”创建应用程序，此项功能会非常有用。标记名计数不包括远程标记名引用或系统标记。选择主页选项卡上的更新使用计数，可以了解远程标记名引用的使用情况。
- 选中锁定窗口大小复选框，以锁定 InTouch 应用程序窗口的大小。

  这样可让您在不调整窗口和图形大小的情况下，将应用程序转换为不同的分辨率。
- 选中透过中空对象选取复选框，以选择中空对象后面的对象。

  这样可达到类似这样的目的：不必将框架置后，便能选择框架内的对象。
- 选中启用快速切换复选框，以使用“快速切换”在 WindowMaker 与 WindowViewer 之间切换。

  快速切换是出现在 WindowMaker 右上角的运行时字样。在 WindowViewer 中，它是开发字样。

  如果启用快速切换，在切换到 WindowViewer 之前，WindowMaker 会自动保存对所有打开的窗口所作的全部更改。
- 选中转换到 **WindowViewer** 时关闭复选框，以在启动 WindowViewer 时自动关闭 WindowMaker。

  此选项旨在节约有限的内存。如果内存足够，并且经常在 WindowViewer 与 WindowMaker 之间切换，请不要选择此选项。

  选择转换到 **WindowViewer** 时关闭时，同时还会选择 **WindowViewer** 属性屏幕中常规属性选项卡上相应的关闭 **WindowViewer** 命令。
- 选中显示网格复选框，以显示网格。
- 选择禁用脚本自动完成，以禁用自动完成建议列表框的显示。
- 在行选择精度框中，输入光标可以远离某行多少个像素但仍能选中它。

  在大多数情况下，缺省设置 4 的效果就很好。
- 在撤消级别框中，输入要保留的撤消与恢复级数。

  最多可以有 25 级。如果输入零，则会关闭撤消/恢复功能。

  一个级别代表一个操作。创建新的窗口或是打开现有的窗口时，撤消与恢复堆栈都是空的。关闭窗口时，两个堆栈都会清空。
- 在网格间距框中，输入网格坐标之间的像素数。

5. 您也可以分别使用文本字体、按钮字体和脚本编辑器字体字段来配置文本、按钮和脚本编辑器的字体属性。
6. 通过使用各自屏幕上的字体设置，可以在任何窗口中覆盖这些缺省值。
7. 选择确定。
8. 重新启动 WindowMaker 以应用所作的任何更改。

鼠标快捷方式

以下快捷方式可用于打开对话框与执行其它常见任务。

访问 WindowMaker 中项目的菜单命令

- 使用鼠标右键单击该项目。这些项目包括图形对象、视图中的文件夹名，等等。

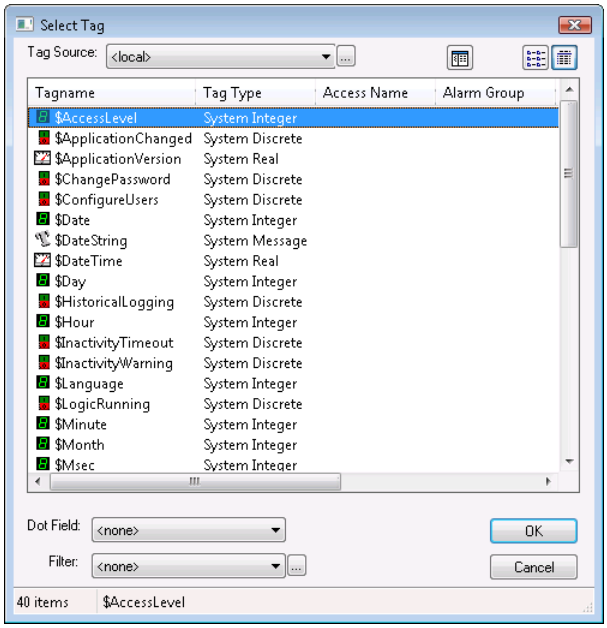
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

打开“动画链接”对话框

- 双击对象或符号。

打开“选择标记”对话框（标记浏览器）

- 双击链接定义对话框中空白的表达式输入区域。此时出现选择标记对话框。



访问标记点域

- 在任何“标记名”或“表达式”输入框中，输入标记名加上一个英文句点，然后双击英文句点的右侧。您也可以仅输入一个英文句点，然后双击它的右侧。此时出现选择域名对话框，其中显示所有的标记名点域。

在“标记名字典”中打开标记名定义

- 双击标记名。

使用全屏模式

全屏模式隐藏除了打开的窗口和浮动工具栏之外的所有程序元素。

在打开或关闭全屏之间切换

- 在查看工具栏上，选择全屏幕，以从正常模式切换到全屏幕模式。

  此时查看工具栏自动更改为恢复工具栏，并浮动在顶部。

从文件菜单管理窗口

使用文件菜单打开、保存、关闭或删除窗口时，对所选命令有效的所有窗口的名称都会显示在列表中。

查看窗口列表

您可以在图标视图、列表视图或详细信息视图中查看窗口。

- 图标视图：选择图标可查看以网格排列的窗口列表。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 列表视图：选择列表可查看以多列列表形式排列的窗口。
- 详细信息视图：选择详细信息可查看窗口列表以及窗口描述。描述是从窗口属性面板的“注释”字段进行填充的。

选择窗口

- 要选择某个窗口，请选择窗口图标或条目。该窗口即会突出显示。
- 要选择多个窗口，请选择这些窗口图标或条目。这些窗口即会突出显示。
- 要选择所有窗口，请选中全选复选框。

取消选择窗口

- 要取消选择所选窗口，请再次选择该窗口图标或条目。
- 要清除所有窗口的选择，请选择清除全选复选框。

搜索窗口

- 要从列表中搜索窗口，请使用搜索栏。

设置字体缺省值

您可以为文本对象与带有文本的按钮对象设置字体缺省值。通过使用字体部分来自定义窗口或按钮文本，可以覆盖这些缺省值。

设置字体缺省值

1. 在文件菜单上，选择配置，然后选择 **WindowMaker**。

   此时出现 WindowMaker 配置屏幕。
2. 您可以分别使用文本字体、按钮字体和脚本编辑器字体字段来配置文本、按钮和脚本编辑器的字体属性。

   通过使用字体部分，可以覆盖这些缺省设置。
3. 选择确定。

使用方向键移动对象

在 WindowMaker 中，可以使用方向键来移动一个或一组所选的对象。

使用方向键移动对象时，对象的移动距离取决于是否显示网格。

显示网格时，对象移动的像素数取决于网格间距，此间距在 WindowMaker 配置屏幕上设置。缺省设置为网格点之间相距十个像素。

显示网格时：

- 按方向键一次，对象移动一个网格点。
- 按 SHIFT + 方向键，对象移动两个网格点。
- 按 CTRL + 方向键，对象移动四个网格点。

不显示网格时：

- 按方向键一次，对象移动一个像素。

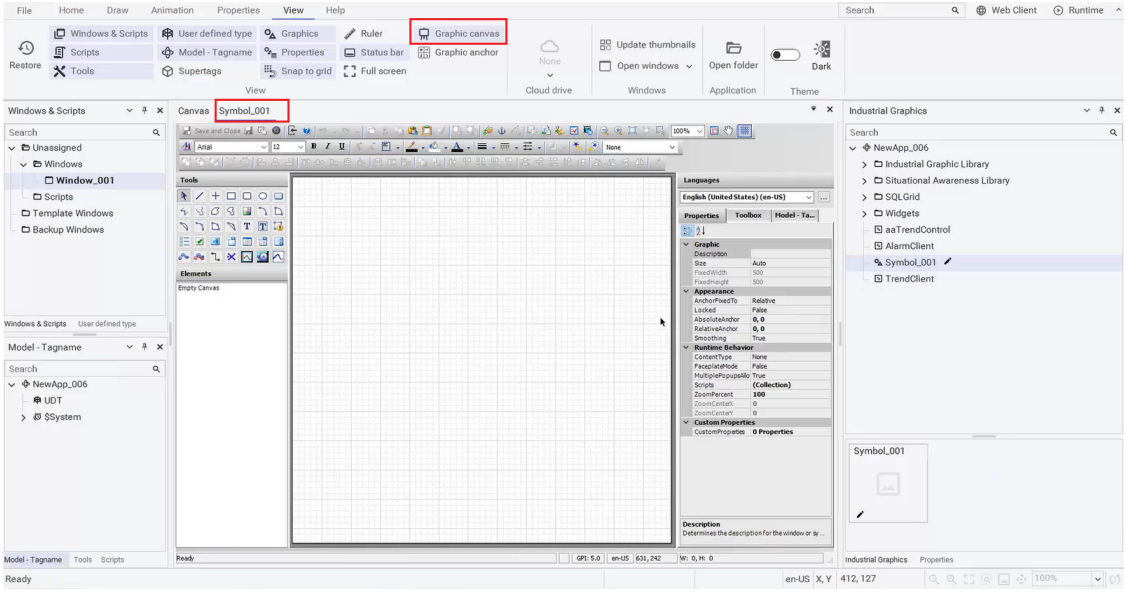
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 按 SHIFT + 方向键，对象移动十个像素。
- 按 CTRL + 方向键，对象移动 50 个像素。

使用图形画布和图形定位点

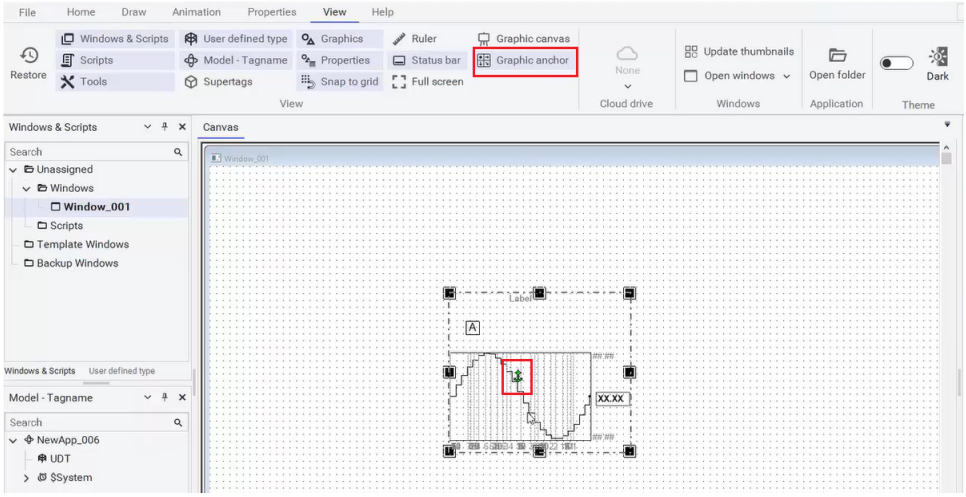
图形画布

在 WindowMaker 的查看菜单下，使用图形画布选项查看作为 WindowMaker 中一个选项卡的工业图形编辑器窗口。即，当选择“图形画布”选项时，如果打开图形，将在 WindowMaker 的一个选项卡中打开工业图形编辑器。



图形定位点

在 WindowMaker 的查看菜单下，使用图形定位点选项在图形上查看定位点。有关定位点的更多信息，请参阅《AVEVA 工业图形编辑器帮助》中的“大小传播和定位点”一节。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用调色板

通过调色板，可以将颜色应用于线条、长方形、圆角长方形、椭圆、折线、多边形以及文本的静态与动态属性。此外，您也可以为窗口选择背景颜色，以及为位图选择透明色（以便能够查看位图后面的对象）。

调色板提供广泛的颜色供您选择，可选的颜色多达 1670 万种。可用的颜色数量可能会受视频卡功能的限制。

您也可以定义和添加自定义颜色。

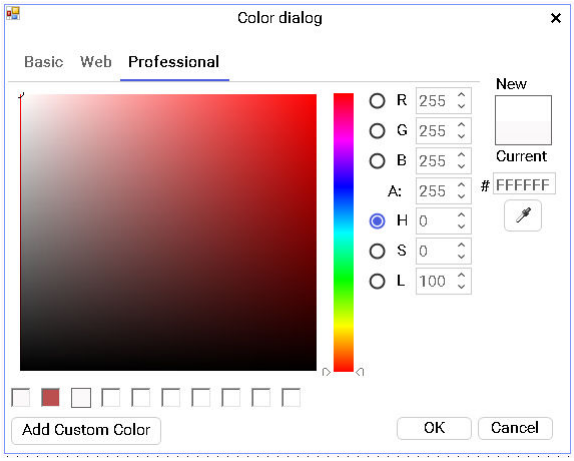
打开调色板

选择“属性”面板中的颜色方块，或选择其中一个颜色工具以便将线条、填充或文本颜色应用于所选的对象时，便会出现调色板。

打开调色板

1. 在配置屏幕中，选择“颜色”字段旁边的颜色方块。

   此时出现颜色对话框。



1. 选择基本和 **Web** 选项卡，以访问其它经典调色板。
2. 执行以下任意操作：
   - 选择颜色显示区域中的任意位置并使用游标调整颜色。要指定不含白色或黑色成分的 100% 的颜色，请按 ALT + O。
   - 在红、绿和蓝框中输入值以定义颜色。您可以通过观察颜色表来对这些值进行试验。注意色调、饱和度以及亮度的值也会随之改变。
   - 在色调、饱和度以及亮度框中输入值可定义颜色。更改这些值中的任何值时，红、绿、蓝的值也会相应改变。

     “色调”是离散的颜色值，其中 0 是红色、60 是黄色、120 是绿色、180 是青色、200 是洋红色、240 是蓝色。

     “饱和度”是指定的色调中的颜色数，最大值为 240。

     “亮度”是颜色的明亮程度。
3. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

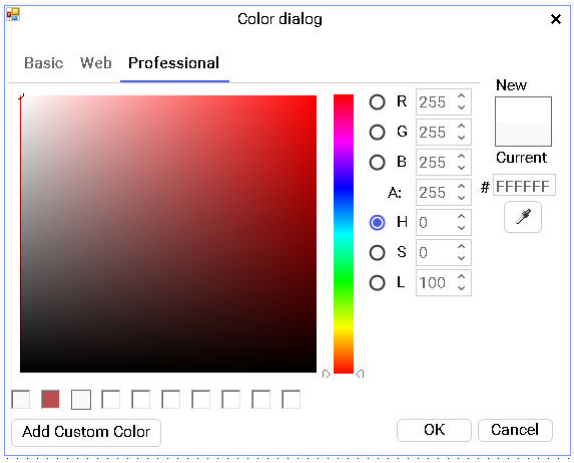
此时关闭颜色对话框，并应用所选的颜色。

创建自定义颜色

您可以创建由自定义颜色构成的调色板。

创建自定义颜色

1. 打开调色板。
2. 在自定义调色板区域中，选择一种颜色并选择添加自定义颜色。



3. 在颜色-纯色框中查看产生的颜色。

   如果显示器设置为显示 256 色，则颜色-纯色框可能会显示两种颜色。右侧显示所选的颜色作为纯色显示时的效果。左侧显示经过抖动处理的颜色，即使用 256 色中的两种来显示特定颜色时的近似效果。

4. 选择确定。

您也可以使用吸管工具来创建自定义颜色。

注意：使用此项功能创建透明位图。

使用吸管工具选择自定义颜色

1. 打开调色板。
2. 选择吸管工具，然后选择希望添加的颜色。

导入与导出自定义颜色

如果定义了自定义的调色板，则可以从 InTouch 应用程序中导出它，然后将它导入另一个 InTouch 应用程序。

要导入自定义的调色板

1. 打开调色板。
2. 单击自定义调色板向下箭头。
3. 单击加载调色板。此时出现标准的 Windows 打开对话框。
4. 找到并选择包含所需颜色定义的 .pal 调色板文件。
5. 单击打开。此时调色板文件中包含的颜色都加载到自定义调色板中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要导出自定义的调色板

1. 打**开调**色板。
2. **单击自定义调色板**向下箭头。
3. **单击导出调色板**。此时出现标准的 Windows 另存为对话框。
4. **为调**色板文件指定名称，然后**单击保存**。

平移与缩放

通过放大与缩小，可以更仔细地查看正在编辑的元素，从而确保各个对象都能精确排列或是位于正确的位置上。

缺省条件下，“平移与缩放”工具栏出现在屏幕的右下角。同其它工具栏一样，它可以浮动或固定在其它位置上。

如果您指定的目标分辨率与屏幕分辨率不同，边界画布将在平移与缩放时相应进行调整。

您可以：

- 在 100% 到 500% 的范围内进行放大与缩小。
- 使用橡皮圈工具缩放特定的区域。
- 按特定的百分比缩放窗口。
- 选择并拖动以平移窗口。
- 返回到正常的缺省视图。

显示或隐藏“平移与缩放”工具栏

- 在查看菜单上，选择状态栏。

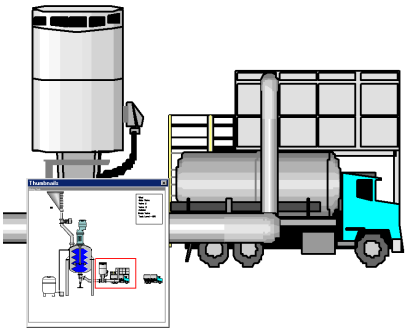
使用缩略图窗口进行平移和缩放

放大应用程序窗口的细部时，略图窗口显示该细部与整个应用程序之间的关系。

略图窗口让您既可以总览开发区域，又可以观察其细部。

通过选择**缩略图**按钮，可以显示或隐藏**缩略图**窗口。

红色长方形是窗口中缩放的区域的边界。



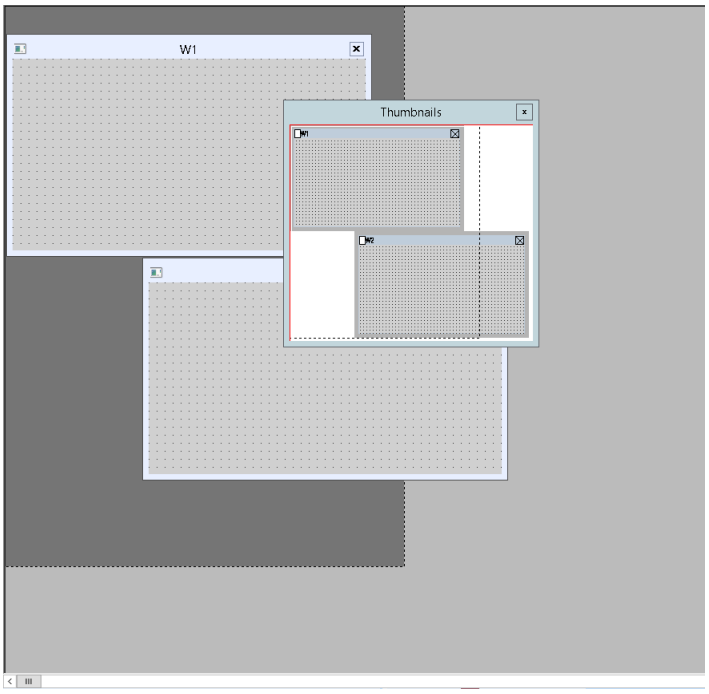
- 拖动红色长方形可以显示窗口的不同部分。
- 选择窗口的不同区域可以将长方形移到该区域。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 调整长方形的大小可以更改显示区域的缩放级别。

对于不能缩放的对象，如 ActiveX 控件，略图视图会显示白色的长方形框。

如果您指定的应用程序目标分辨率与屏幕分辨率不同，则描绘您指定的目标分辨率尺寸的边界画布也将以略图视图显示。



注意：超出目标分辨率边界尺寸的应用程序窗口仍将以略图视图显示，如上面的示例所示。

使用鼠标滚轮进行缩放与平移

如果鼠标带有滚轮，则通过按 CTRL 键并滚动滚轮，可以更改图像的缩放级别。

- 滚动鼠标滚轮时，每一格相当于按 20% 更改缩放级别。
- 将光标放到 InTouch 窗口中，然后按鼠标滚轮，也可以在窗口中浏览。按鼠标滚轮时，会出现带有四个箭头的图标。通过移动鼠标在窗口中浏览。

平移与缩放限制

平移与缩放操作不会应用于以下控件：

- ActiveX 控件
- 分布式报警对象
- 16 笔趋势
- 文本框
- 复选框
- 列表框
- 组合框

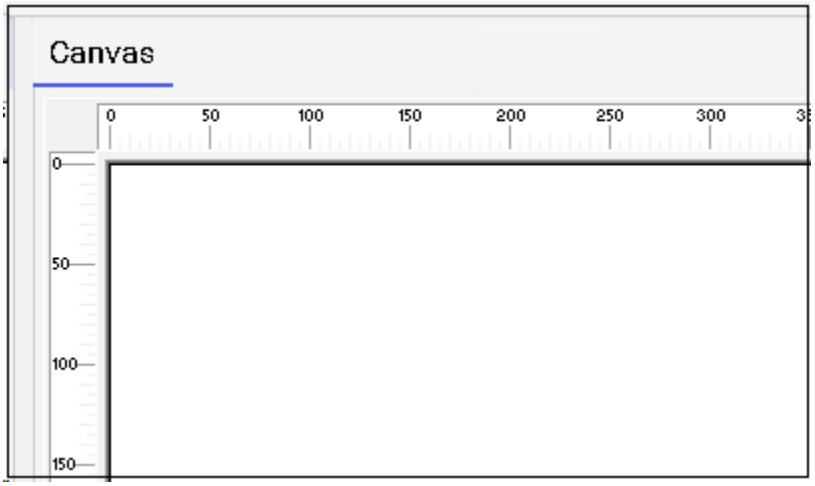
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- “单选按钮组”对象

如果在视图缩放到超过 100% 时，这些控件中的某一个位于可见区域中，则在该控件占据的区域中，会出现一个使用该控件名的长方形。

使用屏幕网格与标尺

您可以在开发区域中显示网格与标尺来帮助排列与对齐对象。



将对象对齐到网格

通过选择对齐网格，可以将对象对齐到预定义的网格点。

缺省条件下，网格设置为 10 个像素，并且设置为在启动 WindowMaker 时可见。网格的像素间隔可以在 WindowMaker 配置屏幕中进行配置。

要查看网格，必须选择 WindowMaker 配置屏幕上的显示网格，并且选择查看菜单上的对齐网格。

配置网格

1. 在文件菜单上，选择配置，然后选择 **WindowMaker**。此时出现 WindowMaker 配置屏幕。
2. 在网格间距框中，输入坐标之间间隔的像素数。
3. 如果希望在选择对齐网格时看到网格，请选中显示网格复选框。

   如果没有选择显示网格，则在选择对齐网格时，窗口中看不见网格。

使用标尺

使用标尺帮助精确对齐窗口中的对象。标尺沿着开发环境窗口的顶部和一侧显示。

显示或隐藏标尺

1. 在查看菜单上的视图组中，选择标尺。
2. 重复此步骤即可隐藏标尺。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

关于 WindowViewer

WindowViewer 为 InTouch 应用程序提供运行时环境。基于应用程序的工作要求，可以配置 WindowViewer 支持应用程序的方式。例如，根据应用程序的安全性要求，可以配置 WindowViewer 向操作员提供的菜单与命令。

自定义运行时环境

您可以通过设置属性来自定义运行时 WindowViewer 环境。

- 常规属性确定运行 InTouch 应用程序的环境条件。
- 窗口配置属性确定用户与 WindowViewer 中运行的 InTouch 应用程序进行交互时可以访问的菜单、命令以及窗口组件。

配置常规 WindowViewer 属性

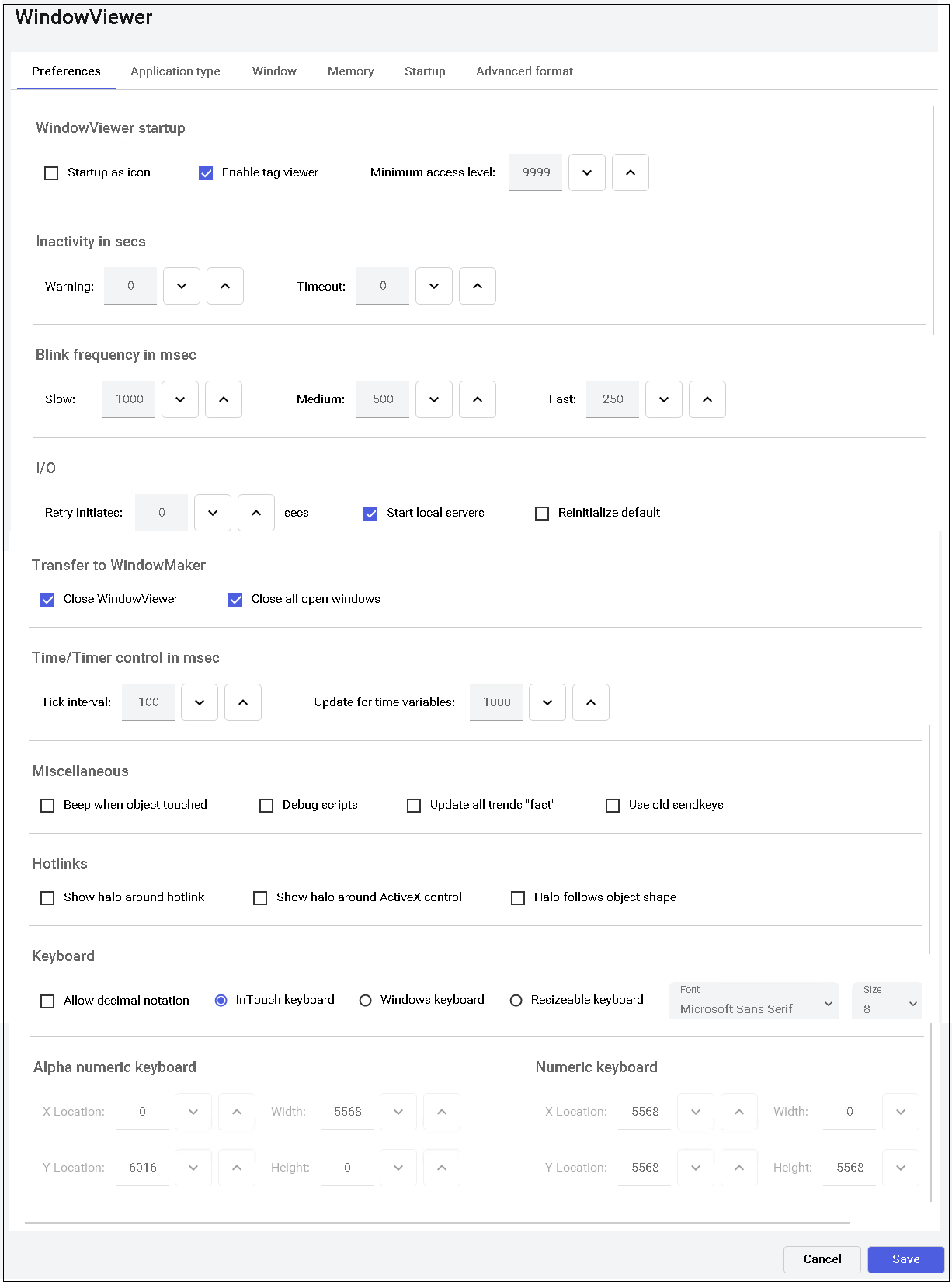
您可以通过配置一组常规属性来确定 WindowViewer 在作为 InTouch 应用程序运行时的特征。修改这些常规属性之后，必须重新启动 WindowViewer 才能使更改生效。

注意：先关闭 WindowViewer，然后更改操作系统的区域格式。

1. 打开 WindowMaker。
2. 在文件菜单上，**选择配置**，然后**选择 WindowViewer**。

   此时出现 WindowViewer 配置屏幕。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**首选项**选项卡的 **WindowViewer 启动**部分中，执行以下操作：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 如果您希望 WindowViewer 在启动时处于最小化状态，请选中**启动后只显示图标**复选框。
- 选中**启用标记查看器**复选框，以允许“标记查看器”应用程序在运行时启动。“标记查看器”可让您在运行时观察和监视标记并修改标记值。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序运行时》中的启用 Tag Viewer。
- 在**最小访问级别**框中，输入运行“标记查看器”应用程序所需的最小安全访问级别。该值必须介于 0 到 9999 之间。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序运行时》中的启用 Tag Viewer。

4. 在**不活动（秒）**部分中，设置操作员不活动警告与超时时段。

   如需有关设置警告和超时时段的详细信息，请参阅《AVEVA™ InTouch HMI 管理》中的配置不活动超时。

5. 在**闪烁频率（毫秒）**部分中，为慢速、中速以及快速闪烁动画链接输入以毫秒为单位的间隔长度。
6. 在 **I/O** 区域中，执行以下操作：
   - 在**重试间隔**框中，输入连接尝试失败之后 InTouch 应用程序重新尝试连接到 I/O 服务器之前等待的秒数。如果 InTouch 第一次就能成功连接到 I/O 服务器，那么 **启动 I/O 重试**框无效。
   - 如果希望在您启动 WindowViewer 而您尝试与之通讯的 I/O 服务器未在运行时显示对话框，请选中**启动本地服务器**复选框。
   - 如果希望使用缺省设置重新初始化“访问名”，请选中**重新初始化缺省值**复选框。此时会忽略指定给“访问名”的当前值，而使用它们的原始设置重新进行初始化。
7. 在**转换到 WindowMaker** 区域中，执行以下操作：
   - 如果希望在启动 WindowMaker 时自动关闭 WindowViewer，请选中**关闭 WindowViewer** 复选框。

     选择此选项时，也会自动选择位于 WindowMaker 配置屏幕上的“转换到 WindowViewer 时关闭”选项。如果内存足够，并且使用快速切换功能在 WindowViewer 与 WindowMaker 之间切换，则应该清除此选项。
   - 如果希望在从 WindowViewer 转换到 WindowMaker 时自动关闭所有打开的窗口，请选中**关闭所有打开的窗口**复选框。
8. 在**时间/定时器控制（秒）**区域中，执行以下操作：
   - 在**计时间隔**框中，输入 InTouch HMI 用于检查内部定时器的间隔。

     此间隔确定“应用程序运行期间”、“窗口显示期间”、“条件为真时（期间）/为假时（期间）”以及键与触动按钮动作“按下期间”QuickScript 可以启动的时间。

     此选项设置 INTOUCH.ini 文件中 TimerTickInterval 参数的值。对于按计划每 100 毫秒运行一次的脚本，应该将“计时间隔”设置为不超过 50 毫秒。在运行 Windows XP 或 Windows 2003 的计算机上，计时间隔下限为 10 毫秒。
   - 在**更新时间变量**框中，输入以毫秒为单位的间隔，系统标记（如 $Msec、$Second 或 $Minute）按此间隔更新时间。

     我们建议使用缺省设置 1000 毫秒。将此选项设置为零则不会更新时间变量。
9. 在**其它**区域中，执行以下操作：
   - 如果希望 InTouch 应用程序在操作员选择窗口上的触控对象时发出蜂鸣声，请选择**接触对象时蜂鸣**。
   - 如果希望每次运行 QuickScript 时将消息写入记录器，请选择**调试脚本**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果从窗口配置属性页中选择调试，则可以从 WindowViewer 的特别菜单中打开或关闭 QuickScript 记录功能。

- 如果希望趋势对象的更新速度更快，请选择更新所有趋势“快速”。

  仅当应用程序窗口上没有任何对象与运行时趋势重叠时，才能选择此选项。如果选择此选项并且有任何对象与趋势重叠，则可能无法正确绘制对象。

- 如果使用以 InTouch 3.26 或更早版本开发的国际化应用程序，请选中使用旧的 **SendKeys** 复选框。

10. 在热链接区域中，执行以下操作：
    - 如果希望在用户将光标移到运行时屏幕上的对象时突出显示该对象，请选中在热链接周围显示光晕复选框。
    - 如果希望在 ActiveX 控件周围显示光晕，请选中在 **ActiveX** 控件周围显示光晕复选框。
    - 如果希望在用户将光标移到对象时沿该对象的轮廓线出现突出显示的光晕，请选中光晕沿着对象形状复选框。
11. 在键盘区域中，选择希望使用的键盘类型（如果有）。

    如需有关在 WindowViewer 中设置键盘选项的详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发》中的动画对象。

12. 选中允许十进制表示法复选框，以在运行时仅允许十进制值，并限制非数字字符。（允许十进制表示法选项尚未在非英语操作系统上进行测试。）
13. 在英文字母/数字键盘和数字键盘区域中，配置 X、Y 位置以及“宽度”和“高度”值。
14. 选择保存。

配置 WindowViewer 的视觉特征

您可以通过配置属性来确定 WindowViewer 运行 InTouch 应用程序时的视觉特征。这些属性确定 WindowViewer 窗口上出现的菜单、命令以及标准控件。

配置 WindowViewer 的视觉特征

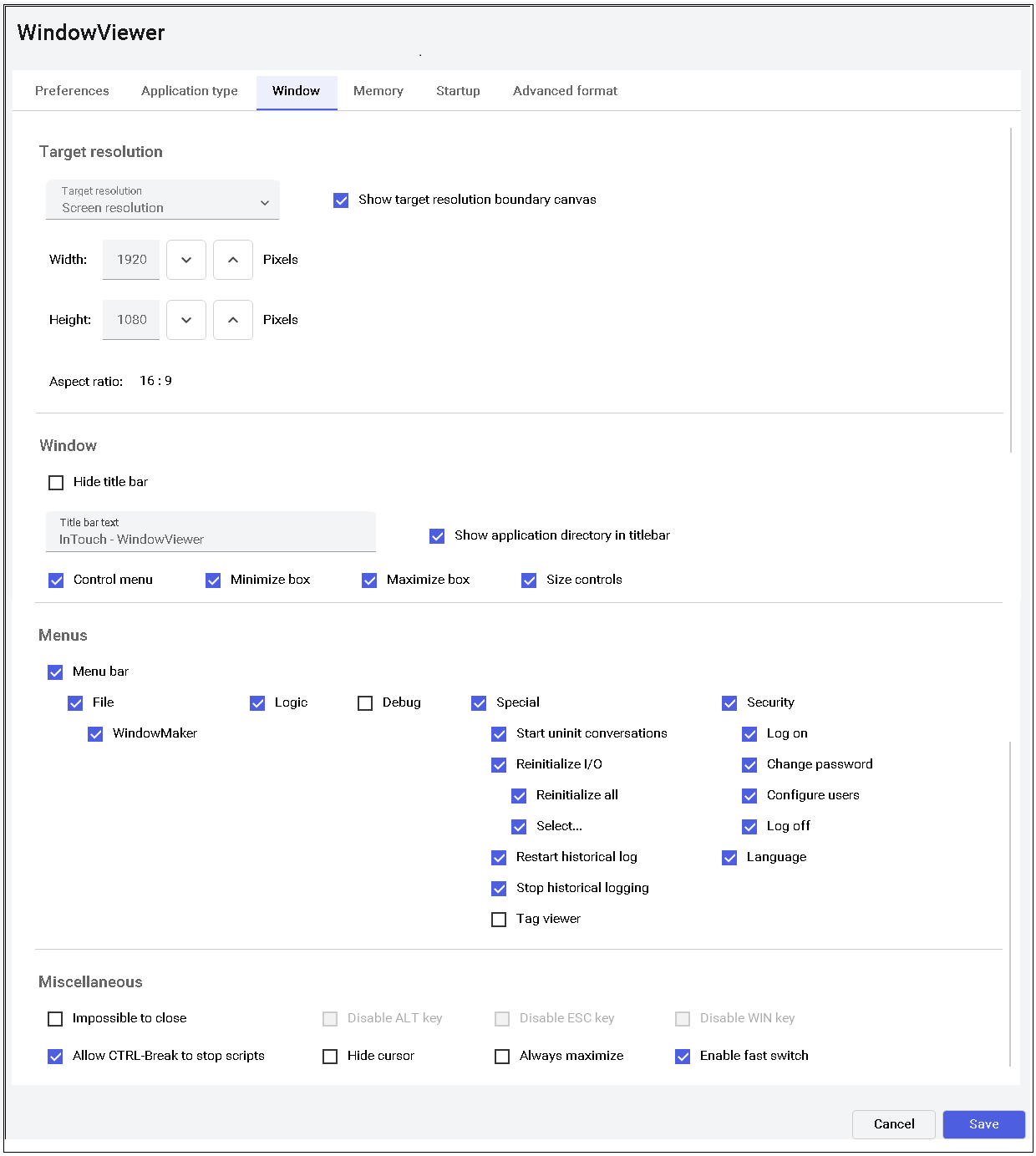
1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。

3. 选择窗口选项卡。

   选中视觉特征对应的复选框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 重新启动 WindowViewer。

配置远程会话中运行的应用程序的用户访问权限

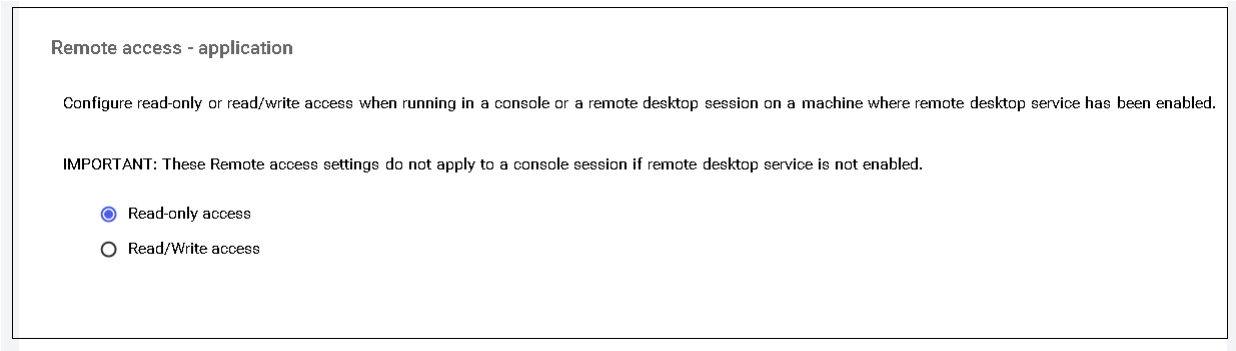
您可以为运行在远程桌面连接或终端服务会话中的分布式 InTouch 应用程序指定只读访问权限。只读访问权限适用于需要查看应用程序，但不应具有写权限的非操作员用户。例如，管理者需要使用 InTouch Access Anywhere 通过不安全的公共网络查看移动设备上的应用程序。使用只读访问权限为可通过公共网络访问的分布式 InTouch 应用程序提供更好的保护。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置远程节点上运行的应用程序的用户访问权限

1. 在 WindowMaker 中打开应用程序。
2. 在文件菜单上，选择配置，然后选择 **WindowViewer**。
3. 在 WindowViewer 配置屏幕中，选择应用程序类型选项卡。

   此时出现远程访问部分，其中包含向 WindowMaker 中打开的应用程序授予“读/写”或“只读”访问权限的选项。“读/写”访问权限是缺省选项。



4. 进行选择，然后选择确定以关闭对话框。

在初始化期间，WindowViewer 将验证应用程序是否运行在远程会话中以及是否指定为只读。此外，还将检查远程节点是否拥有只读许可证。如果上述所有条件均已满足，则可在只读模式下查看应用程序的 InTouch 链接和用户输入动画。

关于管理 WindowViewer 的内存

可以配置 WindowViewer 如何将内存用于窗口和弹出符号以提高运行时的性能。使用 ShowSymbol 动画和 ShowGraphic 脚本函数显示的窗口和弹出符号在运行时于特定条件下可保留在内存中，以便能够快速检索。

还可以为定期内存健康检查指定间隔并为堆栈内存段指定设置。每次打开一个新的符号时，无论预先设置的间隔是多长时间，它都会触发内存健康检查。

快速切换到 WindowViewer 或从 WindowViewer 快速切换到 WindowMaker 会清除图形缓存和窗口内存缓存。

如需有关针对 ShowGraphic 函数显示的符号的内存管理的信息，请参阅《工业图形编辑器用户指南》。

为 WindowViewer 窗口配置内存使用

可以配置 WindowViewer 如何将内存用于应用程序窗口以提高运行时的性能。

在特定条件下，重新打开缓存的已关闭窗口会从内存中检索这些窗口，而不是从磁盘加载窗口。

可以指定特定窗口具有内存使用的较高优先级，并且仅为这些窗口配置单独的内存设置。

修改任何 WindowViewer 内存选项后，您必须重新启动 WindowViewer 才能应用更改。

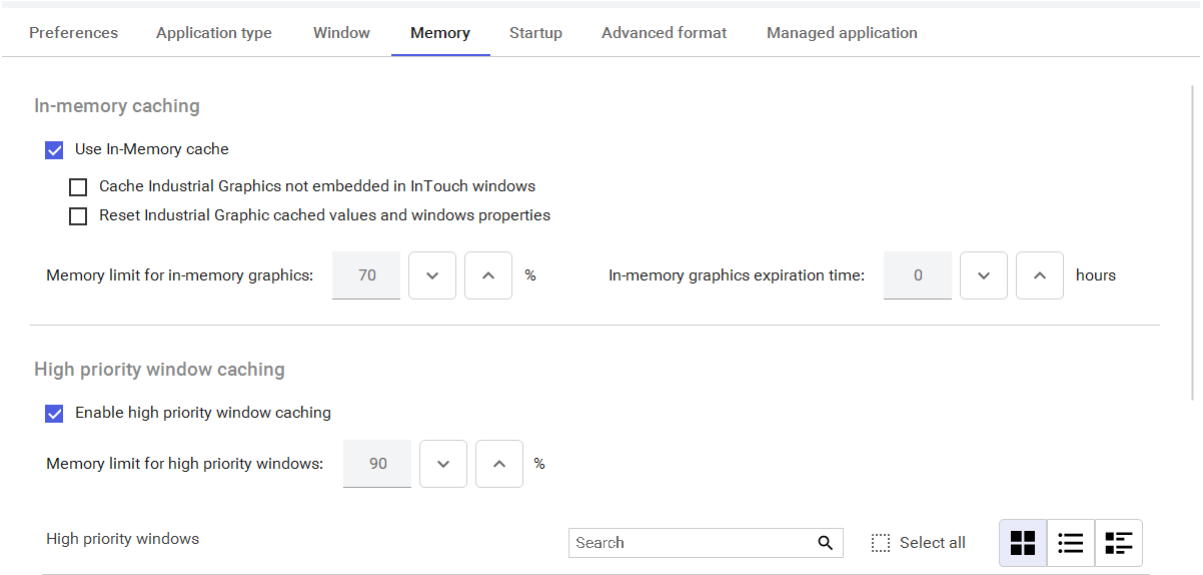
要设置内存属性，请执行以下操作：

1. 打开 WindowMaker。
2. 在文件菜单上，单击配置，然后单击 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 单击内存选项卡。



4. 在内存缓存区域中，执行以下操作：
   a. 如果您想在运行时将所有关闭但要缓存的窗口保存到内存中，请选中使用内存缓存复选框。
   b. 选中缓存未嵌入 InTouch 窗口的工业图形复选框可启用工业图形符号缓存。
   c. 如果希望在关闭并重新打开窗口时重置自定义属性和窗口属性，请选中重置工业图形缓存值和窗口属性复选框。
   d. 在内存窗口的内存限制框中，针对在运行时将已关闭的内存窗口保存在缓存内存中所占用的内存输入相应的上限。处理内存的缺省内存限制为 70%。

      如果超过该内存限制，那么系统就会在运行时自动从缓存中删除最早的已关闭内存窗口，除非该窗口已标记为高优先级窗口。

      内存窗口的内存限制始终低于高优先级窗口的内存限制。
   e. 在内存窗口过期时间框中，针对在运行时已关闭的内存窗口保存在缓存内存中的持续时间输入相应的上限。您可以输入 0 到 8760 小时之间的任意值。缺省值为 0 小时，这指定不存在时间限制。

      注意：InTouch 窗口和工业图形共享内存缓存。如果超过设置的内存限制，那么系统会自动删除缓存中最旧的内存图形。

应用内存限制还是过期时间限制取决于先达到哪个限制。

注意：在运行时，重新打开该窗口时，将显示各个动画中配置的缺省文本，而不是先前缓存的值，从而造成混淆。此行为支持的动画有：
值显示
用户显示
颜色更改填充
颜色更改线
颜色更改文本
工具提示
已命名的样式

5. 在高优先级窗口缓存区域中，执行以下操作：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

a. 选中启用高优先级窗口缓存复选框，以允许某些窗口被标记为高优先级。在运行时关闭这些窗口后，它们会始终保存在缓存内存中。

b. 在高优先级窗口的内存限制框中，针对在运行时将已关闭的高优先级窗口保存在缓存内存中所占用的内存输入相应的上限。内存限制的缺省值为 90%。运行时，如果所用内存与总内存的百分比超过该限制，系统会先删除最早的内存窗口，然后删除最早的高优先级窗口。

c. 在高优先级窗口框中，选择要标记为高优先级的窗口。

注意：当启用使用内存缓存或启用高优先级窗口缓存复选框时，也会启用重置工业图形缓存值和窗口属性复选框。当选中重置工业图形缓存值和窗口属性复选框时，重置操作也将影响使用内存缓存或启用高优先级窗口缓存选项。

6. 单击保存。

配置内存健康检查的时间间隔

系统会定期检查内存和图形设备界面 (GDI) 计数。如果超过了内存或 GDI 计数限制，则系统会开始删除窗口。缺省条件下，该时间间隔为 5 分钟。

如果您要更改间隔，则可以在 InTouch.ini 文件中添加新项目，然后指定新间隔值。

如果您要关闭检查，则可以添加新项目，然后将值设置成 0。

在修改间隔之后，您必须重新启动 WindowViewer 才能应用更改。

如需有关如何删除窗口的详细信息，请参阅为 WindowViewer 窗口配置内存使用。

配置内存健康检查的时间间隔

1. 在应用程序文件夹中打开 InTouch.ini 文件。
2. 在 [Intouch] 节下，添加以下脚本，其中 *<interval>* 是所需间隔（以分钟为单位）：

```
MemoryHealthCheckInterval = <interval>
```

无论预先设置的间隔是多长时间，打开新的弹出符号或新窗口都将触发内存健康检查。

配置 wwHeap 内存设置

wwHeap 是用于管理堆栈内存段的内存管理器。内存管理器可提供让一个或多个程序共享虚拟内存的机制。

警示：仅当您遇到记录器中报告的内存冲突时，才修改 wwHeap 内存设置。

您可以通过指定 wwHeap 的大小和开始位置来配置 wwHeap 内存设置。缺省大小、缺省开始位置和允许的位置范围因操作系统而异。

缺省大小在下表中介绍：

操作系统	缺省大小
32 位	1519 MB
32 位，启用了 /3GB 开关	2048 MB
64 位	2048 MB

缺省位置与允许的位置范围在下表中介绍：

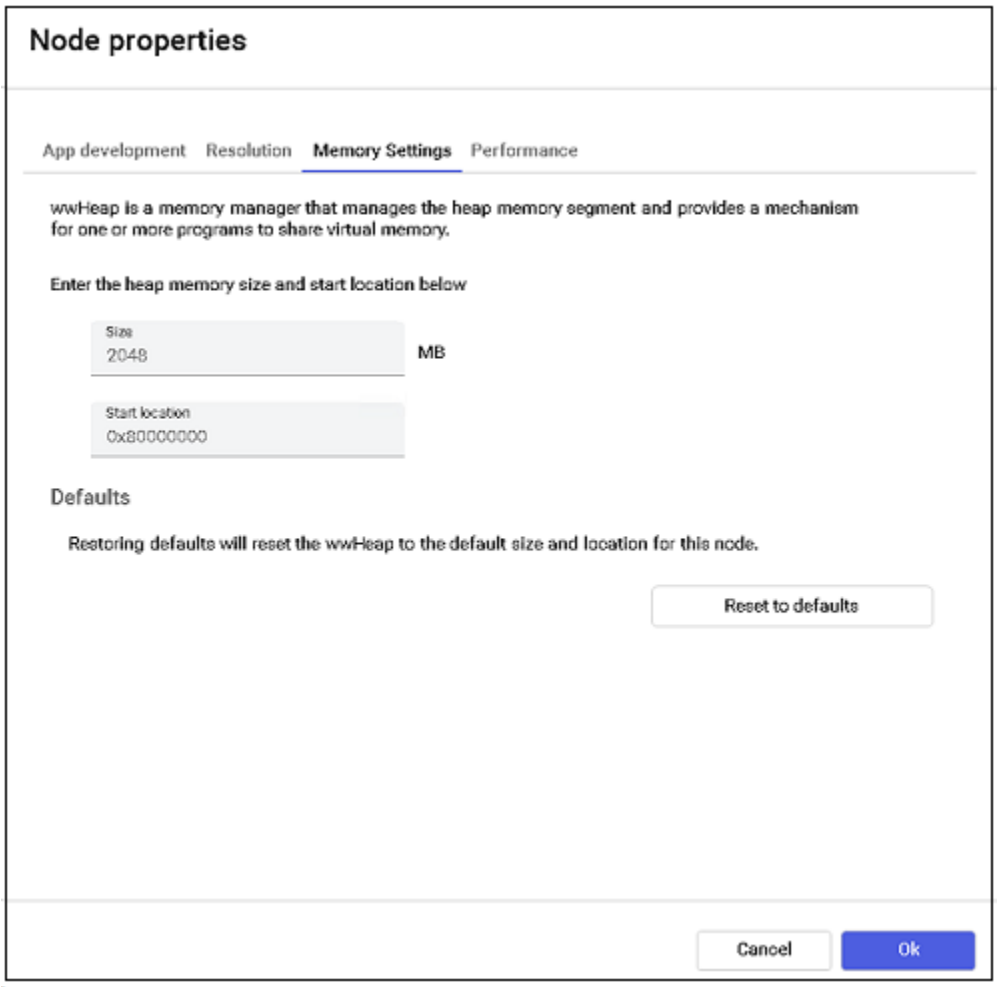
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

操作系统	缺省开始位置	允许的范围
32 位	0x21000000	0x00010000 至 0x7FFEFFFF
32 位，启用了 /3GB 开关	0x40000000	0x00010000 至 0xBFFEFFFF
64 位	0x80000000	0x00010000 至 0xFFFEFFFF

配置 wwHeap 内存设置

1. 启动应用程序管理器。
2. 在工具菜单上，选择节点属性。

   此时出现节点属性屏幕。
3. 选择内存设置选项卡。



4. 执行以下操作：

- 在大小框中，输入 wwHeap 内存的大小。您可以输入 1 MB 到 2048 MB 之间的任意值。
- 在开始位置框中，输入起始地址。

5. 要恢复缺省值，请选择恢复缺省值。
6. 选择确定。

选择区域设置 WindowViewer 选项

InTouch WindowMaker 包括一个 WindowViewer 高级格式区域设置配置选项。

要按区域语言环境启用数字格式，必须在设计时选择区域设置选项以将工业图形数字格式化为在区域设置中选择的国家/地区的格式。缺省条件下，禁用区域设置选项。

WindowViewer 只在启动时检查操作系统区域设置。这意味着如果执行以下任一操作，可能需要重新启动 WindowViewer：

- 在 WindowViewer 运行时选择区域设置选项。
- 在 WindowViewer 运行时且选择了区域设置选项的情况下更改操作系统区域设置。

设置高级格式属性

根据该值的大小，您可以在 WindowViewer 中配置要在运行时显示的小数位数。只有当您在“值显示”或“用户输入”动画中选择“实型”时，该功能才可用。

如果从现场设备中收集的数据质量不佳或因过大而无法显示，那么您可以指定要在运行时显示的特殊字符。

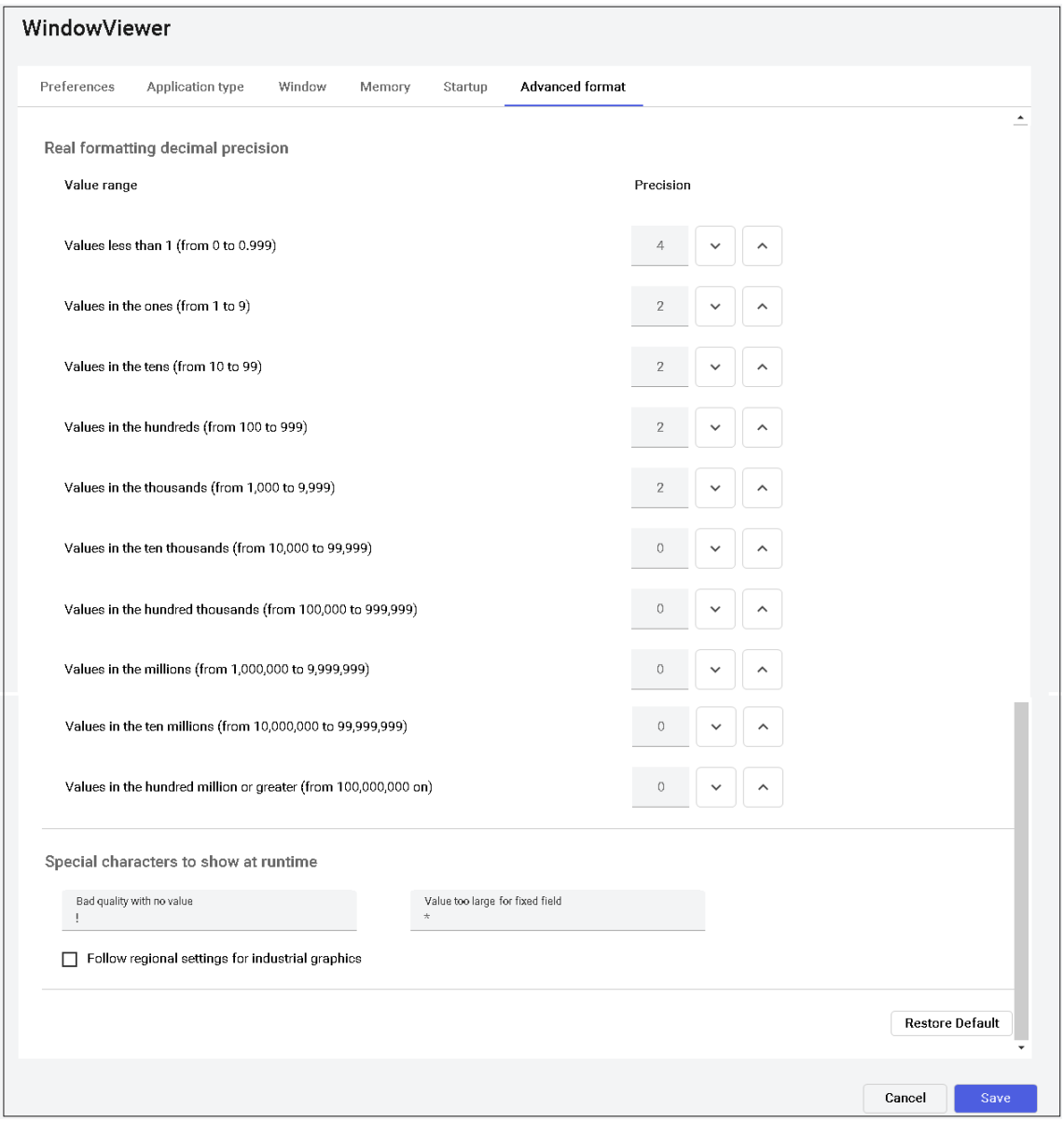
工业图形中出现的数字可以按照 Windows 控制面板的区域设置中设置为当前位置的国家或地区的格式显示。在运行时，随工业图形数值一起显示的千位分隔符和小数点与操作系统“区域设置”中指定的国家或地区的数字格式是一致的。

设置高级格式属性

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。
3. 选择高级格式选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**实时设置小数精度格式**部分中，**针对每**个**实**型数字范**围输**入您希望在运行**时显**示的小数位数。
5. 在**运行时显示的特殊字符**部分中，**执**行以下操作：
   - 在**质量不佳且无值**框中，**输**入当数据点**质**量不佳且没有上一个已知良好**值时**您希望在运行**时显**示的字符。缺省字符**为 !**。您可以**输**入除空格之外的任意 ASCII 字符。
   - 在**对于固定宽度过大的值**框中，**输**入当**值过**大而无法**显**示**时**您希望在运行**时显**示的字符。缺省字符**为 ***。您可以**输**入除空格之外的任意 ASCII 字符。
6. 要以**计**算机的**区域设**置指定的国家或地区的格式**显**示工**业图**形内的数字，**请选**中**遵循工业图形的区域设置**的**复选**框。

   缺省条件下，禁用**区域设置选项**。

   注意： WindowViewer 只在启**动时检查**操作系**统**区域**设**置。**这**意味着如果**执**行以下任一操作，可能需要重新启**动** WindowViewer：

1) 在 WindowViewer 运行时选择区域设置选项。
2) 在 WindowViewer 以所选的区域设置选项运行时更改操作系统区域设置。

7. 要恢复缺省值，请选择恢复缺省值。

设置托管 HMI/SCADA 应用程序的计算机的区域语言环境

要按区域语言环境启用数字格式，请将运行 HMI/SCADA 应用程序的计算机的区域设置为您想要的工业图形数字格式所属的国家/地区。

区域设置可以从 Windows 控制面板访问。如果您想要显示非美国格式的工业图形数字，请选择格式选项卡并在格式字段中选择国家/地区。

为终端服务器环境中的 WindowViewer 配置核心关联

在“终端服务”客户端上运行时，如果计算机具有多个处理器，则 WindowViewer 可以将 CPU 0 之外的 CPU（核心）用于其执行。这样便使得在终端服务器环境中运行的 InTouch 应用程序可以利用终端服务器的多核心功能。但是，当 WindowViewer 在终端服务器控制台上运行时，此选项不可用。InTouch 应用程序始终在单个处理器上运行，无论可用处理器数是多少。

当 WindowViewer 启动时，系统会检查计算机是否具有多个处理器以及允许哪些处理器运行 WindowViewer 实例。WindowViewer 随后按顺序检查处理器，在运行最少 View 实例的处理器上开始运行。

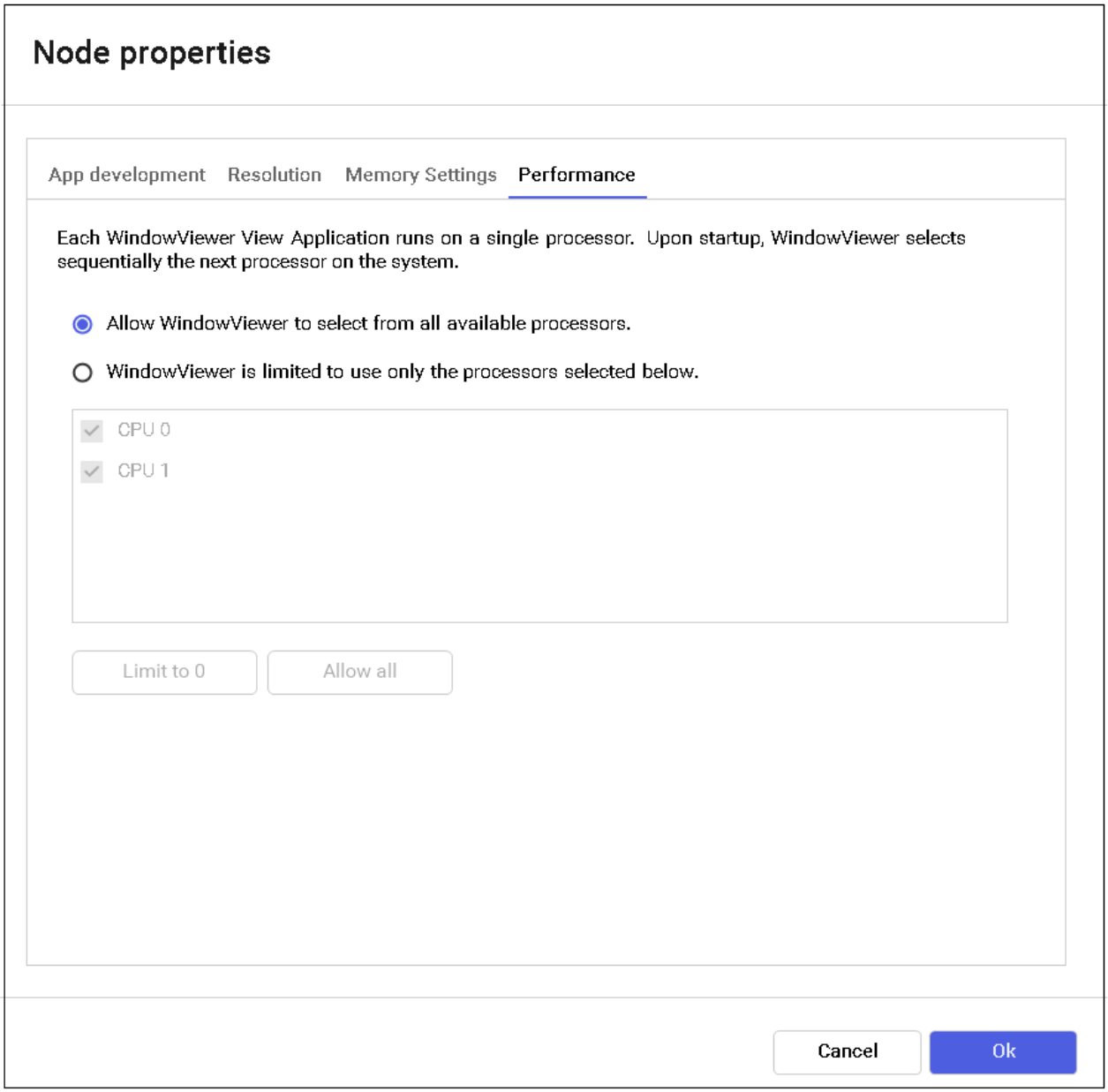
如果您拥有可访问性能选项卡的管理权限，则可以配置处理器“池”，WindowViewer 从中选择要在其上运行的处理器。

可在应用程序管理器为 WindowViewer 设置核心关联。请避免使用任务管理器手动为 WindowViewer 调整核心关联，因为 WindowViewer 核心选择过程不考虑在任务管理器中配置的核心关联设置。

配置处理器“池”

1. 启动应用程序管理器。
2. 在工具菜单上，选择节点属性。此时出现节点属性对话框。
3. 选择性能选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 要允许 WindowViewer 使用任何可用处理器，**请选择允许 WindowViewer 从所有可用的处理器中进行选择**。
5. 要限制 WindowViewer 可以使用的处理器，**请选择 WindowViewer 仅限于使用以下所选的处理器**，然后执行以下任何操作：
   - 确保选中您希望 WindowViewer 能够在其上运行的每个处理器的 **CPU** 复选框。
   - **选择限制为 0** 可仅允许 WindowViewer 在处理器 0 上运行。当您单击此按钮时，会自动选中 **CPU 0** 复选框。
   - **选择允许所有**可选择所有复选框。

   您可以随时清除所选处理器，并从列表中选择新处理器。还可以一次选择多个处理器。如果您清除某个处理器复选框，则 WindowViewer 实例会继续在该处理器上运行。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 选择确定。WindowViewer 会基于其它 View 会话，在下一个 CPU 上启动。

使用 WindowViewer 窗口

典型的 InTouch 应用程序至少包含多个窗口，操作员可以通过与这些窗口进行交互来管理工业过程。根据您在 WindowViewer 配置 屏幕上的窗口选项卡中设置的属性，操作员可以从 WindowViewer 文件菜单中运行标准命令来打开与关闭窗口。

在不同模式下查看 WindowViewer

选择详细信息可从列表视图更改为详细信息视图。详细信息会显示窗口的类型以及上次修改日期和时间。

在详细信息视图中，可以通过单击窗口所在行的任何部分（而不仅是复选框）来选择与取消选择任何未打开的窗口。选中时整行突出显示。

- 要打开所选窗口，请选择确定。
- 要取消选择并关闭对话框，请选择取消。
- 要让对话框返回列表视图，请选择列表。
- 要选择列出的所有窗口，请选择全选。
- 要取消选定的所有窗口，请选择全部清除。
- 要按升序或降序对列表进行排序，请选择列标题。

从 WindowViewer 中打开窗口

如果将 WindowViewer 配置成显示文件菜单，则操作员可以打开或关闭 InTouch 应用程序窗口。操作员从文件菜单中选择打开窗口或关闭窗口命令时，出现所选命令对应的对话框。

适合所选命令的所有窗口的名称都会出现在列表中。例如，选择打开窗口命令之后，出现要显示的窗口屏幕。

如果 WindowViewer 配置成显示文件菜单，则操作员可以打开 InTouch 应用程序窗口。

从 WindowViewer 中打开窗口

1. 在文件菜单上，选择打开窗口。

   此时出现要显示的窗口对话框。

2. 选择要打开的每个窗口的名称旁边的复选框。
1. 选择确定以关闭对话框并打开所选的窗口。

注意：如果选择了“替换”型窗口，它会关闭与之相交的任何窗口。

从 WindowViewer 中关闭窗口

如果 WindowViewer 配置成显示文件菜单，则操作员可以关闭 InTouch 应用程序窗口。

关闭打开的窗口

1. 在文件菜单上，选择关闭窗口。

   此时出现选择要关闭的窗口屏幕。

2. 选择要关闭的一个或多个窗口的名称旁边的复选框。
3. 选择确定以关闭所选窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从 WindowViewer 转换到 WindowMaker

开发 InTouch 应用程序时，通过选择文件菜单中的 WindowMaker 命令或工具栏上的**开发**命令，可以轻松地在 WindowMaker 与 WindowViewer 之间进行转换。这就是所谓的快速切换。

快速切换仅针对快速开发测试。不要在生产环境中使用它。您可以隐藏切换到 WindowMaker 的命令。

从 WindowViewer 转换到 WindowMaker

1. 启动 WindowViewer。
2. 在 文件 菜单上，选择 **WindowMaker**。

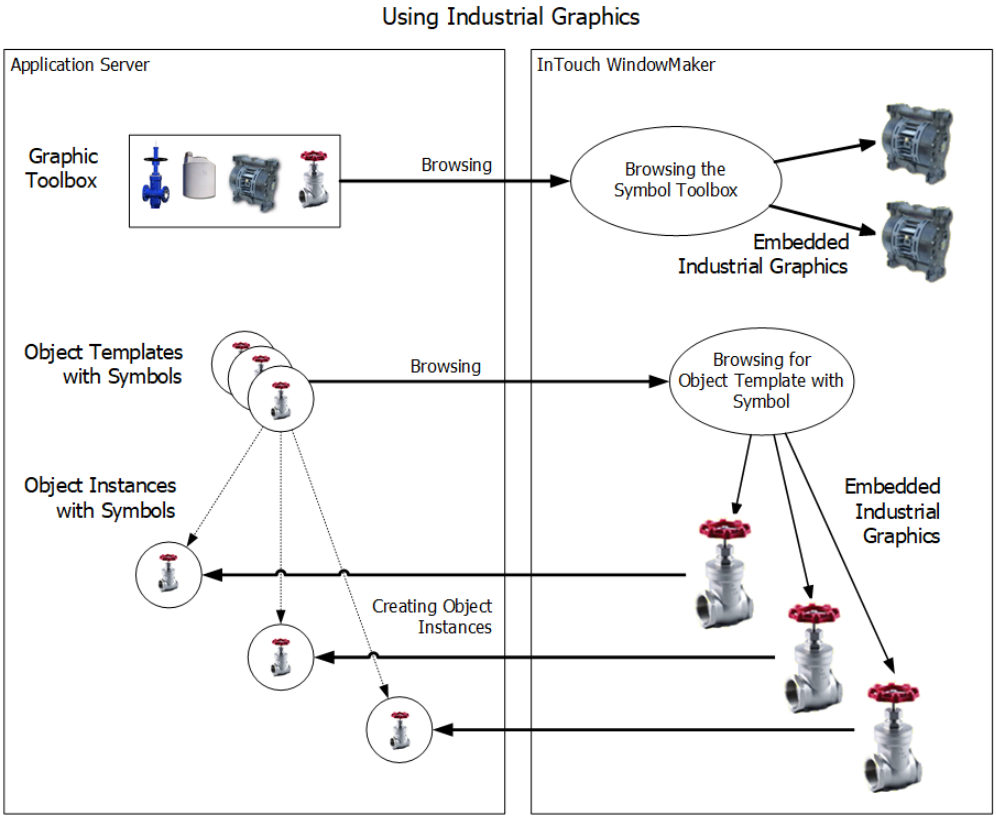
   此时出现要编辑的窗口对话框。
3. 选择转换到 WindowMaker 时要打开的每个窗口的名称旁边的复选框。
4. 选择确定以关闭对话框并转换到 WindowMaker。

注意：如果应用程序开发人员在开发期间配置 WindowViewer 的属性时选择**关闭 WindowViewer** 选项，则转换到 WindowMaker 时 WindowViewer 会自动关闭。

工业图形

System Platform IDE 包含一个“工业图形编辑器”，可用于创建符号来代表生产过程，还可以为 AutomationObject 提供 HMI 界面。

下图显示如何在 InTouch 应用程序中使用以工业图形编辑器创建的符号。



如需了解更多信息，请观看标题为“Introduction to Industrial Graphics（工业图形简介）”的视频。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建工业图形

在 IDE 中，工业图形是使用“工业图形编辑器”创建的。

您可以创建：

- 工业图形（在“工业图形工具箱”中）。这些工业图形不与任何特定的对象模板或对象实例关联。
- 工业图形包含在特定的对象模板或实例中。

工业图形编辑器

除了在 System Platform IDE 中管理 InTouch 应用程序的优点外，还可以在“工业图形编辑器”中创建图形来为生产环境建模。“工业图形编辑器”已完全集成到 System Platform IDE 中，支持强大的图形编辑功能。

“工业图形编辑器”包含向符号元素应用“元素样式”的能力。“元素样式”是针对应用于图形元素的文本、元素填充、元素轮廓和线条定义的一组视觉属性。使用“元素样式”可以轻松地将标准视觉样式应用到图形元素，并建立一致的符号标准。

“工业图形编辑器”还包含“符号向导编辑器”，可用于创建含有不同视觉和功能配置的“符号向导”。例如，可以使用“符号向导编辑器”创建具有单独左进气阀配置和右进气阀配置的可配置泵符号。Situational Awareness Library 符号是包含从符号的“向导选项”选择的预构建配置的“符号向导”的示例。单个 Situational Awareness Library 符号可以轻松配置为表示温度计、流量计或压力计。在托管的 InTouch 应用程序中嵌入“符号向导”时，选择所需配置。

WindowMaker 融合了“工业图形工具箱”，无需打开“工业图形编辑器”即可为 InTouch 应用程序选择工业图形和 Situational Awareness Library 符号。可以通过将符号直接从 WindowMaker 拖放到 InTouch 窗口中来添加符号。如需有关使用工业图形和 Situational Awareness Library 符号的详细信息，请参阅《工业图形编辑器》文档或 WindowMaker 帮助。

注意：在最新版本的 Microsoft 渲染技术中，某些渐变功能已经弃用。为了适应并保证将来支持图形，已将受影响的功能从配置环境中删除。之前使用弃用功能配置的图形将继续按预期渲染。请参阅《工业图形编辑器用户》文档中的“使用弃用的功能加载图形”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 2 应用程序

您可以使用“InTouch 应用程序管理器”或 IDE 来管理 InTouch 应用程序。

InTouch 应用程序可按管理方式、支持的符号类型和发布的来源进行分类：

- 独立的 InTouch 应用程序：独立的应用程序是在“应用程序管理器”中创建的缺省应用程序，允许灵活使用 InTouch 符号和工业图形。
- 托管的 InTouch 应用程序：托管的应用程序是使用 IDE 创建和管理的。
- 发布的 InTouch 应用程序：发布的应用程序是通过从衍生的 InTouchViewApp 模板中导出托管的应用程序进行创建的。
- InTouchView 应用程序：InTouchView 应用程序可从“InTouch 应用程序管理器”或 IDE 进行创建。

InTouch HMI 支持迁移和升级旧版新型应用程序。

独立的 InTouch 应用程序

独立的 InTouch 应用程序由“InTouch 应用程序管理器”管理。它们在“InTouch 应用程序管理器”中显示独立字样。

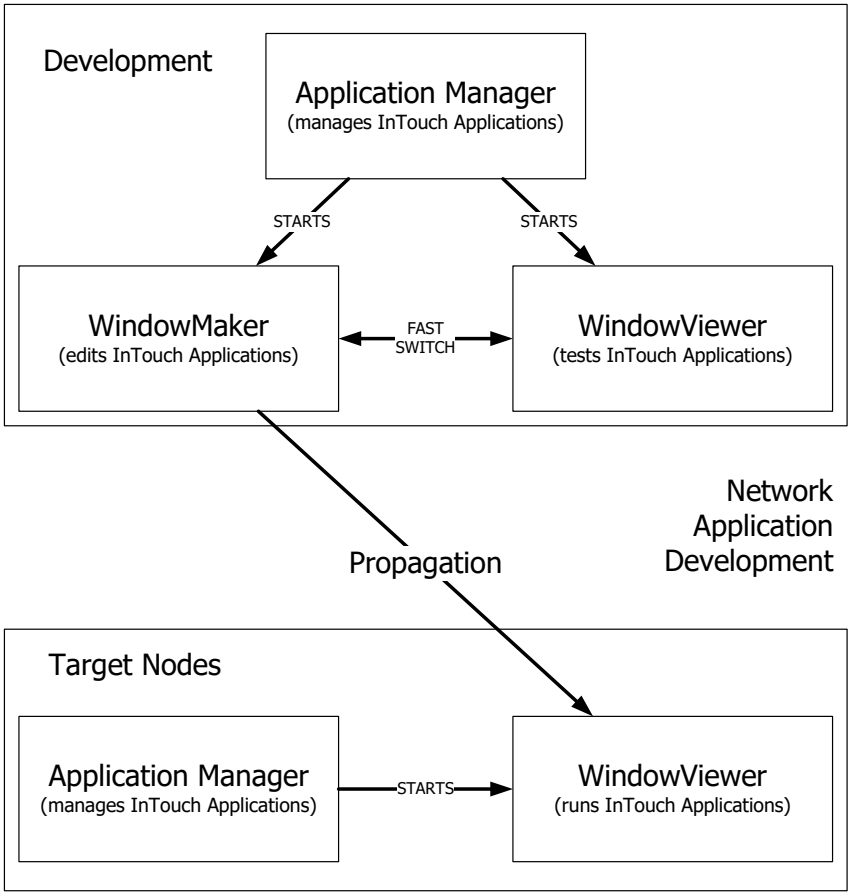
<1>通过使用</1><2>“</2><3>应用程序管理器</3><4>”，</4><5>您可以</5><6>：</6>

- 创建与管理独立的 InTouch 应用程序。
- 启动 WindowMaker 以编辑 InTouch 应用程序。
- 启动 WindowViewer 以运行 InTouch 应用程序。

您也可以直接在 WindowMaker 与 WindowViewer 之间切换以测试或运行应用程序，然后再切换回去以修改应用程序。独立的应用程序由 InTouch HMI 在目录文件系统中维护的一组文件构成。它还可以包含工业图形。独立的应用程序可以部署到多个网络节点上，而不限制于单个节点。它可以导入到 IDE 中并转换为托管的应用程序。

对于将更改从开发节点上的 InTouch 应用程序传播到目标节点上运行的 InTouch 应用程序，由“网络应用程序开发”负责管理。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



托管的 InTouch 应用程序

您可以使用 System Platform 集成开发环境 (IDE) 来管理 InTouch 应用程序。这些应用程序称为托管的 InTouch 应用程序。

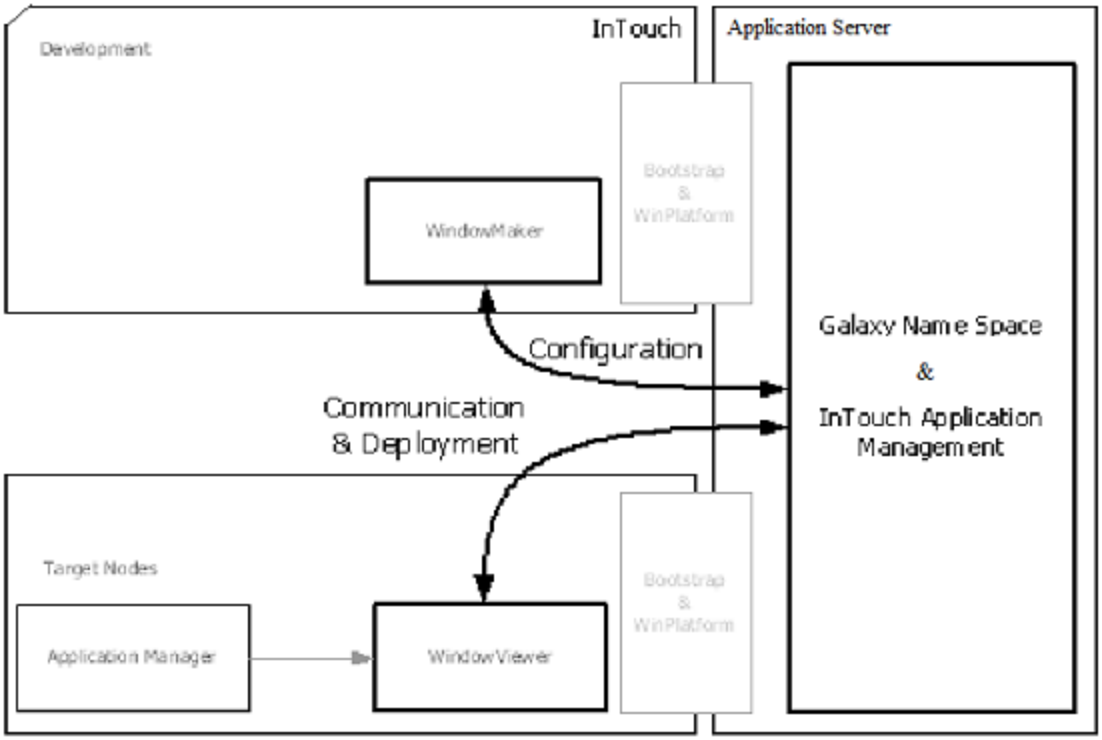
它们在“InTouch 应用程序管理器”中显示托管字样。这些应用程序与 ArchestrA 环境的集成程度更高，并且支持高级图形。

使用 WindowMaker 在 Galaxy 的一个节点上开发 InTouch 应用程序。然后将其部署到一个或多个正在运行 WindowViewer 的目标节点上。

使用 IDE 的系统平台功能来管理 InTouch 应用程序时，您可以：

- 查看哪些 InTouch 应用程序在哪个节点上运行。
- 使用 InTouch 应用程序中心储备库。
- 将更改部署到在远程节点上运行的 WindowViewer。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



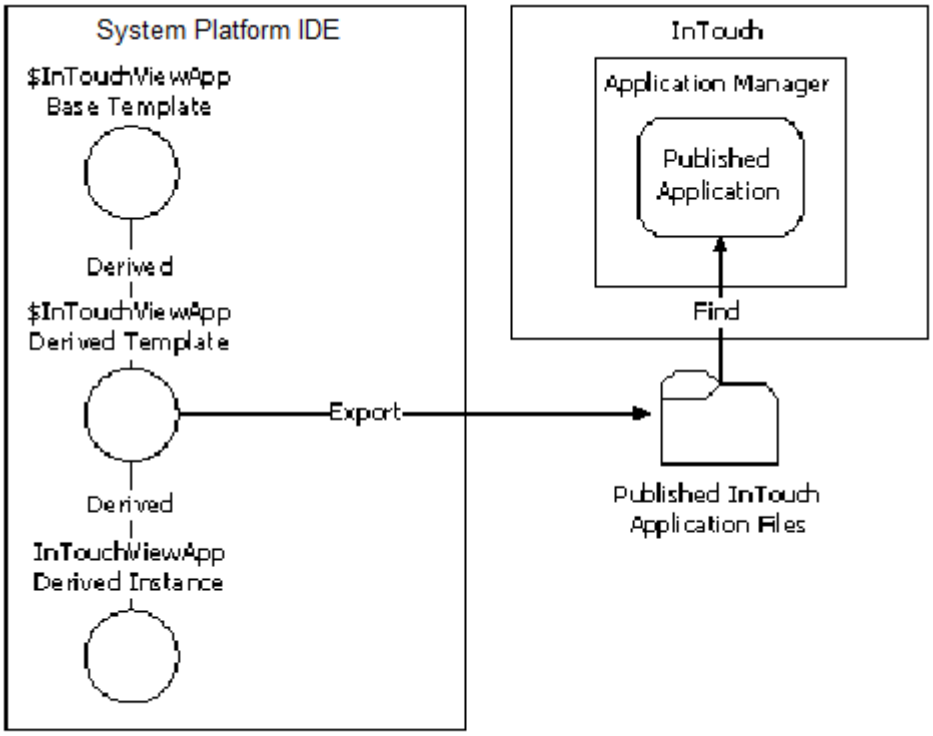
您只能通过“应用程序管理器”在 WindowViewer 中启动托管的应用程序。WindowViewer 启动时，它在运行时将托管的应用程序的文件复制到文件夹中。

每个托管的应用程序都与一个从基本模板衍生的 InTouchViewApp 对象关联。InTouchViewApp 对象仅包含对托管的 InTouch 应用程序文件夹的引用，以及托管的 InTouch 应用程序的特定于行为的其它信息。应用程序文件存储在单独的 Galaxy 文件储备库文件夹中。一个文件夹包含 InTouch 应用程序最近签入的版本，另一个文件夹包含最近签出的版本。使用 IDE 中的“工业图形编辑器”，可以在 InTouch HMI 应用程序中创建代表生产过程的符号。仅当从 IDE 打开 WindowMaker 时，才能快速从 WindowMaker 切换到 WindowViewer，来测试托管的应用程序。

发布的 InTouch 应用程序

编辑托管的 InTouch 应用程序之后，您可以发布它。您可以从衍生的 InTouchViewApp 模板发布托管的应用程序。发布托管的应用程序时，会创建一个用户自定义的文件夹，包含 InTouch 应用程序文件以及应用程序中内嵌的任何工业图形。您使用“应用程序管理器”的查找实用程序来确定该文件夹的位置。此后，转换后的应用程序作为已发布的应用程序出现在“应用程序管理器”中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



发布的 InTouch 应用程序的优点是，它们可以像独立的 InTouch 应用程序那样分发，但是还可以继续支持工业图形的功能。

不过，您不再能：

- 使用 System Platform IDE 来部署 InTouch 应用程序。
- 在 InTouch 应用程序中编辑或添加工业图形。
- 编辑发布的 InTouch 应用程序。

您可以使用 WindowMaker 将已发布的应用程序从 InTouch 11.1 之前的版本（2014 R2 P01）迁移到 V11.1 P01。然后，您可以更新应用程序源文件然后重新发布，来修改已发布的应用程序。发布托管的应用程序之后，仍可以使用内嵌的工业图形将数据写入 Galaxy 或实现数据的可视化。已发布的应用程序无法再次导入 Galaxy。迁移已发布的托管应用程序后，您需要重新发布该应用程序。重新发布时，“内嵌的报警控件”将升级到新版本。

InTouchView 应用程序

InTouchView 应用程序显示用于“应用程序服务器”环境的可视化界面。InTouchView 应用程序在 WindowViewer 中运行，但由 Application Server 提供大多数 HMI 功能。应用程序还可以配置为作为 InTouch 标记服务器，为其它节点提供安全标记信息。InTouchView 应用程序在客户端节点只需要访问数据而不需要开发节点完整功能的情形中非常有用。

InTouchView 应用程序仅提供全功能 InTouch 应用程序中的某些标准功能。InTouchView 应用程序：

- 无法连接到 Application Server Galaxy 或 Intouch 标记服务器之外的其它 I/O 数据源。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 无法生成报警。不过，您可以显示与确认来自远程报警供应器（如 Application Server 对象和 InTouch 报警）的报警。
- 不记录应用程序数据或事件。InTouchView 应用程序仅生成与 SYS 和 USER 相关的事件。
- 当应用程序使用来自 Galaxy 的数据时，只能使用 ArchestrA 安全性对其进行保护。
- 无法引用内嵌工业图形中的标记。

您使用 WindowMaker 开发 InTouch 应用程序，然后在 WindowViewer 中运行它们。随后可以将 InTouch 应用程序更改为 InTouchView 应用程序，这样可以通过 Application Server 管理 InTouch 应用程序。同样，您可以将 InTouchView 应用程序更改为 InTouch 应用程序。

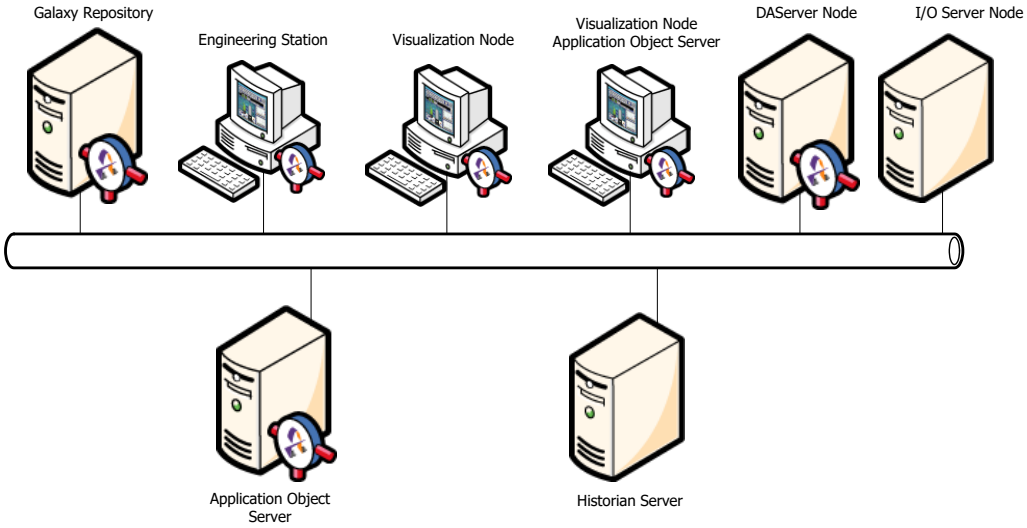
以下列表显示了在创建 InTouchView 应用程序时不能使用哪些 WindowMaker 命令和“标记名字典”选项。

- 不可用的命令：
  - 访问名
  - 报警组
  - 配置...报警
  - 配置...历史记录
  - 配置...分布式名称管理器
- 不可用的“标记名字典”选项：
  - 报警
  - 详细和报警
  - 记录数据
  - 记录事件
  - 优先级

应用程序服务器架构

“应用程序服务器”依靠 ArchestrA 技术为分布式 InTouch HMI 应用程序提供服务。“应用程序服务器”服务分布在一组节点上，InTouch HMI 安装在“可视化节点”上以提供应用程序界面。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



“应用程序服务器”Galaxy 代表基于单个逻辑域名空间的系统配置，该域名空间纳入了分布在多个节点上的各种组件。

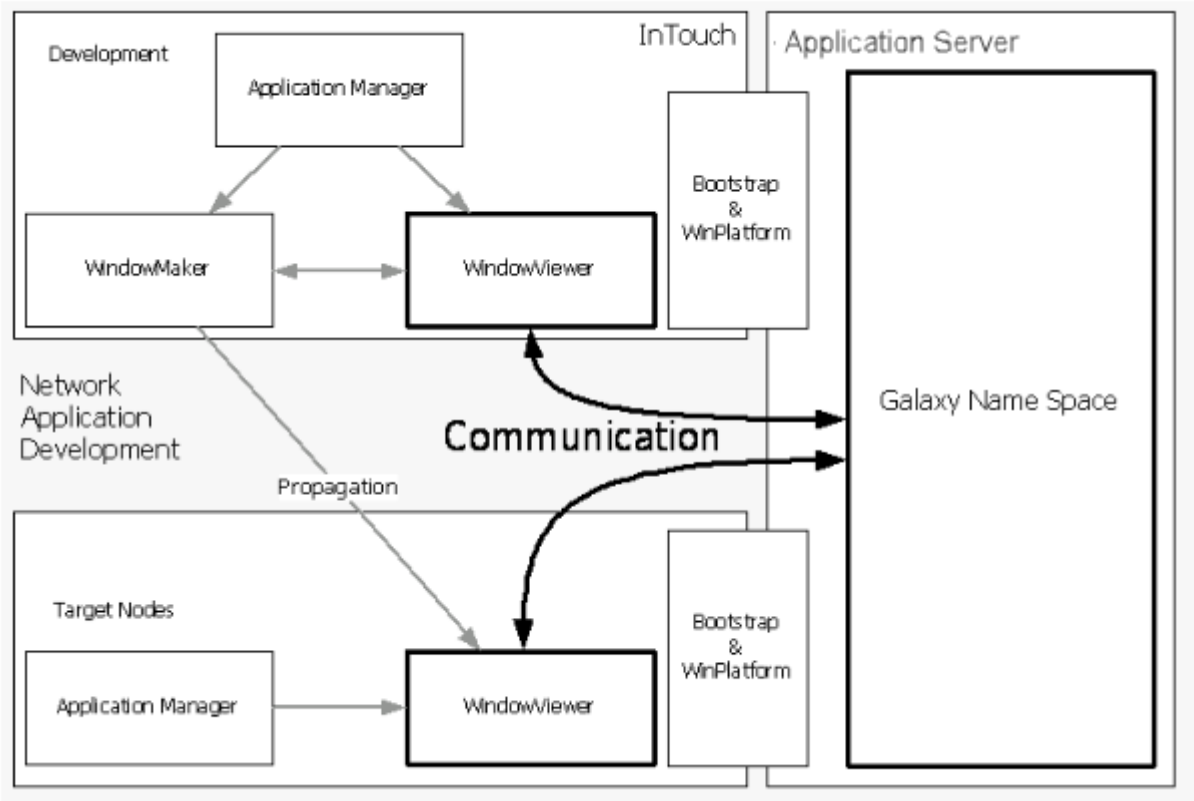
“应用程序服务器”可以存储与 InTouch 标记类似的产品数据，但它支持更多的数据类型与数组。可以提供与 InTouch 报警系统兼容的报警功能。“应用程序服务器”提供与 InTouch HMI 兼容的脚本语言，对 .NET 函数的支持有显著提高。要读取与写入 Galaxy 数据，需要在 InTouch 节点上安装“应用程序服务器”Bootstrap 组件。要浏览 Galaxy，需要安装“应用程序服务器”IDE 组件。

如需详细信息，请参阅 Application Server 文档。

Galaxy 内的通讯

ArchestrA 使您可以使用 Galaxy 范围的域名空间来包含与处理同生产相关的数据。它还可以从生产环境中运行 InTouch 的各个节点进行高级别的可视化与数据访问管理。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



比较独立、托管及发布的 InTouch 应用程序

独立、托管及发布的 InTouch 应用程序有不同之处，也有相似之处，具体如下表所述：

	独立的 InTouch 应用程序	**托管的 InTouch 应用程序**	**发布的 InTouch 应用程序**
创建应用程序	应用程序管理器	System Platform IDE • 新建应用程序 • 导入独立应用程序 • 导入 SmartSymbol	不可能
编辑应用程序	从“应用程序管理器”中启动 WindowMaker	从 IDE 中启动 WindowMaker	不可能
删除应用程序	从“应用程序管理器”中删除。	删除 InTouchViewApp 模板	从“应用程序管理器”中删除。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

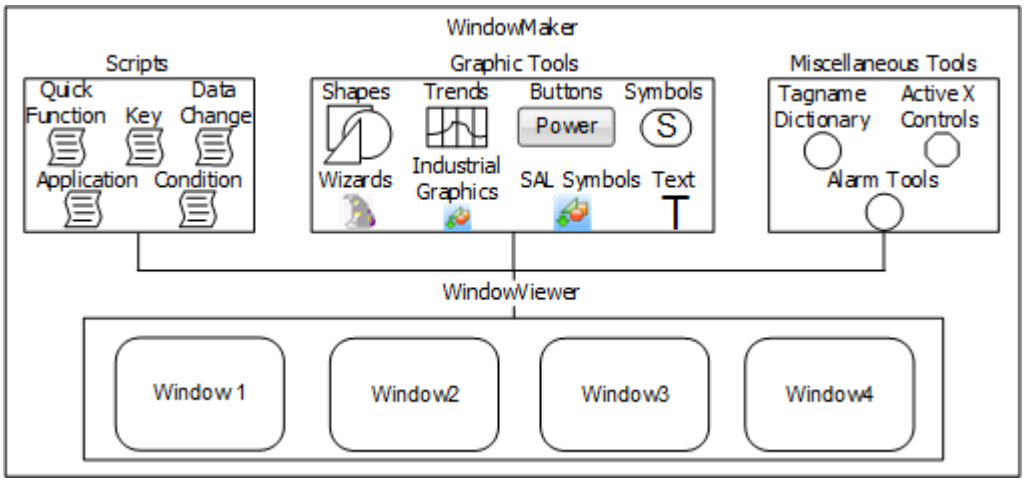
	独立的 InTouch 应用程序	托管的 InTouch 应用程序	发布的 InTouch 应用程序
支持工业图形，包括“符号向导”	是	所有操作都支持	是，但仅能查看，不能创建与编辑
对 DB Dump 与 DB Load 的支持	<1>是</1><2>, “</2><3>应用程序管理器</3><4>”</4><5>中的功能</5>	是，IDE 中的功能	<1>是</1><2>，“</2><3>应用程序管理器</3><4>”</4><5>中的功能</5>
按原始分辨率编辑应用程序时要求进行转换	是	否	是
分布式应用程序的管理	网络应用程序开发 (NAD)	System Platform IDE	网络应用程序开发 (NAD)
配置如何接受新版本的 InTouch 应用程序	在“应用程序管理器”中配置（网络应用程序开发）	在 WindowMaker 中配置	在“应用程序管理器”中配置（网络应用程序开发）
使用“快速切换”来测试应用程序	是	是	是
使用标记值与标记参数保留	是	是，还要求配置本地工作目录	是

构建应用程序

WindowMaker 提供图形工具、脚本语言及标记管理实用程序，来定义应用程序窗口中出现的对象的行为。通过使用 WindowMaker，可以创建标记来代表与窗口对象关联的数据点。生产过程中的数据最终与标记值关联。此标记数据可以在应用程序中用于监视报警、创建趋势，以及确定运行时应用程序的行为。

下图显示创建 InTouch HMI 应用程序的一些 WindowMaker 工具。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您可以使用许多种图形工具：从可以结合起来创建更复杂对象的简单形状，到附带预定义属性的标准工业图形。您可以根据触发机制创建不同类型的脚本。您还可以将预定义的 InTouch 函数插入脚本中。您可以使用“标记名字典”定义各种标记值阈值，确定标记处于正常还是报警状态。

运行应用程序

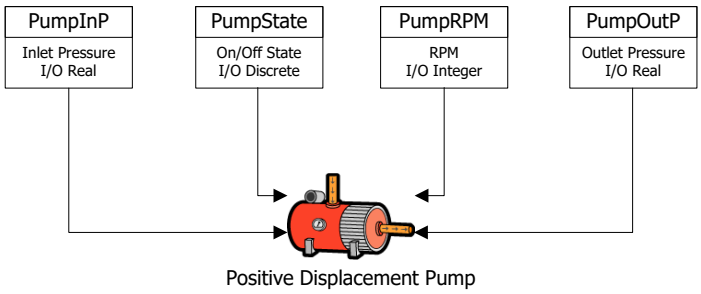
使用 WindowViewer 可以运行所有类型的 InTouch 应用程序。从 System Platform IDE 中部署托管的应用程序后，可以从“应用程序管理器”将其在 WindowViewer 中打开。在应用程序运行时，可以使用多种多样的运行时触发器来启动脚本。您可以配置 WindowViewer 将应用程序数据与报警存储到文件或 SQL Server 数据库。您可以通过要求操作员登录到 WindowViewer 并防止操作员对运行 WindowViewer 的计算机进行任何更改来实施安全机制。操作员可以通过选择 WindowViewer 菜单命令来启动与停止应用程序的历史记录功能。根据标记数据的存储与分发方式，可以将运行 WindowViewer 的计算机配置成客户端或服务器。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 3 标记

InTouch 人机界面 (HMI) 应用程序是生产环境中各种组件的图形化表示。工厂操作员使用这种图形界面来监视和管理生产过程。

下图显示一个泵的示例，它是生产过程中的一个组件。该泵有一些属性及关联的值。压力、RPM 及状态都是泵的属性，这些属性的值通过 HMI 进行监控。

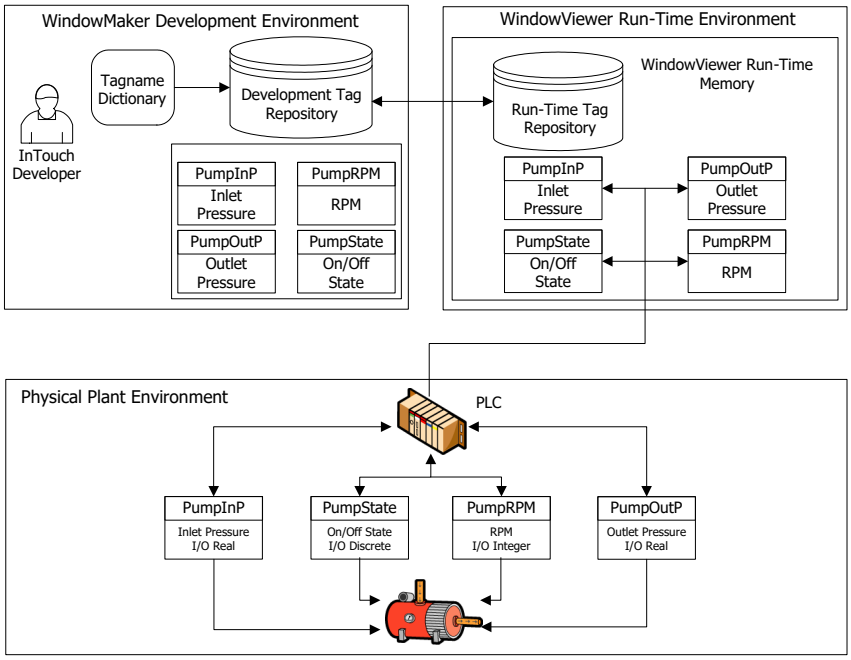


在 InTouch HMI 应用程序中，*标记*代表数据项。通过使用标记，可以将特定的组件属性当作数据项从生产环境中访问。在上图中，PumpState 标记指出泵是打开还是关闭的。对于生产环境中的各种组件，如果要在 InTouch 应用程序中监视或控制其属性，则可以为它们创建标记。

对于从生产组件中采集的不同类型的数据，可以使用不同类型的标记。例如，PumpState 标记返回 On/Off 布尔值，以指出泵是正在运行还是已经停止。对于要成为应用程序一部分的数据类型，需要指定适当类型的 InTouch 标记。

使用 InTouch 标记

首先开始创建一个 InTouch 应用程序。然后使用 WindowMaker 中的“标记名字典”工具为应用程序定义标记。下图显示 InTouch 开发与运行时环境。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您可以使用“标记名字典”来指定标记的名称与类型。对于某些类型的标记，在“标记名字典”中还有其它一些选项，可以用于指定标记的附加属性。例如，I/O 型标记包含一些附加选项，用于指定与远程数据源的连接。

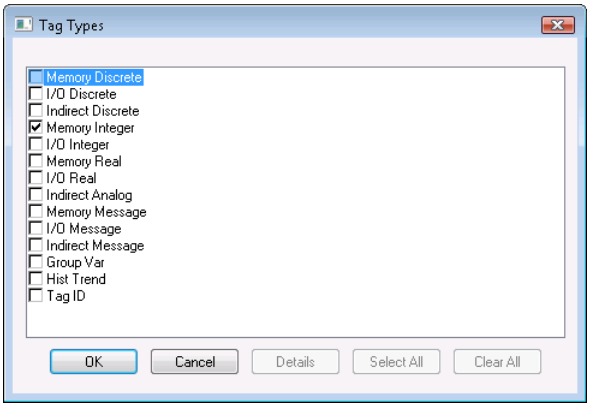
InTouch 应用程序在 WindowViewer 环境中运行。WindowViewer 启动应用程序时，它从开发储备库中读取标记，然后将它们放入运行时内存。

InTouch 应用程序使用动画链接或脚本与放入运行时内存中的标记进行通讯。InTouch 应用程序通过指定给标记的组件属性来跟踪当前值及其它状态信息。

InTouch 标记的类型

定义标记时，您根据所要关联的过程数据给标记指定特定的类型。例如，如果某个标记显示泵的 RPM，则将该标记的类型指定为整型标记。

在“标记名字典”中，使用**标记类型对话**框给创建的任何标记指定标记类型。



在指定标记类型之后，“标记名字典”列出所选标记类型的特定选项。

内存标记

内存标记定义 InTouch 应用程序中的内部系统常数与变量。例如，可以将某个内部常数定义为实数 3.414。在过程模拟中，内存标记可以充当计数器来控制后台 QuickScript 的动作。基于同该标记关联的计数，QuickScript 可以触发各种动画效果。内存标记也可以充当由其它程序来访问的计算变量。

基于同标记关联的过程数据，有四种类型的内存标记可以选择。

- **内存离散**

  内存离散标记与过程组件的状态属性关联。指定给内存离散标记的值是两个可能的布尔状态，如：

  - 0 或 1
  - 假或真
  - **开**或**关**
  - 高或低

- **内存整型（模拟）**

  您可以将内存整型标记指定为 -2,147,483,648 与 2,147,483,647 之间的 32 位带符号整数。

- **内存实型（模拟）**

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您可以将内存实型标记指定为 -3.4 x 1038 与 3.4 x 1038 之间的十进制浮点数。在比较两个实型标记值时，两个实型标记的差异应大于 FLT_EPSILON（值 1.19209290E-07F，十进制 0.0000001192092896）。所有浮点都按 64 位精度进行计算，但结果存储为 32 位十进制数。如需有关实数最大精度的详细信息，请参阅 IEEE 十进制单位。

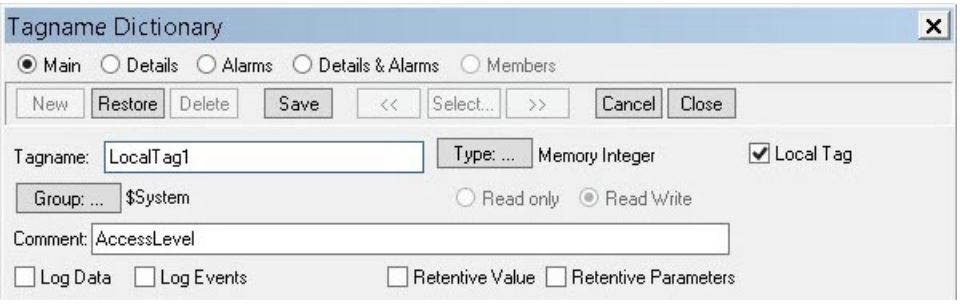
- 内存消息

  您可以将内存消息型标记指定为长度不超过 131 个单字节字符的文本字符串。

本地标记

用户可以通过本地标记来创建每个会话的内存标记以便在 Web 客户端中使用。例如，用户可以配置用于导航的本地内存标记。在运行时，当用户选择链接到该标记的导航符号时，每个会话都会独立运行。

注意：.Dot 字段不支持本地标记。



创建本地标记

1. 创建内存标记，然后选中本地标记复选框。配置其它标记属性。如需有关创建新标记的详细信息，请参阅创建新标记
2. 选择保存。

I/O 标记

I/O 标记从外部数据源中读取 InTouch 应用程序数据，或向其中写入数据。外部数据包括来自可编程控制器、过程计算机以及网络节点的输入与输出。I/O 标记数据值可以通过以下协议远程访问：

- Microsoft“动态数据交换”（Dynamic Data Exchange，简称 DDE）
- SuiteLink™

运行时内存中的 I/O 标记的值发生改变时，InTouch HMI 更新远程应用程序。反之，只要远程应用程序中相应数据项的值发生改变，便会更新 InTouch 中的 I/O 标记值。

根据与标记关联的过程数据，InTouch HMI 提供四种类型的 I/O 标记。这四种类型的 I/O 标记与内存标记类型非常类似。

- **I/O 离散**

  I/O 离散标记与组件过程属性关联，这些属性的值使用两个可能的状态表示，如：

  - 0 或 1
  - 假或真
  - 开或关
  - 高或低

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- **I/O 整型**（模拟）

  I/O 整型标记可以指定为 -2,147,483,648 与 2,147,483,647 之间的 32 位带符号整数。

- **I/O 实型**（模拟）

  I/O 实型标记可以指定为 -3.4 x 1038 与 3.4 x 1038 之间的十进制浮点数。在比较两个实型标记值时，两个实型标记的差异应大于 FLT_EPSILON（值 1.19209290E-07F，十进制 0.0000001192092896）。所有 I/O 实型标记浮点都按 64 位精度进行计算，但结果存储为 32 位数。如需有关 I/O 实数最大精度的详细信息，请参阅 IEEE 十进制单位。

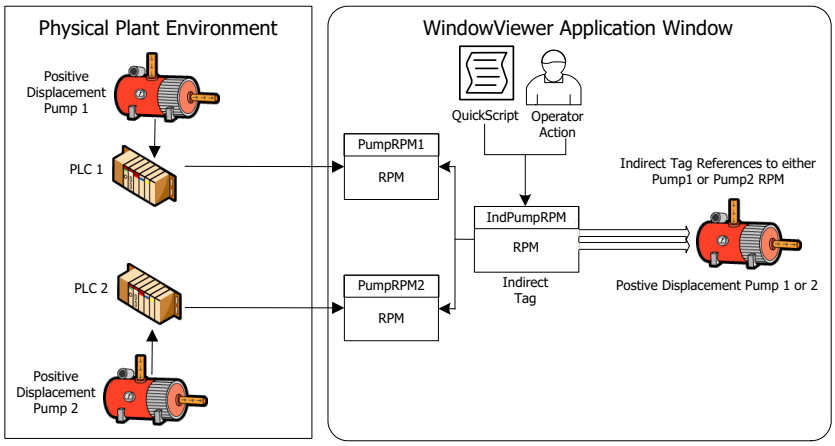
- **I/O 消息**

  I/O 消息标记可以指定为最大长度不超过 131 个单字节字符的文本字符串。

间接标记

<1>间接标记用作其它标记的</1><2>“</2><3>指针</3><4>”。</4> 例如，您可以创建单个 InTouch 窗口，使用间接标记去显示来自多个不同组的标记的数据。

下图显示一个应用程序窗口示例，该窗口能够显示多个泵。您可以在一个窗口中使用间接标记来显示与单独的泵关联的不同源标记的值，而不必为每个泵创建单独的窗口。



QuickScript 或操作员动作将间接标记指向源标记。例如，以下脚本语句根据 PumpNo 标记的值，将两个 PumpRPM 标记指定给 **IndPumpRPM** 间接模拟标记。

```
IF PumpNo == 1 THEN
   IndPumpRPM.Name = "PumpRPM1";
   ELSE
   IndPumpRPM.Name = "PumpRPM2";
ENDIF;
```

将某个间接标记的值赋给另一个源标记时，该间接标记的作用就好像它自己是源标记。同原始的源标记和间接重复标记关联的这些值会一起进行同步。如果源标记的值发生改变，间接标记便会反映出这个变化。如果间接标记的值发生改变，源标记的值也会相应地改变。

您可以使用离散、模拟以及消息型的间接标记。这三种类型的间接标记类似于内存与 I/O 类型的标记。

如需有关间接标记的详细信息，请参阅定义间接标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

其它标记

您可以使用针对特定、有限用途而设计的其它 InTouch 标记类型。这些标记可用于创建动态报警显示、创建历史趋势以及更改指定给历史趋势笔的标记。

历史趋势标记

<1>“</1>历史趋势<2>”</2>标记可用于引用历史趋势图<3>。</3> 与历史趋势关联的所有“点域”都可以应用于“历史趋势”标记。

如需有关定义与使用“历史趋势”标记的详细信息，请参阅绘制标记数据的趋势。

“标记 ID”标记

“标记 ID”标记从标记值绘制在 InTouch 历史趋势图上的标记中检索信息。通常，“标记 ID”标记可用于显示指定给特定趋势笔的标记的名称，或更改给趋势笔指定的标记。

您可以在 QuickScript 中包含一条语句<1>, </1>使用<2>“</2>标记 ID<3>”</3>类型的标记将某个新的标记指定给任何历史趋势中的笔<4>。</4> 例如，以下 QuickScript 语句更改与历史趋势笔关联的标记：

```
HistTrend.Pen1=MyLoggedTag.TagID;
```

此 QuickScript 运行时，历史趋势的 Pen1 开始绘制 MyLoggedTag 的历史记录数据的趋势。

如需有关定义与使用“历史趋势”标记的详细信息，请参阅使用历史趋势向导。

SuperTag

SuperTag 是包含一组相关标记的模板。例如，您可以创建一个 SuperTag 模板，其中包含一组指定给某个泵所有属性的标记。

在生产过程中有完全相同的设备时使用 SuperTag。您可以将 SuperTag 模板的实例指定给每一个完全相同的过程项，而不必为每件设备都创建一组标记。

如需有关 SuperTag 的详细信息，请参阅定义可复用的标记结构。

废弃标记

通过使用“组变量”标记，可以利用 InTouch 的标准报警系统来创建动态报警显示、动态磁盘日志、动态打印。保留“组变量”标记仅为了与使用 InTouch 7.11 及更早版本开发的应用程序保持后向兼容性。请勿在使用比 InTouch 7.11 更新的版本开发的应用程序中使用“组变量”标记。

系统标记

通过使用系统标记，可以提供与系统相关的信息和用于 InTouch 脚本的标准函数，例如日期与时间。系统标记是所有应用程序的一部分。您也可以在脚本中使用系统标记来管理正在运行的程序，如：

- 监视与管理应用程序安全性。
- 建立与远程节点的 I/O 通讯。
- 检测何时发生报警。
- 启动与停止历史记录。
- 使用 NAD 将应用程序更接到新版本。

在“标记名字典”中，系统标记通过将美元号 ($) 用作标记名的第一个字符进行标识。系统标记无法删除。您只能更改与系统标记关联的注释。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

系统标记引用

下表介绍 InTouch 系统标记：

系统标记	用途	详细信息
$AccessLevel	只读整型标记，指定与当前登录的操作员关联的访问级别。 此信息可以在动画链接或脚本中用于控制操作员对特定 InTouch 功能的访问。	请参阅保护 InTouch 安全
$ApplicationChanged	只读离散标记，指出 NAD 环境中的主应用程序是否已发生更改。	请参阅分发应用程序
$ApplicationVersion	只读实型标记，指定 WindowViewer 中运行的应用程序的当前版本号。	请参阅分发应用程序
$ChangePassword	只写离散标记，设置为 1 时显示改变口令对话框。	请参阅保护 InTouch 安全
$ConfigureUsers	只写离散标记，显示通用的配置用户对话框，以编辑安全性用户名列表。	请参阅保护 InTouch 安全
$Date	只读整型标记，显示自 1970 年 1 月 1 日以来已过天数的整数部分。	请参阅内置函数
$DateString	只读消息标记，按照 Windows 区域和语言选项对话框中指定的格式显示日期。	请参阅内置函数

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

系统标记	用途	详细信息
$DateTime	只读实型标记，显示自 1970 年 1 月 1 日以来已过天数的小数部分。	请参阅内置函数
$Day	只读整型标记，显示本月的当前日期值 (1-31)。	请参阅内置函数
$False	离散只读标记在表达式中返回 FALSE 值。 从较早版本的 InTouch 应用程序更新到当前版本时，$False 系统标记用于替换任何废弃系统标记的实例。	无进一步信息。
$HistoricalLogging	可读写离散标记，用于在运行 InTouch 应用程序时开始与停止历史记录功能。	请参阅记录标记值。
$Hour	只读整型标记，按 0 到 23 的形式显示当前小时值。	请参阅内置函数
$InactivityTimeout	只读离散标记，表示用户不活动期限已过。设置为 1 时，不活动期限已过，用户会自动从 WindowViewer 注销。	请参阅保护 InTouch 安全
$InactivityWarning	只读离散标记，表示不活动状态警告期限已过。$InactivityWarning 的值可用于向操作员发出不活动状态警告。	请参阅保护 InTouch 安全
$LogicRunning	可读写离散标记，用于启动与停止应用程序脚本。	请参阅保护 InTouch 安全
$Minute	只读整型标记，显示当前的分钟值 (0-59)。	请参阅内置函数

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

系统标记	用途	详细信息
$Month	只读整型标记，显示当前月份 (1-12)。	请参阅内置函数
$Msec	只读整型标记，显示当前毫秒值 (0-999)。	请参阅内置函数
$NewAlarm	可读写离散标记，表示发生新本地报警的时间。	请参阅在运行时控制标记与组的报警属性
$ObjHor	只读整型标记，显示屏幕上所选对象中心的水平像素位置。	请参阅《AVEVA™ InTouch HMI 应用程序开发》中的动画对象
$ObjVer	只读整型标记，显示屏幕上所选对象中心的垂直像素位置。	请参阅《AVEVA™ InTouch HMI 应用程序开发》中的动画对象
$Operator	只读消息标记，显示登录到 InTouch 应用程序的操作员的姓名。	请参阅保护 InTouch 安全
$OperatorDomain	只读消息标记，包含应用程序在使用基于操作系统的安全性进行登录时所指定的域或机器名。	请参阅保护 InTouch 安全
$OperatorDomainEntered	只写消息标记指定了操作员登录尝试 InTouch 应用程序的域。 您只有在将值指定给 $PasswordEntered 系统标记后，登录尝试才会启动。	请参阅保护 InTouch 安全

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

系统标记	用途	详细信息
$OperatorEntered	可读写消息标记，指定尝试登录 InTouch 应用程序的操作员用户帐户名。 您只有在将值指定给 $PasswordEntered 系统标记后，登录尝试才会启动。	请参阅保护 InTouch 安全
$OperatorName	只读消息标记，在使用基于操作系统或 ArchestrA**(R)** 的身份验证时显示操作员的全名。	请参阅保护 InTouch 安全
$PasswordEntered	只写消息标记，指定尝试登录 InTouch 应用程序的操作员的口令。 将值写入此标记时，使用 $OperatorDomainEntered、$OperatorEntered 及 $PasswordEntered 系统标记尝试进行登录。	请参阅保护 InTouch 安全
$Second	只读整型标记，显示当前秒值 (0-59)。	请参阅内置函数
$StartDdeConversations	可读写离散标记，用于在运行时启动未初始化的对话。	请参阅使用 I/O 进行数据访问。
$System	只读标记，用于确定根报警组。	请参阅*关于报警与事件*
$Time	只读整型标记，显示自当天午夜起经过的毫秒数。	请参阅内置函数
$TimeString	只读消息标记，按照 Windows 区域和语言选项对话框中指定的格式显示当前时间。	请参阅内置函数

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

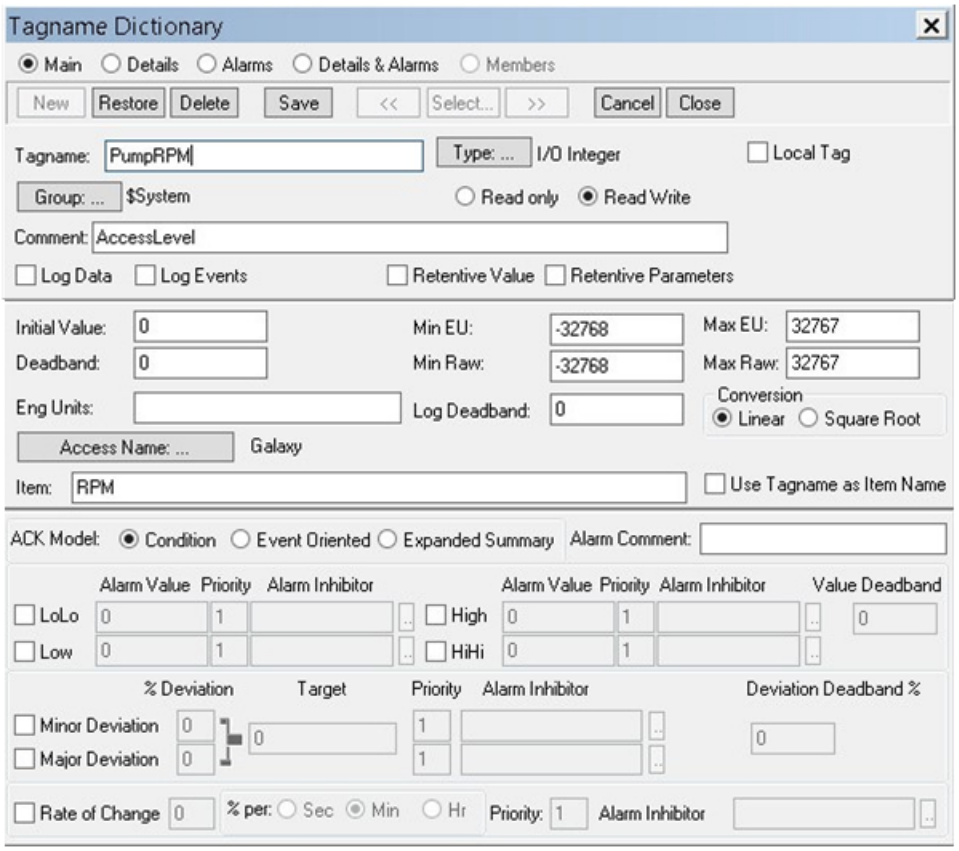
系统标记	用途	详细信息
$VerifiedUserName	只读消息标记，包含经验证的用户的全名，或包含空值。	请参阅保护 InTouch 安全
$Year	只读整型标记，按四位数字的格式显示当前年份。	请参阅内置函数

标记属性

每个 InTouch 标记类型都有一组属性，用于描述与标记关联的数据特性。与 InTouch 标记关联的四种主要数据类型有：

- 离散值
- 整数
- 实数
- 文本消息

在使用<1>“</1>标记名字典<2>”</2>创建标记时<3>，</3>所有的标记属性都会赋一个初始值<4>。</4> 下图显示 I/O 整型标记的属性。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从“标记名字典”中设置标记属性的初始值之后，可以使用点域来动态更改大多数标记属性。点域确定 InTouch 应用程序运行时可以由脚本监视或修改的标记属性。在脚本中可以将点域附加到标记的名称。

如需有关使用点域来动态更改标记属性的详细信息，请参阅使用标记点域来查看或更改标记属性。

内存标记属性

下表列出四种内存标记的属性。作为<1>“</1>标记名字典<2>”</2>的选项<3>，</3>可以选择或修改每种属性<4>。</4> 如需有关详细信息，请参阅创建新标记。

标记属性	离散	整型	消息	实型
% 偏差		•		•
%		•		•
确认模型	•	•		•
报警注释	•	•	•	•
报警组	•	•	•	•
报警约束	•	•		•
报警状态	•			
报警值		•		•
注释	•	•	•	•
死区		•		•
工程单位		•		•
偏差死区百分比		•		•
高		•		•
上上限		•		•
初始值	•	•	•	•
记录数据	•	•		•
记录死区		•		•
记录事件	•	•	•	•
下限		•		•
下下限		•		•
最大长度			•	
主偏差		•		•
最大值		•		•

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

标记属性	离散	整型	消息	实型
最小值		•		•
副偏差		•		•
关闭消息	•			
打开消息	•			
优先级	•	•	•	•
变化率		•		•
只读	•	•	•	•
读写	•	•	•	•
保留参数		•		•
保留值	•	•	•	•
目标		•		•
值死区		•		•

I/O 标记属性

与内存标记类似<1>，</1>作为<2>“</2>标记名字典<3>”</3>的选项<4>，</4>可以选择或修改 I/O 标记属性<5>。</5> 如需有关详细信息，请参阅创建新标记。

标记属性	离散	整型	消息	实型
% 偏差		•		•
%		•		•
访问名	•	•	•	•
确认模型	•	•		•
报警注释	•	•	•	•
报警组	•	•	•	•
报警约束	•	•		•
报警状态	•			
报警值		•		•
注释	•	•	•	•
转换		•		•
死区		•		•

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

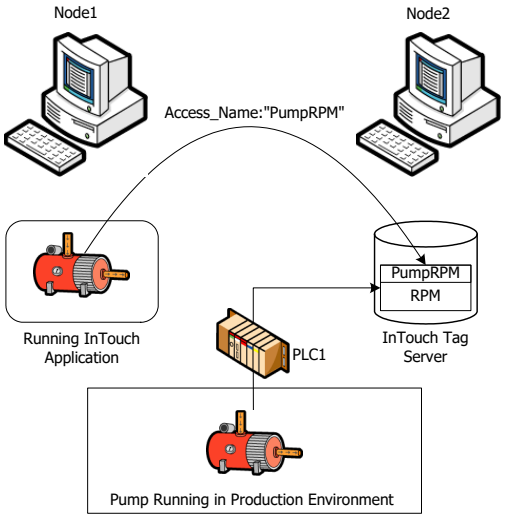
标记属性	离散	整型	消息	实型
偏差死区百分比		•		•
工程单位		•		•
高		•		•
上上限		•		•
初始值	•	•	•	•
输入转换	•			
项目	•	•	•	•
记录数据	•	•		•
记录死区		•		•
记录事件	•	•	•	•
下限		•		•
下下限		•		•
最大长度			•	
主偏差		•		•
最大工程单位		•		•
最大原始数据		•		•
最大值		•		•
最小工程单位		•		•
最小原始数据		•		•
最小值		•		•
副偏差		•		•
关闭消息	•			
打开消息	•			
优先级	•	•	•	•
变化率		•		•
只读	•	•	•	•
读写	•	•	•	•
保留参数		•		•

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

标记属性	离散	整型	消息	实型
保留值	•	•	•	•
平方根转换		•		•
目标		•		•
将标记名用作项目名	•	•	•	•
值死区		•		•

远程标记引用

您可以创建分布式 InTouch 应用程序，让标记服务器在运行 InTouch 应用程序的节点之外的单独节点上运行。下图显示一个远程引用 PumpRPM 标记的 InTouch 应用程序，该标记来自另一个节点上运行的标记服务器。



创建 InTouch 应用程序引用远程节点上的标记有两种方法：

- 将 I/O 标记关联到“访问名”，由访问名将远程服务器确定为标记源。如需有关为 I/O 标记定义“访问名”的详细信息，请参阅设置访问名。
- 使用直接指向该标记的远程引用。例如，PLC1:PumpRPM。

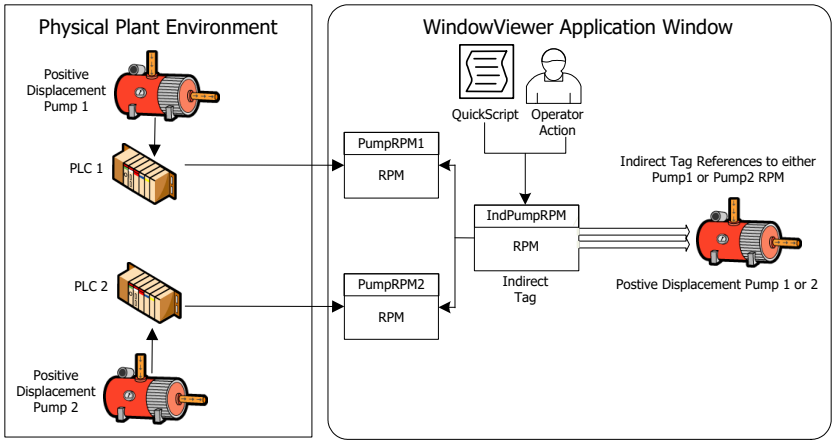
如需有关详细信息，请参阅通过远程引用访问 I/O 数据。

定义间接标记

使用间接标记，可以创建包含可显示多个标记值的窗口对象的应用程序。

下图显示应用程序窗口中的一个泵对象。该泵对象根据从间接标记中设置的值来代表两个可能的过程泵。QuickScript 或操作员动作负责选择与间接标记关联的源标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



间接标记可以最大限度缩短开发时间。由于单个窗口对象可以代表生产环境中运行的多个过程，因此可以创建更少的应用程序窗口。

在脚本中使用间接标记

您可以使用脚本将输入源标记赋值给间接标记。通过将源标记的名称指定给间接标记的 .**Name** 点域，可以将输入源标记指定给该间接标记。

例如，如果创建一个间接模拟标记 **IndPumpRPM**，则通过使用与下例类似的脚本语句，可以将两个 **PumpRPM** 源标记指定给该间接模拟标记：

```
IF PumpNo == 1 THEN
   IndPumpRPM.Name = "PumpRPM1";
   ELSE
   IndPumpRPM.Name = "PumpRPM2";
ENDIF;
```

间接标记赋值脚本可以由应用程序事件或操作员动作（如选择窗口按钮）触发。

将另一个源标记赋给某个间接标记时，该间接标记的作用就好像它自己是源标记。如果源标记的值发生改变，间接标记便会反映出这个变化。如果间接标记的值发生改变，源标记的值也会相应地改变。

由于间接标记的 .Name 点域是一个简单的字符串，因此可以在运行时动态指定间接标记目标。例如，如果创建一个“数据改变”QuickScript，每次 **Number** 标记的值发生改变时该脚本都会运行，则指定给间接 **IndPumpRPM** 标记的源标记也会相应地改变：

```
IndPumpRPM.Name = "PumpRPM" + Text(Number, "#" );
```

此脚本运行时，模拟标记 **Number** 的值转换为文本，并附加到字符串 **PumpRPM** 上。如果 **Number** 等于 **1**，则这将间接标记 **IndPumpRPM** 的名称设置为 **PumpRPM1**。

间接模拟型标记同时可用于整型与实型标记。只要标记类型相同，间接标记可以映射到任何其它标记。

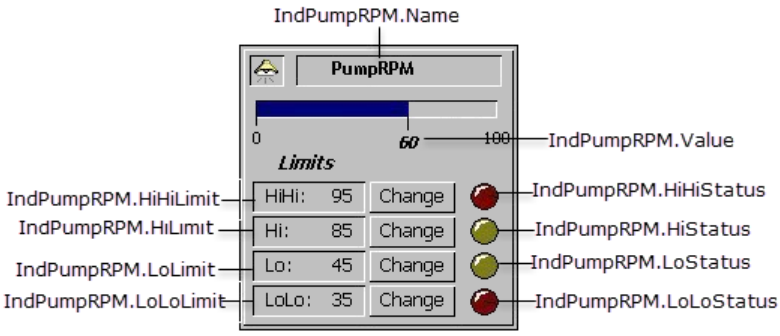
您也可以将保留属性指定给间接标记。利用这种保留功能，当应用程序再次启动时，间接标记会保留最新的标记赋值情况。

为本地标记使用间接标记

间接标记通常用于本地<1>“</1>标记名字典<2>”</2>中定义的标记<3>。</3> 间接本地标记可用于创建显示本地标记的多个属性的可视化对象。例如，您可以在应用程序窗口中创建一个面板。该面板包含一些可选项，这些可选项链接到指定给不同点域的间接本地标记。在下例中，操作员修改链接到本地间接标记 **PumpRPM** 的点域报警限。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

下图显示一个面板，该面板包含一些指向泵 RPM 报警限的动画链接。间接标记将不同本地标记的属性指定给报警限面板。



要将面板重定向到适当的标记，请在 QuickScript 中包含一条语句。

```
Indirect_tag_name.Name = "tag_name";
```

在这个示例脚本中，tag_name 是本地“标记名字典”中定义的实际标记的名称。该脚本运行时，与这个本地标记关联的所有点域值都可以通过间接标记由应用程序对象进行访问。

对远程引用使用间接标记

远程间接标记引用与本地标记引用不同。远程引用的语法是：

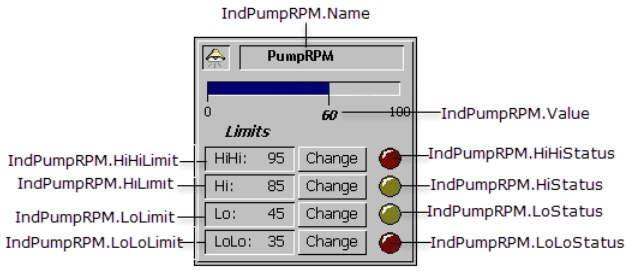
AccessName:Item

其中

- *AccessName* 是任何有效的 InTouch“访问名”。
- *Item* 是任何有效的“项目名”，此“项目名”受“访问名”定义中指定的“I/O 服务器”支持。

使用远程引用时，服务器给客户端返回一个值，而不是一个标记结构。该值包含时间标签与质量标签。因此，除与值、时间以及质量相关的标记点域之外，指定给远程引用的间接标记无法访问任何其它标记点域。例如，间接标记无法通过远程引用来访问标记属性以指定报警限。

一个可能的解决方案是创建包含一组间接标记的面板。下图显示一个可修改泵的报警限的面板。



在本例中，面板使用 10 个间接标记，这些标记都与隐式 **.Value** 点域关联。报警面板重定向到远程 InTouch 节点 TagServer1 上的远程引用标记 IndPumpRPM。InTouch“访问名”配置如下：

访问名：	TagSource1
节点名：	TagServer1
应用程序名称：	查看
主题名：	标记名

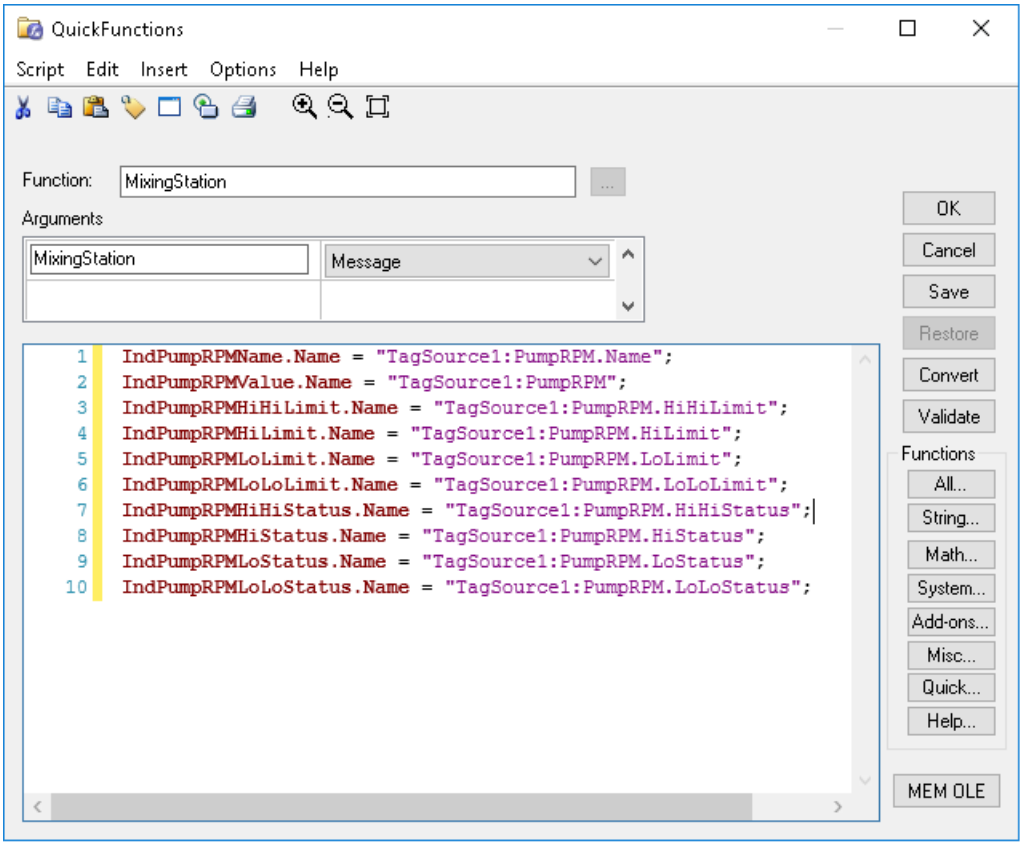
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要将该面板重定向到远程引用标记 PumpRPM，请运行以下 QuickScript：

```
IndPumpRPMName.Name = "TagSource1:PumpRPM.Name";
IndPumpRPMValue.Name = "TagSource1:PumpRPM";
IndPumpRPMHiHiLimit.Name = "TagSource1:PumpRPM.HiHiLimit";
IndPumpRPMHiLimit.Name = "TagSource1:PumpRPM.HiLimit";
IndPumpRPMLoLimit.Name = "TagSource1:PumpRPM.LoLimit";
IndPumpRPMLoLoLimit.Name = "TagSource1:PumpRPM.LoLoLimit";
IndPumpRPMHiHiStatus.Name = "TagSource1:PumpRPM.HiHiStatus";
IndPumpRPMHiStatus.Name = "TagSource1:PumpRPM.HiStatus";
IndPumpRPMLoStatus.Name = "TagSource1:PumpRPM.LoStatus";
IndPumpRPMLoLoStatus.Name = "TagSource1:PumpRPM.LoLoStatus";
```

每次重定向面板时都必须运行此脚本。另一个解决方案是创建一个 InTouch QuickFunction，允许您编写单个脚本并将远程引用的名称传递给它。通过使用调用相同 QuickFunction 的多个面板，可以减少脚本的编码量。

例如，通过使用一组类似的脚本命令，可以定义一个 QuickFunction **RedirectAlarmFacePlate**：



您可以调用 **RedirectAlarmFacePlate** 函数来处理整个重定向过程。要完成这项工作，必须使用另一个 InTouch QuickScript 来调用该函数。例如：

```
CALL RedirectAlarmFacePlate ("TagSource1:PumpRPM");
```

定义可复用的标记结构

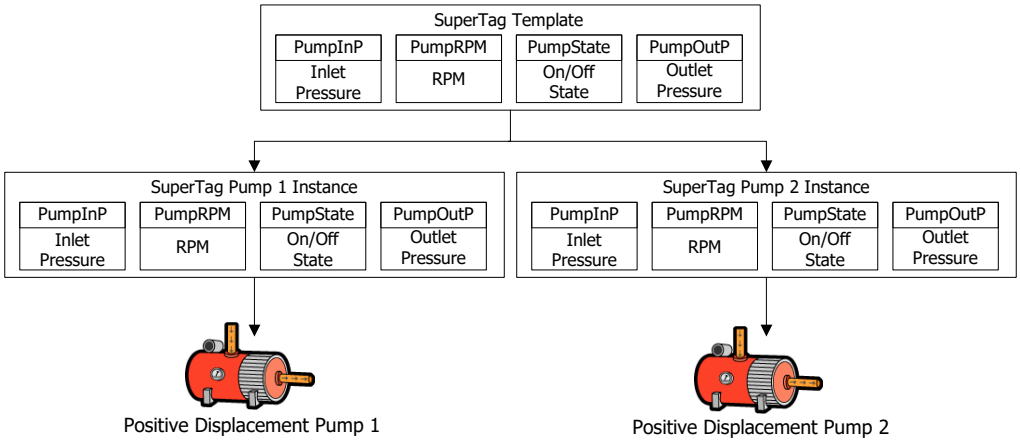
SuperTag 是一组相关标记的模板。属于 SuperTag 模板的标记与制造过程中某个组件的公共属性关联。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

SuperTag 可以节省开发时间。您可以复制一个 SuperTag 模板，然后为具有相同属性的所有过程组件创建单独的实例，而不必为制造过程中的每个组件都创建一组标记。

SuperTag 模板

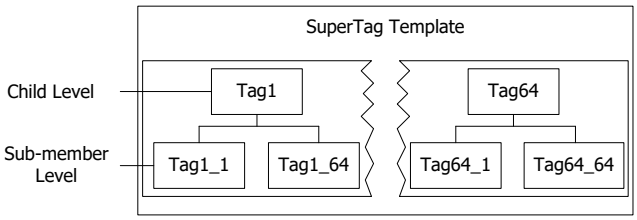
下图显示一个 SuperTag 模板，其中包含一组与泵数据关联的相关标记。您可以复制该模板，以便为过程中的相同泵创建 SuperTag 实例。



属于 SuperTag 实例的所有标记的行为都与普通标记完全相同。成员标记可以指定为 InTouch 离散、整型、实型及消息等数据类型。这些标记支持趋势、报警以及所有的标记点域。

SuperTag 模板可以按两个嵌套级别来组织自己的成员标记。SuperTag 模板最多可以包含 64 个内嵌的子标记。每个子标记最多可以包含 64 个子成员标记。这可以在 SuperTag 模板中总共提供 4095 个标记。

下图显示标记在 SuperTag 模板中的组织方式。



一个 SuperTag 父模板嵌入另一个 SuperTag 模板时，内嵌的标记就变成了子成员。

创建 SuperTag 父模板之后，“标记名字典”在**标记类型对话**框中将它列为一种标记类型。在创建新的标记时，可以立即选择该模板作为标记类型。您不必重新启动 WindowMaker，便可以定义一些标记来使用新创建的 SuperTag 类型。

保存 SuperTag 模板

缺省条件下，所有 SuperTag 模板都保存在 **C:\ProgramData\Wonderware\InTouch** 文件夹中的 Supertag.dat 文件内。您可以选择将 SuperTag 模板保存在其它位置。

指定用于保存 SuperTag 模板的位置

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择 **WindowMaker**。

   此时出现 WindowMaker 配置屏幕。
3. 在 **SuperTag 路径**文本框中，指定用于保存 SuperTag 模板的位置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 选中在应用程序目录中存储 **SuperTag** 数据复选框，以将 SuperTag 数据存储在应用程序目录中。这将覆盖在 SuperTag 模板路径字段中指定的路径。

   注意：如果要导出 SuperTag 模板，则可以选中此复选框并导出相应的 InTouch 应用程序。有关更多信息，请参阅导出 InTouch 应用程序以用作模板。

导入 SuperTag

要在 InTouch 应用程序迁移期间导入 SuperTag，请参阅在 InTouch 应用程序迁移期间导入资产。

管理 SuperTag 模板与成员标记

随着 InTouch HMI 2023 的发行，Template Maker 实用程序已被弃用。您可以使用画布上的 SuperTag 窗口来管理 SuperTag。这样便可以创建、编辑以及删除 SuperTag 模板及其成员标记。创建 SuperTag 模板之后，“标记名字典”在**标记类型对话**框中将它与间接标记形式的模板列为标记类型。

您可以随时修改 SuperTag 模板或成员标记。从模板衍生的现有 SuperTag 实例不会继承对该模板所作的更改。所有新的实例都会继承修改后模板的更改。

您可以删除整个 SuperTag 模板，也可以删除属于该模板的所选成员标记。已删除的模板不再列在“标记名字典”的**标记类型对话**框中。

创建 SuperTag 模板

1. 打开 **SuperTag** 窗格。
2. 右键单击 **SuperTag** 模板，然后选择新建 **SuperTag**。

   此时 SuperTag 模板创建完成。您可以根据需要重命名 SuperTag 模板。

编辑现有的 SuperTag 模板或成员标记

注意：我们建议您使用 SuperTag 窗格来管理 SuperTag。

1. 打开 **SuperTag** 窗格。
2. 导航到要编辑的 SuperTag 模板名或成员标记。
3. 使用鼠标右键单击 SuperTag 模板，然后选择**编辑 SuperTag**。

   或者，双击 SuperTag 模板。

   此时在画布上出现 SuperTag 窗口。
4. 您可以重命名 SuperTag 模板或修改描述。
5. 根据需要进行更改。
6. 选择确定。

删除 SuperTag 模板或成员标记

1. 打开 **SuperTag** 窗格。
2. 导航到要删除的 SuperTag 模板名或成员标记。
3. 使用鼠标右键单击 SuperTag 模板，然后选择删除。

   此时出现一条消息，要求确认是否删除所选的项目。
4. 选择是以删除所选的模板。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要点：SuperTag 模板实例的成员标记无法删除。例如，如果 PumpRPM 是 SuperTag 模板 TankPump 的成员标记，则无法从任何 TankPump 实例中删除它。您只能从 SuperTag 模板中删除标记。

SuperTag 实例

SuperTag 模板与模板实例不同。实例是在 InTouch HMI 应用程序中实际实施的 SuperTag 模板版本。

模板与实例最主要的区别是父模板名将替换为实例标记名。子模板名与子成员标记不发生改变。

创建 SuperTag 实例

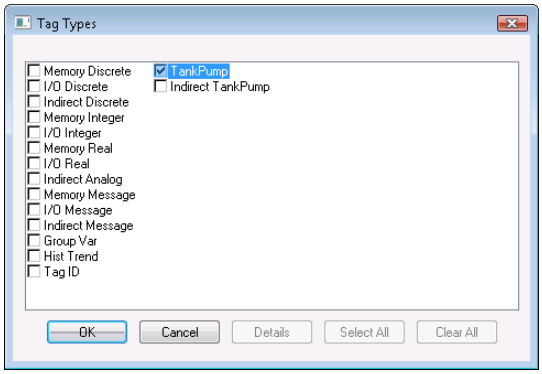
您可以使用“标记名字典”来创建 SuperTag 实例。定义新的 SuperTag 实例时，“标记名字典”自动创建所有的成员标记与子成员标记。

从模板创建 SuperTag 实例

1. 在 **SuperTag** 窗格中，选择要为其创建实例的 SuperTag 模板。
2. 使用鼠标右键单击 SuperTag 模板，然后选择新建实例。

   此时即会创建新的 SuperTag 实例。
3. 双击这个新实例可打开“标记名字典”。
4. 在**标记名**框中，为新的 SuperTag 实例输入所需的名称。

   SuperTag 实例名最多可包含 32 个字符。实例名与普通的 InTouch 标记遵循相同的命名规则。
5. 选择**类型**以显示**标记名类型**对话框。
6. 从列表中选择 SuperTag 模板名。有关如何创建 SuperTag 的详细信息，请参阅管理 SuperTag 模板与成员标记。



7. 选择确定。此时**标记名字典**对话框展开，以显示附加选项。

   在**标记名**框中输入的新标记将变成所选 SuperTag 模板中所有成员标记的父对象。
8. 设置标记的属性。执行以下操作：
   a. 在**成员列表**框中，从 SuperTag 模板列表中选择一个标记。
   b. 从**数据访问**中选择内存或 **I/O**，以显示相应的内存或 I/O 详细信息对话框。
   c. 输入详细信息，方法与标准 InTouch 标记相同。
   d. 从列表中选择其余的成员标记并配置它们。
9. 为 SuperTag 实例的成员标记指定所有的详细资料之后，选择**关闭**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

复制 SuperTag 实例

您可以使用“标记名字典”来复制现有实例。在复制实例之后，这些标记立即就可以在动画链接与 InTouch QuickScript 中使用。

从“标记名字典”中复制 SuperTag 实例

1. 打开“标记名字典”。
2. 单击**选择**以显示**选择标记**对话框，列出为应用程序定义的一列标记。
3. 从列表中选择要用作新实例的模板的 SuperTag 实例。
4. 选择确定。

   此时所选模板的名称出现在**标记名**框中。
5. 选择新建。此时出现一条消息，要求确认是否复制 SuperTag 实例。
6. 选择是。此时出现**输入名称对话**框，提示输入新 SuperTag 的名称。

   根据标准的标记命名惯例输入最多包含 32 个字符的名称。
7. 选择确定。新的 SuperTag 实例即会出现在“标记名字典”中。
8. 根据需要来编辑成员标记，方法与普通 InTouch 标记相同。
9. 选择**关闭**。

将标记添加到 SuperTag 实例

您可以使用“标记名字典”将标记添加为现有 SuperTag 实例的成员。

添加标记时，输入 SuperTag 实例的准确名称，后跟反斜杠 (\) 分隔符以及新成员标记的名称。

例如：

```
Pump_8\PumpSTS
```

注意：如果您计划使用批量导入实用程序将标记从 InTouch HMI 应用程序迁移到 Application Server，请参阅使用批量导入实用程序导入 SuperTag 以获取有关替换标准反斜杠分隔符的详细信息。

将标记添加到 SuperTag 实例

1. 打开“标记名字典”。
2. 要将标记添加到 SuperTag 实例，请执行以下操作：
   a. 选择新建。
   b. 在**标记名**框中，输入 SuperTag 实例的准确名称，后跟反斜杠 (\) 分隔符和新成员标记的名称。
   c. 选择**类型**。
   d. 为要添加到实例的新成员标记选择标记类型。
   e. 选择确定。此时出现该成员标记类型的详细信息对话框。
   f. 输入所需的详细信息，方法与普通 InTouch 标记相同。
3. 选择保存。
4. 单击**选择**。
5. 选择添加了成员标记的 SuperTag 实例。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 选择确定。

   在成员列表框中，属于该 SuperTag 模板的所有成员标记都会列出。

   此时新成员标记出现在列表中。

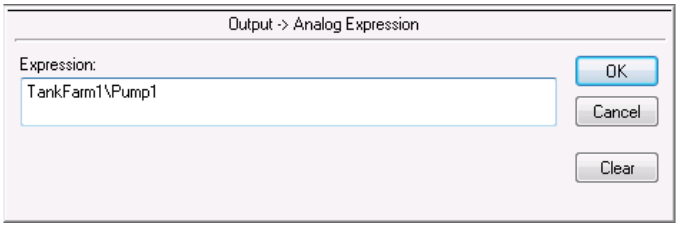
创建 SuperTag 的其它方法

您还可以使用以下方法来创建 SuperTag：

- 在“动画链接表达式”输入框中
- 在 InTouch QuickScript 中
- 在使用 InTouch DBLoad 实用程序加载到应用程序的外部文件中

注意：使用其它方法创建成员时，SuperTag 配置屏幕不在 SuperTag 模板定义中显示该成员。

通过动画表达式或 InTouch QuickScript 创建 SuperTag 时，必须使用有效的 SuperTag 格式。例如：



注意：如果在动画表达式或 QuickScript 中指定的 SuperTag 实例与成员标记当前尚未定义，则程序会提示您定义标记。选择“确定”。此时出现“标记名字典”，并显示所创建的 SuperTag 实例与成员标记。

下面是有效语法的示例：

```
ParentInstance\ChildMember ParentInstance\ChildMember\Submember
```

下面是无效语法的示例：

```
ParentInstance\
ParentInstance\ChildMember\
```

如果使用的格式无效，则出现错误消息框，指出 SuperTag 语法包含错误。

将 SuperTag 实例设置为父对象

父对象可以用于相对引用。当您将一个图形嵌入到另一个图形中时，可以使用 OwningObject 属性。您可以在“工业图形编辑器”中属性配置选项卡的运行时行为部分下设置父对象属性。为托管和独立 InTouch 应用程序均输入带 `InTouch:` 前缀的实例名称。例如，InTouch:Tank01。您也可以使用脚本设置父对象。OwningObject 属性可设置 ShowGraphic() 脚本函数所显示的图形的父对象。它可以是常量字符串和引用字符串的串联。示例：graphicInfo.OwningObject = "UserDefined_001";

注意：OwningObject 属性可为该图形设置引用，但是如果符号属于对象向导的一部分，那么它与 GraphicName 属性没有关联。因此，如果编写具有父对象的符号的脚本，请指定父对象名称作为 GraphicName 属性的一部分，例如：UserDefined_001.Pump_001。

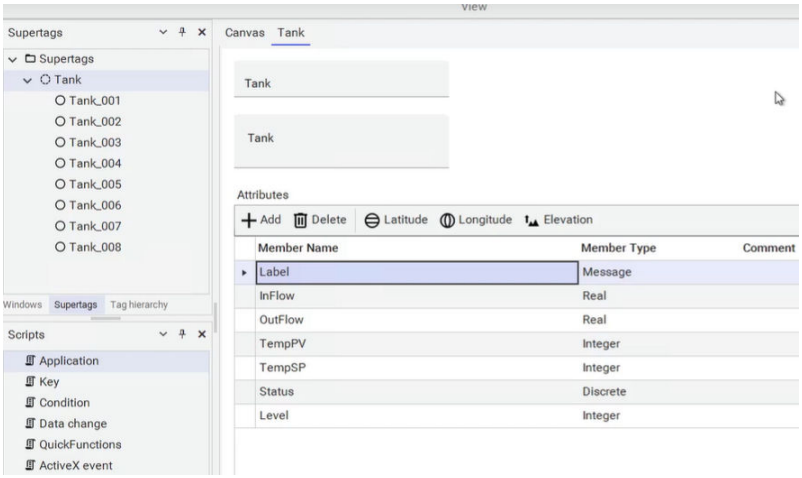
SuperTag 中的父对象

您可以将相对引用与 SuperTag 一起使用。您可以创建符号并使用基于 SuperTag 的父对象属性。通过此功能可以将 SuperTag 实例设置为图形的父对象。这有助于快速开发应用程序，而无需配置各个符号。

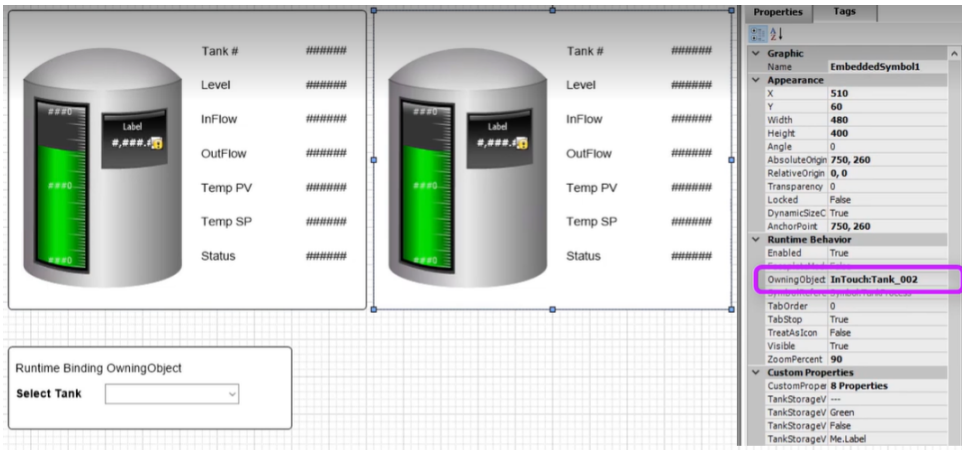
例如：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

假设一个名为 Tank 的 SuperTag 添加了 7 个属性，并创建了 8 个实例，即 Tank_001、Tank_002 直到 Tank_008，如下所示：

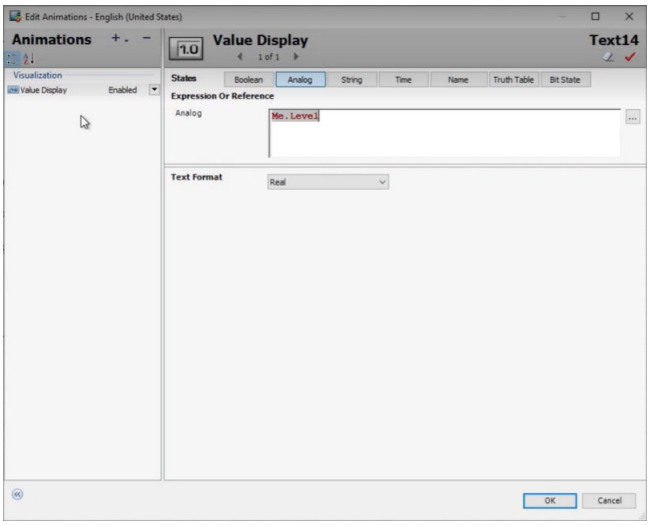


这些贮料罐 SuperTag 实例中的每一个都被设置为父对象。为了表示这些贮料罐实例，在“工业图形编辑器”中添加了这些贮料罐符号。

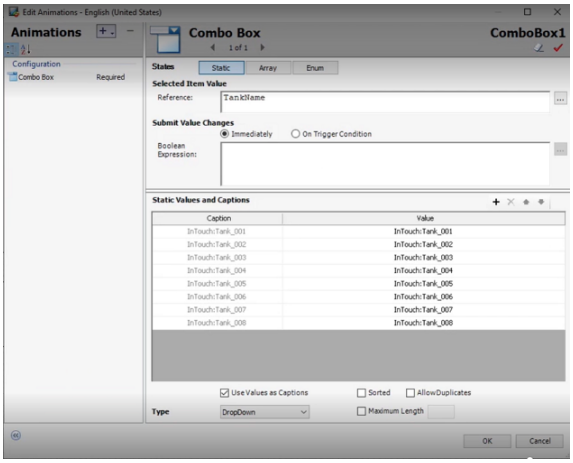


对于每个贮料罐符号，将属性的值显示设置为相对引用。（嵌入的图形应包含“me.Attribute”，相对引用才会起作用。被引用的属性（例如“me.Level”）将在运行时绑定到该 SuperTag 实例的属性。）

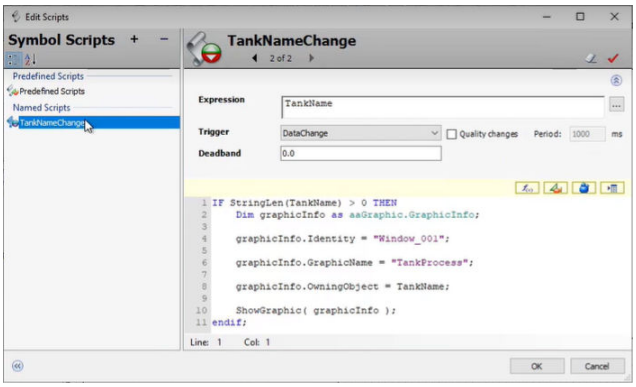
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



将引用自定义属性名称作为贮料罐名称的组合框添加到图形中，并将组合框中的值设置为 SuperTag 实例的名称。

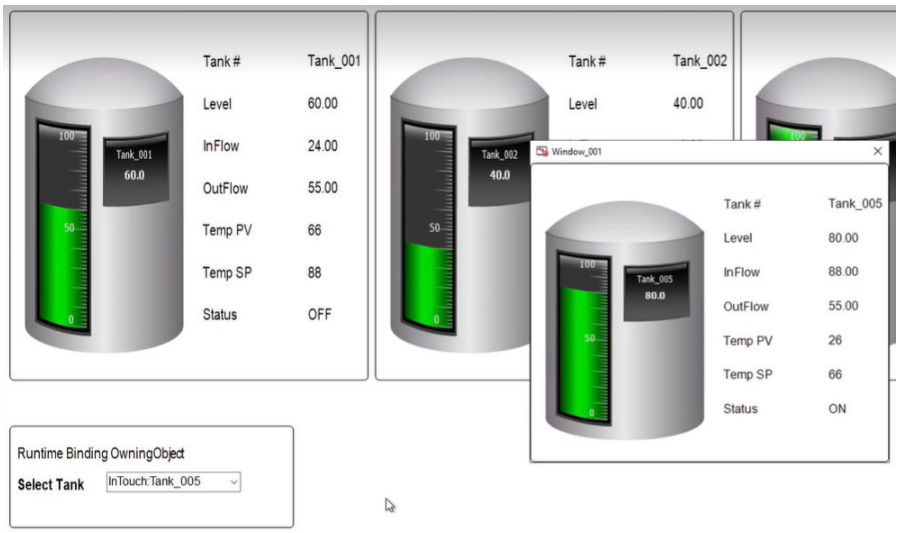


在此图形的脚本中，将数据更改脚本配置为在贮料罐名称自定义属性更改时执行。此脚本使用显示图形选项来显示贮料罐，并将父对象设置为自定义属性贮料罐名称的值。



切换到运行时之后，在组合框中可以选择任何贮料罐，并显示具有不同属性值的相应贮料罐。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



SuperTag 属性

使用画布上的 SuperTag 窗口，可以向 Supertag 模板添加属性。对于每个属性，还可以配置成员名和成员类型等属性。

添加或修改成员属性

1. 打开画布中的 SuperTag 窗口。
2. 在属性部分中，选择添加。

   此时将向网格中添加一个新属性。
3. 双击成员名单元格，以修改名称。
4. 双击成员类型单元格，以从可用的选项“整型”、“实型”、“消息”和“离散”中进行选择。缺省条件下，“成员类型”字段设置为“整型”。
5. 双击注释单元格，以修改描述。

添加或修改位置属性

1. 打开画布中的 SuperTag 窗口。
2. 在属性部分中，选择纬度、经度或海拔。

   此时将向网格中添加纬度、经度和海拔的新属性。
3. 对于所选的位置属性，“成员名”分别显示纬度、经度或海拔。

   注意：您无法修改位置属性的“成员名”。
4. 双击成员类型单元格，以从可用的选项“整型”、“实型”、“消息”和“离散”中进行选择。缺省条件下，位置属性的“成员类型”设置为“实型”。
5. 双击注释单元格，以修改描述。

删除属性

1. 打开画布中的 SuperTag 窗口。
2. 在属性部分中，选择要删除的属性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

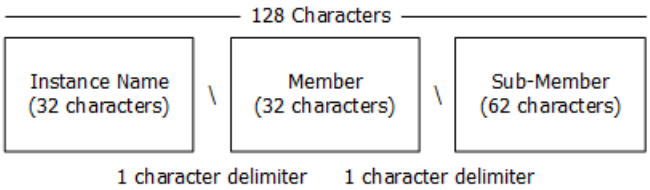
3. 选择删除。

   此时将删除所选的属性。

引用 SuperTag 成员

InTouch 标记名最多可以包含 128 个字符。每个 SuperTag 的“父实例\子成员\隶属成员”最多可以包含 128 个字符。标记名的最大长度使得在对 SuperTag 的引用上存在着一定的限制。

SuperTag 引用最多可以包含两个模板（父实例\子成员），引用深度为一个成员，具体如下图所示。



注意：如果属性类型未设置为任何现有 SuperTag，SuperTag 隶属成员最多可以包含 62 个字母数字字符。基于 SuperTag 类型的隶属成员最多只能包含 32 个字母数字字符。

SuperTag 模板中的每个成员都可以使用当前用于访问标准 InTouch 标记类型的点域的标准格式进行访问。InTouch 中凡是可以使用标准标记的场合也都支持 SuperTag 引用语法。例如，有效的 SuperTag 点域引用是：

```
TankFarm\Tank1\Pump1RPM.RawValue
```

远程标记引用同样支持 SuperTag。例如，有效的 SuperTag 远程引用是：

```
PLC1:"TankFarm\Tank1\Pump1RPM.RawValue"
```

使用批量导入实用程序导入 SuperTag

您可以使用批量导入实用程序将标记定义从 InTouch 转换为 Application Server 对象结构。使用批量导入实用程序可以更高效地将 InTouch 标记迁移到 Application Server。除标准的 InTouch 标记之外，批量导入实用程序还可以迁移 SuperTag。

如果打算将 SuperTag 从 InTouch 应用程序迁移到 Application Server，请将 SuperTag 引用中的反斜杠字符 (\) 替换为受支持的字符，如下划线 (_)。

示例：

```
TankFarm_Tank1_Pump1RPM.RawValue
```

注意：在此发行版发布时，DBDump 和 DBLoad 过程存在已知限制。通过从 .CSV 文件导入创建的 SuperTag 实例只会显示在标记字典中，不会显示在 SuperTag 窗格中。

在 InTouch 应用程序迁移期间导入资产

从 InTouch SP2020 R2 或更低版本迁移 InTouch 应用程序时，不会自动导入资产。

要导入在较旧版本的 InTouch 应用程序中创建的资产，请从较旧版本复制 supertag.dat 文件，并将其粘贴到新 InTouch 应用程序的相同位置。Supertag.dat 文件中包含在应用程序中创建的所有资产。

supertag.dat 文件的文件路径如下：

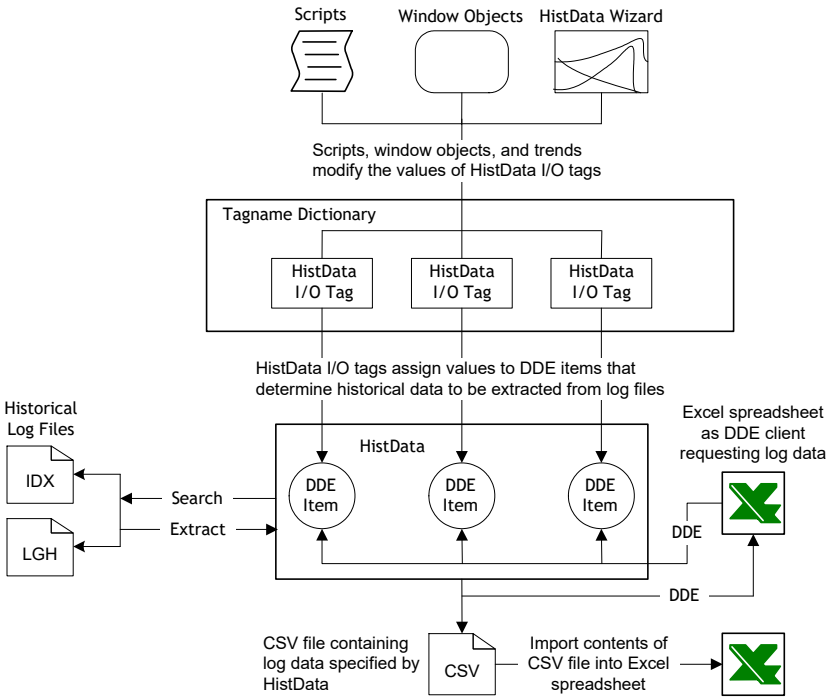
C:\ProgramData\Wonderware\InTouch\supertag.dat

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从其它应用程序访问历史标记值

您可以使用 InTouch HistData 实用程序将数据从历史日志文件中提取到逗号分隔值 (.csv) 文件。Excel 这类的应用程序可以作为 DDE 客户端直接从 HistData 中提取 InTouch 记录数据，或从 HistData 实用程序创建的输出文件中导入记录数据。

下图显示将所选历史记录数据保存到文件或 DDE 客户端应用程序的过程。



使用 DDE 项目显示历史数据

HistData 程序包含一组 DDE 项目，指定如何从日志文件中提取历史数据。这些项目是 HistData 内部数据库的一部分。您可以将值指定给每个项目。

下表概括了在 HistData 程序中定义的 HistData 项目。

项目	数据类型	描述
DATADIR	消息	包含历史日志文件的文件夹的路径。
DBDIR	消息	包含“InTouch 标记名字典”内容的文件夹的路径。
STARTDATE	消息	从日志文件中提取数据的开始日期。开始日期的格式为 MM/DD/YY。
STARTTIME	消息	从日志文件中提取数据的开始时间。开始时间的格式为 24 小时制的 HH:MM:SS 格式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

项目	数据类型	描述
DURATION	消息	从日志文件中进行数据采集的间隔长度。**DURATION** 可以表示为： • 星期 (w) • 天 (d) • 小时 (h) • 分 (m) • 秒 (s) **DURATION** 周期可以指定为小数。例如，`DURATION=0.5m` 相当于 30 秒。如果要求提供一个样本，请将 **DURATION** 设置为 0。
INTERVAL	消息	数据采集间隔的时间长度。**INTERVAL** 可以表示为星期、天、小时、分以及秒。**INTERVAL** 周期的时间单位与 **DURATION** 周期的时间单位相同。 间隔可以指定为小数值。例如，`INTERVAL=0.25d` 表示 6 个小时。 **DURATION** 或 **INTERVAL** 的最大周期为六个星期。六个星期的最大周期适用于指定给 **DURATION** 或 **INTERVAL** 的任何时间值。例如，42 是 **DURATION** 或 **INTERVAL** 周期的最大天数。
FILENAME	消息	某个文件的名称与文件夹位置，该文件包含从历史日志文件中提取的数据。
WRITEFILE	整型	一个标帜，指出输出文件的 HistData 写入操作状态。设置为 1 时，HistData 将请求的数据写入 **FILENAME**“项目名”所指定的文件。文件更新完成时，**WRITEFILE** 自动重置成 0。
ERROR	消息	一个字符串，它包含从日志文件中提取数据时发生的最后一个错误的描述。**STATUS** 设置为 1 时，**ERROR** 字符串设置为“无”。**STATUS** 设置为 0 时，**ERROR** 字符串包含错误消息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

项目	数据类型	描述

TAGS1, TAGS2,...	消息	包含一个或多个标记名的字符串，这些标记的数据从将日志文件中提取。
		TAGS 字符串在 WindowViewer 中最多可以包含 131 个字符，在 Excel 中最多可以包含 255 个字符。
		通过添加名称为 Tagsn 的标记项，可以给该字符串附加内容以满足更长的请求，其中 *n* 代表一个递增的整数。
		如果某个标记需要额外的标记文本，请在字符串末尾输入一个加号 **(+)**。
		例如：
		`TAGS="$Date,ProdLevel,ProdTemp,+"`
		`TAGS1="ReactLevel,Temp,GasLevel,+"`
		`TAGS2="MotorStatus"`
		不允许使用重复的标记名，每个标记字符串的最大长度为 512 个字节。

PRINTTAGNAMES	离散	一个标帜，指出标记的名称是否放到关联的值列的上方。设置为 1 时，打印标记名。设置为 0 时，不打印标记名。
DATA	消息	此项目按逗号分隔值的格式在 HistData 程序中保存请求的数据。它由其它应用程序用来通过 DDE 进行数据的 **ADVISE** 或 **REQUEST**。
状态	离散	HistData 最新操作的状态。 值为 1 表示 HistData 从日志文件中成功地提取了历史数据。值为 0 表示发生了一个错误。
SENDDATA	整型	表示 HistData 更新操作状态的标帜。设置为 1 时，HistData 使用所请求的数据更新 **DATA** 项。更新完成时，**SENDDATA** 自动重置为 0。 如果使用 **SENDDATA** 时收到错误消息，指出请求的数据过多，请缩短 **DURATION** 期限或减少请求的标记数。不允许使用重复的标记名，每个标记字符串的最大长度为 512 个字节。

使用 DDE 访问记录数据

您可以使用两种方法将记录数据提取到输出文件中。

- 如果要将八个或更多标记的历史记录数据保存到输出文件，则可以使用手动方法。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 如果仅需要保存映射到当前指定给历史趋势的笔的记录数据，请使用“历史数据向导”。

使用 HistData 手动提取记录数据

您可以将历史数据手动提取到输出文件中。按以下顺序完成各个步骤：

1. 创建 HistData 访问名
2. 为 HistData 创建 I/O 标记
3. 创建 HistData 窗口
4. 运行 HistData

创建 HistData 访问名

为了让 InTouch 从 HistData 程序中索取数据，必须定义一个“访问名”。

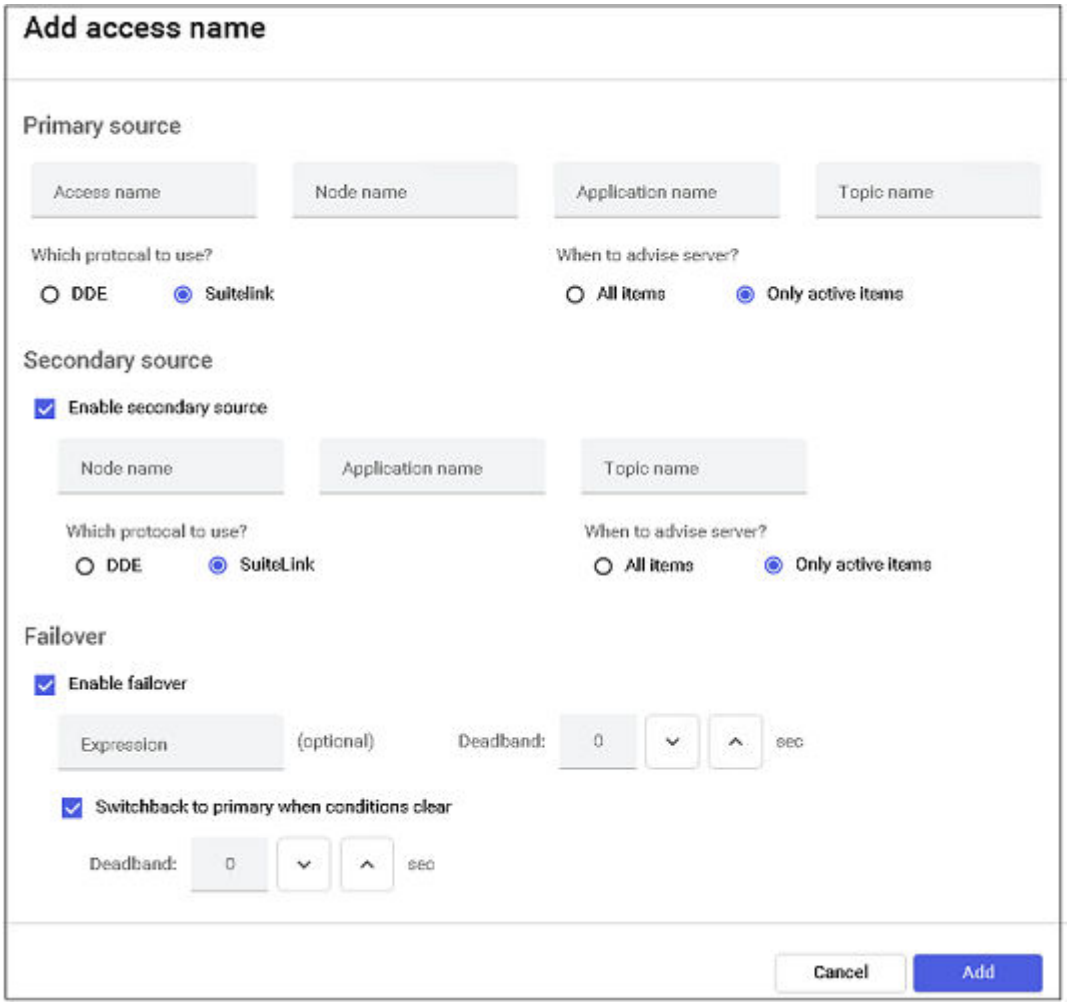
定义访问名

1. 在主页菜单上的标记组中，选择访问名。

   此时出现访问名对话框。
2. 选择添加。

   此时出现添加访问名对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**访问名**框中，输入一个最多包含 32 个字母数字字符的名称。指定**给访问名**与**主题名**的**值应该**相同。
4. 在**节点名**框中，输入日志文件当前所在**节点**的名称。
5. 在**应用程序名称**框中，输入不带 .exe 文件扩展名的 HistData。
6. 在**主题名**框中，输入在**访问名**框中指定的名称。**访问名**与**主题名应该**相同。
7. 从可用**选项**中**选择**相应的通**讯协议**：**DDE** 和 **Suitelink**。
8. 在**要对服务器提示时**区域中，只要使用 HistData，便**选择所有项**。
9. 选择添加。

创建 HistData 标记

在定义<1>“</1>访问名<2>”</2>之后<3>，</3>创建以下 I/O 型标记<4>，</4>以生成包含日志文件数据的输出文件<5>。</5> 将上一步中创建的“访问名”指定给标记。

标记	I/O 标记类型	项目
HDWDATADIR	消息	DataDir
HDWDBDIR	消息	DbDir

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

标记	I/O 标记类型	项目
HDWDURATION	消息	持续时间
HDWERROR	消息	错误
HDWFILENAME	消息	FileName
HDWINTERVAL	消息	间隔
HDWSTARTDATE	消息	StartDate
HDWSTARTTIME	消息	StartTime
HDWSTATUS	消息	状态
HDWTAGS、HDWTAGS1、HDWTAGS2	消息	标记
PRINTTAGNAMES	离散	PrintTagNames
HDWWRITEFILE	整型	WriteFile

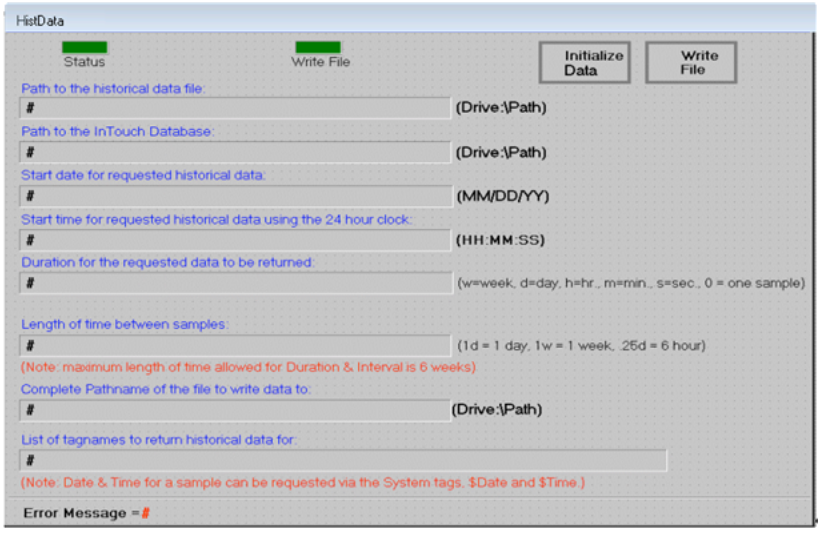
注意：HistData 向导自动创建除 PRINTTAGNAMES 标记以外的这些标记。

如果要将记录数据发送到 Data 项目中，以便可以从其它应用程序访问它，请创建两个额外的标记。而且<1>“</1>历史数据向导<2>”</2>不会自动创建 HDWSendData 及 HDWData 标记<3>。</3>

标记	I/O 标记类型	项目
HDWSendData	离散	**SendData**
HDWData	消息	数据

创建 HistData 窗口

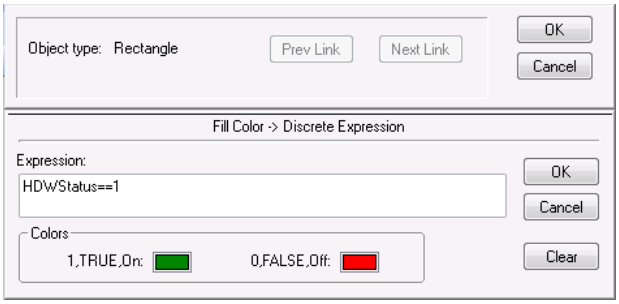
在创建 I/O 型标记之后，创建一个名为 **HistData** 的新窗口，类似于下例：



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

符号链接到一个用户输入链接。例如，# 符号有一个指向 HDWDataDir 标记的“用户输入/字符串”链接。该用户输入链接可用于在运行时更改标记的值。

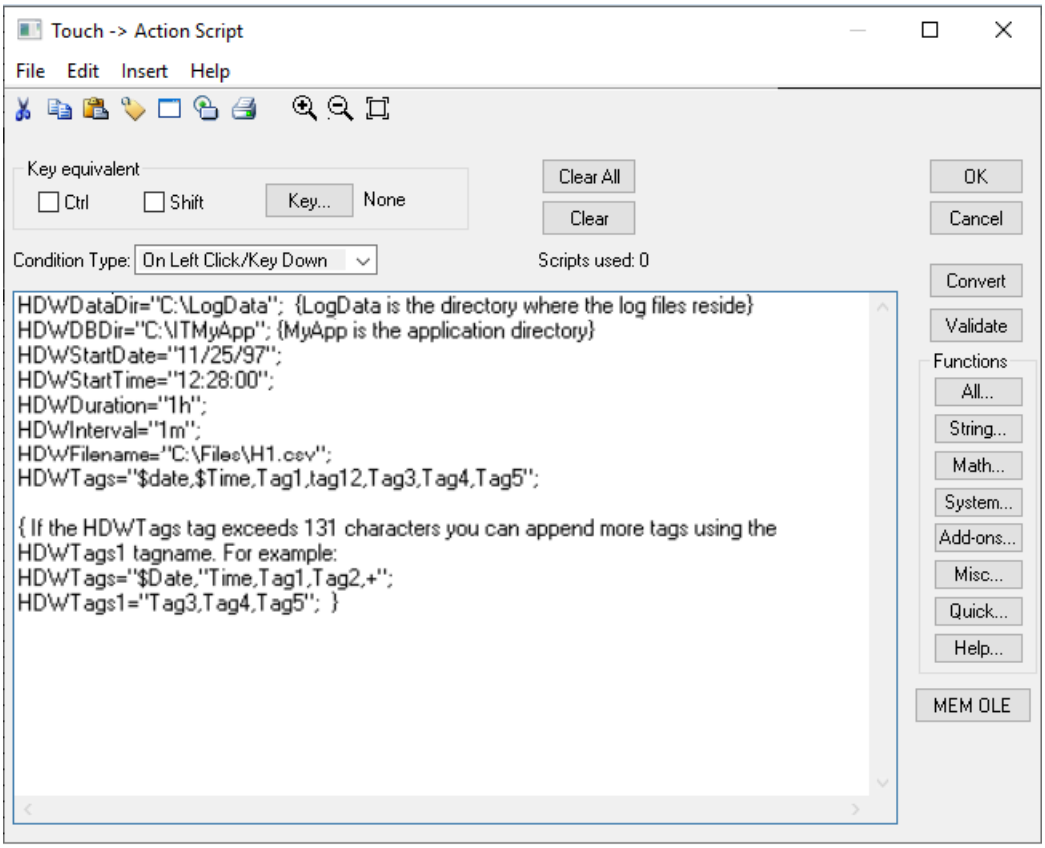
状态按钮链接到一个基于 HDWStatus 标记的“填充颜色 -> 离散表达式”。



写入文件按钮链接到一个基于 HDWWriteFile 标记的“填充颜色 -> 离散表达式”。

初始化数据按钮链接到“触动按钮”动作脚本。例如：

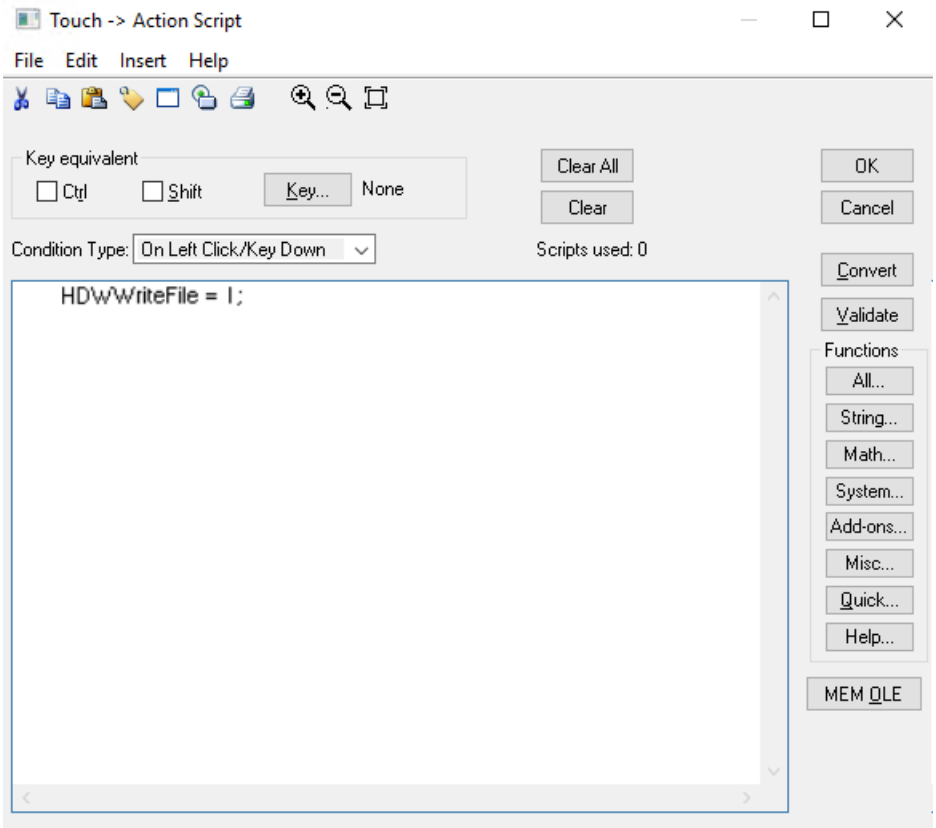
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



选择初始化数据按**钮时**，HistData **项**将使用所需的**值进**行初始化。如果需要，也可以通**过**使用“用**户输**入**链**接”在运行**时**更改**这些值**。

“写入文件”按**钮链**接到“触**动**按**钮**”**动**作脚本。**选择写入文件**按**钮时**，生成**输**出文件。**“写入文件”**按**钮**的配置如下：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



运行 HistData

在创建 HistData 窗口之后，完成以下步骤以便在 WindowViewer 中运行它。

运行 HistData 窗口

1. 启动 HistData 并将它最小化。
2. 启动 WindowViewer 并打开 HistData 窗口。
3. 选择初始化按钮，并对 HistData 项目进行更改。
4. 选择写入文件按钮。

   如果操作成功，Status（状态）值将变为 ON（打开），并呈现与 ON（打开）状态关联的颜色。如果操作不成功，Status（状态）的值将变为 OFF（关闭），并显示 Error Message（错误消息）以指明故障原因。

有关详细信息，请参阅创建 HistData 窗口。

使用“HistData 向导”来提取记录数据

您可以创建一个输出文件，包含在历史趋势中出现的记录数据。InTouch 包含“历史数据向导”，可以将从日志文件中提取数据的步骤自动化。

由于 HistData 输出文件写入历史趋势中显示的日志数据，因此该文件只能包含当前指定给历史趋势笔的标记的数据。

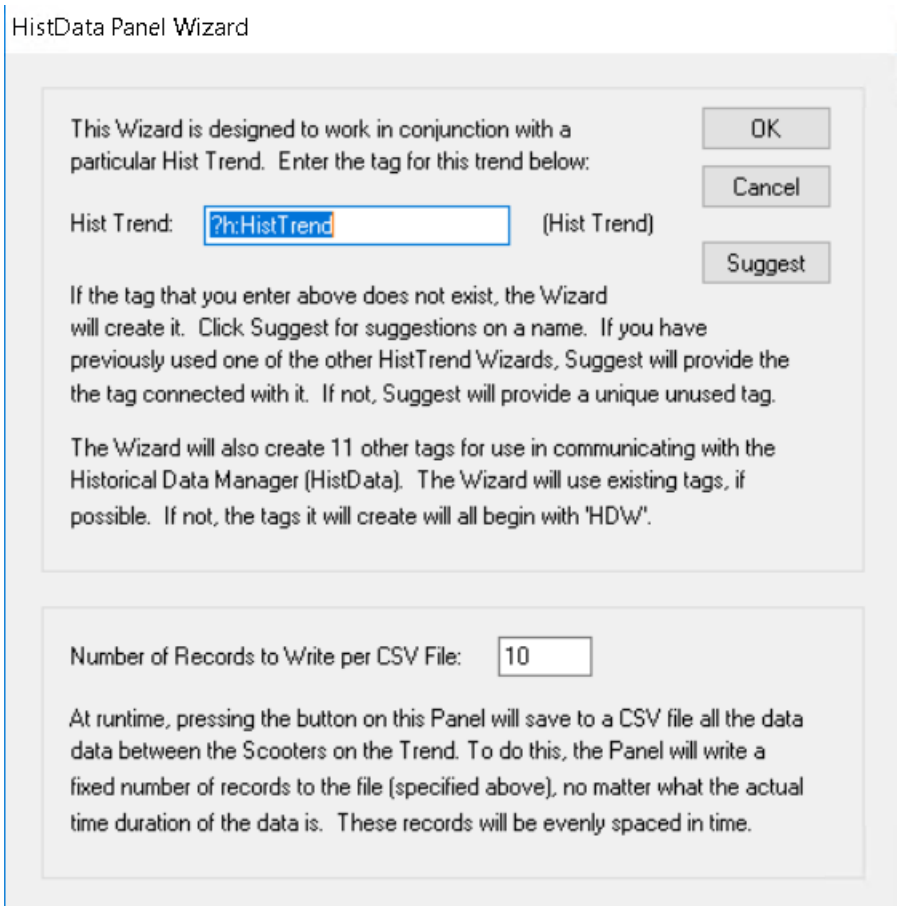
使用“HistData 向导”来提取记录数据

1. 启动 WindowMaker。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 打开一个包含历史趋势的窗口。
3. 在绘图菜单上的插入组中，选择向导。

   此时出现向导选择对话框。
4. 从左侧面板中选择趋势。
5. 从右侧面板中选择 HistData 向导图标，然后选择确定。
6. 将鼠标指针移到要放置 HistData 对象的趋势窗口区域。
7. 选择将 HistData 对象放置在趋势窗口中。“历史数据向导”创建一个窗口对象，其中包括一个按钮，以及文件名框，它显示创建输出文件的位置的路径。
8. 双击趋势窗口中放置的“历史数据向导”对象。此时出现历史数据面板向导对话框。



9. 在历史趋势框中，输入“历史趋势”标记的名称。
10. 在写入每一 CSV 文件的记录数框中，输入要写入输出文件的记录数。
11. 选择确定。“历史数据向导”创建一组使用 HDW 前缀的标记。

    “历史数据向导”创建创建 HistData 标记中列出的标记。“历史数据向导”将这些标记指定给 HisDataViewSt“访问名”。
12. 使用 WindowViewer 运行历史趋势窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

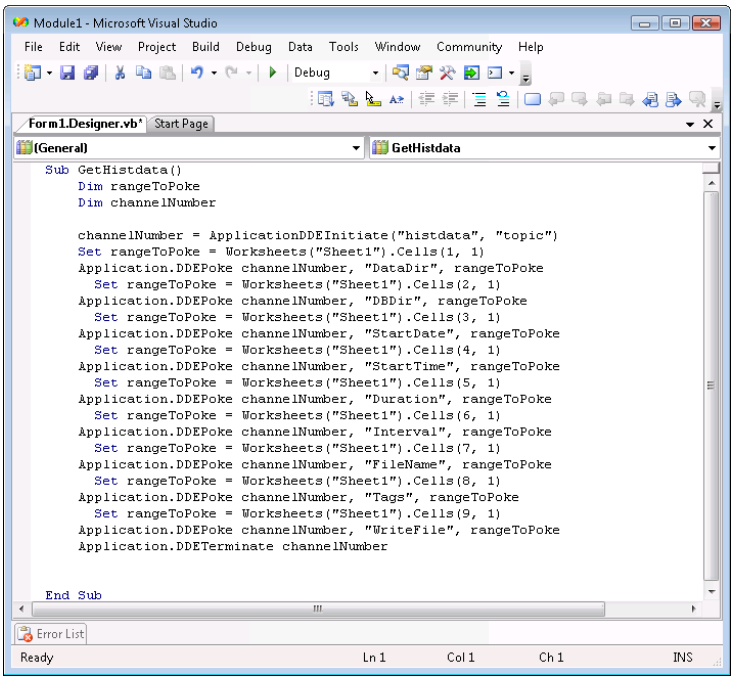
13. **选择保存至文件**（它是 HistData 窗口对象的一部分）。HistData 在窗口对象中显示的文件夹位置创建输出文件。

注意：如果标记值在以新创建的 *.idx 和 *.lgh 文件启动 InTouch 后的一小时内不发生变化，HistData 不会正确填充 .csv 文件。
HistData 数据的分辨率与历史趋势指示器的分辨率不同。HistData 分辨率基于在时间跨度内请求的值数量。它不能准确反映 *.idx 和 *.lgh 文件中的值。

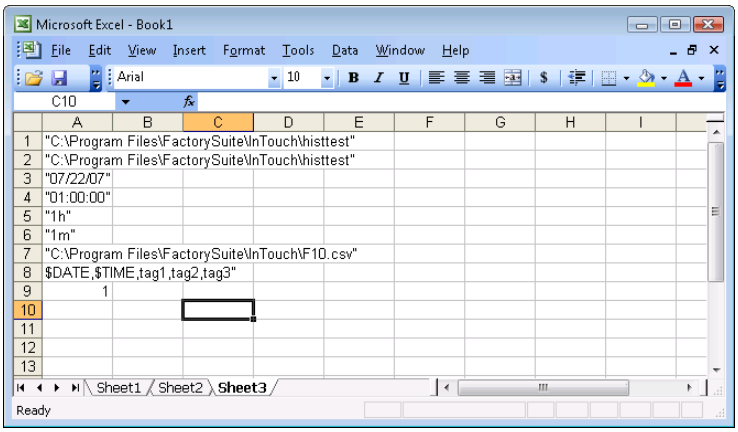
从其它应用程序中访问历史数据

您可以编写 Excel 宏，以便从历史数据文件提取数据。HistData 程序会对宏中的 INITIATE、POKE 及 TERMINATE 函数作出响应。带关键字（内部数据库项目）的 POKE 函数设置定义查询的参数。在正确指定查询之后，会运行宏向 HistData 文件请求提供所选的历史数据。

下例显示使用 VBA 编写的一个宏：



在上述示例中，要插入的数据在 Sheet1 中。下例显示要插入的数据：



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

排解 HistData 错误

使用 HistData 提取记录数据时，可能会看到发生一些错误。下表在左侧一栏列出典型的 HistData 问题或错误消息。右侧一栏描述该问题的可能原因及解决方案。

错误消息或条件	原因与/或解决方案
错误消息： 请求的数据太多 -- 请缩短持续时间或减少标记名数。	SendData 项目请求过多的数据时，便会发生此错误。 如果唯一目的是创建一个输出文件，用它来储存日志文件的数据，请不要使用 **SendData** 项目。
错误消息： 无法打开文件 `C:\FILES1\HISTDATA.CSV`	文件夹路径不存在，或文件夹路径的拼写不正确。
错误消息： 无法打开文件 `C:\FILES\`	未定义输出文件。
错误消息： `DATADIR` 项目无效	**DataDir** 项目指定的目标文件夹路径不存在。验证文件夹路径的拼写是否正确。
错误消息： `STARTDATE` 项目无效	**StartDate** 项目包含无效的开始日期格式。在 Windows 中，将计算机的日期格式更改为 mm/dd/yy。
错误消息： 未找到日志文件	**DataDir** 项目指定的路径中没有所请求的日期的日志文件。
错误消息： 在数据库中找不到标记名 `TAG• •`	在应用程序的“标记名字典”中不存在所请求的标记。验证标记名的拼写是否正确。
错误消息： 在 `C:\IT6.0B\HISTEST` 中找不到 `tagname.x`	**tagname.x** 文件不存在或已损坏。
未创建 .csv 输出文件且没有出现错误。	• HistData 没有运行。 • **Tags** 项没有列出已指定要记录的任何标记。 • HDWWritefile 定义不正确。确保标记是整型标记，“DDE 访问名”正确且项目是 **WriteFile**。此外，确保 **MinEU**=**MinRaw** 且 **MaxEU**=**MaxRaw** 时未进行缩放。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

错误消息或条件	原因与/或解决方案
.csv 输出文件包含日期与时间标签，但不包含所请求的标记的任何记录数据。	在所请求的时段内<1>，</1>这些记录的<2>“</2>历史<3>”</3>记录中没有任何项目<4>。</4> 显示一个历史趋势，以验证日志文件在所请求的时段内是否包含数据。
WWLogger 包含以下消息： `DDE HistData|Viewstream1 错误！WriteFile：插入被服务器拒绝。`	由于错误会给 HistData 项目指定值，因此每次创建 .csv 文件失败时，都会将错误消息写入 WWLogger。如果尝试将 **WriteFile** 项目的值设置为 0，或尝试写入错误项目，则也会出现此错误。
日志文件中应该包含许多条记录时，.csv 文件只包含一个记录。	可能给 **Interval** 项目指定了不正确的值，导致采集间隔非常小。 也可能由于 **Duration** 项目使用了不正确的格式，如 `HDWDuration=1-` （未指定增量）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

IEEE 十进制单位

Historian HMI 采用电气与电子工程师协会 (IEEE) 754 标准，将 32 位二进制值转换为十进制浮点数。

在 16 位可编程逻辑控制器 (PLC) 中，IEEE 754 标准的 32 位数字存储为两个 16 位字。对于低位与高位十六进制字，PLC 中的浮点寄存器通常采取顺序编号机制。目前这代 32 位个人计算机使用单个 32 位寄存器。寄存器的位编号机制与两个顺序的 16 位寄存器采用相同的格式。

要在 Historian 应用程序中使用浮点数，Historian 必须能够转换两个 16 位 PLC 寄存器中存储的值。Historian 始终将原始 PLC 寄存器值当作单独的整数，因此必须进行位转换。对于这两个寄存器整数，不能执行布尔 **AND** 运算，也不能将它们转换为实数。Historian 无法对双整数寄存器执行类型转换。

在 Historian HMI 中显示浮点数

Historian HMI 使用 IEEE 32 位浮点格式在应用程序中显示实数。IEEE 浮点格式只是实际实数的一个近似值。除非实数是二的偶数次方，否则不能用 IEEE 32 位浮点格式完全准确地表示。IEEE 32 位浮点数的精度约为十进制小数点后八位。

要在 Historian 应用程序中显示实数时，确保该数字不超过八位。以下浮点数格式显示 Historian 应用程序中有效的实数：

#

#,###

#.##

0#

###.##

#.#######

###.#####

######.##

ABCDEF

###.####

任何超过八位的浮点数都可能会发生舍入错误。

如果在文本格式中不包括小数，则按照 WindowViewer 的高级格式属性中配置的实数格式小数精度用小数显示数字。

注意：如果在小数左侧添加“#”，或者没有小数，则不要限制显示的位数。

示例 1

Historian 应用程序应显示实数 2.3。但 2.3 这个数字不是二的偶数次方，因此不能使用 IEEE 32 位浮点格式精确地表示为 8 个以上小数位的十进制数。

要确保从应用程序中将 2.3 这个值显示为 ASCII 字符 2.3，该数字不得超过八位。如果该数字超过最大值八位，则产生得数字可能显示为 2.29999999 或 2.30000001。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例 2

在比较两个实型标记值时，两个实型标记的差异应大于 FLT_EPSILON（值 1.19209290E-07F，十进制 0.0000001192092896）。但如果该数字超过 8 位，则结果值可能不正确。为了更正此错误，可以乘以 1000 或更大的 10 的倍数。执行此操作后，该值大于 1e-7。执行必要的比较运算，然后再除以 1000 或者刚才乘以的数字。

配置和使用 InTouch OPC UA 服务器

InTouch HMI 支持用于机器到机器通讯的 OPC UA（统一架构）协议。

此 OPC UA 服务器功能允许第三方客户端与 InTouch HMI 进行交互并采用行业标准，如 OPC UA。在 InTouch HMI 上启用 OPC UA 服务器功能时，客户端可以利用内置的 Gateway Communication Driver 或第三方客户端连接到 InTouch HMI，然后使用 Gateway Communication Driver 或该客户端安全浏览 OPC UA 域名空间并与 InTouch HMI 进行交互。

OPC UA 配置检查清单

对 OPC UA 服务器和 OPC UA 客户端进行端到端配置所需的任务

配置任务将按照必须完成它们的顺序进行显示。

1. **配置系统管理服务器：**系统管理服务器用于在机器之间建立信任关系，并且必须对其进行配置以确保在节点之间进行安全通讯。通常在初始 System Platform 安装期间对系统管理服务器进行配置。如需详细信息，请参阅《*System Platform 安装*》中的“配置系统管理服务器”。

   注意：必须在具有管理权限的用户的环境中运行 InTouch HMI，该环境允许 InTouch HMI 访问用于实现安全通讯的加密证书。

2. **配置 OPC UA 服务器：**设置配置选项，测试 OPC UA 服务器连接，并激活 Gateway Communication Driver。
3. **IT 合规性/防火墙验证：**必须在此配置点完成防火墙配置和验证。已在其上部署 OPC UA 服务器的节点必须具有“入站规则”才能对防火墙进行配置和验证。

   重要事项：必须先成功执行防火墙测试，才能继续进行其余的配置任务。

4. **配置 OPC UA 客户端：**客户端配置可能包括以下内容：
   - 定义 OPC UA 服务器地址（格式为 `opc.tcp://<ServerName>:<PortNumber>`）。
   - 选择正确的 OPC UA 服务器安全策略 (`Basic256Sha256`)。
   - 将用户添加到 InTouchHMIOPCUAWriteUsers 用户组。
   - 输入已配置的 OPC UA 用户凭证（用户名和密码）。
   - 仅支持使用匿名连接来读取 InTouch 标记。为了避免任何安全风险，建议使用经过身份验证的凭证来访问数据。
5. **安全证书：**在运行时节点上下载并配置 OPC UA 安全证书。
6. **验证连接性：**打开 OPC UA 客户端，并验证您是否可以连接到 OPC UA 服务器，以及是否可以在域名空间中查看项目。

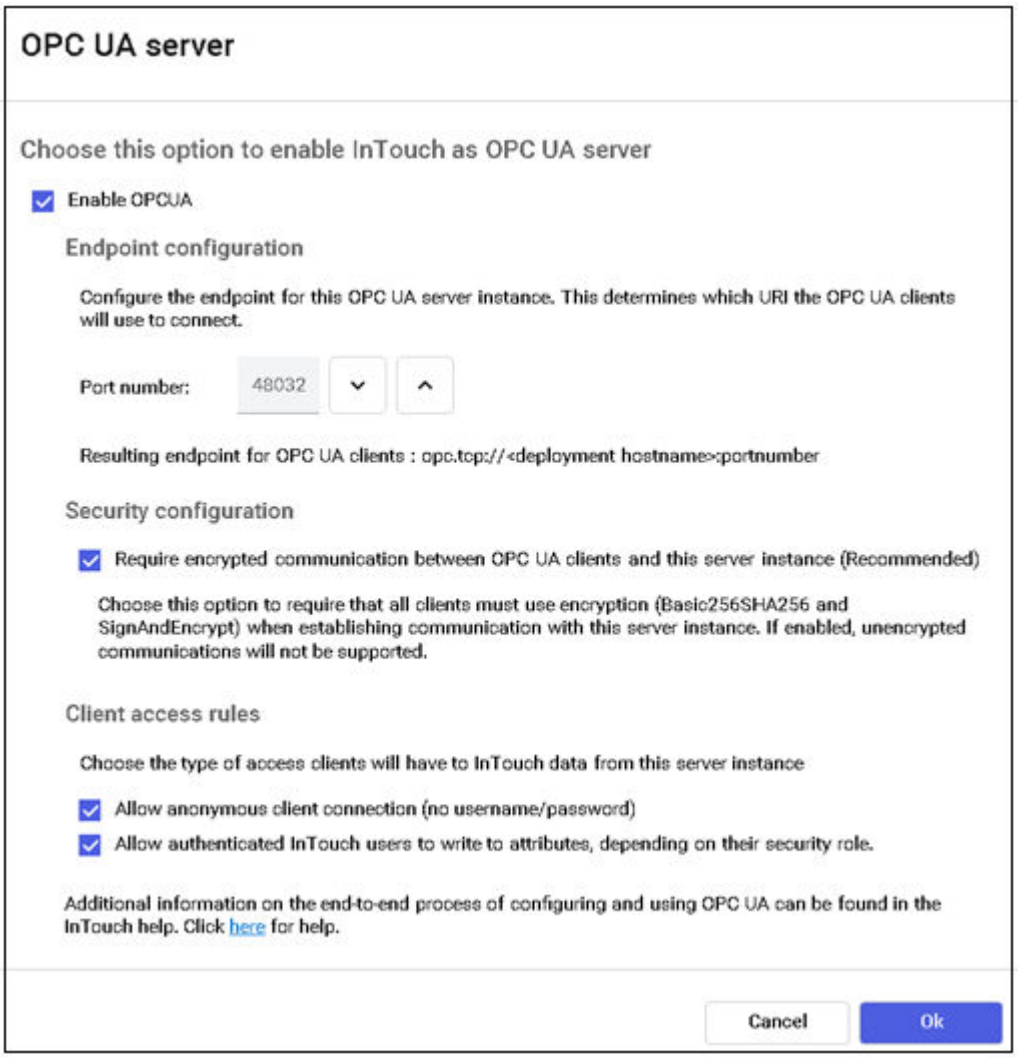
配置 InTouch OPC UA 服务器

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch OPC UA 服务器提供了从 OPC UA 客户端访问 InTouch HMI 数据的权限，而无需网关或其它协议转换机制。

1. 启动 **AVEVA InTouch HMI 应用程序管理器**。
2. 在“工具”选项卡上，选择 **OPC UA 配置**。

   此时出现“OPC UA 服务器”对话框。
3. 缺省条件下，将会禁用 OPC UA 服务器。要配置该服务器，请选择启用 **OPCUA**。



4. **编辑端口号**。缺省条件下，端口将设置为 48032。

   注意：端口号对于用户和 RDS 会话是唯一的。为其它 RDS 会话分配替代端口号。
5. **安全配置**：强烈建议您启用此选项，因为这样将会通过连接加密负载。请注意，客户端必须与此配置相匹配。
6. “客户端访问规则”选项可用于确定客户端对 InTouch 数据将具有的访问类型。
   a. 选中**允许匿名客户端连接（无用户名/密码）**复选框，以允许匿名访问。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

b. 选中**允许经过身份验证的 InTouch 用户根据其安全角色写入属性复选**框，以仅允许经过身份验证的用户。

7. 选择确定。
8. 在配置之后，启动 WindowViewer 以启动 OPC UA 服务器。OPC UA 客户端现在可以访问 InTouch HMI 数据。

配置 OPC UA 服务的防火墙

InTouch HMI 中的 OPC UA 通讯要求运行时节点防火墙允许与 OPC UA 客户端节点建立连接。然而，在更改防火墙设置之前，建议您执行防火墙测试。

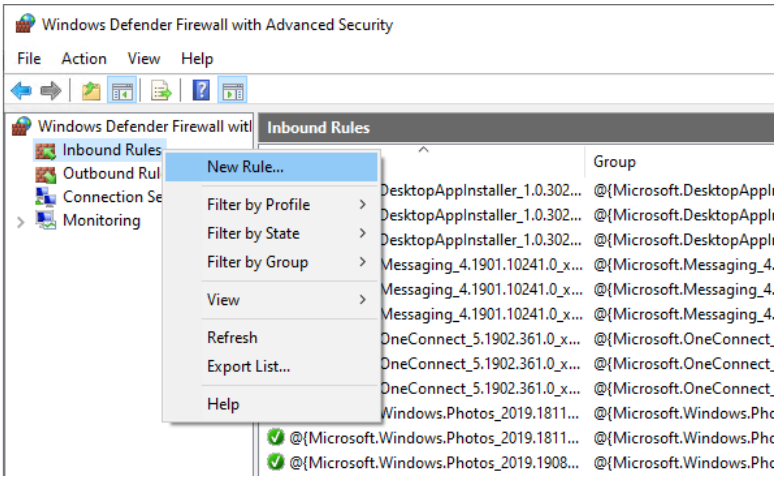
配置运行时节点防火墙

要点：必须将防火墙规则添加到在其上部署了 OPC UA 服务器服务的节点。

配置运行时节点防火墙

在部署了 OPC UA 服务器服务的运行时节点上，打开 Windows 防火墙并对其进行如下配置：

1. 在 Windows 搜索栏中，打开 **Windows 防火墙**。
2. 选择高级设置，并创建入站规则。

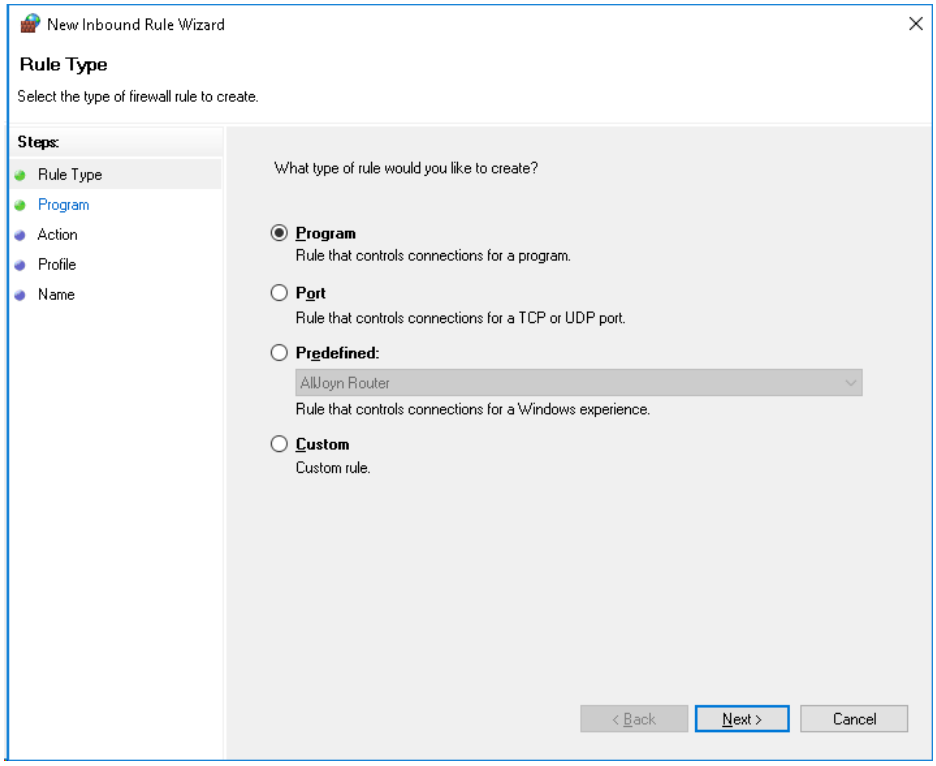


3. 选择新建规则。

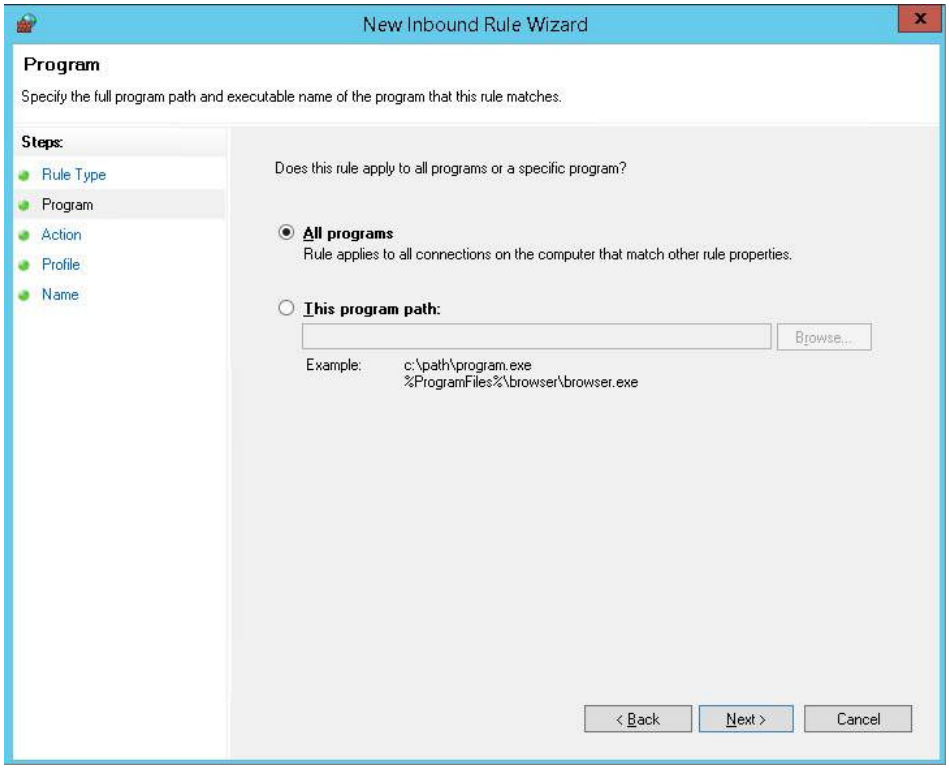
   规则向导即会打开。

4. 对于“规则类型”，选择程序，然后选择下一步。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

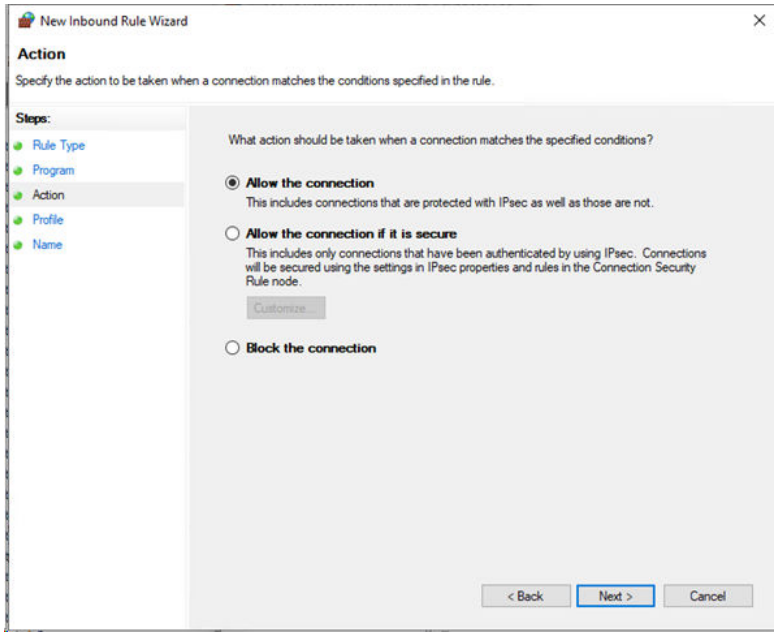


5. 指定所**选规则**的程序路径。

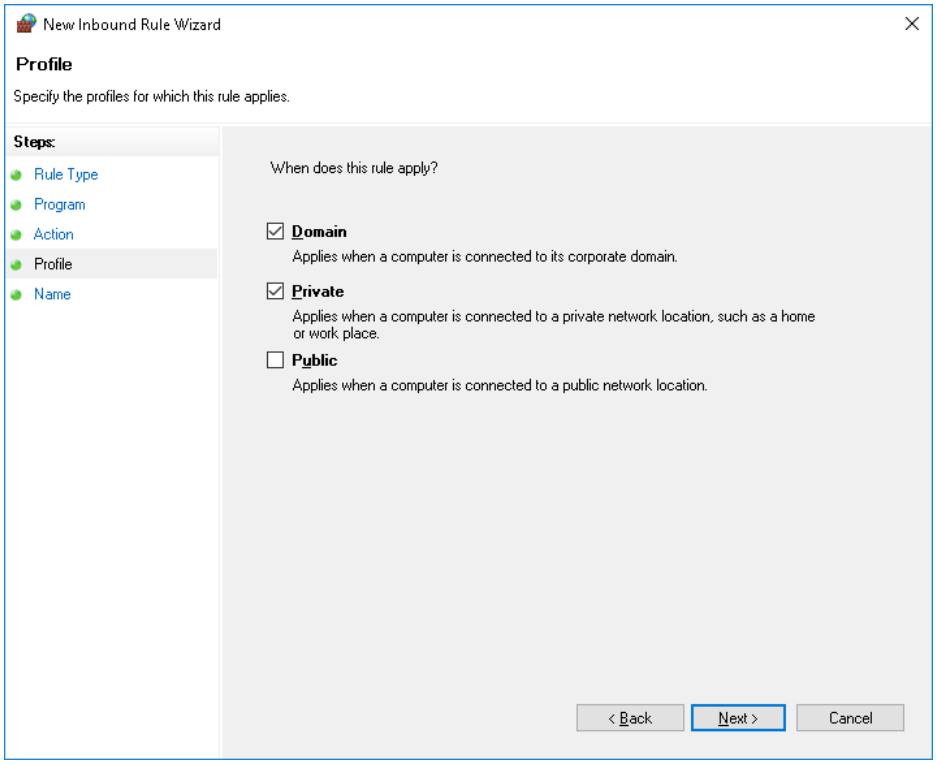


6. 在下一个屏幕上，**选择“允许连接”选项。选择下一步**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

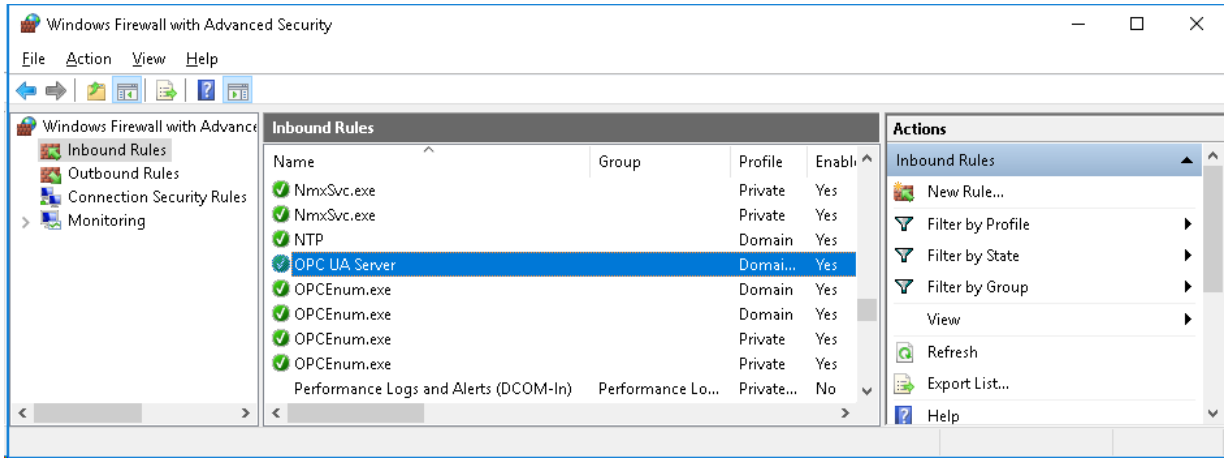


7. 该向导将询问规则何时适用。
   - 对于 域 环境：选择域和专用。建议您取消选择公共。
   - 对于工作组环境：选择公共。“域”和“专用”设置在“工作组”环境中不起作用。



8. 最后，为此规则提供一个名称（例如“OPC UA 服务器”）。如果您将要配置多个 OPC UA 服务，请确保使用能够将每个服务与其它服务进行区分的名称。
9. 现在，检查是否已将新规则添加到 Windows 防火墙中的“入站规则”列表，并且是否已启用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



10. 通过重复防火墙测试，验证是否可以从 OPC UA 客户端节点连接到运行时节点。

防火墙测试

通过 Telnet 执行防火墙测试

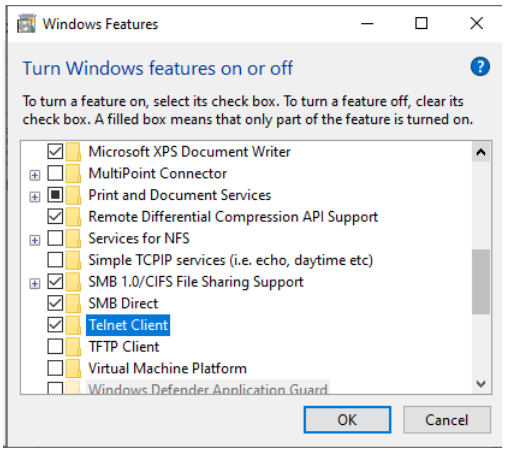
1. 从 OPC UA 服务器服务运行所在的节点之外的单独节点中，通过打开 **Telnet 客户端** Windows 功能启用 **Telnet**。

   注意：理想的情况是，从 OPC UA 客户端节点运行此测试。

   a. 打开 Windows 控制面板。

   b. 转到程序和功能，然后选择打开或关闭 **Windows** 功能。

   c. 向下滚动功能列表找到 **Telnet 客户端**并启用它。缺省条件下，Telnet 处于禁用状态。



2. 在运行此测试之前，验证 WindowViewer 是否正在运行。如果配置了 OPC UA 服务主机，WindowViewer 即会启动 InTouch OPC UA 服务。如需有关详细信息，请参阅配置 InTouch OPC UA 服务器。

3. 通过输入以下内容，在 OPC UA 客户端节点上的命令窗口中运行 Telnet：

   telnet <nodeName 或 ipAddress> <portNumber>

   其中

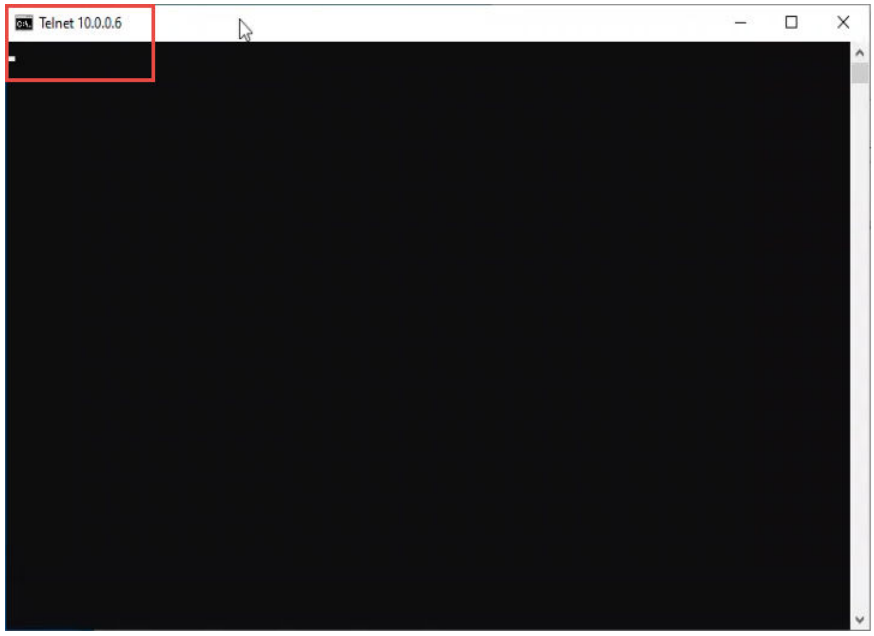
   - **nodeName** 是 InTouch HMI 运行时节点的机器名称。使用 nodeName 或 ipAddress，不能二者同时使用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- **ipAddress** 是 InTouch HMI 运行时节点的 IP 地址。使用 nodeName 或 ipAddress，不能二者同时使用。
- **portNumber** 是您在 InTouch HMI 中为 OPC UA 服务配置的端口号。缺省端口号为 48032。

示例：`telnet 10.10.10.06 48031`

- 如果命令没有成功，它将会超时，并显示一条消息，指出连接失败。在这种情况下，转到配置运行时节点防火墙。
- 如果 telnet 命令成功，命令提示即会更改为 Telnet 提示。



- 如果防火墙测试成功，请配置 OPC UA 客户端和 OPC UA 服务器证书。设置 OPC UA 连接的下一个步骤取决于您使用的是第三方 OPC UA 客户端应用程序，还是 Gateway Communication Driver 提供的 OPC UA 连接。

为第三方 OPC UA 客户端应用程序配置服务器和客户端证书

重要事项：只有您使用的是第三方 OPC UA 客户端时，这些步骤才适用。如果您是使用 Gateway Communication Driver 作为 OPC UA 客户端，请跳到使用 OI Gateway 配置客户端安全证书。

要在 InTouch OPC UA 服务器和第三方 OPC UA 客户端之间配置加密的通讯，这两台计算机都将需要访问以下证书：

- <计算机名称> ASB OPC UA 服务器
- 来自您的 OPC UA 客户端的客户端证书。

要完成此设置，您将需要执行三个步骤：

1. 将证书从 OPC UA 服务器节点复制到 OPC UA 客户端节点。此步骤包括以下内容：
   - 导出 OPC UA 服务器证书。
   - 将该证书安装在客户端节点上。
2. 将证书从 OPC UA 客户端节点复制到 OPC UA 服务器节点。
3. 确保已将防火墙配置成允许 InTouch.OPCUA.ServiceHost.exe 应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将 OPC UA 服务器证书导出到 OPC UA 客户端节点

导出 OPC UA 服务器证书

1. 打开 OPC UA 服务器节点上的 Windows 证书管理器。

   要打开证书管理器，请在 Windows 搜索框中输入管理计算机证书并选中，或者打开命令提示并运行 `certlm.msc`。

2. 在树视图中，展开个人节点，然后选择证书。
3. 找到<计算机名称> ASB OPC UA 服务器证书。
4. 使用鼠标右键单击该证书并选择“所有任务\导出...”。

   证书导出向导即会打开。

5. 根据您的客户端应用程序使用的证书类型，您需要将证书导出为以下两种证书文件类型中的一种：
   - **.cer 文件**（如果客户端使用 Windows 证书存储区）
   - **.der 文件**（如果客户端使用基于文件的证书）。

   尽管文件扩展名不同，但文件格式相同。

6. 在“证书导出向导”中，选择以下选项。
   - **导出私钥：** 选择“不，不要导出私钥”（缺省值），然后选择下一步。
   - **导出文件格式：** 根据您的客户端应用程序的要求，选择“DER 编码二进制 X.509 (.CER)”或“Base64 编码 X.509 (.CER)”。进行选择，然后选择下一步。
   - **要导出的文件：** 输入文件名以导出根 CA（例如“`c:\temp\<本机名> OPC UA Server.cer`”），然后选择下一步。
7. 导出向导完成后，根据您的客户端应用程序的期望内容，可能需要将证书的文件扩展名从“.cer”更改为“.der”。如果您的 OPC UA 客户端应用程序将证书存储在特定的文件夹，而不是 Windows 证书存储区中，那么将需要执行此操作。

在客户端计算机上导入 OPC UA 服务器证书

要完成将 OPC UA 服务器证书复制并安装到 OPC UA 客户端节点的过程，您需要将 OPC UA 服务器证书导入到 OPC UA 客户端计算机。

每个 OPC UA 客户端应用程序都有其自己的证书管理机制。通常，OPC UA 客户端将使用以下两种机制中的一种来管理证书：

- 利用 Windows 证书存储区
- 将证书存储在指定的文件夹中，具体由 OPC UA 客户端应用程序定义。

如需有关如何导入服务器证书的详细信息，请参阅您的 OPC UA 客户端应用程序的文档。

将证书导入 OPC UA 客户端的 Windows 证书存储区

1. 将证书文件（<机器名称> OPC UA Server.cer）复制到 OPC UA 客户端节点。复制文件的位置并不是很重要。
2. 使用鼠标右键单击该证书文件，然后从上下文菜单中选择安装证书。

   这样即会打开“证书导入向导”。

   注意： 需要管理员权限才能导入证书。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 在“证书导入向导”中，选择以下选项。
   - 存储位置：选择“本地计算机”，然后选择下一步。
   - 证书存储：浏览至“个人”部分，然后选择下一步。
   - 正在完成证书导入向导：查看设置，然后选择完成。

将证书复制到 OPC UA 客户端上的指定位置

1. 将证书文件（<机器名称> OPC UA Server.cer）复制到由您的 OPC UA 客户端应用程序指定的文件夹。下表显示许多常见的 OPC UA 客户端的证书文件夹位置。

OPC UA 客户端	制造商	文件夹
Datafeed OPC UA 客户端	Softing	C:\ProgramData\Softing\OpcClient\pki\trusted\certs
UaExpert	UnifiedAutomation	C:\Users\Admin\AppData\Roaming\unifiedautomation\uaexpert\PKI\trusted\certs
UA 客户端入门	UnifiedAutomation	C:\ProgramData\unifiedautomation\UaSdkNetBundleEval\pkiclient\trusted\certs
Matrikon	Matrikon	C:\Users\Admin\Appdata\Local\Matrikon\OPCUAExplorer\pki\DefaultApplicationGroup\trusted\certs
KEPServer	Kepware	C:\ProgramData\Kepware\KEPServerEX\V6\UA\Client Driver\cert
顶级服务器	软件工具箱	C:\ProgramData\Software Toolbox\TOP Server\V6\UA\Client Driver\cert

   如需有关管理证书的其它信息，请参阅 OPC UA 客户端应用程序的文档。

2. 配置 OPC UA 证书，如下所述。

在 OPC UA 服务器上配置 OPC UA 客户端证书

配置 OPC UA 服务器和客户端证书过程中的下一个步骤是信任已安装在 OPC UA 服务器节点上的 OPC UA 客户端证书。

最简单的方法是尝试将 OPC UA 客户端连接到服务器。由于尚未建立客户端证书的信任，因此连接将失败。

一旦连接失败，就会将未安装在 OPC UA 服务器上的任何 OPC UA 客户端证书放入 OPC UA 服务器上的“Rejected Certificate”文件夹中。

缺省条件下，文件夹位置是：

C:\ProgramData\AVEVA\PCS\OPC UA Rejected Client Certificates\certs

注意：访问此文件夹需要管理员权限，而且缺省条件下该文件夹处于隐藏状态。

导入放在 Rejected Certificate 文件夹中的证书

1. 要导入 OPC UA 客户端证书，请浏览到 Rejected Certificate 文件夹。
2. 右键单击要信任的 OPC UA 客户端的证书，然后选择“安装证书”。这样即会打开导入证书向导。
3. 在该向导中选择以下选项：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- **存储位置：选择**“本地计算机”，然后**选择下一步**。
- **证书存储：选择**“受信任人”，然后**选择下一步**。
- 完成**证书导入向导：查看设置**，然后**选择完成**。

端口使用情况

OPC UA 通过一个 TCP 端口进行通讯。这是在 OPC UA 服务器配置的**端点连接设置**中指定的，缺省使用端口 48032。如需有关详细信息，请参阅配置 InTouch OPC UA 服务器。

如果您将在使用 RDS 会话的同一台计算机上运行多个 OPC UA 服务器，则必须为每个后续服务器指定不同的端口号。

确认此端口没有被安装在您的计算机上的任何防火墙软件阻止。如需有关通过防火墙进行测试和配置的通讯信息，请参阅配置 OPC UA 服务的防火墙。

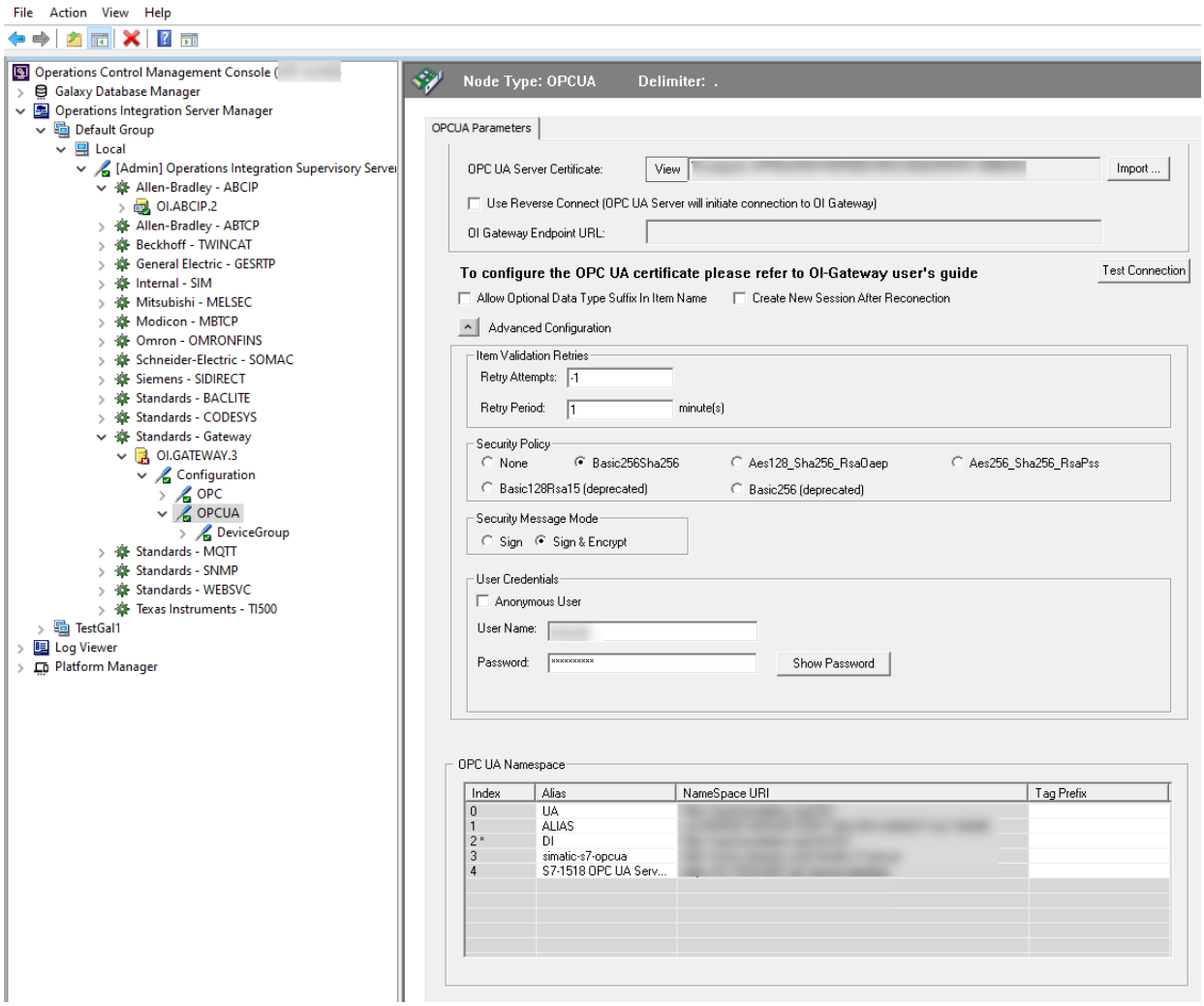
使用 OI Gateway 配置客户端安全证书

Gateway Communication Driver 提供了一种用于在服务器和客户端 OPC UA 节点上配置安全证书的简便方法。

通过 Gateway Communication Driver 配置安全证书

1. 从运行时节点上的“开始”菜单中打开“操作控制管理控制台”。（**开始 > AVEVA >操作控制管理控制台**）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在控制台**树**中，**导**航到 Operations Integration Supervisory Servers 下面的 **OI.GATEWAY.3** 节点。
3. **创**建 OPC UA **连**接。
4. 配置 OPC UA 服**务**器**详细**信息。
   - **服务器节点：输**入运行**时节**点的机器名称。
   - **OPC UA 服务器：这**是 OPC UA 服**务**器（运行**时节**点）的 URI（**统**一**资**源**标识**符）。**因为**当前无法**发现**地址，所以必**须**手**动进**行**输**入。以 opc.tcp://<机器名称>:<OPC UA 端口号> 格式**输**入地址。

     使用您在 InTouch HMI **应**用程序管理器中配置 InTouch OPC UA 服**务**器**时**所**输**入的 OPC UA 端口号。缺省端口号**为** 48032。
5. **输**入授**权**和身份**验证**凭**证**。

   必**须**与在 OPC UA 服**务**器**对话**框中配置的授**权设**置相匹配。如果**选**中“需要安全身份**验证**”**复选**框，**则**必**须选择**以下**设**置：
   - **安全策略：**Basic256Sha256。
   - **安全消息模式：**符号和加密。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 在“用户凭证”下，选择匿名用户以允许匿名访问。如果在 OPC UA 配置期间已选择相应的选项，那么也可以提供经过身份验证的用户的用户凭证。所提供的用户凭证必须属于 InTouchHMIOPCUAWriteUsers 用户组。

6. 选择**测试**按钮。测试将会失败，但是它将下载 OPC UA 证书。

   重要事项：这次初步测试失败的原因是，必须信任客户端和服务器应用程序之间的证书。通过安装证书可解决此问题。

7. 转到下一节。

信任证书后，将需要验证 OPC UA 客户端配置。

有关详细信息，请参阅配置 OPC UA 数据源对象。

信任 OPC UA 服务器和 OPC UA 客户端之间的证书

在此发行版的 OPC UA 服务器服务中，必须手动执行创建 OPC UA 服务器和 OPC UA 客户端之间的信任这一操作。**测试**操作会导致 Gateway Communication Driver 将它自己的证书提交到 OPC UA 服务器节点，以便它可以得到信任。以下步骤显示接下来如何信任来自 OPC UA 服务器节点的客户端证书。如果您使用的不是 Gateway Communication Driver，请按照为第三方 OPC UA 客户端应用程序配置服务器和客户端证书中列出的过程进行操作。

1. 访问 **C:\ProgramData\AVEVA\PCS\OPC UA Rejected Client Certificates** 文件夹。

   这是缺省条件下最初放置来自客户端的证书的位置，以尝试连接到 OPC UA 服务器。

   注意：缺省条件下，ProgramData 文件夹处于隐藏状态。您可能需要在 Windows Explorer 中启用该隐藏的项目选项，才能对其进行查看。

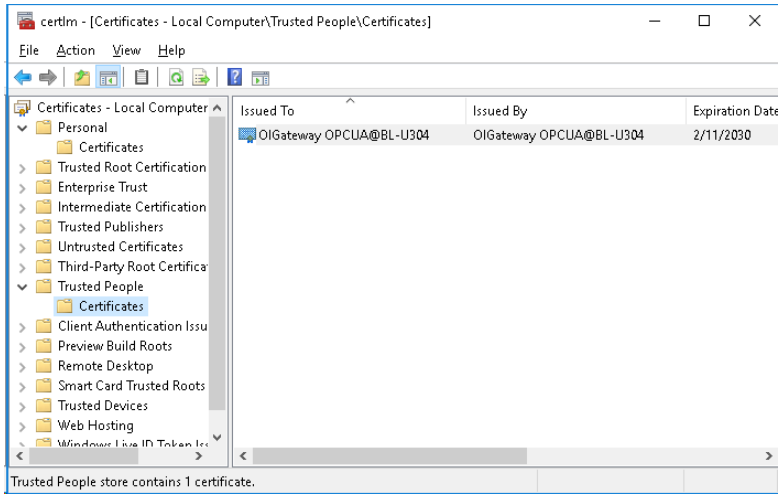
2. 使用鼠标右键单击证书名称，例如 **OIGatewayOPC UA@OPCUA client node{long hex ID}.der**。
3. 从上下文菜单中选择安装证书。这样即会打开证书导入向导。
4. 对于“存储位置”选择本地计算机，然后选择下一步。
5. 从**选择证书存储**列表中，选择受信任人作为证书存储。这是将使用 OPC UA 证书的唯一选项。
6. 关闭该向导以完成安装。
7. 最后，由于现在已安装该证书，因此将它从 **OPC UA Rejected Client Certificates** 文件夹中删除。

验证 OPC UA 证书安装

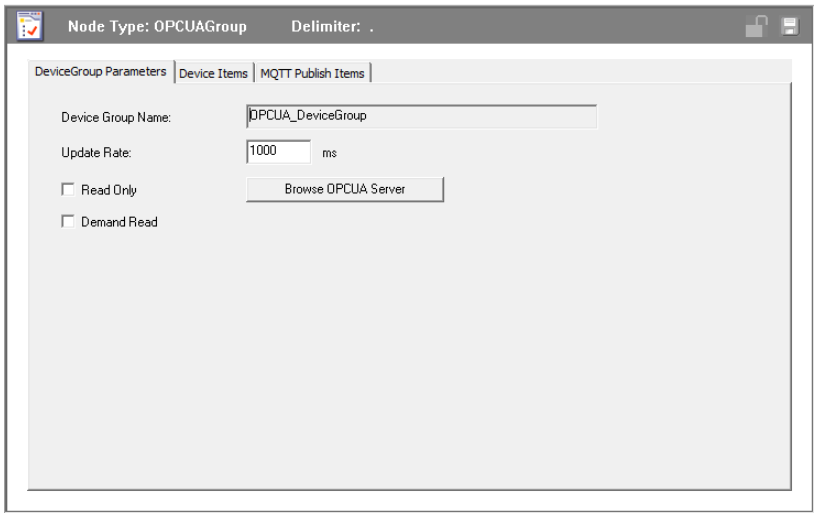
验证证书安装

1. 通过在 Windows 搜索窗口中键入 **certificate** 打开证书管理器，然后从搜索结果中选择管理计算机证书。**证书管理器**即会打开。
2. 导航受信任人文件夹，并检查是否已安装 OPC UA 证书。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

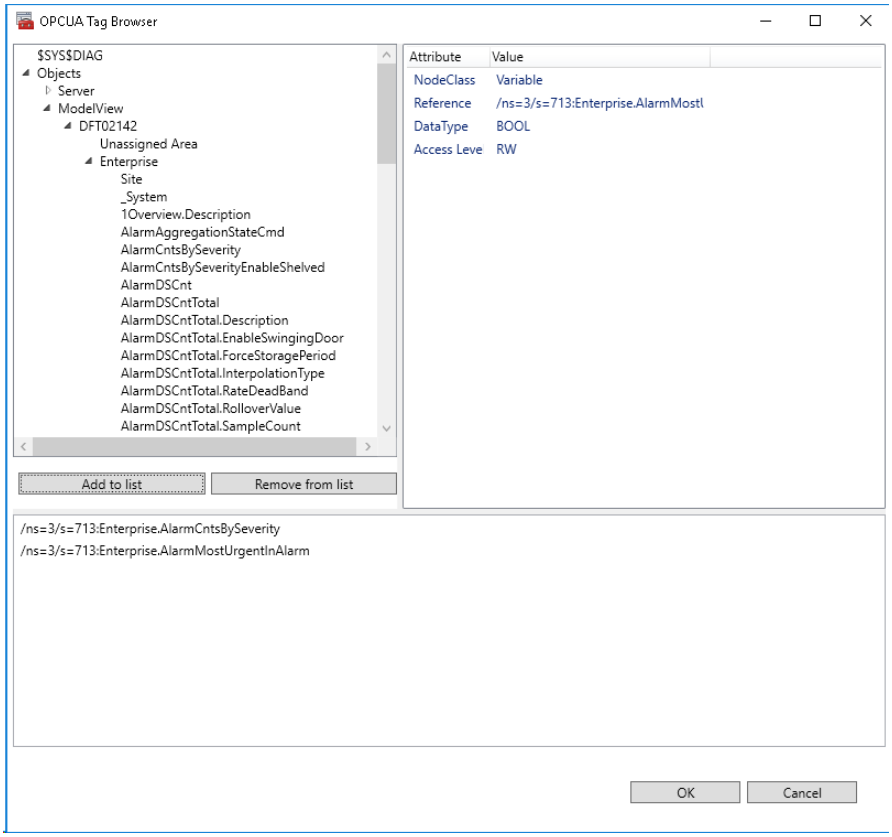


3. 从 OPC UA 客户端节点中，再次打开操作控制管理控制台，然后在 OPC UA 参数窗口中选择测试连接按钮。在该窗口的底部，将自动填充 **OPC UA 域名空间别**名列表，这表示连接已成功。
4. 在 OPC UA 连接下添加一个组。设备组名称将 **OPCUA_** 作为前缀添加到您所输入的组名称。



5. 选择浏览 **OPC UA 服务器**按钮以浏览 OPC UA 层次结构。
6. 可选：在别名列表中向监视器添加 OPC UA 标记，以有助于确保您可以浏览域名空间。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



7. 在**操作控制管理控制台**中切**换**回 OI Gateway 配置，可注意到**设备项目**窗口中**现**在列出了 OPC UA **标记**。
8. 缺省条件下，列表中的**项**目名称会重**复**完整的**项**目引用路径。根据需要将**项**目重命名**为**更好**记**的名称。

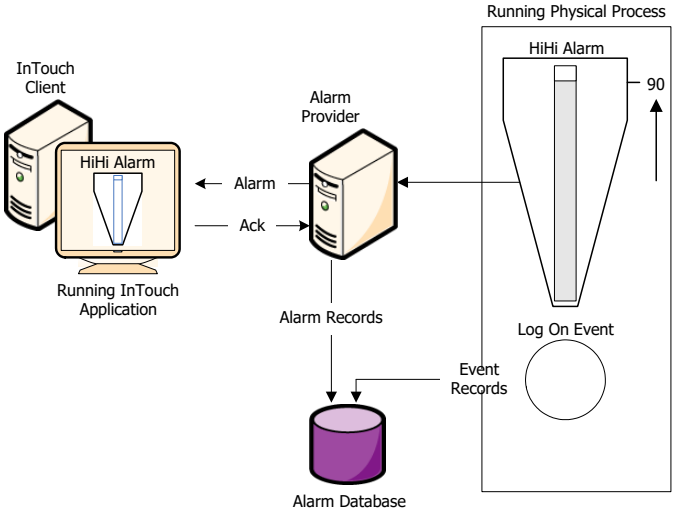
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 4 报警

通过创建可生成报警与事件的 InTouch 应用程序，可以通知操作员有关生产过程活动的状态。

- 报警向运行时操作员警告可能导致潜在问题的过程条件。通常，您设置一个在过程值超过定义的极限时触发的报警。操作员通常必须确认该报警。
- 事件代表正常的系统状态消息。通常事件在发生某种系统条件时触发，如操作员登录到 InTouch 应用程序。操作员不必确认事件。

下图显示 InTouch HMI 在应用程序运行期间如何处理报警与事件。报警与事件数据保存到报警数据库。



您可以配置任何标记来监视事件。每次标记值改变时，都有一个事件消息记录到报警系统。事件消息包含：值是如何改变的，是操作员、I/O、脚本还是系统促使了这个改变。

关于 InTouch 报警

报警代表可能导致问题并要求操作员作出响应的过程条件警告。报警通常在过程值超过用户定义的极限（如模拟值超过阈值上限）时触发。这会触发一个报警，以通知操作员有问题发生。操作员确认该报警之后，InTouch HMI 可识别出该报警已经得到确认。

您可以将 InTouch HMI 配置成即便引起报警的条件已消失但仍要求确认报警。这确保操作员可以了解到那些导致了临时报警状态但又已恢复正常的事件。

主要的报警状态在下表中介绍：

报警状态	条件
已确认	报警已确认。
ALM	报警已发生。
RTN	标记已从报警状态返回到正常状态。
LATCHED	已确认“UNACK_RTN”状态的报警，或报警从“ACK”状态返回到正常状态。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

报警优先级

您可以给报警指定一个优先级，或者说是严重度。例如，锅炉温度超出极限时，要求发出高优先级的报警，需引起立即关注。对于已到换班时间的报警，严重程度则可以低很多。报警优先级通常取决于环境 - 工厂应用、设备性质、安全性、备份系统的可用性、损害或停机的潜在成本等。

在定义标记时可以指定报警优先级。优先级范围可以是从 1 到 999，其中 1 表示最严重。

您可以指定报警优先级范围来代表报警的分类。例如，如果某个过程要求使用四种严重程度级别，则可以创建四个优先级范围。

报警严重程度	优先级范围
关键	1 – 249
主要	250 – 499
次要	500 – 749
提示性	750 – 999

范围对于过滤报警非常有用。例如，您可以配置一个报警显示，从所有的报警中过滤出关键报警。您可以创建动画链接、确认脚本以及过滤的查看与打印，所有这些都可以基于报警优先级范围。

报警子状态

多状态报警包含一系列报警子条件。
例如，模拟报警通常有多个极限。

- High 与 Low 阈值设置正常操作范围的边界。
- HiHi 与 LoLo 极限表示极度偏离正常值范围的偏差。

锅炉温度水平可以由于任何这些子状态之一而处在报警条件下。在继续处于整体报警条件时，锅炉温度还可以在任何两个子状态之间转换。

报警确认

报警发生时，运行时操作员（或系统）必须确认报警。确认只是表示有人注意到该报警。这与采取修正操作没有关系，后者可能不会立即发生。与报警条件是否返回到正常也没有什么关系 — 有时即便没有任何外界干预，它也可能自行恢复正常。

较高或中等优先级的报警通常需要立即确认，而优先级非常低的报警可能不需要。尽管生成报警的条件可能已消失（例如，温度上升得过高，然后又降了下来），但在确认之前，报警本身并不会被认为已得到解决。

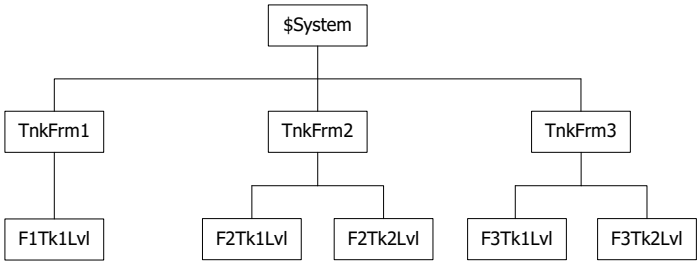
一旦确认报警，操作员即可解除报警。在启用锁存功能后，会将返回到正常状态的已确认报警置于 LATCHED 报警状态。操作员可以解除 LATCHED 报警，以从报警客户端控件网格的当前模式中删除 LATCHED 报警，但这些报警在报警客户端控件的最近模式中仍然可见。

报警组

您可以将报警分组，以便更轻松地进行跟踪与管理。报警组是工厂的不同区域、设备的各个部件、操作员的责任或生产过程的逻辑表示。

例如，下图显示贮油站应用的三层报警组层次结构。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



对于过滤报警显示、Alarm Printer 以及确认脚本，“报警组”都非常有用。

每个标记都与一个报警组关联。缺省情况下，标记指定给 $System 主组。您可以在 $System 组下创建其它报警组层次结构，最多可达 32 级。

在“标记名字典”中定义标记时，可以创建报警组，并将标记同它们关联起来。

报警组与组变量同 SmartSymbol 不兼容。您无法在 SmartSymbol 中使用对报警组或组变量的引用。

关于 InTouch 事件

事件是系统中发生的某些事情的可检测到的实例，它不一定要与报警关联。进入或脱离某种报警状态的转换过程便是事件的一种。事件也可以是操作员的动作、系统配置的改变或某种系统错误。

事件和条件不同。条件可以持续几分钟、几小时、几天，甚至几周。事件则是瞬时的；它在发生之后便立即结束。报警是一种条件；而报警通知则是一个事件。

事件代表正常的系统状态消息，不要求操作员作出响应。

定义标记进行事件监视时，可以选择在每次标记值改变时，都打印事件消息或将其记录到报警系统。事件消息包含：值是如何改变的，是操作员、I/O、脚本还是系统促使了这个改变。

事件可以是以下类型之一：

事件	条件
OPR	操作员使用“值输入”修改了标记值。
LGC	QuickScript 修改了标记值。
DDE	从 DDE 客户端插入了标记值。
PROT	从工业图形（工业图形中的自定义属性或按钮）启动了标记值改变。
SYS	发生了系统事件。
USER	$Operator 发生了改变。

不论是否启用了任何标记的事件记录功能，系统都将生成 SYS 与 USER 事件。DDE、OPR 和 LGC 事件与标记值相关，只有那些已启用事件记录的标记才会生成这些事件。

InTouch 报警的类型

在 InTouch HMI 中，报警根据其特性分为一些常见的类别。这些类别也就是所谓的“类”与“类型”。“分布式报警”系统将所有报警归类到五种基本条件下：离散、值、偏差、变化率以及 SPC。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

报警条件	分布的类	分布的类型
离散	DSC	DSC
值 - LoLo	值	LOLO
值 - Low	值	LO
值 - High	值	HI
值 - HiHi	值	HIHI
偏差 - 主	DEV	MAJDEV
偏差 - 副	DEV	MINDEV
变化率	变化率	变化率
SPC	SPC	SPC

在定义标记时，您可以将每个 InTouch 标记关联到某个报警条件。根据标记的类型，可以给它定义一个或多个报警类或类型。

离散报警

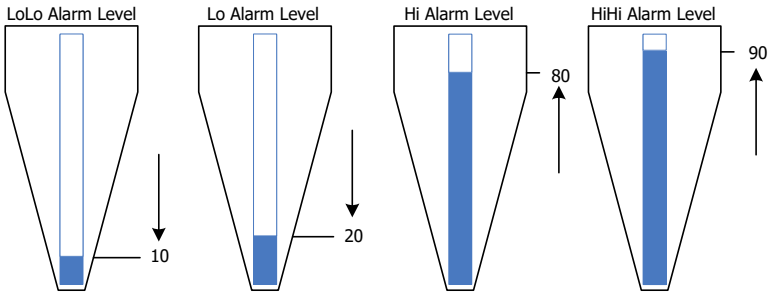
离散报警对应于有两种可能状态的离散标记。创建离散标记时，您可以将报警状态配置成对应于离散标记的“真”还是“假”状态。

模拟报警

模拟报警对应于同整数或实数关联的模拟标记。在模拟报警类型中，存在多个子类型：值、偏差及变化率。

值报警

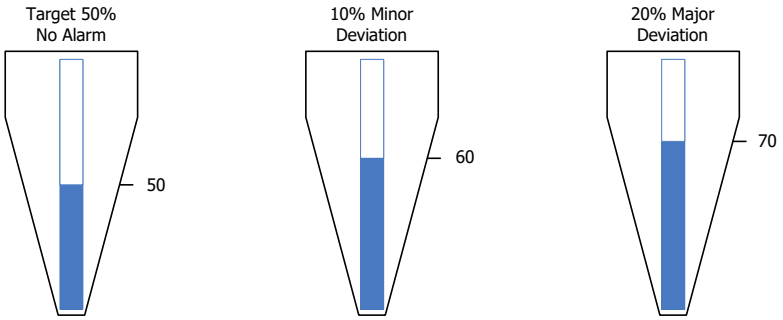
当前标记值与一个或多个预先确定的极限进行比较。如果值超过极限，则声明处于报警状态。您可以分别配置“LoLo”、“Lo”、“Hi”及“HiHi”等极限的值与优先级，并指明是否要使用每个极限。



偏差报警

当前标记值与目标值进行比较，然后将差异的绝对值与一个或多个极限进行比较，这些极限使用标记值范围的百分比来表示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您可以分别配置副偏差极限与主偏差极限的值与优先级，并指明是否要使用每个极限。您也可以配置偏差死区的值，它也是使用标记范围的百分比来表示的。这可以控制标记值必须改变多少个百分点（占总计范围的）才能被认为处于报警状态。

例如，您可以按以下方式配置极限：

- 范围从 0 到 100
- 目标 50
- 副偏差 10%，即将副偏差阀值设置为 40 与 60。
- 主偏差 20%，即将主偏差阀值设置为 30 与 70。
- 死区 10%

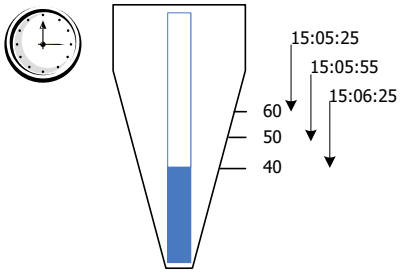
如果标记值是 39，则发生副偏差报警。不过在评估并清除报警之前，标记值至少必须变为 50（40 加上死区值 10）。

如果标记值是 72，则发生主偏差报警。在该报警清除之前，标记值至少应该降到 61。

变化率报警

在一个测量周期中，比较标记的当前值与前一个值。标记值的变化率使用前一个标记值、前一次变化的时间、当前标记值以及当前时间来计算。

只要标记值发生变化，就会对标记进行测试以确定是否发出变化率报警。一个例外是应用程序在 WindowViewer 中开始运行时的初始标记值。在这种情况下，将忽略初始值，并且第一次变化率比较在标记值的第二次和第三次变化之间进行。之后，在连续的标记值变化之间进行变化率测量。



如果连续标记变化的绝对差值超过指定极限，将发出变化率报警。变化率极限以一段时间间隔（可以是每秒、每分钟或每小时）内标记值的百分比来表示。您可以配置变化率极限的值与优先级，以及是否要使用该极限。

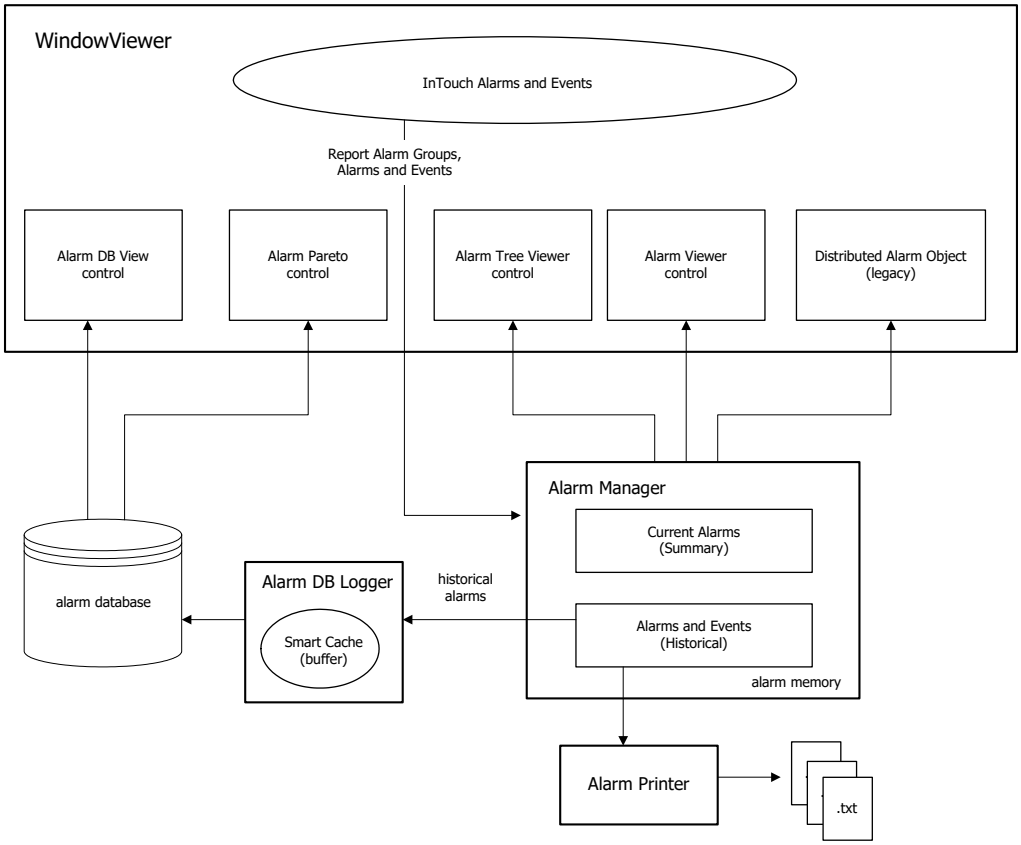
InTouch 分布式报警系统

“分布式报警”系统由以下部分组成：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- “报警管理器”，管理当前活动报警（摘要报警）以及历史报警与事件。摘要与历史报警保存在 InTouch 内部报警内存中。
- Alarm DB Logger，将历史报警与事件存储到报警数据库。报警数据库是一个 SQL Server 数据库。
- Alarm Printer，打印历史报警与事件。
- 一组 ActiveX 控件，在运行时从内部报警内存或报警数据库中检索报警与事件。

下图显示系统概况。



要点：“分布式报警”系统作为一组 Windows 服务运行。为减少以管理员权限运行“分布式报警”系统时的安全隐患，针对这些服务，用户帐户权限已设置为非交互式。

运行时操作员可以使用“分布式报警”系统：

- 显示、记录及打印由本地 InTouch 应用程序以及其它网络应用程序的报警系统生成的报警与事件。
- 在本地或从远程网络节点上确认报警。
- 在 InTouch 应用程序中使用报警 ActiveX 控件显示预先配置的报警显示。
- 使用单独的报警注释字段提供关于报警的更多反馈。

作为应用程序开发人员，您可以：

- 通过点域来控制报警。
- 配置报警，以便在应用程序的完全控制下，直接或间接启用或禁用它们。您可以将报警抑制应用于单个报警类、标记或报警组，以禁止在特定的视图节点上显示报警信息。系统范围的禁用可以从源头阻止报警活动。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 配置要记录到历史中的报警信息。Alarm DB Logger 可以作为 Windows 服务运行，也可以根据需要手工启动。报警记录功能使用 UTC (GMT) 时间标签，并提供夏令时与跨时区兼容性。
- 设置故障转移报警供应器。如果主报警供应器出现故障，“分布式报警”系统可以顺利地从备份系统获取报警信息。重新连接主节点之后，在恢复使用主系统之前，“分布式报警”系统会确保已经重新同步报警确认。

“分布式报警”系统：

- 通过 SuiteLink 协议发送数据，使用最少的 CPU 与网络资源。
- 在报警发生时，而不是在接收器接收报警时，给报警加上时间标签。时间标签包含毫秒。

报警供应器与接收器

在任何给定的节点上，都会有一组“报警供应器”（发布者）与“报警接收器”（预订者）。InTouch“分布式报警系统”提供通讯链接，在节点与软件组件之间传递报警信息。

报警供应器

报警供应器：

- 跟踪报警项 – 即可进入报警状态的项目 – 并向“分布式报警系统”提供这些项目的列表，包括有关任何项目分组层次结构的信息。
- 在报警项的状态改变时，通知“分布式报警系统”。状态改变包括项目是进入还是脱离报警状态，以及是否已确认最新的报警。
- 跟踪报警项是否已禁用。

InTouch HMI 支持外部报警供应器，如 QI Analyst、Application Server Galaxy，以及使用 Alarm API 工具包构建的其它软件。这些报警记录的日期/时间标签由报警感应器提供，而不是由“分布式报警”系统生成。

报警接收器

报警接收器：

- 向“分布式报警”系统提供一组查询，确定它希望收到其通知的报警项。在报警接收器更改或删除查询之前，它保持活动状态，并指定报警供应器或报警组 - 这与使用“通配符”的 SQL 查询非常相似。只要报警供应器发出变化通知，“分布式报警”系统便检查与注册的任何查询匹配的报警，然后将更新传递给相应的报警接收器。
- 在收到更新时，显示或记录同项目的状态或其转换有关的信息。
- 确认报警。报警接收器发送确认通知给“分布式报警系统”，确定报警与报警供应器。该通知传递给报警供应器，后者随之将项目状态更新为已确认（如果适用），并进而通知“分布式报警系统”，从而确保将更新发布给所有感兴趣的报警接收器。

注意：“分布式报警系统”中的大部分通讯活动是将报警查询与报警记录从一个节点发送到另一个节点。在节点内，报警查询与报警记录由内部报警内存进行跟踪并缓存，以最大限度减少网络流量。

分布式报警组列表

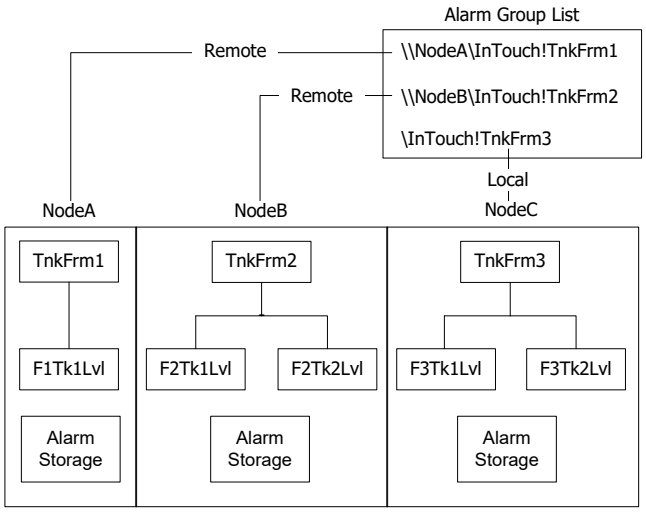
“分布式报警系统”使用报警组将报警整理到本地目录树视图中。分布式报警显示使用目录树视图过滤报警。您可以从网络上的多个节点来查看这些报警组。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

“分布式报警系统”使用一个报警组列表将本地与远程节点上的报警组合并到一起。报警组列表是一个有名称的列表，它由 InTouch 节点以及那些节点中的每一个上定义的报警组所组成。它也可以包含其它报警组列表名与本地报警组。报警接收器（如 Alarm Viewer 控件）使用此列表查询报警。

在查询别名下，Alarm Viewer 控件可以显示合并到一起的报警，这些报警来自属于该列表的各个组。这些报警可以在本地 InTouch 节点或是从网络中的远程节点上进行确认。

下图显示一个报警组列表，它将三个节点上的报警组合并到一起。报警组列表在 NodeC 本地进行定义。剩余的报警组都来自远程节点。



“分布式报警显示”显示对列表中所有的报警组进行查询所产生的报警。例如，如果有兴趣显示多个 InTouch 节点上所有的贮油站报警，则可以创建一个名称为 TankFarmAlarms 的列表。对于此列表，您可以将运行贮油站 InTouch 应用程序的所有节点上的报警组都添加到其中。

如需有关创建报警组的详细信息，请参阅创建报警组。

摘要报警与历史报警

摘要报警是当前活动的报警。历史报警是当前不活动并且通常存储在报警数据库中的报警。

例如，您可能希望查看在等待确认的所有当前报警的摘要，而所有其它报警信息则属于过去的历史，并非十分紧要。

报警禁用、约束及抑制

您可以“关闭”报警或者忽略它们，但并不实际删除报警配置。您可以禁用、约束或抑制某个报警。

报警禁用与约束在报警供应器上进行控制。抑制则在报警接收器上进行控制。如需有关供应器与接收器的详细信息，请参阅报警供应器与接收器。

- **禁用**。通过设置标帜将报警标记为已禁用，可以在报警供应器上禁用报警。无论发生什么报警条件，该项都绝不会进入报警状态。如需有关可用于禁用报警的点域的详细信息，请参阅启用与禁用标记或报警组的报警。

  您可以一次禁用或启用某个标记的所有报警。此外，对于包含子状态的报警，也可以分别禁用每个子状态。

- **约束**。您可以按照以下方式来约束报警：

  a. 在 WindowMaker 中将“约束”标记添加到报警配置。约束标记用于在运行时将报警标记为受约束。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

b. 在运行时将约束标记设置为“真”或“假”。约束标记为“假”时，报警按正常方式进行处理。约束标记为“真”时，该项目无法报警。

每个报警子状态都可以使用不同的标记进行约束，您可以不给一些子状态指定约束标记。

将标记指定为某个报警的约束标记时，会增加其交叉引用使用计数。

- **抑制**。抑制会导致报警接收器忽略特定的报警。如果某个报警与排除标准匹配，则它是不可见的。这就是说，它不出现在该特定报警接收器的显示中，也不能在其中进行打印或记录。

  实际的报警生成完全不受抑制的影响。报警记录仍然可以记录到报警历史中。

如果某个报警变为禁用或有效约束状态，而项目仍处于报警状态，则该项目会被强制转换到一个不同（有效）的状态。具体的状态应取决于当前有哪些可用的状态，以及它们是否也被禁用。此活动由报警供应器根据报警的类型、极限值等进行处理。

受禁用或有效约束的报警不会等待确认。如果报警有子状态，它仅可以在仍旧可用的子状态上等待确认。

终端服务报警支持

通过在 Terminal Services for InTouch 中使用“分布式报警系统”，运行在不同终端会话上的报警客户端可以选择要显示哪些报警数据以及如何显示。

“报警供应器”通过唯一确定应用程序及应用程序实例的名称来识别它们。“报警供应器”或“报警接收器”向“分布式报警系统”注册之后，此信息将可供“分布式报警系统”使用。

运行“报警供应器”的节点则通过在系统中唯一确定该计算机节点的名称进行识别。Terminal Services Edition (TSE) 客户端会话所生成的报警记录中包括扩展的节点名称，其格式为：节点:IP 地址；例如，serverAlarm:192.168.1.23。在计算机节点上启动“分布式报警系统”的实例之后，此信息便可供它使用。

在记录报警事件时，节点与完整的“报警供应器”名可以确定报警的来源。

在“终端服务”环境中确认报警时，记录的“操作员节点”将是客户端机器的名称，相应的操作员从这些客户端机器中建立“终端服务”会话。如果节点名无法检索，将使用节点的 IP 地址代替。

终端服务器客户端会话不能是 InTouch 报警供应器。

分布式报警系统数据储存

“分布式报警系统”使用多种数据存储形式：

- **内部报警内存（缓冲区）**

  有关当前与最新报警的大多数信息都保存在各计算机节点的内存中。InTouch 使用两个内存位置：一个用于摘要（当前）报警，另一个用于历史报警与事件。此模型也用在“分布式报警系统”中。

  摘要报警的内存可根据需要进行扩展，以容纳所有的当前报警，直至达到可用内存的限制。历史报警的内存只能增长到预设的限制。在历史内存达到此限制之后，随着新的报警记录添加进来，最旧的报警记录会被丢弃。在多节点环境中，各节点上的报警内存构成一个报警内存集合。

  如需有关设置该限制的信息，请参阅配置报警缓冲区大小。

- **报警数据库**

  Alarm DB Logger 创建一个数据库，以跟踪报警发生、子状态转换、确认及恢复正常的时间。实际上，这些记录会形成系统中的报警历史。

“分布式报警系统”以查询为基础，因而支持使用一个计算机节点来记录多个其它节点的报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Build

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 5 管理 InTouch HMI 应用程序

管理 InTouch HMI 应用程序时，您可以：

- 创建或删除 InTouch 应用程序。请参阅创建 InTouch 应用程序和从应用程序管理器中删除 InTouch 应用程序。
- 在 WindowMaker 或 WindowViewer 中打开应用程序。请参阅在 WindowMaker 与 WindowViewer 中打开应用程序。
- 搜索应用程序。请参阅查找 InTouch 应用程序。
- 将应用程序移到不同的计算机上。请参阅将应用程序发布到远程节点。
- 将应用程序发布到多个计算机上。请参阅《*AVEVA™ InTouch HMI 应用程序部署*》中的分发应用程序。
- 管理 InTouch 服务。请参阅关于管理 InTouch 服务。
- 导入或导出标记定义、窗口及脚本。请参阅导出与导入 InTouch 组件。
- 配置安全性。请参阅《*AVEVA™ InTouch HMI 管理*》中的保护 InTouch 安全。

您可以按照以下方式扩展应用程序：

- 将文本字符串与报警注释翻译成不同的语言。请参阅《*AVEVA™ InTouch HMI 应用程序运行时*》中的在运行时切换语言。
- 将应用程序与 Application Server 集成。请参阅托管的 InTouch 应用程序和关于在运行时查看应用程序。
- 将应用程序显示到多个监视器上。请参阅关于设置多监视器系统。
- 在 Tablet PC 上使用应用程序。请参阅关于在 Tablet PC 上使用 InTouch。

对于 Windows Vista、Windows 7 和 Windows Server 2008 操作系统，标准用户可以使用“InTouch 应用程序管理器”查找应用程序和打开 WindowViewer。应用程序属性对于标准用户为只读。从应用程序管理器进行的所有读取/写入或配置操作都需要管理权限。

启动应用程序管理器

您可以从开始菜单或使用计算机桌面的快捷方式来启动“应用程序管理器”。

第一次启动应用程序管理器

1. 在任务栏上，选择开始，指向程序，指向 **AVEVA InTouch HMI**，然后选择 **InTouch HMI 应用程序管理器**。

   此时打开 **AVEVA 应用程序管理器**。

   在配置器中启用 **Operations Control 连接体验**模式后，首次启动“应用程序管理器”时，系统将提示您使用 AVEVA Identity Manager 进行身份验证。

2. 使用 AVEVA Identity Manager 进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

   窗口显示您使用“应用程序管理器”在计算机上创建或找到的 InTouch 应用程序。

确定上一次用于保存应用程序的 InTouch 版本

1. 打开“应用程序管理器”。
2. 在查看菜单上的视图组中，选择详细信息。

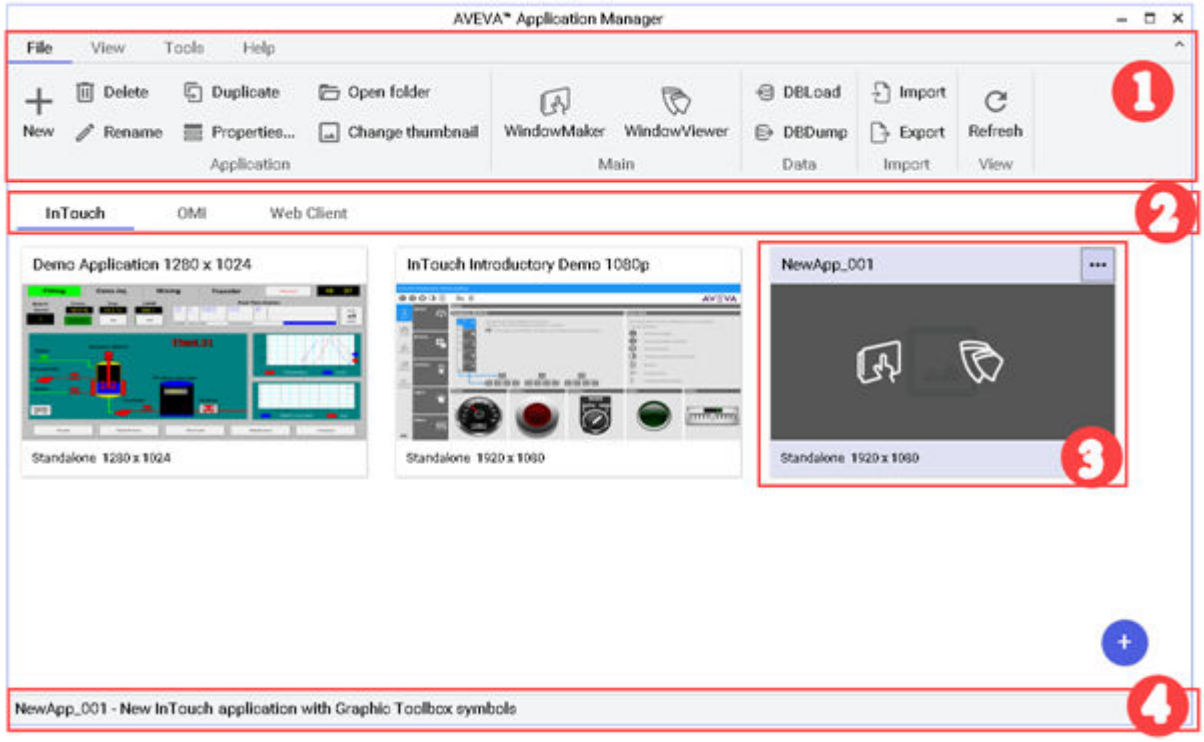
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时“详细信息”网格视图中出现应用程序列表。

3. 请参阅应用程序的版本列。

使用应用程序管理器

“应用程序管理器”是创建 InTouch HMI 应用程序的第一步，并且提供可用于应用程序管理的多个选项。界面分为功能区和工作区/画布。



标识符	元素	描述
1	功能区栏	顶部包含一些菜单项，每个菜单项上都有按功能分组的命令。
2	应用程序栏	显示 InTouch、OMI 和 Web 客户端的选项卡。 功能区中每个菜单项上的命令取决于所选应用程序的类型。
3	应用程序磁贴	代表每个应用程序。
4	状态栏	显示所选项的描述消息。

功能区包含以下菜单项：

- 文件：文件菜单是缺省功能区菜单。它有五组相关命令：应用程序、主、数据、导入和视图。您可以使用诸如创建、复制、删除、在 WindowMaker 或 WindowViewer 中打开、导入、导出等命令以及许多其它常见功能来修改应用程序设置。
- 查看：视图菜单提供了用于更改画布上列表视图的选项，如列表、平铺或详细信息。
- 工具：工具菜单包含重要的“应用程序管理器”工具，如节点属性、OPC UA 配置、查找和 Insight Publisher。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- **帮助**： **帮助**菜**单**允**许访问**“InTouch HMI 帮助”文档，以及**查**看“**应**用程序管理器”版本和版**权**版本。

工作区/画布分**为**多个**选项**卡，**每**个**选项**卡代表一**种应**用程序**类**型 - InTouch、OMI 和 Web 客**户**端。

创建 InTouch 应用程序

您可以使用**应**用程序管理器**创**建新的 InTouch **应**用程序。**应**用程序路径不得超**过** 114 个字符，包括网**络驱动**器**盘**符、冒号及所有的反斜杠。如果超**过**限制，**则**无法在 WindowMaker 中打**开应**用程序。**应**用程序名称必**须为** 32 个字符或更少。

创建**应**用程序**时**会**创**建 INTOUCH.ini 文件。INTOUCH.ini 文件包含**应**用程序的缺省配置**设**置。配置**应**用程序**时**，新**设**置会写入 INTOUCH.ini 文件。

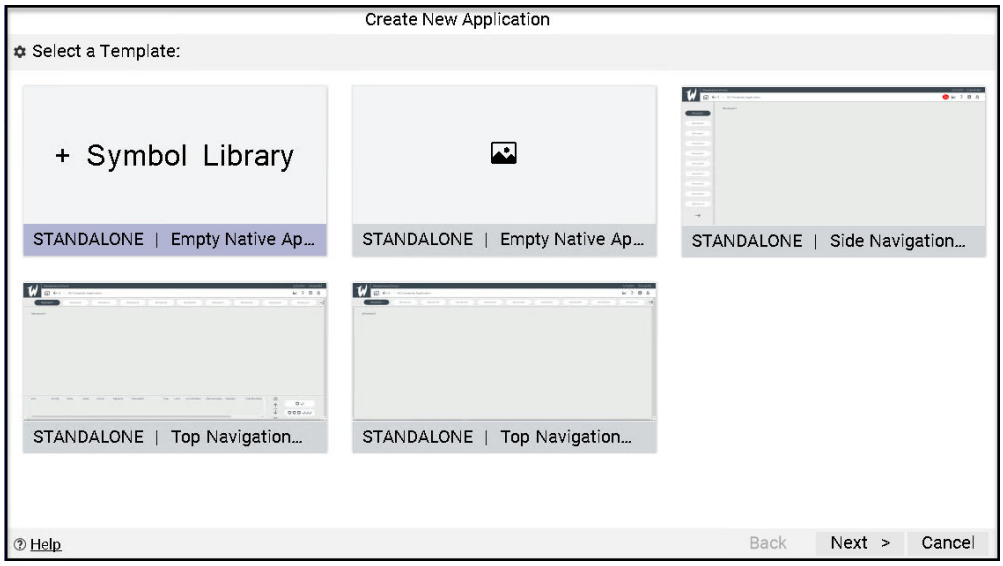
创建**应**用程序之后，您可以将**现**有的 INTOUCH.ini 文件**复**制到**应**用程序文件**夹**。**这样**，您就不必在**每**次**创**建新的**应**用程序**时**配置自定**义**的 InTouch 参数。

注意：您必**须**是 InTouchDevelopers OS 用**户组**的成**员**才能**创**建独立的 InTouch **应**用程序。

创建新的应用程序

1. 您可以使用不同**选项**在“**应**用程序管理器”中**创**建新的**应**用程序。
   - 在**文件选项**卡上的**应用程序组**中，**选择新建**。
   - **选择**右下角的 。
   - 按 Ctrl+N。
   - 使用鼠**标**右**键单击**空白区域，然后**选择新建...**。

   此**时**出**现创建新应**用程序 - **选择**模板页面。



2. 突出**显**示新**应**用程序的模板，然后**选择下一步**。

   空模板将不包括缺省的**图**形工具箱符号和 SAL 符号。在所有模板中，都可以在**应**用程序中同**时**使用 InTouch 符号和工**业图**形。如果您已**创**建任何**应**用程序模板并且在正确的文件**夹**中提供**这**些模板，**则**也将**显**示**这**些模板。

   此**时**出**现创建新应**用程序 - **输入应**用程序**详细**信息页面。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 输入以下详细信息：
   - **类型**：显示所选应用程序的类型。
   - **应用程序名称**：输入应用程序的名称
   - **目录名称**：输入应用程序文件夹名称。
   - **应用程序路径**：选择省略号 (...) 按钮可浏览非缺省位置的文件夹。
   - **设置缺省目录**：使用该开关将应用程序路径设置为缺省目录。
   - **分辨率**：从下拉列表中选择应用程序目标分辨率。

     可用的分辨率选项为：
     - 屏幕分辨率（缺省）
     - 1024 x 768
     - 1280 X 1024
     - 1366 X 768
     - 1440 X 900

     要编辑宽度和高度，请从“分辨率”下拉列表中选择自定义分辨率。
   - **宽度**：指定应用程序的宽度。
   - **高度**：指定应用程序的高度。
   - **描述**：输入可选的描述，最多可以输入 255 个字符。

4. 选择完成。

   应用程序创建完成后，将显示在应用程序管理器的应用程序列表中。

如需了解更多信息，请观看标题为“Creating an Application（创建应用程序）”的视频。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建新的 InTouchView 应用程序

通过在创建应用程序时设置“应用程序管理器”中的某个选项，可以确定 InTouchView 应用程序。您可以将 InTouch 应用程序配置为 InTouchView 应用程序，反之亦然。运行从 InTouchView 转换到 InTouch 的应用程序时，要求使用完整版的 InTouch 许可证。当应用程序更改为 InTouchView 应用程序时，获得的许可证将根据所选数据源的不同而有所不同。如需详细信息，请参阅 InTouchView 应用程序。

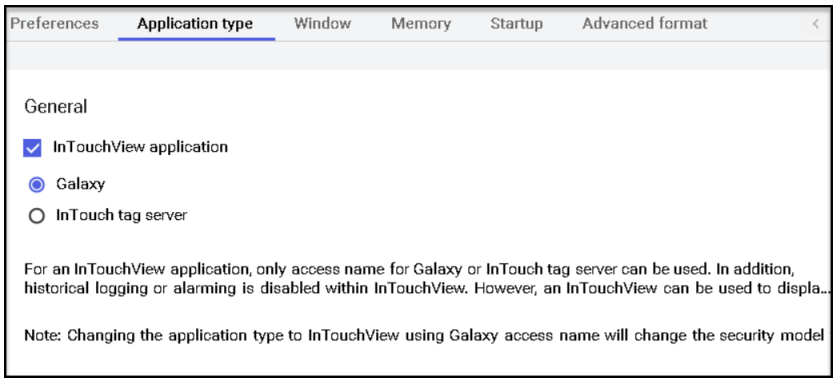
将 InTouch 应用程序更改为 InTouchView 应用程序

如果应用程序仅需要连接到 Application Server 或 InTouch 标记服务器的数据，则可以将 InTouch 应用程序更改为 InTouchView 应用程序。某些 WindowMaker 功能在 InTouchView 应用程序中不可用。

将 InTouch 应用程序更改为 InTouchView 应用程序

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在文件菜单上，突出显示配置，然后选择 **WindowViewer**。

   此时出现 **WindowViewer** 配置屏幕。
3. 在应用程序类型选项卡中，选中 **InTouchView 应用程序复选**框。



   - 要连接到 Galaxy 中的数据，请选择 **Galaxy**。
   - 要连接到 InTouch 标记服务器中的数据，请选择 **InTouch 标记服务器**。
4. 在主页菜单上的标记组中，选择访问名。
   a. 如果选择 **Galaxy**，您必须删除 Galaxy 之外的所有“访问名”，才能将 InTouch 应用程序转换为 InTouchView。如果这些“访问名”没有删除，则会在尝试转换期间显示一条消息。
   b. 如果选择 **InTouch 标记服务器**，您可以配置新的访问名。
      选择添加，并提供访问名和节点名称。

      应用程序名称和主题名变灰。在将此应用程序更改为 InTouchView 之前，必须删除除 Galaxy 或引用 InTouch 标记服务器以外的所有访问名。只有来自其中应用程序被配置为*视图*的访问名的 I/O 引用将被绑定，来自访问名 *Galaxy* 的 I/O 引用不会被绑定。
   c. 选择确定。

      如需有关配置访问名的详细信息，请参阅设置访问名。
5. 选择关闭。
6. 选择确定。出现消息时，选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在 WindowMaker 与 WindowViewer 中打开应用程序

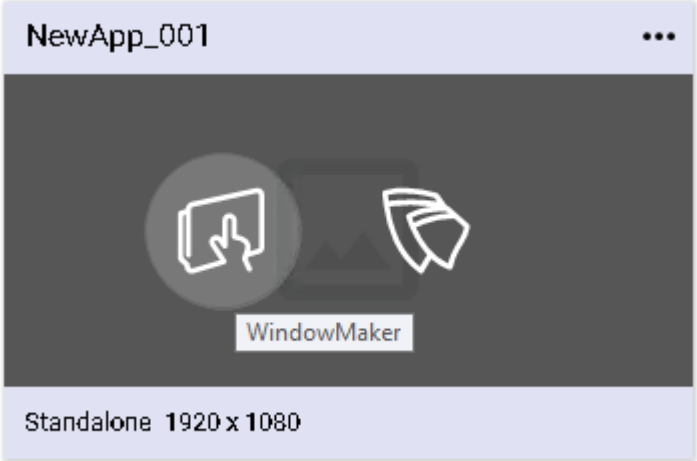
在 WindowViewer 中打开新的应用程序之前，必须先在 WindowMaker 中打开它。

注意：您需要是 InTouchDevelopers 组的成员才能启动 WindowMaker。

在 WindowMaker 中打开一个应用程序

1. 在<1>“</1>应用程序管理器<2>”</2>窗口中选择应用程序<3>。</3>
2. 在文件菜单上的主组中，选择 **WindowMaker**。

   或者，将光标悬停在应用程序磁贴上，然后选择 **WindowMaker**。



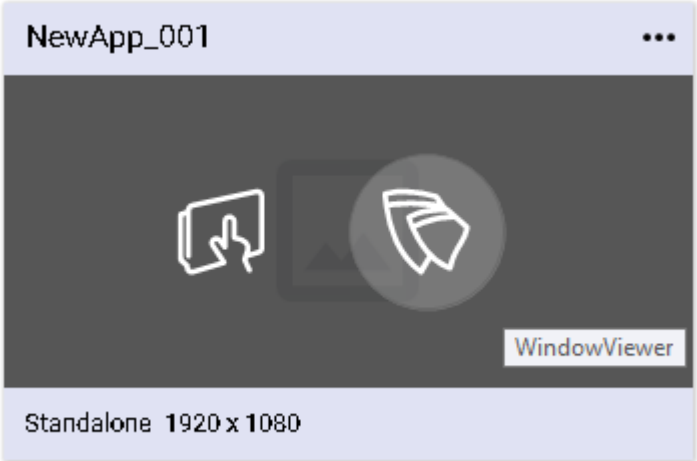
   应用程序即会在 WindowMaker 中打开。

提示：您也可以双击应用程序以便在 WindowMaker 中打开它。

在 WindowViewer 中打开应用程序

1. 在<1>“</1>应用程序管理器<2>”</2>窗口中选择应用程序<3>。</3>
2. 在文件菜单上的主组中，选择 **WindowViewer**。

   或者，将光标悬停在应用程序磁贴上，然后选择 **WindowViewer**。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

应用程序即会在 WindowMaker 中打开。

自定义应用程序管理器窗口

您可以隐藏工具栏与状态栏。您也可以更改在“应用程序管理器”窗口中列出应用程序的方式。应用程序可以以“详细信息”、“列表”和“图标”形式显示。

- **详细信息** - 此视图以表格形式列出应用程序及所有应用程序相关属性（如路径和分辨率）。
- **列表** - 此视图显示应用程序名称、类型、缩略图和功能区。
- **图标** - 此视图显示应用程序名称、类型、分辨率、描述、缩略图和功能区。

显示/隐藏工具栏

- 在查看菜单上，选择工具栏以显示或隐藏所选应用程序的状态。

显示/隐藏状态栏

- 在查看菜单上，选择状态栏以显示或隐藏所选应用程序的状态。

更改应用程序列表的视图

- 在查看菜单上，突出显示适当的命令或选择工具栏上的对应按钮。

重新排列工具栏选项的顺序

1. 选择 ⋮ 图标，然后选择自定义...
2. 在项目选项卡下，突出显示一个选项（例如 WindowMaker），然后选择向上移动或向下移动按钮更改工具栏上的选项显示顺序。

删除工具栏上显示的工具

- 在查看菜单上，选择该选项对应的复选框。

1. 例如，清除 WindowMaker 选项对应的复选框，然后选择关闭。Window Maker 选项将不会显示在工具栏上。

将主题应用到“应用程序管理器”

- 在查看菜单上的主题组中，选择浅色主题/深色主题以在这两个主题之间进行切换。

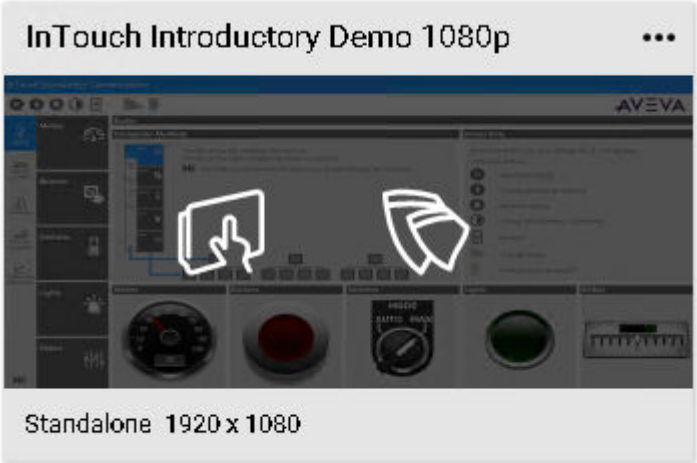
使用应用程序磁贴

在图标或列表视图中，每个应用程序都显示为一个含有应用程序特定选项的磁贴。

启动 WindowMaker 或 WindowViewer

将光标悬停在应用程序磁贴上，以查看用于在 WindowMaker 或 WindowViewer 中启动应用程序的选项。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



其它设置

选择省略号图标可查看其它设置。

- **WindowMaker**：选择以启动 WindowMaker。

  注意：首次启动 WindowMaker 时，可能需要使用 AIM 进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

- **WindowViewer**：选择以启动 WindowViewer。

  注意：首次启动 WindowViewer 时，可能需要使用 AIM 进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

- **应用程序文件夹**：选择导航到应用程序文件夹。
- **属性**：单击可启动应用程序属性窗口。
- **更改缩略图**：突出显示图标，然后从“浏览”对话框中选择图像。图像将显示在图标和列表视图中。
- **重命名**：选择并输入应用程序的新名称。
- **DBLoad**：选择可启动 DBLoad 实用程序。
- **DBDump**：选择可启动 DBDump 实用程序。
- **导出为模板**：选择以导出 InTouch 应用程序。请参阅导出 InTouch 应用程序以用作模板。
- **删除**：选择以删除应用程序。

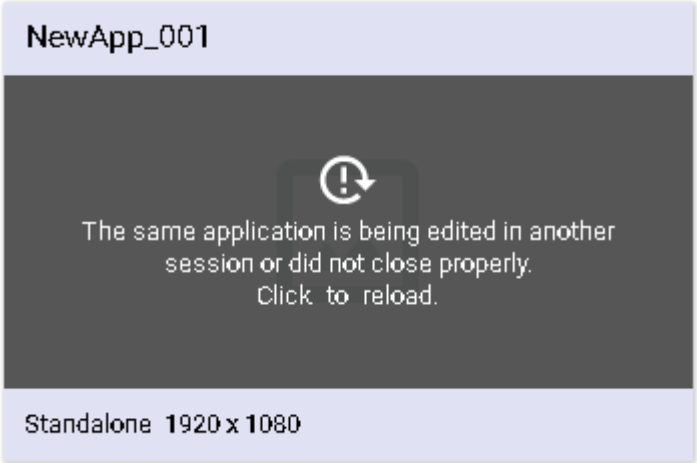
锁定与解锁 InTouch 应用程序

如果正在另一个会话中对应用程序进行编辑，或者未正确关闭应用程序，则该应用程序可能被锁定。您必须使用“应用程序管理器”解锁应用程序。

解锁 InTouch 应用程序

- 启动“应用程序管理器”。
- 选择当前锁定的应用程序。
- 选择应用程序磁贴以解锁并重新加载应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



应用程序将解锁并可供使用。如需详细信息，请参阅应用程序编辑锁定。

修改 InTouch 应用程序

您可以重命名应用程序以及更改应用程序属性。

重命名应用程序

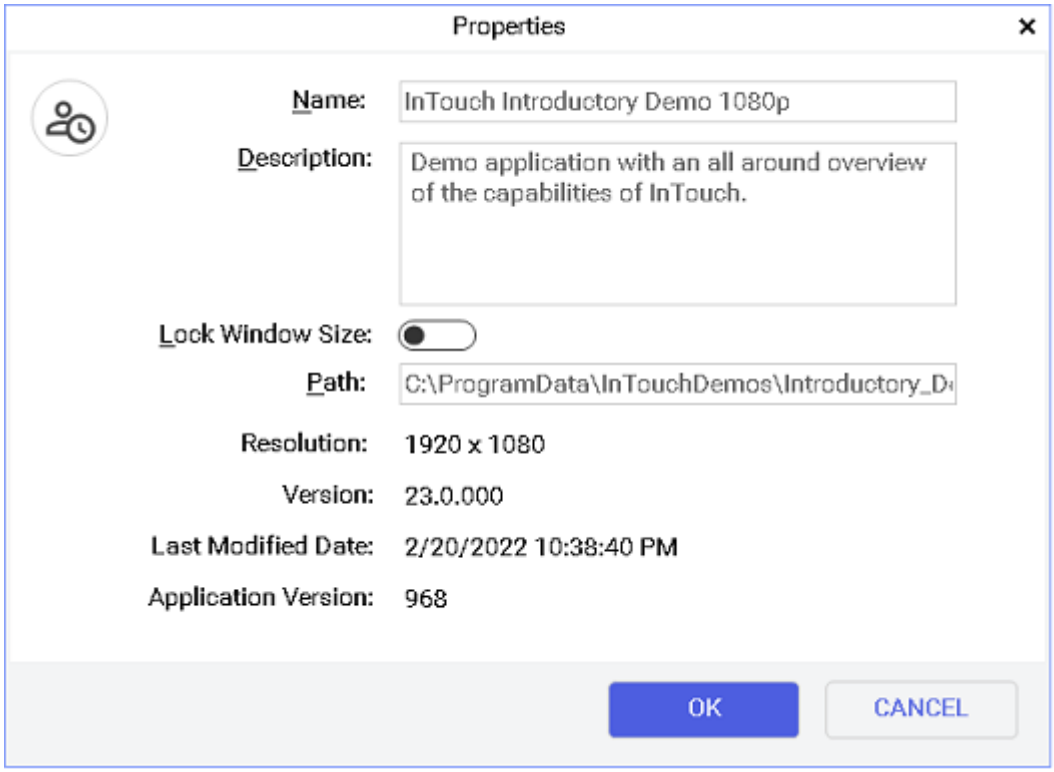
1. 选择列表中的应用程序。
2. 在文件选项卡上的应用程序组中，选择重命名。

修改应用程序属性

1. 选择列表中的应用程序。
2. 在文件选项卡上的应用程序组中，选择属性。

   此时出现属性对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 配置应用程序属性。
   - 在名称框中，为应用程序输入新的名称。
   - 在描述框中，为应用程序输入另一段描述。
4. 选择确定。

更改目标分辨率

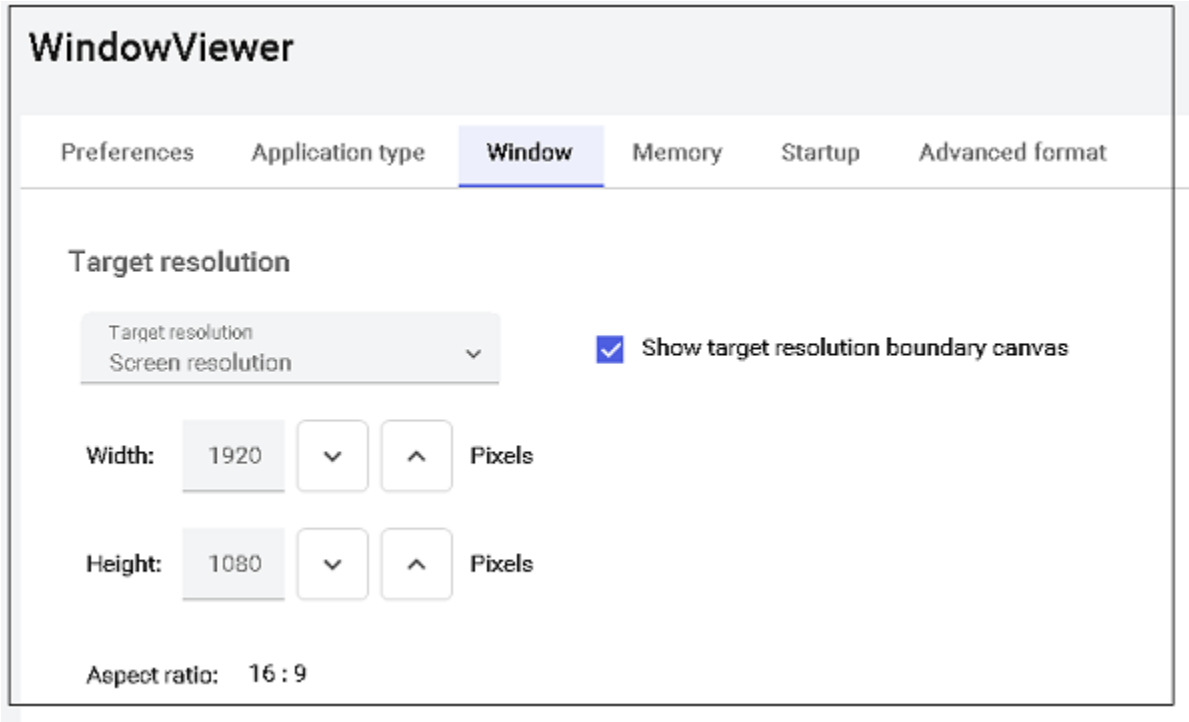
您可以在 WindowMaker 中编辑应用程序时更改指定的目标分辨率。在命令行中不能使用此功能。

执行以下操作：

1. 打开 WindowMaker。
2. 在文件菜单上，突出显示配置，然后选择 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。
3. 选择窗口选项卡。
4. 在目标分辨率部分，根据需要编辑目标分辨率。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 选择确定。

   边界画布将修改以反映更改。图形、窗口大小和窗口控件将保持不变。

   注意：缺省条件下，显示目标分辨率的边界画布处于选中状态。

从应用程序管理器中删除 InTouch 应用程序

从“应用程序管理器”中删除应用程序时，应用程序文件保留在计算机上。

删除应用程序

1. 选择列表中的应用程序。
2. 在文件选项卡上的应用程序组中，选择删除。
3. 在出现的消息中，选择是。
4. 您也可以按照以下方式删除应用程序：
   - 使用鼠标右键单击应用程序，然后从上下文菜单中选择删除。
   - 选择应用程序功能区中的删除图标。
   - 选择应用程序，然后按键盘上的 **delete** 键。

   您可以选择和删除多个应用程序。

删除较早的新型应用程序

如果节点包含旧版新型应用程序并且该节点已升级到当前产品版本，则可以选择删除旧版新型应用程序。此操作将永久删除新型应用程序文件夹和储备库。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

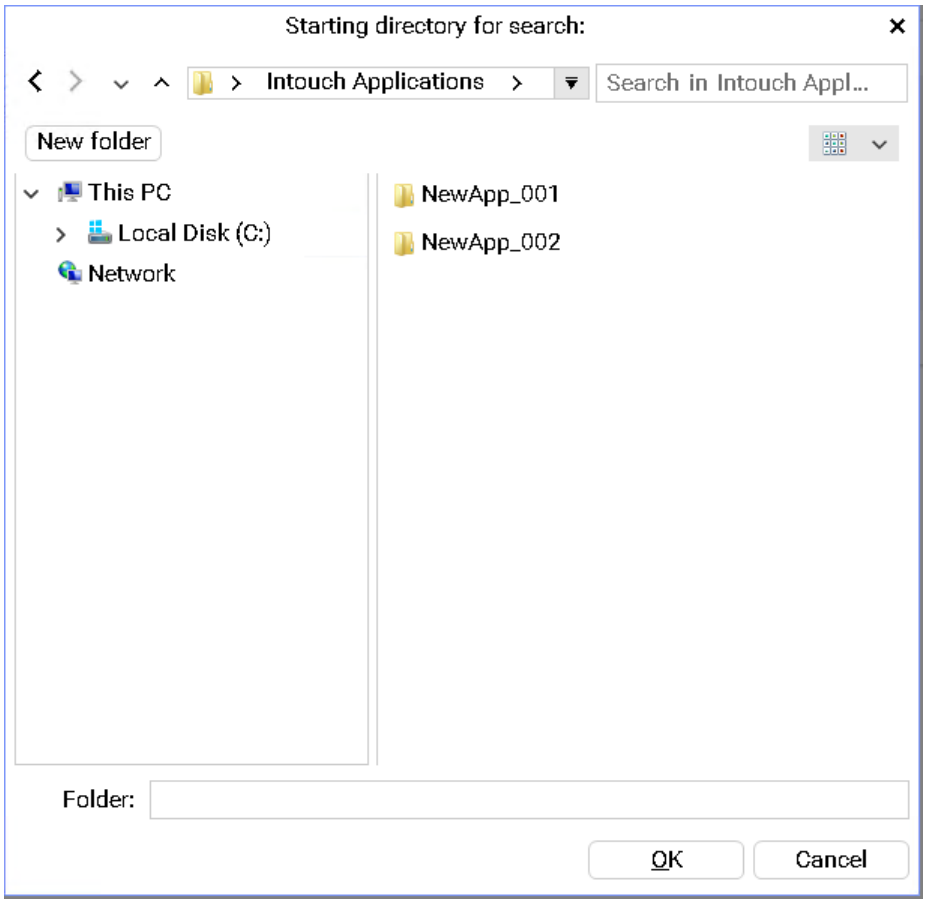
查找 InTouch 应用程序

您可以搜索现有的 InTouch 应用程序。“应用程序管理器”显示找到的任何应用程序。

如果完整路径超过 114 个字符（包括网络驱动器盘符、冒号及所有的反斜杠），则无法在 WindowMaker 中打开应用程序。如果应用程序超过此字符限制，请给路径中的各个文件夹重命名，或是将应用程序移到另一个位置。

查找应用程序

1. 在工具选项卡上，选择查找。此时出现“起始搜索目录”对话框。



2. 选择要在其中搜索应用程序的文件夹。
3. 选择确定。

使用应用程序模板

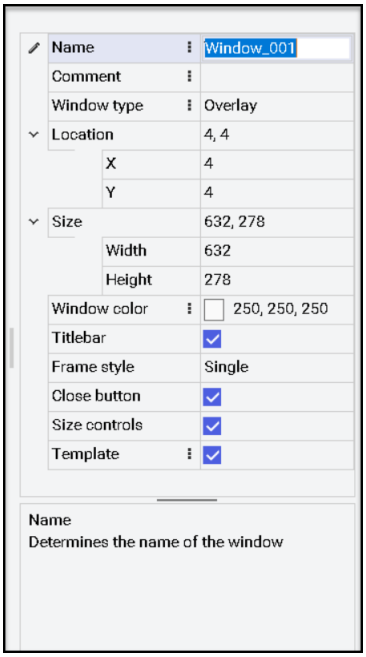
您可以从独立应用程序开发自己的应用程序模板。开发应用程序模板的过程分三步。

1. 创建一个应用程序，并将应用程序中的适当窗口设置为模板窗口。
2. 创建用于在应用程序模板浏览器中供预览的应用程序缩略图。
3. 将应用程序导出为一个 .aaPKG 文件，使其可用作浏览器中的模板。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将应用程序窗口设置为模板窗口

1. 创建一个应用程序。
   a. 使用图形、脚本和窗口开发应用程序。
2. 对应用程序中的所有窗口执行以下操作：
   a. 右键单击每个窗口，然后选择属性。
   b. 在窗口属性面板中，选中模板复选框。



选择确定时，每个窗口都会自动放置在“窗口”窗格的“模板窗口”文件夹中。

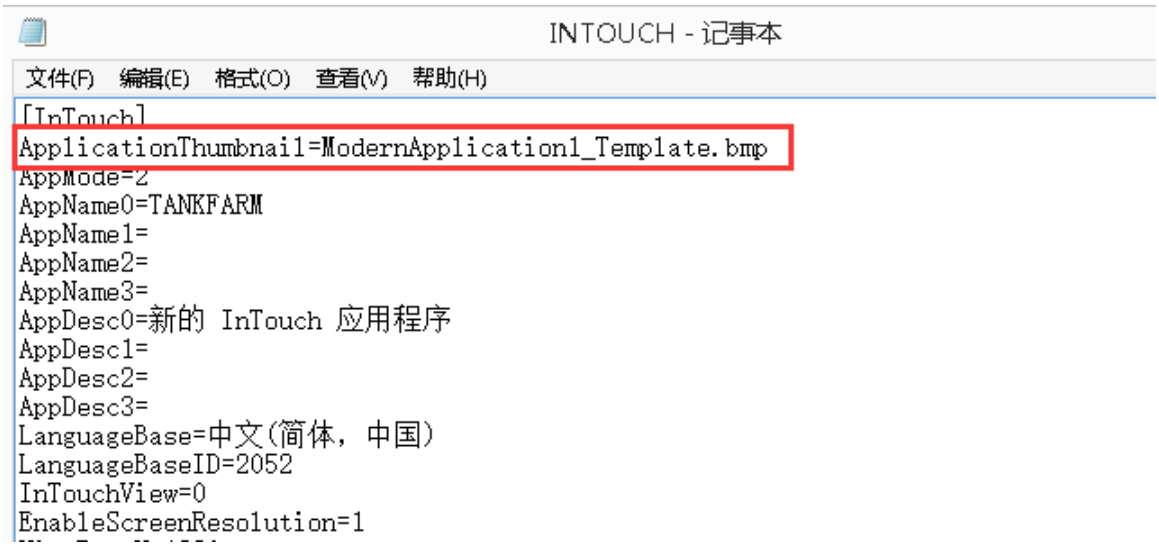
现在必须为您要将其制作为模板的应用程序创建并分配缩略图。

创建和分配应用程序缩略图

1. 使用任意屏幕截图程序，抓取应用程序在配置或运行时的屏幕截图。
2. 将该图像保存为任意图片文件格式，如 .bmp 或 .png 文件，然后将其复制到应用程序文件夹中。
3. 从应用程序文件夹中，用标准文本编辑器（如 Notepad）打开 INTOUCH.INI 文件。
4. 编辑 INTOUCH.INI 文件，在应用程序缩略图字段中加入该图像的文件名。

注意：**ApplicationThumbnail** 字段区分大小写，且必须完全符合缩略图图像的名称和扩展名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 保存更改并关闭。

现在必须将应用程序导出为 .aaPKG 文件，该文件将填充到**应用程序模板浏览器**中。

导出应用程序并创建模板

1. 在“应用程序管理器”中，导出应用程序以创建 .aaPKG 文件。

   如需有关导出应用程序的详细信息，请参阅导出 InTouch 应用程序以用作模板。

2. 将导出的 .aaPKG 文件复制到以下目录中：

   ```
   C:\Program Files (x86)\Wonderware\InTouch\ApplicationTemplates
   ```

   在应用程序模板浏览器中，您的应用程序现在就可以作为模板使用了。

注意：您在先前步骤中创建的应用程序模板缩略图会从导出的 .aaPKG 文件中抽取出来。如果在**应用程序模板浏览器**中，您的应用程序模板显示空白缩略图，则表明图像无效，无法抽取。请确保用于保存缩略图的图像文件格式有效，并且在 ITOUCH.INI 中输入的文件名准确无误。

查看 InTouch 应用程序文件夹

通过使用 WindowMaker，可以查看 InTouch 应用程序文件夹中已部署且在运行时可用的文件。

打开应用程序文件夹

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在**查看**菜单上的**应用程序组**中，选择**打开文件夹**。

   此时出现标准 Windows 资源管理器窗口，显示 InTouch 应用程序文件夹。对于托管应用程序，显示“Galaxy 储备库”节点上的应用程序文件夹。

导出 InTouch 应用程序以用作模板

使用“导出为模板”选项可以导出包含所有图形相关内容（符号、客户端控件、脚本库、语言、样式）的 .aapkg 文件并将其另存为模板。然后可以使用该模板文件创建其它独立或托管的应用程序。

将应用程序导出为模板

1. 在“应用程序管理器”的“文件”选项卡上的“主”组中，选择**导出为模板**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现导出 **InTouch** 应用程序对话框。

2. 提供用于存储 .aapkg 文件的位置。

   此时出现“导出 InTouch 应用程序”进度窗口。

3. 导出完成之后，选择**关闭**。
4. 选择**查看详细信息**以检查是否有任何错误。

   .aapkg file 文件将出现在指定的位置，并且现在可用作应用程序模板来创建新应用程序。

使用应用程序管理器更新 InTouch Web 客户端设置

AVEVA 应用程序管理器的“Web 客户端”选项卡为用户提供了用于配置与 InTouch Web 客户端相关的设置的选项。

更新 InTouch Web 客户端设置

1. 启动“应用程序管理器”，然后选择 **Web 客户端选项**卡。

   此时出现 **Web 客户端设置**屏幕。

2. 配置以下设置：
   - *当前应用程序*：从列表中选择一个应用程序以修改 Web 客户端设置。
   - *图形刷新率（毫秒）*：设置 Web 浏览器将在 Web 服务器中查询图形数据的频率。缺省值为 1 秒。如需详细信息，请参阅《System Platform 安装》。
   - *报警刷新率（毫秒）*：设置 Web 浏览器将在 Web 服务器中查询报警数据的频率。缺省值为 1 秒。如需详细信息，请参阅《System Platform 安装》。
   - *显示标题*：选择该复选框，以显示标题栏。
   - *启用导航栏*：选中该复选框以显示“导航栏”。

     如果未选择**显示标题**设置，将会禁用此设置。
   - *允许匿名访问*：选中该复选框以允许用户无需经过身份验证即可访问 Web 客户端。
   - *启用工作区*：选中该复选框以启用工作区。
3. 在“高级设置”部分下，选择**自定义**以配置以下设置：
   - *网站名称*：提供将替换标准 URL 的字符串。
   - *应用程序标题*：提供将在应用程序栏中显示为应用程序标题的字符串。
   - *网站标题*：提供将在浏览器的标题栏上显示为标题的字符串。
   - *网站图标*：提供将替换浏览器标题栏上的图标的图像文件。
     i. 选择**选择文件...**，以选择图标图像。
     ii. 选择一个文件。
     iii. 选择打**开**。
4. 选择**保存**，以保存这些高级设置。
5. 修改设置后，选择**应用**。
6. 要查看 InTouch Web 客户端，请选择**启动**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：如果网站名称或地址发生更改，请确保配置 AVEVA Identity Manager 以注册新网站。如需详细信息，请参阅向 AVEVA Identity Manager 进行注册, 向 AVEVA Identity Manager 进行注册。

如需有关这些设置的详细信息，请参阅*在 Web 浏览器中查看应用程序图形*。

启动 InTouch OMI ViewApp

使用应用程序管理器中的 OMI 选项卡可启动 InTouch OMI ViewApp 的“Viewer”。

1. **开发**和部署 InTouch OMI 应用程序。

   在部署 ViewApp 之后，它将显示在应用程序管理器的 OMI 选项卡中。

2. **选择** ViewApp。
3. 在 OMI 的“文件”菜单中，**选择 OMI Viewer**。

   此时即会启动“Viewer”并显示 ViewApp。

向 AVEVA Identity Manager 进行注册

使用 AVEVA Identity Manager (AIM)，可以将 InTouch Web 客户端配置为使用单一登录，而不是缺省的基于 Windows 操作系统的身份验证。

注意：InTouch Web 客户端仅通过 AIM 配置支持基于 ArchestrA 的安全性。

配置标识服务器

1. 在 System Platform 配置器中，配置**通用平台 > 系统管理服务器**。
2. 在 AVEVA 应用程序管理器中，使用用户凭证注册 AIM 服务器。

“AIM 注册”对话框也可以用来配置反向代理服务器：

1. 设置反向代理服务器：
2. 在 System Platform 配置器中，配置**通用平台 > 系统管理服务器**。
3. 在“AIM 注册”对话框中提供“安全网关”地址。

您可以选择远程或本地系统管理服务器。如需有关配置“系统管理服务器”的详细信息，请参阅 *System Platform 安装*。

向标识服务器进行注册

1. 在应用程序管理器中，**选择 Web 客户端选项**卡。
2. 依次**选择工具**和 **Identity Manager**。

   此时出现**标识服务器设置**屏幕。

3. **选择使用 AIM 服务器作为身份验证服务器复选**框，以允许使用 AVEVA Identity Manager。

   标识服务器字段显示在配置器中所配置的标识服务器。

4. 更新以下设置：
   a. **用户名：提供用户名以连接到标识服务器。**用户必须属于 Administrators 用户组。
   b. **密码**：提供相应用户名的密码。
5. 要完成反向代理设置，请在**安全网关**字段中提供反向代理或 DMZ 服务器的 URL。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

这是一个可选设置，只有将 Web 客户端托管在反向代理服务器的后面时，才需要进行此设置。有关更多信息，请参阅 AVEVA DMZ 安全链接。

6. 或者，也可以选择允许将工业图形嵌入任何网站复选框，以在运行时在 HTML iframe 中查看 Web 客户端。
7. 选择确定。
8. 选择应用。

注意：如果网站名称或地址发生更改，您必须配置 AVEVA Identity Manager 才能注册新网站。

使用凭据管理器

凭据管理器允许您将凭据存储在安全的位置。您可以使用存储的凭据来验证对 WindowMaker、WindowViewer 和报警实用程序中其它组件的访问。

要启动凭据管理器

1. 以管理员身份启动应用程序管理器。
2. 在工具菜单上的工具组中，单击凭据管理器。

   此时出现“凭据管理器”屏幕。

   注意：必须是 Windows 系统中 InTouchDevelopers 组的成员，才能访问凭据管理器。

“凭据管理器”由两部分组成：

- 命名凭据
- 凭据的详细信息

命名凭据部分以网格格式显示凭据列表，其中包含以下列。

- 名称：表示凭据的名称。
- 类型：表示凭据的类型 - 用户名和密码或域用户名和密码。
- 组：表示凭据所属的 Windows 用户组。单击“组”列中的省略号图标 (...) 以选择一个或多个组。如需详细信息，请参阅管理命名凭据。

凭据的详细信息部分显示所选凭据的以下字段：

- 域
- 用户名
- Password
- 密钥

管理命名凭据

使用“凭据管理器”屏幕来添加/删除凭据。您必须为每个凭据分配组。这些组中的用户将能够从各种 InTouch 组件（例如 WindowMaker、WindowViewer 和报警实用程序）访问凭据。

要添加凭据

1. 在凭据管理器的命名凭据部分中，执行以下操作：
   a. 单击添加 (+)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时将在“命名凭据”网格中添加一个新行。

b. 在名称列中，输入凭据的名称。

c. 在类型列中，从下拉列表中选择凭据的类型。

- 用户名和密码：允许您将 SQL Server 帐户分配给凭据。
- 域用户名和密码：允许您将 Windows 帐户分配给凭据。
- 密钥：允许用户指定最大长度为 1024 个字符的密钥字符串。没有用户名与密钥关联，只有一个字符串。它可以包含任何特殊字符，与地图应用程序 API 密钥一起使用。

d. 在组列中，单击列末尾的省略号。

此时出现选择组屏幕。

e. 在从列表中选择下拉列表中选择域。

f. 滚动浏览可用操作系统组列表以找到该组，然后单击 + 图标。

所选操作系统组即会出现在所选组部分中。

缺省条件下，所有命名凭据都会添加到管理员组中。此外，我们建议您添加运行应用程序的用户所属的组。这样，用户即使在非管理员模式（如 WindowViewer）下运行应用程序时也可以访问凭据。

注意：要删除某个组，请在所选组部分中选择该组，然后单击删除 (-)。

g. 单击确定。

2. 在详细信息部分中，执行以下操作：

a. 在域字段中，输入用户域。

注意：“域”字段仅针对域用户名和密码类型启用。如果您选择“凭据类型”作为用户名和密码，那么将禁用“域”字段。

b. 在用户名字段中，输入所选凭据的用户名。

c. 在密码字段中，输入给定用户名的密码。

d. 在确认密码字段中，再次输入该密码。

e. 在密钥字段中输入一个可用作密钥的值。

注意：将类型设置为密钥后，密钥字段将启用，其他字段将被禁用。

3. 单击保存。

要删除凭据

1. 从网格中选择一个凭据。
2. 单击删除 (-)。

此时将删除所选凭据。

导出和导入命名凭据

您可以导出所选应用程序的凭据，并将其导入到要导入应用程序的节点上。

导出凭据

您可以通过以下方式之一导出凭据：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 仅导出凭据
- 在导出应用程序时导出凭据。

导出凭据时，必须配置安全密钥。安全密钥会对凭据进行加密，在导入凭据时需要使用它进行身份验证。

要使用凭据管理器导出凭据

1. 在凭据管理器屏幕中，单击导出。
2. 此时出现导出凭据窗口。
3. 从网格中选择要导出的凭据。
4. 在下一部分中，设置安全密钥。

   注意：安全密钥的长度必须至少为 12 个字符，并且必须至少包含一个大写字符、一个小写字符、一个数字字符和一个特殊字符。

5. 单击导出。

   此时出现导出凭据详细信息文件浏览器。

6. 选择用于保存凭据的文件夹，然后单击保存。

   凭据将导出为 .bin 文件格式。

要在导出应用程序时导出凭据

1. 在“应用程序管理器”中，使用鼠标右键单击要导出的应用程序，然后单击导出为模板。
2. 选中导出凭据复选框。

   此时将导出应用程序以及凭据列表。

导入凭据

您可以导入凭据以便在已导入应用程序的特定节点上使用。

要导入凭据

1. 在凭据管理器屏幕中，单击导入。
2. 使用文件浏览器选择 .bin 文件，然后单击打开。
3. 输入安全密钥。

   提示：您必须使用导出应用程序时所配置的相同安全密钥。

   凭据将导入到“凭据管理器”中。

要导入 NAD 的凭据

在此发行版中，NAD 储备库上的命名凭据不会自动传播到 NAD 客户端。也就是说，选择要与 NAD 一起运行的应用程序（使用查找应用程序操作）不会导入命名凭据。

要在 NAD 储备库上导入命名凭据，请通过每个 NAD 客户端上的应用程序管理器导入命名凭据。

导入凭据情形

情形 1：导入的凭据和计算机上的现有凭据有所不同。

凭据将导入到“凭据管理器”中，并且新条目将添加到“凭据管理器”网格中。

情形 2： 导入的凭据和现有的凭据相同。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

凭据将导入到“凭据管理器”中，并且新条目将添加到“凭据管理器”网格中。新导入的凭据将被重命名。

情形 3：导入的凭据和现有凭据的名称相同，但详细信息不同。

凭据将导入到“凭据管理器”中，并且新条目将添加到“凭据管理器”网格中。新导入的凭据将以现有凭据的增量进行重命名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

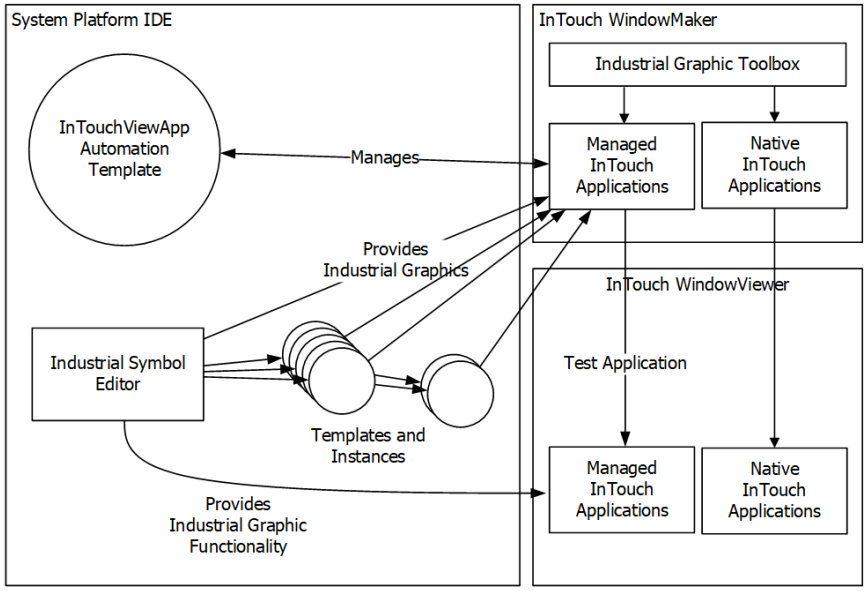
章 6 使用 IDE 管理 InTouch 应用程序

您可以使用 System Platform IDE：

- 创建新的托管 InTouch 应用程序。
- 导入现有的 InTouch 应用程序以用作托管的 InTouch 应用程序。
- 启动 WindowMaker。
- 将在 WindowMaker 中所作的更改提交给托管的 InTouch 应用程序。
- 连同其 InTouchViewApp 对象一起导出与导入托管的 InTouch 应用程序。
- 发布托管的 InTouch 应用程序。
- 删除托管的 InTouch 应用程序。
- 导入与导出托管的 InTouch 应用程序中使用的标记数据。
- 将标记数据导出到 .csv 文件，或从中导入标记数据。
- 切换托管的 InTouch 应用程序的语言。
- 向托管的 InTouch 应用程序添加文件。
- 将所有 Galaxy 图形与 InTouchViewApp 关联。
- 使用应用程序模板创建托管的 InTouch 应用程序

您可以从“InTouch HMI 应用程序管理器”中启动 System Platform IDE。

下图显示如何导入、导出、管理及发布应用程序。



创建托管的 InTouch 应用程序

通过创建并配置 InTouchViewApp 对象，可以创建托管的 InTouch 应用程序。

您还可以直接从 InTouchViewApp 对象中查看应用程序的版本、分辨率及描述信息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 应用程序目录作为共享目录进行创建：

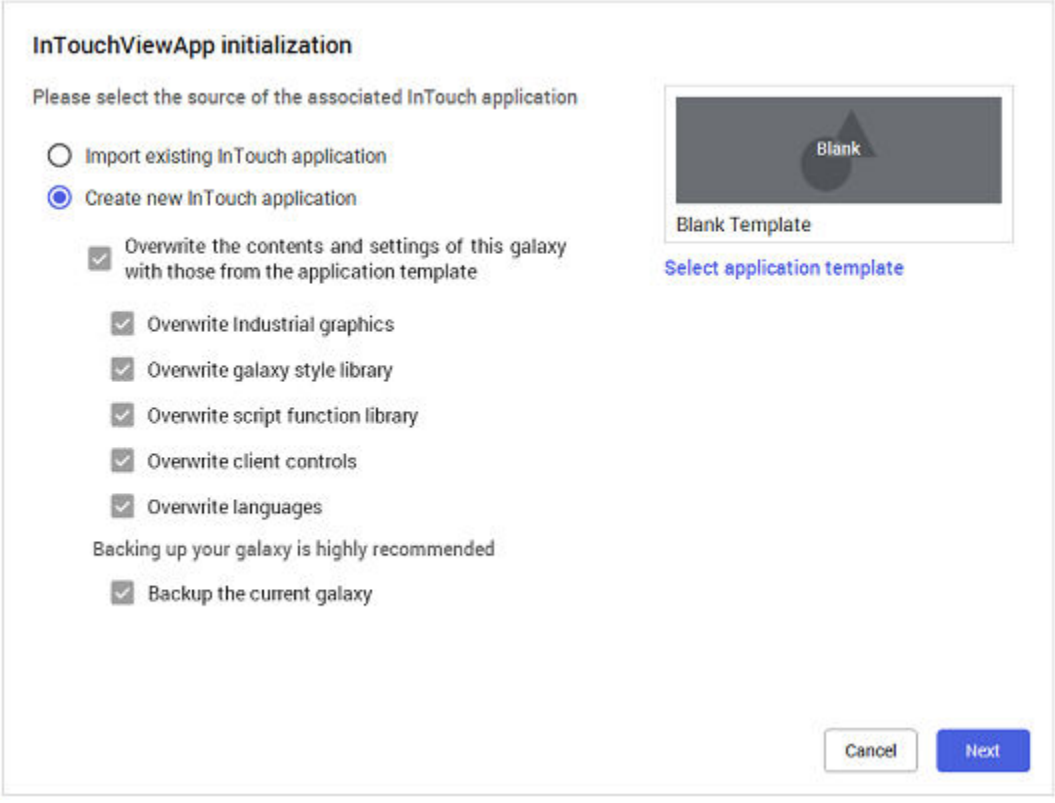
\\GRNodeName\GalaxyName-$InTouchViewAppObjectName

此目录由 IDE 而不是“InTouch HMI 应用程序管理器”来管理。

创建托管的 InTouch 应用程序

1. 打开 System Platform IDE。
2. 在**模板工具箱**中，展开**系统**工具包。
3. 从 **$InTouchViewApp** 基本模板中衍生一个模板。执行以下操作：
   a. 使用鼠标右键单击 **$InTouchViewApp** 基本模板，指向**新建**，然后选择**衍生模板**。此时出现一个使用缺省名称的新衍生模板。
   b. 如果需要，重命名新模板。
4. 双击衍生模板。

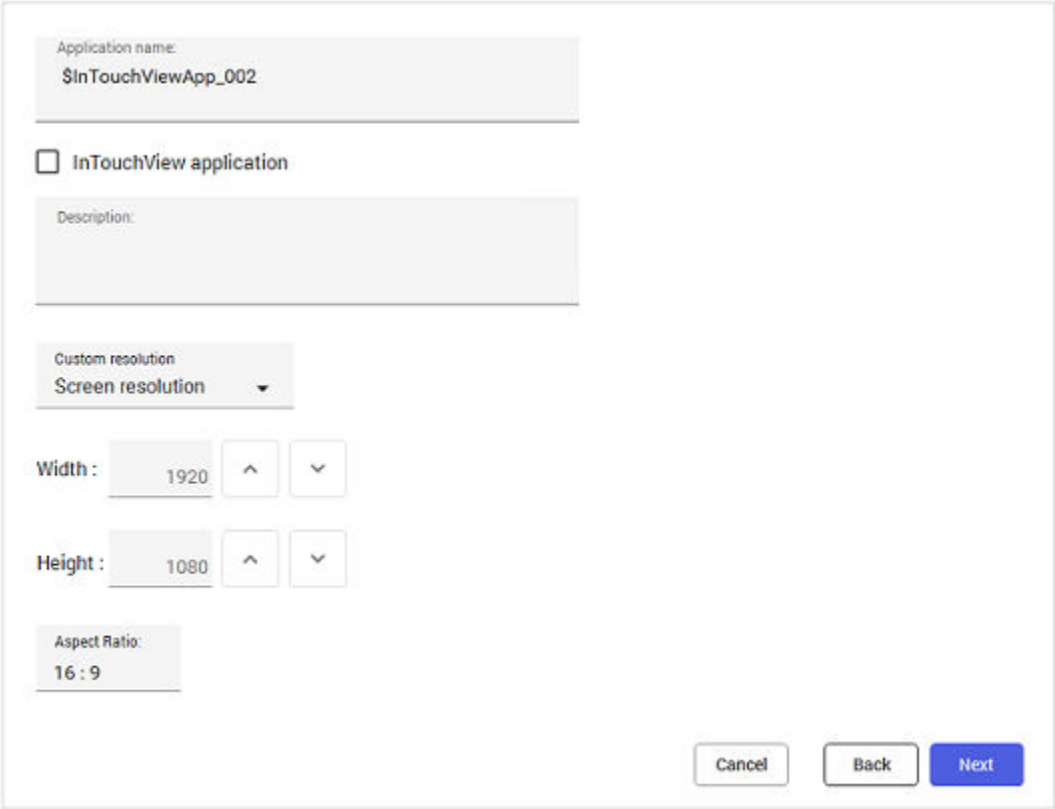
   此时出现 **InTouchViewApp Initialization**（**InTouchViewApp** 初始化）对话框。



5. 突出显示**创建新的 InTouch 应用程序**，然后选择**下一步**。

   此时出现下一个面板。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



6. 输入 InTouch 应用程序的名称与描述。
7. 选择 **InTouchView 应用程序**来创建一个仅将 Application Server 引用用作外部数据源的 InTouch 应用程序。
8. 选择应用程序目标分辨率（如果与缺省屏幕分辨率选项不同）。此字段的选项如下：
   a. 选择**选择目标分辨率**下拉菜单以查看预定义目标分辨率列表。
   b. 突出显示**选择目标分辨率**下拉菜单，然后选择**自定义**。像素宽度和高度字段变为可编辑。边界限制为 150x150 和 10000x10000。
9. 选择下一步。

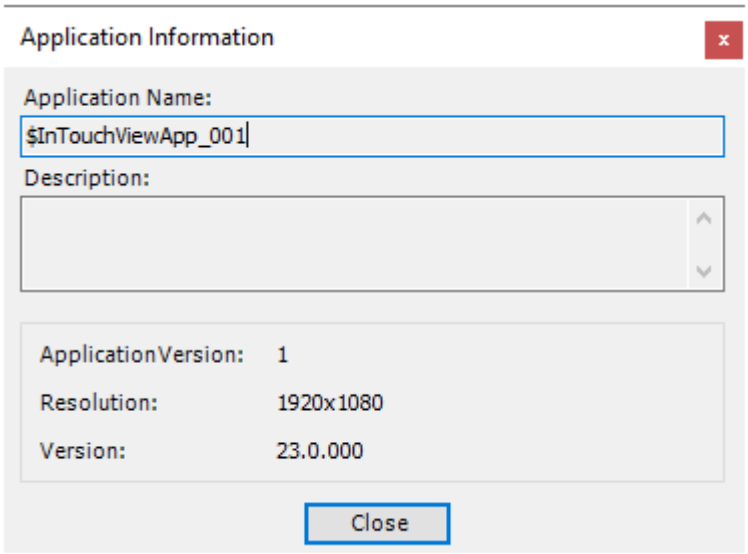
   WindowMaker 启动。

注意：如果选择的目标分辨率与屏幕分辨率不同，在 WindowMaker 中编辑应用程序时系统不会提示您转换为屏幕分辨率。

查看应用程序的版本、分辨率及描述

1. 打开“模板工具箱”。
2. 使用鼠标右键单击 InTouchViewApp 模板，然后选择**应用程序信息**。此时出现**应用程序信息**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



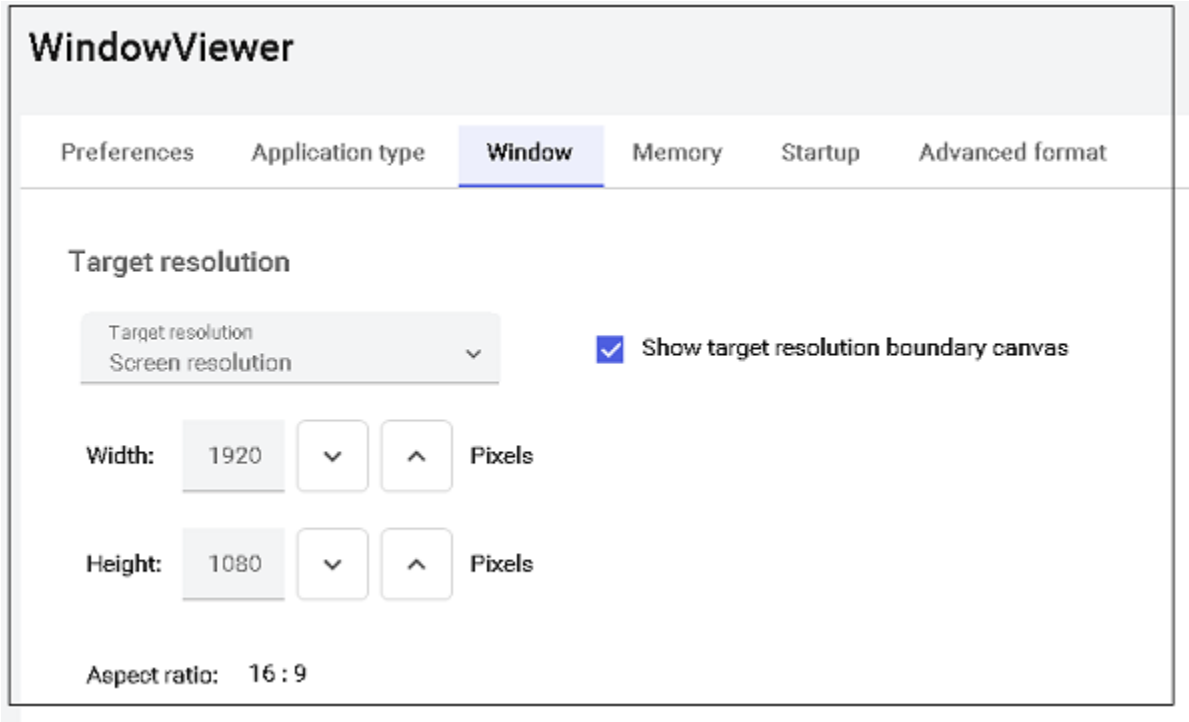
更改目标分辨率

您可以在 WindowMaker 中编辑应用程序时更改指定的目标分辨率。在命令行中不能使用此功能。

执行以下操作：

1. 打开 WindowMaker。
2. 在文件菜单上，突出显示配置，然后选择 **WindowViewer**。
   此时出现 WindowViewer 配置屏幕。
3. 选择窗口选项卡。
4. 在目标分辨率部分中，根据需要编辑目标分辨率。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 选择确定。

   边界画布会更新以反映更改。窗口、窗口大小和窗口控件将保持不变。

   注意：缺省条件下，显示目标分辨率的边界画布处于选中状态。

从应用程序模板创建托管的应用程序

根据应用程序模板创建托管的 InTouch 应用程序可显著减少设计时间并使新应用程序以现有标准为基础。

可以使用应用程序模板浏览器选择您希望托管的 InTouch 应用程序所依据的模板。应用程序模板浏览器中提供的模板是导出的 .aaPKG 文件，从 InTouch 安装目录中的“应用程序模板”文件夹加载这些文件。例如：

```
C:\Program Files(x86)\Wonderware\InTouch\ApplicationTemplates
```

应用程序模板浏览器中的文件夹层次结构将模仿此目录中的层次结构。

根据应用程序模板创建托管的应用程序

1. 打开 IDE 并从 $InTouchViewApp 对象创建衍生模板。

   您可以重命名衍生模板。

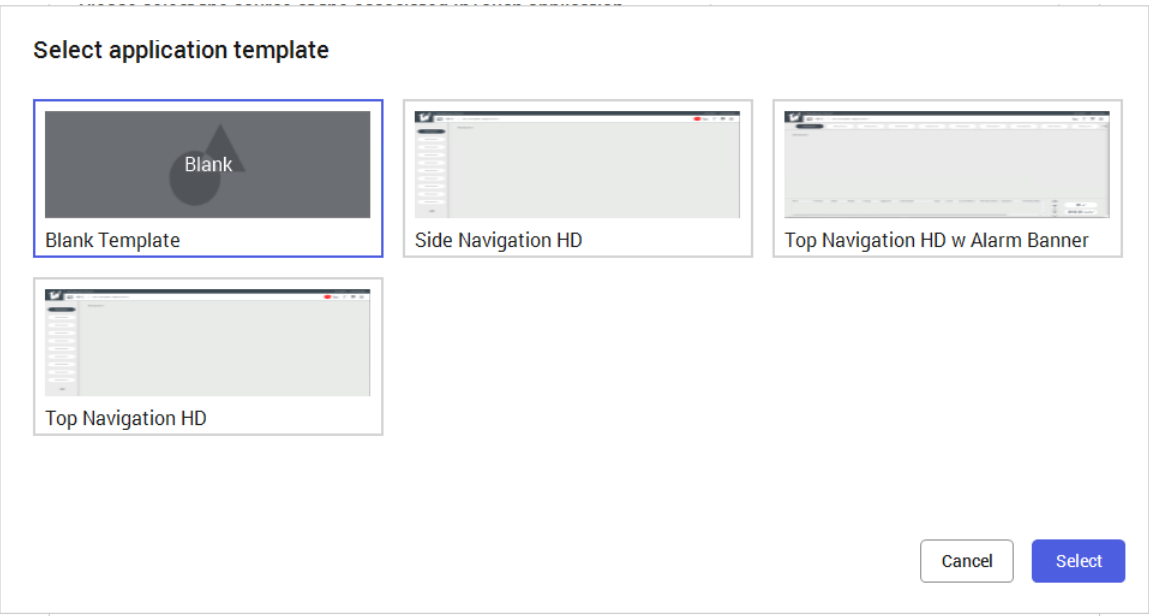
2. 双击衍生模板。

   此时出现 **InTouchViewApp** 初始化对话框。缺省条件下，**创建新的 InTouch 应用程序选项为选中状态**。

3. 选择**选择应用程序模板**。

   此时出现**选择应用程序模板**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



浏览器中每个略图都保持其纵横比。

4. 突出显示首选应用程序模板，然后选择确定。

   您的选择将填充 **InTouchViewApp** 初始化对话框。

5. 仔细查看并选择您首选的 Galaxy 设置覆盖选项。

   注意：缺省条件下，使用应用程序模板中的内容和设置覆盖此 **Galaxy** 的内容和设置选项处于选中状态。将被覆盖的特定 Galaxy 设置分成多个子选项。此时可以取消选中任意或所有这些选项。

   这些覆盖选项如下：

   - **覆盖工业图形**

     如果选中，出现冲突时 Galaxy 符号将被应用程序模板符号替换。

   - **覆盖 Galaxy 样式库**

     如果选中，应用程序模板样式库将替换整个 Galaxy 样式库。如果未选中，将跳过此步骤。不存在样式库合并。

   - **覆盖脚本函数库**

     如果选中，出现冲突时 Galaxy 脚本函数库将被应用程序模板脚本库替换。

   - **覆盖客户端控件**

     如果选中，出现冲突时 Galaxy 客户端控件将被应用程序模板客户端控件替换。

   - **覆盖语言**

     如果选中，应用程序模板中定义的语言设置将替换 Galaxy 中定义的语言设置。

     如果未选中，出现冲突时 Galaxy 语言设置将保持不变。如果未选中且未出现冲突，应用程序模板设置将附加到 Galaxy 设置。

6. 选择下一步。

   如果选择了**备份当前 Galaxy 选项**，系统提示您备份 .cab 文件的文件名的位置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要点：强烈建**议备**份 Galaxy。

7. 确**认应**用程序名称并**选择 InTouchView 应用程序**。

   缺省条件下，目**标**分辨率**设置为**模板的分辨率。**像素**字段被禁用。**这**可以在**应**用程序**开发**期**间调**整。

8. **选择下一步**。

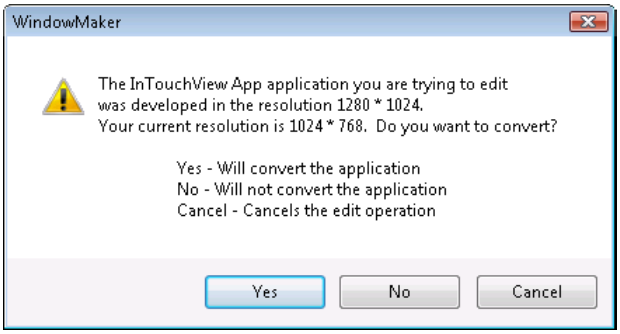
   初始化完成后，可以部署您的**应**用程序。

从 IDE 启动 WindowMaker

通**过**从 IDE 中启**动** WindowMaker，可以**编辑**托管的 InTouch **应**用程序。

您可以从 InTouchViewApp 模板或**实**例中启**动** WindowMaker。

打**开**托管的 InTouch **应**用程序**时**，如果它**创建时**使用的屏幕分辨率不同于当前系**统**分辨率，**则**会出**现**一个消息。



- **选择是**将 InTouch **应**用程序**转换**成当前的系**统**分辨率，然后打**开**它。
- **选择否**在原始分辨率下打**开**并**编辑** InTouch **应**用程序。

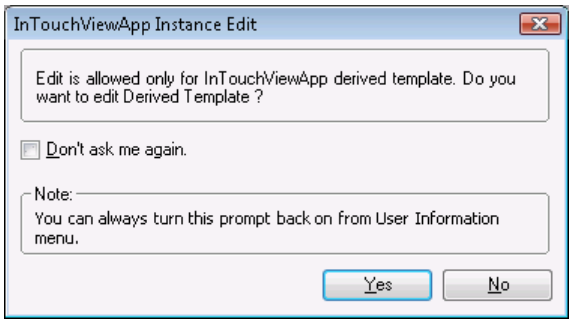
从 InTouchViewApp 模板中启动 WindowMaker

1. 打**开** IDE。
2. 找到包含要修改的托管的 InTouch **应**用程序的 InTouchViewApp 模板。
3. 双**击** InTouchViewApp 模板。WindowMaker 作**为对**象的缺省**编辑**器启**动**，并打**开** InTouch **应**用程序。您已**经**准**备**好**编辑**托管的**应**用程序。

从 InTouchViewApp 实例中启动 WindowMaker

1. 打**开** IDE。
2. 找到 InTouchViewApp **对**象，其父**对**象存放要修改的托管的 InTouch **应**用程序。
3. 双**击** InTouchViewApp **对**象。此**时**出**现 InTouchViewApp 实例编辑**对**话**框。

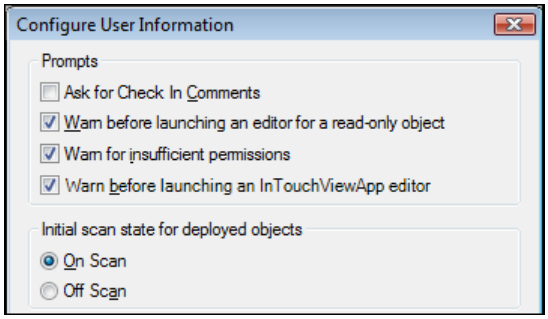
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有**权**利。



4. 执行以下任意操作：
   - **选择否**不打**开**模板。
   - **选择是**在 WindowMaker 中打**开关联**的 InTouchViewApp 模板。

     如果启用**不要再次询问**复选框并**选择是**，则下次打**开** InTouchViewApp 实例时，会自动从**关联**的 InTouchViewApp 模板中打**开**托管的 InTouch 应用程序。

     您可以使用从**编辑**菜单中打**开**的**配置用户信息对话**框来更改此设置。



托管的 InTouch 应用程序的语言切换

对于托管的 InTouch 应用程序，某些语言切换功能是由 IDE（而不是 InTouch HMI）处理的。

使用 IDE 可以：

- 导出与导入图形文本以进行翻译。
- 配置托管的应用程序使用的语言。

如需详细信息，请参阅 Application Server 用户文档。

提交 InTouch 应用程序的更改

修改托管的 InTouch 应用程序之后，您可以提交更改以部署到目标节点上。

对托管的 InTouch 应用程序作出更改之后，任何衍生的 InTouchViewApp 对象都将出现 Pending Changes（待处理的更改）图标。

这表示您必须将更改重新部署到目标节点，以便那些节点上的 WindowViewer 能够反映出这些更改。

如需有关如何将更改部署到操作员节点的详细信息，请参阅部署托管的 InTouch 应用程序和在操作员节点上接受新的应用程序版本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

提交 InTouch 应用程序的更改

1. 在 WindowMaker 中像处理独立的 InTouch 应用程序那样修改托管的 InTouch 应用程序。
2. 保存 InTouch 窗口。
3. 在文件菜单上，选择退出。WindowMaker 关闭，焦点返回到 IDE。此时出现签入对话框。
4. 如果需要，请输入签入注释，然后选择确定。此时出现签入进度对话框。
5. 选择关闭。

导入 InTouch 应用程序

您可以导入现有的 InTouch 应用程序以用作托管的 InTouch 应用程序。此操作分为两个步骤：

- 创建 InTouchViewApp 对象以关联到所导入的 InTouch 应用程序。
- 导入 InTouch 应用程序。

导入的 InTouch 应用程序变成一个托管的 InTouch 应用程序。

IDE 会在其管理的文件夹中为现有 InTouch 应用程序创建一个副本。IDE 保持现有的 InTouch 应用程序与位置不变。

如果导入的 InTouch 应用程序为 6.0 或更高版本，则会出现一条转换消息。在 WindowMaker 中打开应用程序之前，会对它进行转换。6.0 之前版本的 InTouch 应用程序无法进行转换。

将 InTouch 应用程序导入到 Galaxy 中

1. 打开 IDE。
2. 在模板工具箱中，从 $InTouchViewApp 基本模板中衍生一个模板。
3. 打开该衍生模板。此时出现 **InTouchViewApp** 初始化对话框。
4. 突出显示导入现有的应用程序，然后选择下一步。
5. 找到 InTouch 应用程序。执行以下任意操作：
   - 选择省略号按钮以浏览包含托管的 InTouch 应用程序的文件夹。
   - 要搜索应用程序，请选中查找应用程序复选框。指定要开始搜索的搜索根目录，然后选择查找。从列表中选择 InTouch 应用程序。
6. 选择下一步。
7. 如果需要，请在应用程序名称框中输入新名称，在描述框中输入描述。部署托管的 InTouch 应用程序时，名称与描述都出现在“应用程序管理器”中。
8. 选择与缺省屏幕分辨率选项不同的应用程序目标分辨率。此字段的选项可按如下方式查看：
   a. 选择选择目标分辨率下拉菜单以查看预定义目标分辨率列表。
   b. 选择选择目标分辨率下拉菜单，然后选择自定义。像素宽度和高度字段将变为可编辑。

   注意：如果在导入现有应用程序时选择不同的目标分辨率，则不存在从源应用程序分辨率到目标分辨率的应用程序分辨率转换。
9. 选择下一步。此时出现下一个面板，显示导入进度。
10. 选择关闭。此时 InTouch WindowMaker 启动，您可以将 InTouch 应用程序当作托管的 InTouch 应用程序进行编辑。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

导入带图形的独立 InTouch 应用程序

如果要转换为托管应用程序的独立 InTouch 应用程序包含图形，那么会导入应用程序，但不会导入图形。

导入包含在独立应用程序中的图形

1. 将独立应用程序导出为模板。
2. 将该模板保存在应用程序模板位置。
3. 在 IDE 中，通过选择这个创建的应用程序模板来创建一个新的 $InTouchViewApp 衍生模板。

导入与导出 InTouchViewApp 对象

您可以在 IDE 中导入与导出 InTouchViewApp 对象。InTouchViewApp 对象包含存放托管的 InTouch 应用程序所需的全部信息，可用于在 Galaxy 之间交换托管的 InTouch 应用程序。

如需有关导入与导出自动化对象的详细信息，请参阅 Application Server 文档。

导入 InTouchViewApp 对象

1. 在 **Galaxy** 菜单上，指向**导入**，然后选择**对象**。此时出现**导入自动化对象**对话框。
2. 突出显示其中包含要导入的 InTouchViewApp 对象的 .aaPKG 文件，然后选择**打开**。此时会导入对象及其托管的 InTouch 应用程序。
3. 选择**关闭**。

导出 InTouchViewApp 对象

1. 在 **Galaxy** 菜单上，指向**导出**，然后选择**对象**。此时出现**导出自动化对象**对话框。
2. 突出显示要导出到的文件夹，指定 .aaPKG 文件名，然后选择**保存**。此时会导出对象及其托管的 InTouch 应用程序。
3. 选择**关闭**。

导出与导入标记数据

您可以将托管的 InTouch 应用程序的标记数据导出到 .csv 文件。您可以将此数据导入到另一个托管的 InTouch 应用程序或独立的 InTouch 应用程序。

从托管的 InTouch 应用程序中导出标记数据

1. 打开 IDE。
2. 选择衍生的 InTouchViewApp 模板，该模板包含要从中导出标记数据的托管的 InTouch 应用程序。
3. 在 **Galaxy** 菜单上，指向**导出**，然后选择 **DB Dump**。

   此时出现要转储到的 **CSV** 文件对话框。
4. 指定 .csv 文件的位置与文件名，然后选择**保存**。

   此时出现确认对话框。
5. 如果希望按数据类型给 .csv 文件中的标记数据分组，请选择按类型的组输出。
6. 选择**确定**。出现成功消息时，选择**确定**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将标记数据从 .csv 文件导入到托管的 InTouch 应用程序

1. 打开 IDE。
2. 选择衍生的 InTouchViewApp 模板，该模板包含托管的 InTouch 应用程序，您正是要将标记数据导入到其中。
3. 在 **Galaxy** 菜单上，指向**导入**，然后**选择 DB Load**。

   此时出现要从中加载的 **CSV** 文件对话框。
4. 突出显示 .csv 文件，然后**选择打开**。

   在导入期间，可能会出现一个或多个消息，警告您存在重复的名称。为每个重复的标记选择适当的选项。
5. 出现成功消息时，选择确定。

保留标记值与参数

在一个节点上使用多个托管的 InTouch 应用程序时，所有的应用程序都使用相同的目录。如果配置要保留的标记数据或参数，并切换应用程序，则会丢失运行时数据。

在托管的 InTouch 应用程序中配置保留

1. 从 IDE 中使用 WindowMaker 打开托管的 InTouch 应用程序。
2. 在文件菜单上，指向配置，然后选择 **WindowViewer**。

   此时出现 **WindowViewer** 属性屏幕。
3. 选择托管的应用程序选项卡。
4. 在本地工作目录框中，输入唯一的现有目录名称。这是目标运行时节点上目录的名称。还可以使用特殊字符来指定本地工作目录路径。下表说明了可以使用的特殊字符：

特殊字符	本地工作目录路径
%SystemDrive%	系统驱动器。例如，“C:”
%USER%	已登录用户的用户名。如果 WindowViewer 作为服务运行，应用程序将复制给“所有用户”
%APPDATA%	已登录用户的应用程序数据。如果 WindowViewer 作为服务运行，应用程序将复制给“所有用户”的应用程序数据
%ITUSER%	已登录用户。如果 WindowViewer 作为服务运行或在控制台会话中运行，应用程序将复制给“所有用户”
%ITAPPDATA%	已登录用户的应用程序数据。如果 WindowViewer 作为服务运行或在控制台会话中运行，应用程序将复制给“所有用户”的应用程序数据

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

删除托管的 InTouch 应用程序。

通过删除关联的 InTouchViewApp 模板，可以在 IDE 中删除 InTouch 应用程序。

执行此操作时，模板及其关联的 InTouch 应用程序目录将完全删除。

只有当关联的 InTouchViewApp 对象没有任何衍生的实例时，才可以删除 InTouch 应用程序。

删除 InTouchViewApp 实例时，不会删除关联的 InTouch 应用程序。

删除 InTouchViewApp 模板

1. 打开 IDE。
2. 选择包含要删除的托管的 InTouch 应用程序的 InTouchViewApp 模板。
3. 在主页菜单上的编辑菜单中，选择删除。

   此时出现删除对话框。
4. 选择是。

   此时会删除 InTouchViewApp 模板与关联的 InTouch 应用程序文件夹。

在 InTouchViewApp 中关联所有 Galaxy 图形

将所有 Galaxy 图形与 InTouchViewApp 模板关联可使部署与发布的 InTouch 应用程序能够执行由 Galaxy 中的任何图形组成的“显示图形”请求，而不必在应用程序中嵌入这些图形。

- ShowGraphic() 函数使用图形作为参数。关联所有 Galaxy 图形可确保图形在运行时显示且可用（无论 InTouchViewApp 是否引用它）。
- 关联所有 Galaxy 图形可确保 ShowGraphic() 函数引用的模板符号在运行时进行部署且可用。

注意：术语“图形”包括“图形工具箱”中存在的任何符号或客户端控件，以及模板与实例拥有或继承的任何符号。

如需有关将所有 Galaxy 图形与 InTouchViewApp 关联的详细信息，请参阅 Application Server 文档。

通过 InTouchViewApp 右键单击上下文菜单可访问“包括所有 Galaxy 图形”选项。

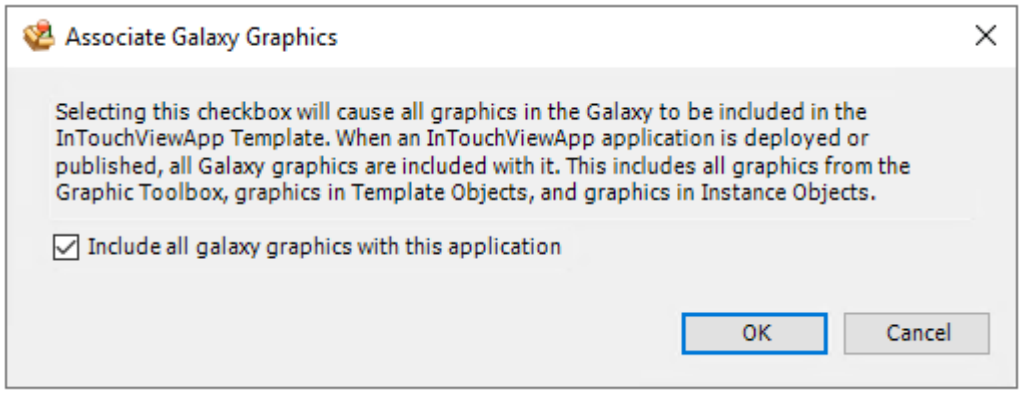
随 InTouchViewApp 模板包括所有 Galaxy 图形

1. 使用鼠标右键单击要配置的 InTouchViewApp 模板。

   此时出现 InTouchViewApp 上下文菜单。
2. 选择**关联 Galaxy 图形**菜单项。

   此时出现**关联 Galaxy 图形**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



出现**关联 Galaxy 图**形对话框时，会签出 $InTouchViewApp 模板。

如果 $InTouchViewApp 模板尚未初始化或签出，则使用此应用程序包括所有 **Galaxy 图**形复选框不可用。

3. 启用**复选**框，然后**选择确定**。

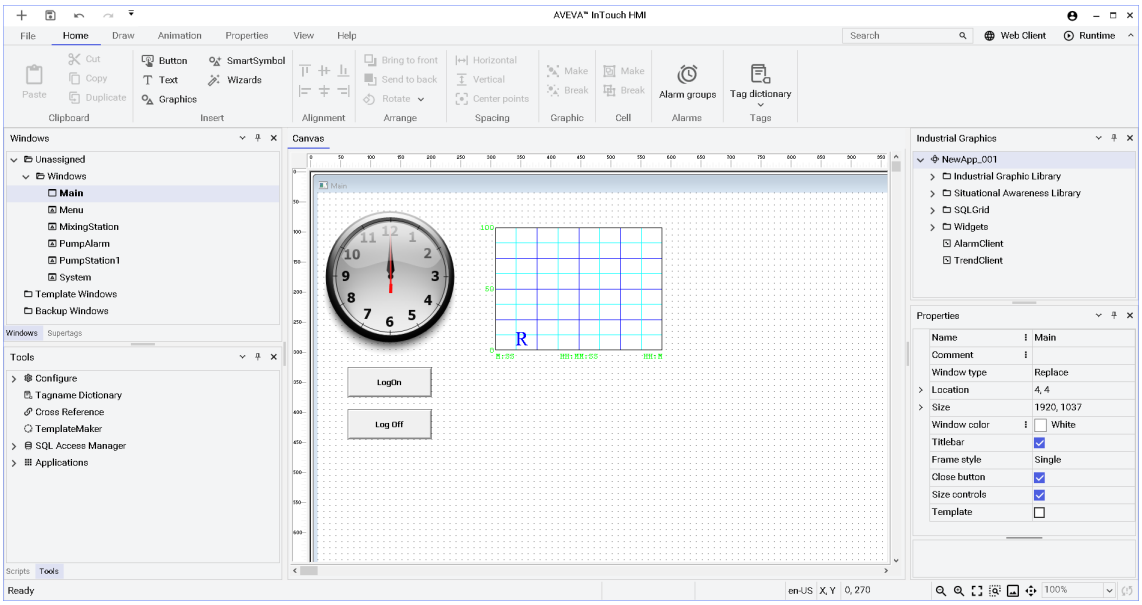
   一个状态框会显示签入进度。

   如果在尝试修改包括所有 **Galaxy 图**形设置时部署了 InTouchViewApp 实例，则会看到一条提示性消息，指示要取消部署所有部署的实例。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 7 窗口

应用程序窗口是一个容器，其中包含一个或多个图形，可在生产过程中用做模型。例如，您可以在一个窗口中显示一套设备，在另一个窗口中显示与该套设备有关的报警信息表。



您可以创建任意数量的窗口，并可以定义窗口属性，如背景颜色、屏幕位置、窗口标题，等等。

创建应用程序窗口

创建新的应用程序窗口时，其缺省设置采用先前创建的或当前活动的 InTouch 窗口的设置。以下设置将反映先前创建的窗口的设置：

- 窗口名：最多可以包含 32 个字符，并且不能包含除美元符号 ($)、井号 (#) 和下划线 (_) 之外的任何特殊字符。创建窗口时，仅需要提供窗口名。所有其它项目都是可选的。

  注意：如果窗口名包含不支持的字符，则下划线 (_) 将替换每个不支持的字符，包括空格字符。如果迁移某个应用程序，那么在编辑窗口或修改任何窗口属性时，将会用下划线 (_) 替换窗口名中的任何特殊字符。这适用于所有类型的窗口，包括应用程序、框架和模板。

  如果您不希望它替换为 “_”，请在 .ini 文件中更新 WindowNameWithSpecialCharacters = 1。

- 注释：可以为窗口包含，但仅用于提供信息。注释出现在文件列表中，但应用程序并不使用它。
- 窗口尺寸值：缺省条件下，窗口的尺寸值设置为先前创建的窗口的尺寸。如果通过拖动窗口边框来手工更改窗口的大小，这些值也会自动修改。

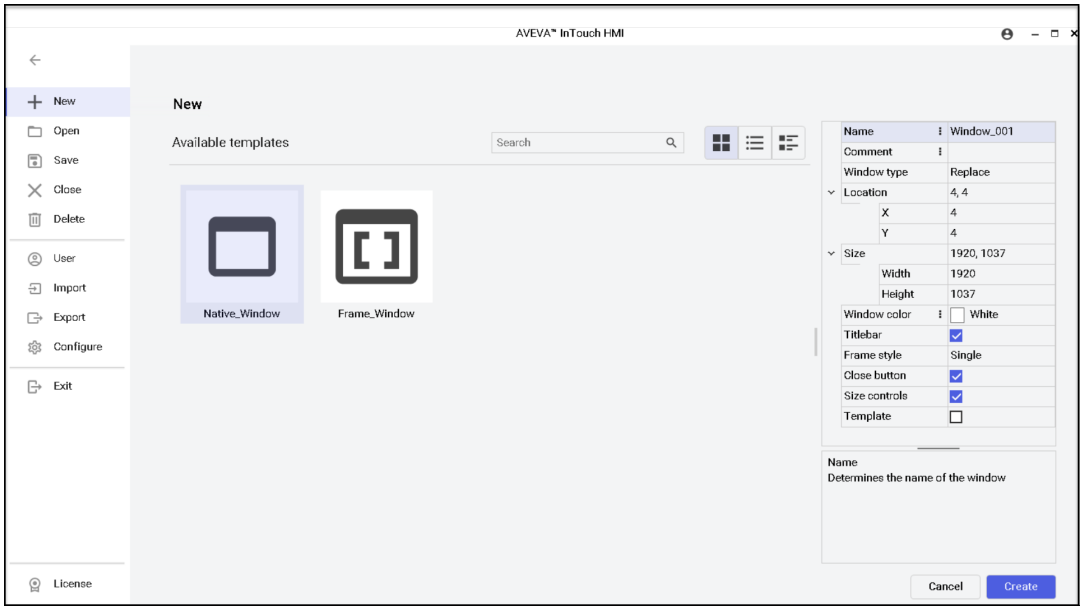
  注意：如果您指定的应用程序目标分辨率与屏幕分辨率不同，应用程序窗口可能超出画布边界。但是，只有目标分辨率画布边界内的窗口会在运行时显示。

创建新窗口

1. 在文件菜单上，选择新建。

   新建窗口显示可用于创建新的本机窗口和框架窗口的模板。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 从可用模板中**选择**要**创**建的窗口**类**型。

   或者在搜索框中，**输**入要据以**创**建窗口的模板名称，然后按 Enter **键**。

3. 要配置模板的**视图**布局，**请选择图标**、**列表**或**详细信息**。
4. 使用“窗口属性”面板配置基本窗口属性。

   窗口属性分**为**三个**类别**：外**观**、布局和窗口**样**式。

5. 要配置新窗口的外**观**：
   - 在**名称**文本框中，**输**入**标识**窗口的唯一名称。
   - 在**注释**文本框中，**输**入要与窗口**关联**的任何注**释**。注**释**必**须为** 60 个字符或者更少。
   - **选择窗口颜**色按**钮**，以**选择**窗口的背景**颜**色。
   - **选择**窗口的 **FrameStyle**，以在窗口周**围显**示**边**框。
     - **选择单一**可以得到**单**个一**维边**框。
     - **选择加倍**可以得到具有三**维边**框的窗口。
     - **选择无**可以得到不**带边**框的窗口。

   选择名称、**注释或窗口颜色**字段旁**边**的省略号，以**执**行以下操作之一：
   - 重置 - 将字段**值**恢**复为**其缺省**值**。
   - **编辑** - 允**许**修改字段的**详细**信息。
   - 排序 - 按字母、**归类**或**归类**字母**顺**序**对**“窗口属性”字段**进**行排序。
   - **显**示描述 - **选择显**示/**隐**藏描述
   - **显**示工具**栏** - **选择显**示/**隐**藏工具**栏**。

6. 要配置新窗口的布局：
   - 从**窗口类型**下拉菜**单**中，**选择**不同的**选项**来看一看窗口在运行**时**如何打**开**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 如果将窗口的“窗口类型”选择为替换，那么当该窗口出现在屏幕上时，与它相交的任何窗口都自动关闭。
- 如果将窗口的“窗口类型”选择为覆盖，那么该窗口会出现在当前打开的窗口的上方。它可以比自己所覆盖的窗口更大。覆盖窗口关闭时，它后面的任何窗口都会重新出现。选择重叠窗口后面的窗口的任何可见部分，便可以将该窗口置前，并使之成为活动窗口。
- 如果将窗口的“窗口类型”选择为弹出，那么该窗口将始终显示在所有其他窗口的上方。弹出窗口通常需要用户作出响应才能删除。

- 展开位置字段以指定窗口的位置和大小。
  - 在 **X** 框中，输入设计区域的左侧边缘与要定义的窗口的左侧边缘之间的像素数。
  - 在 **Y** 框中，输入设计区域的顶部边缘与要定义的窗口的顶部边缘之间的像素数。
- 展开大小字段，以指定窗口的高度和宽度（以像素为单位）。

7. 要配置新窗口的窗口样式：
   - 选中关闭按钮复选框可在标题栏上包含一个按钮，用于在设计时或运行时关闭窗口。即使将这些窗口导出到其他应用程序中，它们也仍会保留关闭窗口按钮。
   - 选中最大化按钮复选框可在标题栏上包含一个按钮，用于在设计时或运行时最大化窗口。
   - 选中最小化按钮复选框可在标题栏上包含一个按钮，用于在设计时或运行时最小化窗口。
   - 选中大小控制复选框可允许用户在 WindowMaker 中调整窗口的大小。
   - 选中标题栏复选框可包含标题栏。
   - 选中模板复选框可将设置保存为模板以供将来使用。
   - 从 **WindowState** 下拉菜单中，选择窗口的缺省状态：正常、最小化或最大化。
8. 选择确定。

将应用程序窗口设置为模板窗口

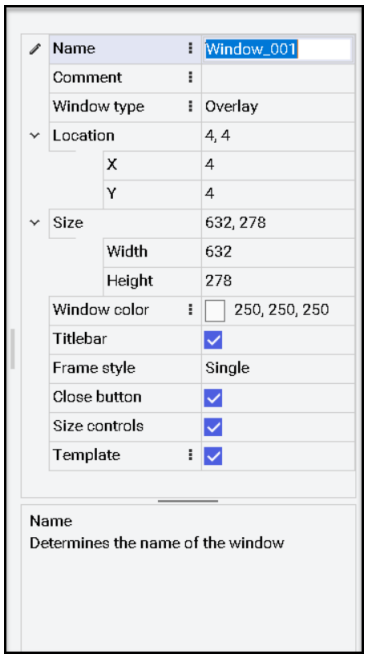
您可以使用“窗口属性”设置将成为模板窗口的应用程序窗口。这样可以复用窗口并减少配置时间。

将应用程序窗口设置为模板窗口

1. 通过执行以下操作之一打开窗口属性窗格：
   a. 要将新窗口设置为模板窗口，在文件菜单上选择新建。
   b. 要将现有窗口设置为模板窗口，在“窗口”窗格上选择该现有窗口。

   窗口的“属性”窗格显示在屏幕的右侧。
2. 选中窗口属性窗格中的模板复选框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 选择确定。

   该窗口现在显示在“窗口”窗格下的“模板窗口”文件夹中。它也显示在模板窗口浏览器中。

注意：您可以取消选中“窗口属性”窗格中的“模板”复选框来将模板窗口更改回应用程序窗口。还可以在“模板窗口”文件夹之间拖放窗口来更改此属性。

从模板窗口创建应用程序窗口

您可以从“模板窗口浏览器”中提供的任一模板窗口创建 InTouch 应用程序窗口。

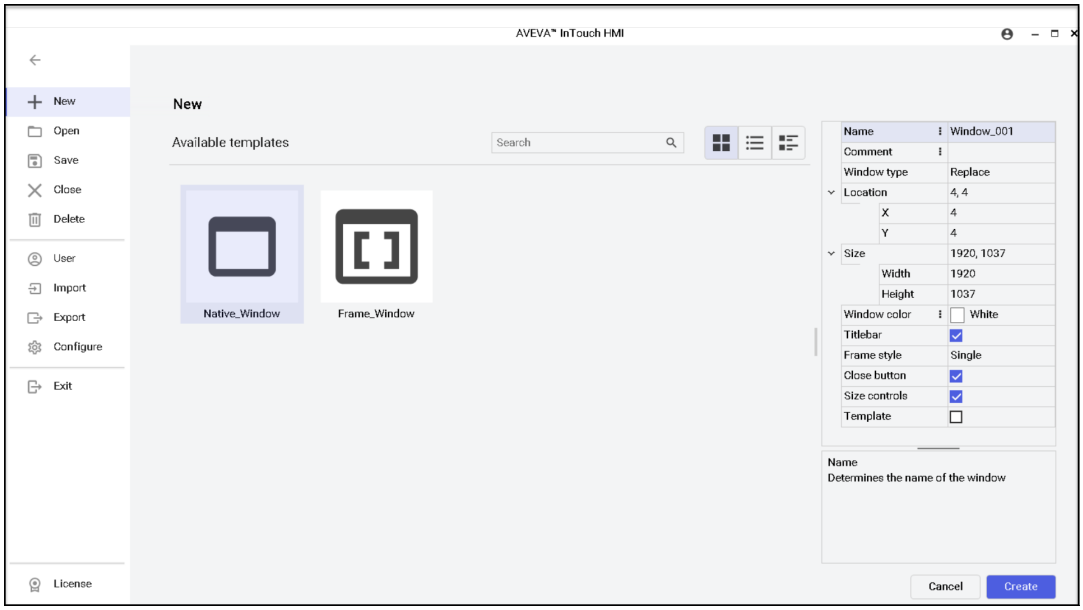
注意：只有选中“模板”属性的窗口才会显示在“模板窗口浏览器”中。如需详细的操作程序，请参阅将应用程序窗口设置为模板窗口。

从模板窗口创建应用程序窗口

1. 在文件菜单上，选择新建。

   此时显示可用模板。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 浏览模板窗口缩略图，然后选择窗口。

   注意：每个模板窗口缺省都以略图视图显示。窗口略图在与整个应用程序有关的设计时间内与实际窗口位置成比例显示。

3. 选择确定。

   注意：如果已迁移的应用程序中的某个模板窗口的窗口名包含不支持的特殊字符，将用下划线 (_) 替换新应用程序窗口中的每个不支持的字符，包括空格字符。

使用框架窗口

如果您使用独立或托管 InTouch 应用程序，可以创建和开发框架窗口来补充或取代应用程序窗口。通过框架窗口，您可以存放支持平移、缩放和触摸功能的工业图形。

框架窗口支持的属性大多数与应用程序窗口相同。但有几个限制。

框架窗口不能：

- 支持多个工业图形
- 存放本机 InTouch 控件
- 存放 ActiveX 控件
- 存放 SmartSymbol
- 支持窗口组件的撤消/恢复操作
- 翻转或旋转嵌入符号的操作
- 支持剪切、复制、粘贴或重复操作
- 提供对框架窗口中嵌入的工业图形的交叉引用支持
- 支持将窗口转换为工业图形
- 支持为嵌入自动化对象图形创建新实例、编辑其实例或选择替代实例

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建框架

1. 在文件菜单上，选择新建。

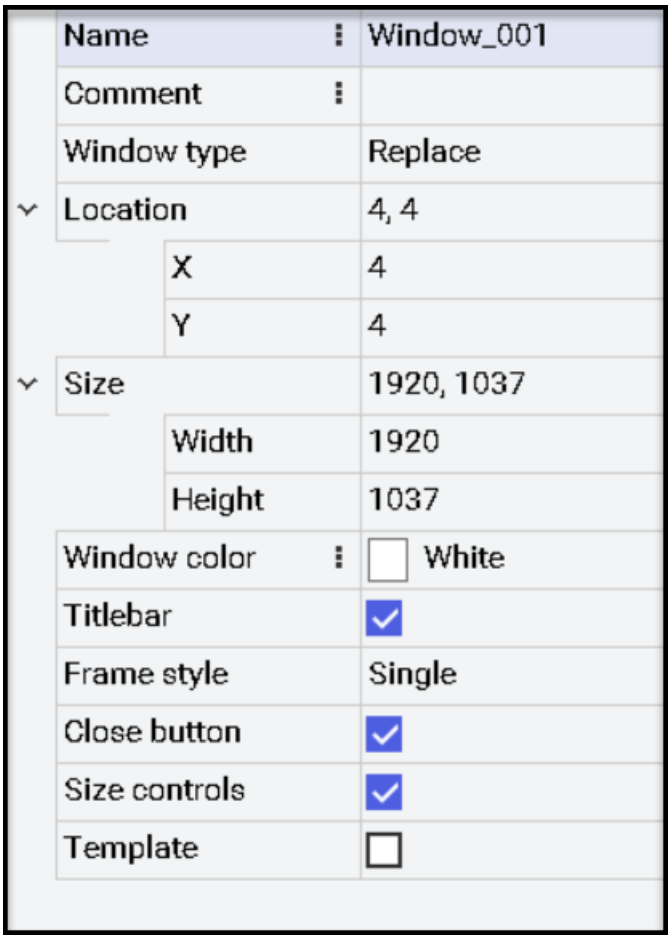
   此时出现含有可用模板列表的新建窗口。

2. 选择框架窗口。
3. 配置窗口属性，然后选择确定。

使用属性面板

您可以直接从 WindowMaker 画布上的“属性”面板编辑框架属性。创建窗口时，“属性网格”中会填入框架属性：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



缺省情况下，“工业图形工具箱”和“属性网格”显示在画布的右侧。您可以将“图形工具箱”或“属性网格”固定到画布的左侧。要将 WindowMaker 恢复为缺省值，请选择查看、恢复布局。

注意：应用程序窗口属性不会填充在属性网格中。

框架窗口支持的属性与应用程序窗口相同，还增加了三个属性。对这些属性的说明如下。

如需每个窗口属性的完整说明，请参阅创建应用程序窗口。

框架属性	功能
MaximizeButton	启用框架右上角的“最大化”按钮。 缺省值为 **False**。
MinimizeButton	启用框架右上角的“最小化”按钮。 缺省值为 **False**。
WindowState	显示窗口的初始状态：正常、最小化或最大化。 缺省值为正常。

对框架属性的更改会立即反应在框架中，但 **FrameStyle** 属性除外。要配置此属性，必须先将 **SizeControl** 和 **TitleBar** 属性设置为 **False**。

FrameStyle 下拉菜单将变为启用状态。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用框架窗口中嵌入的图形

工业图形可以嵌入到框架窗口中，方式与嵌入到应用程序窗口中相同。System Platform IDE 和 InTouch WindowMaker 中均集成了“工业图形工具箱”。该“图形工具箱”包含多个单独的文件夹，分别含有预定义符号的工业图形库和 Situational Awareness Library。工业图形库包含逼真的标准工业对象符号。

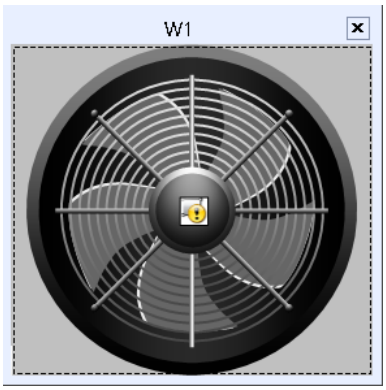
Situational Awareness Library 符号看起来很简单，提供最少的外观细节，以便可以有效地让操作员了解其功能用途和状态，而不显示无关信息。大多数 Situational Awareness Library 符号都设计为“符号向导”，其每个符号都融入了外观和功能的多种配置。要为符号选择功能配置，只需从符号的“向导选项”中选择选项。

执行以下任一操作将符号嵌入框架：

- 创建框架窗口后，选择画布上的嵌入图形按钮 并浏览图形。
- 创建框架窗口后，选择画布上的嵌入图形按钮 ，以在“工业图形编辑器”中创建图形，并保存该图形。创建的图形将嵌入到窗口中。
- 将工业图形从“图形工具箱”拖放到框架上
- 选择“嵌入工业图形”工具栏按钮 并浏览图形
- 使用框架窗口的上下文菜单嵌入图形

您可以使框架自动适合嵌入的工业图形。要使框架适合嵌入的符号，请使用鼠标右键单击窗口，然后选择适合符号。

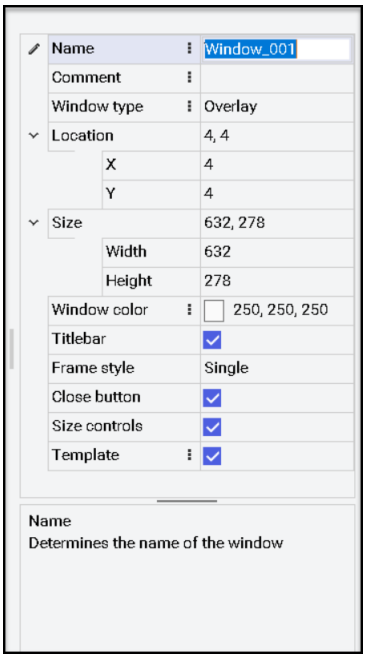
框架将调整大小以适合符号：



注意：适合符号不支持撤消。如果您手动更改框架窗口的尺寸或更改大小属性，则需要再次执行**适合符号**操作。

在框架中嵌入工业图形时，符号将填充属性网格下拉菜单，如下所示：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您可以在框架窗口和符号属性之间切换。执行以下任意操作：

- 从下拉菜单中选择符号名或框架窗口名
- 使用鼠标右键单击框架窗口，然后选择窗口属性。
- 选择标题栏或框架边框以在网格中显示框架窗口属性

  选择嵌入的符号以显示符号属性。

必须配置符号属性才能启用支持键盘、鼠标和触摸手势的平移和缩放功能。下面列出了可配置的符号属性。

符号属性	功能
维持纵横比	调整新型框架的大小时，维持符号的纵横比。缺省值为 **=True**。
符号名称	设置新型框架托管的符号的名称
交互模式	设置交互模式。选项包括： • **无**：禁用平移与缩放 • **PanZoom**：缺省值，启用平移与缩放
显示缩放控件	显示符号与缩放控件。选项包括： • **自动**：根据需要显示控件 • **可见**：始终显示控件

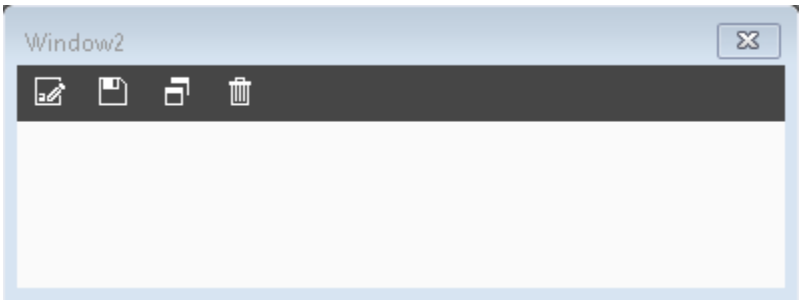
如需有关在运行时使用平移与缩放功能的详细信息，请参阅《*AVEVA™ InTouch HMI 为 InTouch HMI 组件创建标准*》中的“平移与缩放”。

使用框架窗口上的功能区

创建框架窗口时，窗口的功能区中提供了以下选项。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 编辑符号 - 启动图形编辑器，可以在其中编辑符号。如需有关详细信息，请参阅《工业图形编辑器用户指南》。
- 保存窗口 - 保存窗口，请参阅打开、保存和关闭窗口。
- 窗口另存为 - 创建重复窗口，请参阅创建窗口的副本。
- 删除窗口 - 删除窗口，请参阅删除窗口。



使用鼠标右键单击框架窗口的画布时，功能区会突出显示一些可用选项。

修改应用程序窗口

开发应用程序时，可以随时修改窗口的属性。

修改应用程序窗口的属性

1. 右键单击窗口名称，然后选择属性。
2. 在窗口属性面板中进行更改。
3. 选择确定。

如需有关窗口选项的详细信息，请参阅创建应用程序窗口。

打开、保存和关闭窗口

打开窗口

在开发应用程序期间，只要计算机内存足够，便可以打开任意多的窗口。

打开窗口

1. 在文件菜单上，选择打开。

   此时出现选择要打开的窗口对话框，其中列出应用程序中所有窗口的名称。
2. 执行以下任一操作：
   - 要打开一个窗口，请双击相应的窗口名。
   - 要打开多个窗口，请选中要打开的各个窗口对应的复选框，然后选择“确定”。

保存窗口

保存窗口时，与该窗口关联的所有图形、QuickScript、属性等同样会保存下来。

保存窗口

1. 在文件菜单上，选择保存。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现**选择要保存的窗口对话**框，其中列出所有窗口的名称。

2. 选择需要保存的窗口。
3. 选择确定。

关闭窗口

关闭经过修改的窗口时，程序会提示保存所作的更改。

关闭窗口

1. 在文件菜单上，选择**关闭**。

   此时出现**选择要关闭的窗口对话**框，其中列出所有当前打开的窗口的名称。

2. 选中窗口名称旁的复选框。
3. 选择确定。

查看窗口的缩略图预览

在 WindowMaker 中可以查看窗口的略图预览。当应用程序有多个窗口时，该功能尤其有用。您可以在打开窗口前先查看窗口的略图预览，来确认它是否为需要的窗口。

您可以查看窗口的缩略图预览。仅当窗口已保存后，系统才会提供并更新略图预览。如果是迁移应用程序或从 XML 文件导入窗口，那么无需明确地保存窗口即可查看略图预览。如果是从其它应用程序导入窗口，那么无需明确地保存窗口也可以查看略图预览。但如果导入窗口后对其进行了修改，那么必须保存窗口后才能在略图中查看更改。

您还可以更新 InTouch 应用程序中所有窗口的略图。必须先关闭所有窗口，才能更新略图。

更新所有窗口的缩略图

- 在**查看**菜单上的更新组中，选择**窗口缩略图**。

更新特定窗口的缩略图

1. 右键单击该窗口，然后单击**更新缩略图**。

   此时会出现一条确认消息。

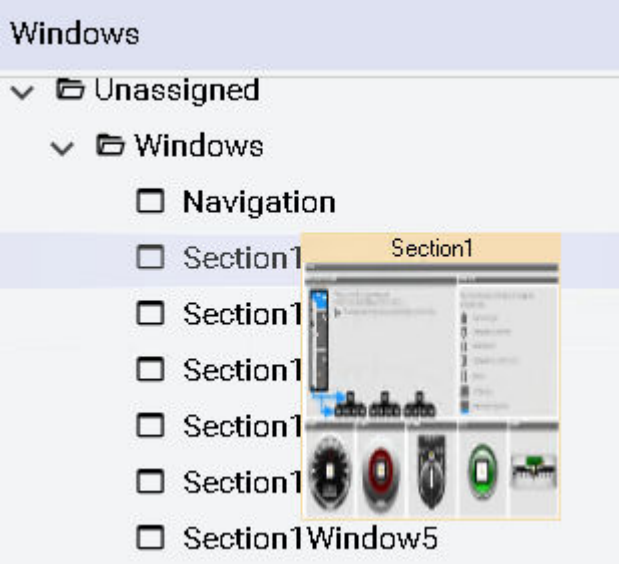
2. 选择**继续**。系统会更新 InTouch 应用程序中所有窗口的略图。

   系统会打开并关闭所有窗口以更新略图。

查看窗口的缩略图预览

1. 打开 WindowMaker。
2. 打开应用程序并保存窗口。
3. 将指针移到包含窗口名和相关图标的长方形区域。此时会出现略图预览。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



创建窗口的副本

有非常类似的过程需要模拟与控制时，您可能会希望创建窗口的副本，然后按照第二个过程或单元对该副本进行自定义。

您可以随窗口一起创建与该窗口关联的所有图形、QuickScript、属性等的副本。

在开始之前，要创建副本的窗口必须处于打开状态，并至少保存过一次。一次只能创建一个窗口的副本。

创建窗口的副本

1. 在“窗口”窗格中，右键单击窗口，然后选择另存为。

   此时出现保存窗口对话框。
2. 在新名框中，输入新窗口的名称。
3. 选择确定。

注意：如果原始窗口名包含不支持的字符，则下划线 (_) 字符将替换重复的窗口名中的每个不支持的字符。

删除窗口

要节约计算机的存储空间时，或是“应用程序浏览器”中的窗口列表太长以致于无法管理时，可以删除未使用的窗口。

警示：确保没有误删窗口。已删除的窗口无法使用撤消来恢复。

删除窗口

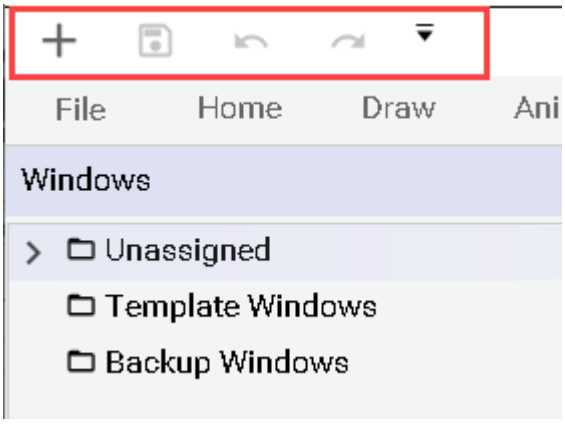
1. 在文件菜单上，选择删除。

   此时出现选择要删除的窗口屏幕。
2. 选择要删除的窗口名，然后选择确定。出现消息时，选择是。
3. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用快速访问工具栏

快速访问工具栏是一个可自定义的工具栏，其中包含一组预定义的或常用的命令，用于从任何屏幕进行快速访问。缺省条件下，快速访问工具栏位于 WindowMaker 屏幕的左上角。



从快速访问工具栏执行命令

您可以从快速访问工具栏执行以下命令。

注意：虽然快速访问工具栏上的大多数命令在其它功能区组中也可用，但有几个命令仅在快速访问工具栏中可用。

快速访问工具栏命令	描述
新建	创建新窗口。
打开	打开现有窗口
保存	保存窗口
另存为	通过指定不同的名称和位置来创建并保存窗口的副本
全部保存	保存所有打开的窗口
关闭	关闭打开的窗口
删除	删除窗口
导入	导入窗口
导出	导出窗口
打印	打印窗口
WindowViewer	在 WindowViewer 中查看窗口
转换为工业图形	将现有的图形转换为工业图形
撤消	取消上次操作
恢复	重复上次的操作

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

退出	退出 WindowMaker

自定义快速访问工具栏

缺省条件下，快速访问工具栏显示以下命令：新建、保存、撤消、恢复

您可以选择在快速访问工具栏上添加命令或从要显示的列表中删除命令。

将命令添加到快速访问工具栏：

1. 选择快速访问工具栏上的向下箭头。
2. 选择要添加的命令。

从快速访问工具栏删除命令：

1. 选择快速访问工具栏上的向下箭头。
2. 突出显示所选命令。

放置快速访问工具栏

缺省条件下，快速访问工具栏位于功能区的上方。

- 要将快速访问工具栏放置在功能区的下方，请选择快速访问工具栏上的向下箭头，然后选择在功能区下方显示。
- 要将快速访问工具栏放置在功能区的上方，请选择快速访问工具栏上的向下箭头，然后选择在功能区上方显示。

使用快速访问工具栏调整功能区的大小

您可以从快速访问工具栏中选择最小化或最大化功能区。

- 最小化功能区：此视图只显示菜单项，而不显示组。要最小化功能区，请选择快速访问工具栏上的向下箭头，然后选择最小化功能区。
- 最大化功能区：此视图显示菜单项和组。要最大化功能区，请选择快速访问工具栏上的向下箭头，然后选择最大化功能区。

打印有关 InTouch 窗口的信息

您可以打印 InTouch 窗口的以下信息：

- 窗口中的图形对象的详细信息。例如，各种类型的对象的窗口位置、用于文本对象的字体、为工业图形定义的自定义属性等。
- 窗口中使用的动画链接的详细信息。
- 与窗口关联的脚本。
- 窗口中使用的标记。

您可以通过打印机打印图形对象的详细信息，或将其打印到 .html 文件。系统会为每个窗口创建一个 .html 文件。该 .html 文件包含：

- 窗口图片，作为引用的 .png 文件。
- 有关窗口中每个图形对象的详细信息的列表。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

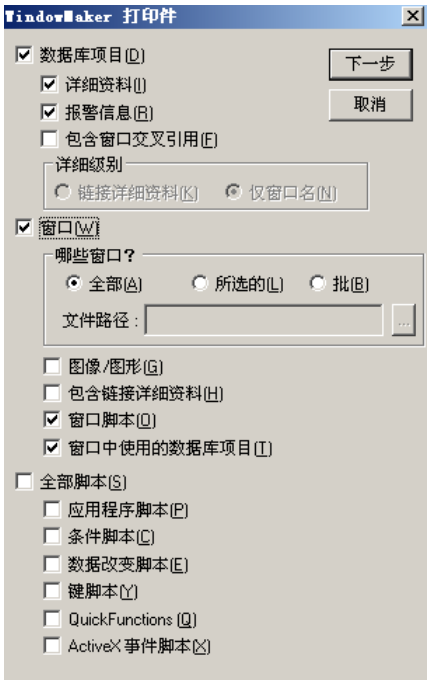
如果打印多个窗口中图形对象的详细信息，则会创建一个 .html 摘要文件，该文件包含一些链接，指向具体窗口的 .html 文件。

如需有关动画链接、脚本或标记的详细信息，您可以通过打印机打印出来或打印到 .txt 文件。

打印有关 InTouch 窗口的信息

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在快速访问工具栏上，选择打印。

   此时出现 **WindowMaker 打印件**对话框。



3. 选择窗口。
4. 选择要打印的窗口：
   - **全部**：打印应用程序中所有窗口的信息，包括任何嵌入图形的名称。
   - **所选的**：仅打印特定窗口的信息，包括所选窗口上任何嵌入图形的名称。此时会出现要打印的窗口对话框。选择应用程序中要打印的窗口，然后选择确定。
   - **批**：仅打印在 .csv 文件中指定的窗口的信息，包括任何嵌入图形的名称。

   如需有关 .csv 格式的详细信息，请参阅用于打印窗口的 .CSV 格式。
5. 选择希望为所选的窗口打印的内容：
   - **图像/图形**：打印窗口中的所有图形对象的有关信息。
   - **包含链接详细信息**：打印窗口的链接详细信息。
   - **窗口脚本**：打印与窗口关联的脚本。
   - **窗口中使用的数据库项目**：打印窗口中使用的标记。
6. 选择下一步。此时出现**选择输出目标**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

7. 执行以下操作之一：
   - 选择发送输出到打印机以打印信息。
   - 选择发送输出到文本文件以创建单个 .txt 文件。
   - 选择发送输出到 HTML 文件以创建一个 .html 摘要文件，以及针对您指定的每个窗口的单个 .html 文件和 .png 文件。如果 .html 与 .png 文件已经存在，则系统会自动覆盖这些文件。
8. 选择打印。

从命令提示符中打印窗口信息

您可以从命令提示符中打印窗口信息。要执行该操作，请创建一个包含窗口名的 .csv 文件，然后在打印命令中引用该 .csv 文件。

打印命令运行时，WindowMaker 自动打开缺省的 InTouch 应用程序，执行打印操作，然后关闭。从命令提示符中打印仅适用于独立的 InTouch 应用程序。

从命令提示符中打印窗口信息

1. 创建一个 .csv 文件，包含要打印的窗口的名称。

   如需有关 .csv 格式的详细信息，请参阅用于打印窗口的 .CSV 格式。
2. 退出 WindowMaker 并关闭 .csv 文件。
3. 打开命令提示符。
4. 输入命令以打印相关信息。

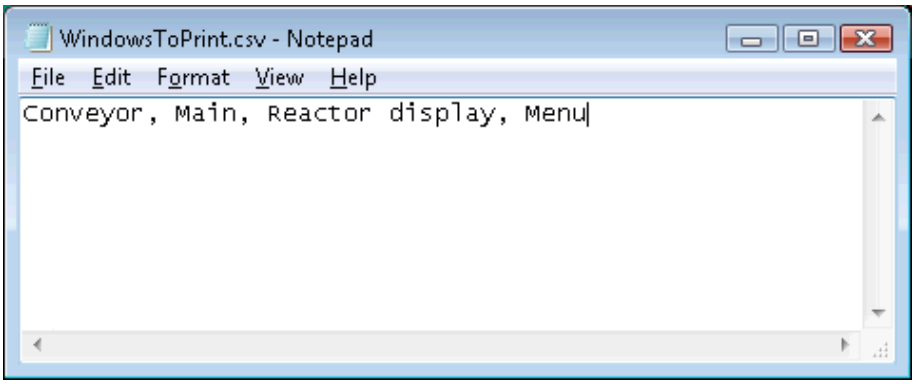
   如需有关命令语法的详细信息，请参阅从命令提示符中进行打印的语法.
5. 按 **Enter** 键。此时 WindowMaker 启动并打印相关信息。

用于打印窗口的 .CSV 格式

可以通过 .csv 文件，将窗口信息用于多种格式，如 Microsoft Excel。在 .csv 文件中创建一行，并包含用逗号隔开的每个窗口名：

Window1,Window2,Window3,WindowN

例如：



如果使用 Microsoft Excel 来创建文件，则应该只有一行，在该行的每个列单元格中包含每个窗口的名称。

不得在窗口名中使用反斜杠字符 (\)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

必须使用逗号，不得使用其它任何分隔符。

从命令提示符中进行打印的语法

命令语法是：

```
"<WindowMaker 路径名>" COMMANDFILE="<CSV 文件>" ALL OUTPUTTARGET=<目标名>
```

其中：

<WindowMaker 路径名> 是 WindowMaker 应用程序 (WM.exe) 的路径。

- <CSV 文件> 是指定要打印的窗口的 .csv 文件的名称。
- <目标名> 是输出位置，可以是打印机，也可以是 .html 文件。
- ALL 是用于打印所有链接、数据库项目以及脚本信息的命令。如果没有包含 ALL 命令，则仅打印图形对象信息。

本例中，所有链接、数据库项目以及脚本信息都会发送到缺省打印机：

```
"C:\Program Files\Wonderware\InTouch\wm.exe" COMMANDFILE="D:\print.csv" ALL OUTPUTTARGET =
PRINTER
```

本例中，图形对象信息会发送到 .html 文件：

```
"C:\Program Files\Wonderware\InTouch\wm.exe" COMMANDFILE="D:\print.csv" OUTPUTTARGET = HTML
<DEMOAPP.html>
```

配置上下文菜单的深度

上下文菜单的深度也称为窗口阴影或上下文菜单的阴影，可为屏幕上的每个浮动窗口创建边框。这有助于将一个窗口与其它重叠的窗口区分开。

您可以通过两种方式为 InTouch 应用程序配置上下文菜单的深度：

- 在控制面板中修改系统属性
- 在注册表编辑器中修改注册表项

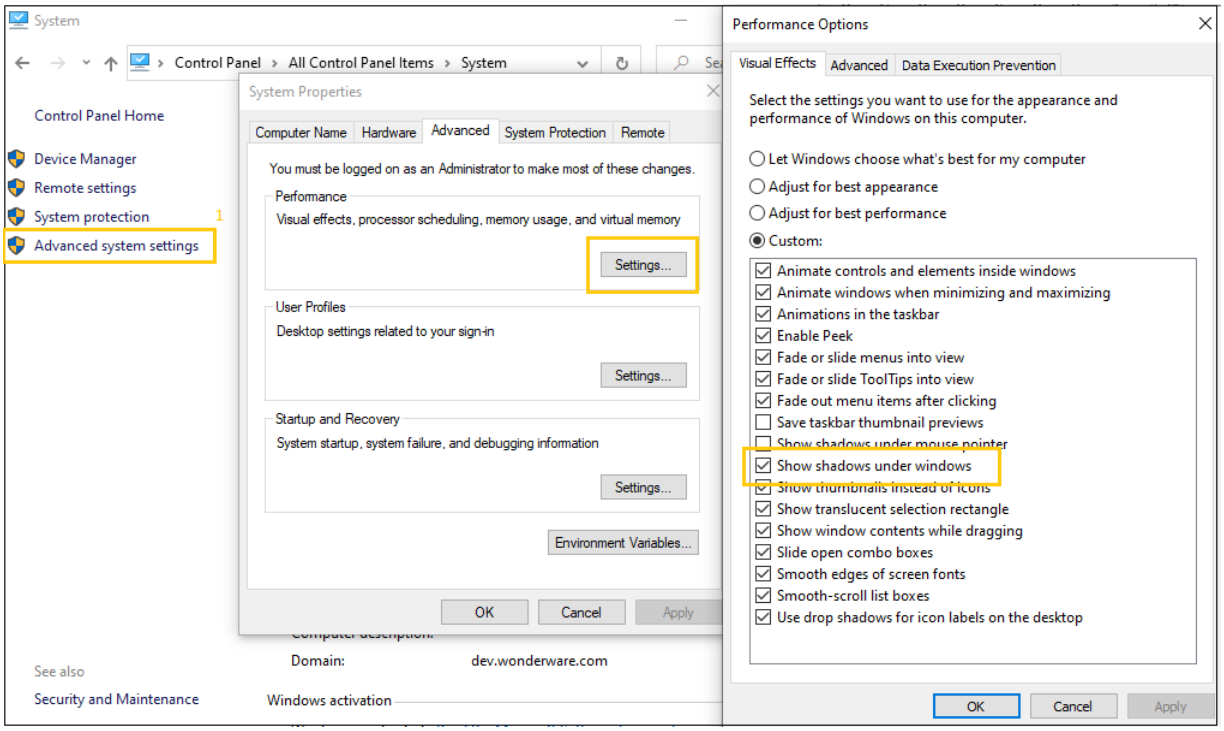
使用控制面板配置上下文菜单的深度

1. 打开“控制面板”。
2. 突出显示系统，然后选择高级设置。

   此时出现系统属性对话框。
3. 在高级选项卡的性能部分下，选择设置。

   此时出现性能选项对话框。
4. 选中在窗口下显示阴影复选框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



使用注册表编辑器配置上下文菜单的深度

1. 打开“注册表编辑器”。
2. 导航到 HKEY_CURRENT_USER\Software\Microsoft\Windows\DWM。
3. 将值 ColorPrevalence" -Type DWORD 设置为 1。
4. 重新启动 WindowMaker。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 8 图形元素

本节包括：

WindowMaker 对象

使用工业图形编辑器 (IGE)

在 WindowMaker 中使用工业图形

将 InTouch 窗口转换为工业图形

使用符号向导编辑器创建符号向导

了解趋势笔历史数据检索

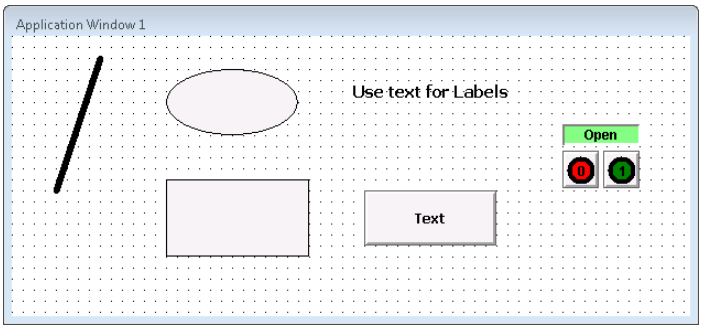
在运行时期间更改趋势笔属性

配置多笔趋势

WindowMaker 对象

图形对象是您构建的人机界面 (HMI) 应用程序的关键部分。

在构建应用程序的过程中，您可以创建简单的对象、将简单的对象合并成更复杂的对象，以及使用一些预先定义好的复杂对象。



复杂对象的各个元素通常组合在一起以：

- 防止复杂对象在编辑时分散开。
- 复制整个对象。
- 给各个单独的元素指定一组公共属性。

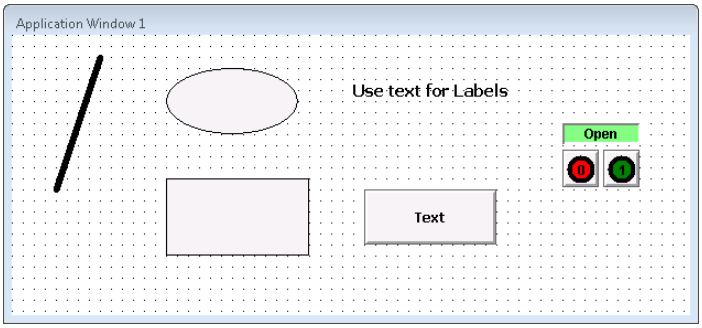
在托管的或高级 InTouch 应用程序中，可以使用由“工业图形编辑器”创建的工业图形。还可以直接从 WindowMaker 的“工业图形工具箱”添加工业图形。如需有关使用工业图形和 Situational Awareness Library 符号的详细信息，请参阅*工业图形编辑器用户文档*或 WindowMaker 帮助。

关于 WindowMaker 对象

图形对象是您构建的人机界面 (HMI) 应用程序的关键部分。

在构建应用程序的过程中，您可以创建简单的对象、将简单的对象合并成更复杂的对象，以及使用一些预先定义好的复杂对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



复杂对象的各个元素通常组合在一起以：

- 防止复杂对象在编辑时分散开。
- 复制整个对象。
- 给各个单独的元素指定一组公共属性。

在托管的或高级 InTouch 应用程序中，可以使用由“工业图形编辑器”创建的工业图形。还可以直接从 WindowMaker 的“工业图形工具箱”添加工业图形。如需有关使用工业图形和 Situational Awareness Library 符号的详细信息，请参阅 *工业图形编辑器用户指南* 或 WindowMaker 帮助。

简单对象

您可以创建以下类型的简单对象：

- 线条
- 形状
- 文本
- 按钮

每种简单对象都有一些控制其外观的属性。

- 线条颜色与粗细
- 填充颜色
- 高度
- 宽度
- 相对方向

创建线条与形状

下表介绍如何执行基本的绘图任务。绘图按钮位于绘图菜单上。

要绘制	选择此选项	按钮
线条	“线条”按钮	
水平或垂直线条	“水平/垂直线”按钮	
长方形	“长方形”按钮	

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要绘制	选择此选项	按钮
圆角长方形	“圆角长方形”按钮 注意：要调整圆角长方形的圆角半径，请参阅更改圆角长方形的圆角半径。	
圆或椭圆	“椭圆”按钮。按住 SHIFT 键可以绘制圆。	

创建按钮

您可以使用按钮来创建与应用程序交互的点。此过程与创建简单的绘图对象非常类似。

如需有关创建多边形的详细信息，请参阅创建折线与多边形。

创建按钮

1. 在**绘图**菜单上的**插入组**中，选择**按钮**。
2. 选择并拖动，以放置按钮及调整其大小。
3. 编辑缺省的按钮文本。执行以下操作：
   a. 右键单击该按钮，然后选择**替换字符串**。
   b. 在**新字符串**框中，输入按钮的文本。
   c. 选择**确定**。

创建折线与多边形

绘制折线与绘制线条略有不同。

创建折线或多边形

1. 在**绘图**菜单上的**形状组**中，选择**多边线**或**多边形**。
2. 选择应用程序窗口以设置第一个点。
3. 再次选择应用程序窗口，以设置更多的点来定义折线或多边形。
4. 双击最后一点。

创建文本

您可以使用文本来给应用程序中的可视化项目贴标签。

创建文本时，文本的格式设置与 WindowMaker 配置屏幕中的设置一致。您可以更改所选文本的外观。如需详细信息，请参阅改文本外观。

输入多行文本时，它们会变成可以独立移动与编辑的对象。您也可以将多个文本对象合并成一个符号，将它们当作一个组进行编辑。

创建文本

1. 在**绘图**菜单上的**插入组**中，选择**文本**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 选择要开始文本的位置。
3. 输入文本，然后按 ENTER 键。此时出现新的文本行。

复杂对象

复杂对象提供比简单对象更多的功能。下表描述了各种类型的复杂对象。

复杂对象	描述
单元	结合成单一整体的两个或更多个对象（包括符号或其它单元）。您可以使用单元来创建虚拟设备，如游标控制器。 对于创建要与不同的标记关联的多个设备，单元非常有用。
符号	结合在一起并视为单个对象的一组简单对象（如线条、图案以及文本）。应用于符号的任何属性更改都会影响该符号中的所有组成对象。符号不得包含位图、按钮、单元、向导或趋势。
SmartSymbol	已经转换成可复用图形模板的 InTouch 单元。在应用程序窗口中可以放置一个或多个 SmartSymbol 模板的实例。对模板所作的任何更改都会传播到这些实例。如需详细信息，请参阅关于 SmartSymbol。
工业图形	使用“集成开发环境”(IDE) 中的“工业图形编辑器”创建的功能非常丰富的图形。
位图容器	可以导入图像（如照片、图画以及屏幕截图）的对象。 您可以旋转位图，也可以给它设置透明背景。如需详细信息，请参阅使用位图容器。
趋势对象	多个标记的实时或历史数据值随时间而变化的图表。如需详细信息，请参阅趋势对象。
向导	只需针对应用程序进行选择、放置以及配置的预先构建的对象。 如需详细信息，请参阅向导。
ActiveX 控件	在应用程序中运行的软件组件。WindowMaker 同时支持 AVEVA 与第三方的 ActiveX 控件。如需详细信息，请参阅使用 ActiveX 控件。

单元与符号

您可以将多个对象合并成两种不同类型的单个整体：单元与符号。单元可包含任何对象。符号只能包含简单对象。符号不得包含单元。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要确定特定的对象是单元还是符号，请双击该对象。

- 如果是单元，则打开替换标记名对话框；如果单元不包含标记名，则出现“替换名”警告消息。
- 如果是符号或简单的图形对象，则打开动画链接选择对话框。

关于单元

使用单元可以合并多个元素，并在它们之间保持固定的间距关系。您也可以使用单元来四处移动多个元素，也可以将它与其它图形对齐。

要更改单元中的元素，必须分解该单元，对元素进行更改，然后重新将这些元素合并成单元。

您可以给单元中的元素设置动画效果，但是无法给单元设置动画效果。单元的大小也无法调整。

关于符号

您可以给符号与简单对象设置动画效果。您也可以使用符号给复杂图形的某些部分设置动画效果。

如果有多个所选对象包含链接，则无法制作符号。

如果将两个符号合并成一个新符号，则原始的符号结构会丢失。如果分解新符号，则它会分解成每个原始符号的各个单独组成部分。两个原始符号都会丢失。

将对象组合成单元

您可以将符号、位图、趋势、按钮、向导以及其它单元合并成一个单元。如果单元中包含符号，则与该符号关联的所有动画链接都会保持不变。

在 SmartSymbol 中使用单元之后，如果分解该 SmartSymbol，则无法调整该单元的大小。

创建单元

1. 选择要包含的对象。
2. 在动画菜单上的单元组中，选择制作单元。

分解单元

1. 选择单元。
2. 在动画菜单上的单元组中，选择分解。

将对象组合成符号

符号不得包含位图、按钮、单元、向导或趋势。如果所选对象之一上附有动画链接，则这些链接会附加到新符号上。

创建符号

1. 选择要包含的对象。
2. 在动画菜单上的图形组中，选择制作图形。

分解符号

1. 选择符号。
2. 在动画菜单上的图形组中，选择分解。

常见操作

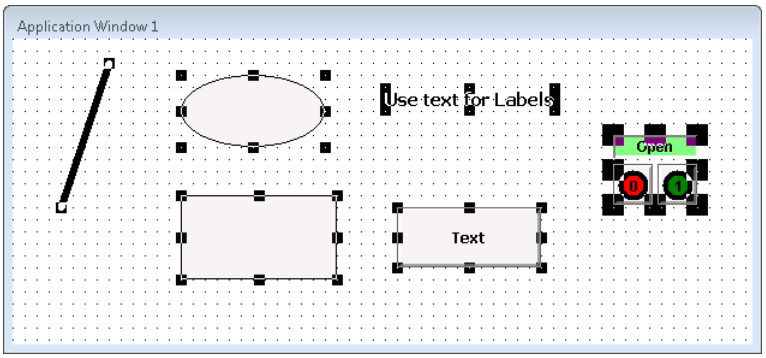
使用鼠标右键单击对象可以看到一个菜单，它显示可应用于该对象的有效命令或动作。您可以：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 选择对象
- 对齐对象
- 层叠对象
- 翻转对象
- 调整对象大小
- 更改字体
- 更改填充
- 控制水平与垂直间距
- 移动对象
- 排列对象
- 撤消更改
- 翻转符号
- 旋转对象
- 更改线条或轮廓
- 删除对象

选择对象

修改对象之前，必须先选择它。选中对象时，它的周边会出现手柄。这些手柄可用于调整对象的大小及/或更改对象的形状。



选择活动窗口中的所有对象

- 在动画菜单上的对象组中，选择全选。

  或者，按 **F2**。

选择一个对象

1. 在动画菜单上的模式组中，单击选择。

   此时会启用“选择模式”。
2. 突出显示要选择的对象。

取消选择一个对象

- 选择窗口的空白区域。

选择多个对象

1. 在动画菜单上的模式组中，单击选择。

   此时会启用“选择模式”。
2. 选择第一个对象，然后按住 SHIFT 键并单击其它对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

选择一组对象

1. 在动画菜单上的模式组中，单击选择。

   此时会启用“选择模式”。

2. 在对象周围拖动一个框。此时会选择完全包含在该长方形内的所有对象。

取消选择一组对象中特定的一个或多个对象

- 按住 SHIFT 键并单击对象。

移动对象

对象可以通过以下方式移动：

- 拖动对象。
- 使用键盘上的方向键。
- 在状态栏的框中输入窗口坐标。

移动对象时，请注意状态栏中坐标的变化情况。

通过拖动来移动对象

- 选择对象并拖动它。

使用方向键移动对象时，对象的移动距离取决于是否显示网格。

显示网格时，对象移动的像素数取决于网格间距，此间距在 WindowMaker 配置屏幕上设置。缺省设置为网格点之间相距十个像素。

显示网格时，

- 按方向键一次，对象移动一个网格点。
- 按 SHIFT + 方向键，对象移动两个网格点。
- 按 CTRL + 方向键，对象移动四个网格点。

不显示网格时，

- 按方向键一次，对象移动一个像素。
- 按 SHIFT + 方向键，对象移动十个像素。
- 按 CTRL + 方向键，对象移动 50 个像素。

要使用方向键移动对象

- 选择对象，然后
  - 按方向键。
  - 按 SHIFT + 方向键。
  - 按 CTRL + 方向键。

对齐对象

您可以根据对象的左侧或右侧边缘、中心、中心点、顶部、中间或底部来对齐它们。

通过使用菜单命令或按钮，可以按多种方式进行对齐。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

选择	或者选择	实现
左对齐		以最左端对象的左边缘为基准，对齐各个对象的左边缘。
水平居中对齐		以对象组的垂直中心线为基准，对齐各个对象的垂直中心线。
右对齐		以最右端对象的右边缘为基准，对齐各个对象的右边缘。
上对齐		以最顶端对象的顶部边缘为基准，对齐各个对象的顶部边缘。
垂直居中对齐		以对象组的水平中心线为基准，对齐各个对象的水平中心线。
下对齐		以最底端对象的底部边缘为基准，对齐各个对象的底部边缘。
中心点对齐		以对象组的中心点为基准，对齐各个对象的中心点。

对齐对象

1. 选择多个对象。
2. 在主页菜单上的对齐组中，选择相应的对齐命令。

层叠对象

您可以将对象放置在其它对象的前面或后面。

将一个对象置于另一对象之后

1. 选择对象。
2. 在主页菜单上的排列组中，选择置后。

将一个对象置于另一对象之前

1. 选择对象。
2. 在主页菜单上的排列组中，选择置前。

   或者，按 SHIFT + F9。

控制水平与垂直间距

您可以在最左端与最右端的所选对象之间沿水平方向分布对象。

您也可以在最顶部与最底部的所选对象之间控制垂直间距。

也可以将对象隔开，使其在中心点对齐。

水平或垂直分布对象

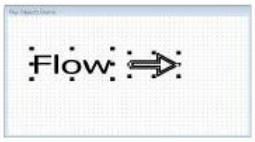
1. 选择对象。
2. 在绘图菜单上的间距组中，选择水平、垂直或中心点。

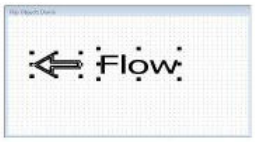
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

翻转对象与单元

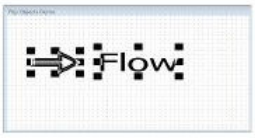
大多数对象都可以水平或垂直翻转。您可以翻转单个对象或一组对象。

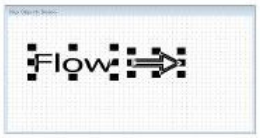
翻转对象相当于将它变换成自己的镜像。文本无法翻转。





翻转单元并不会将单元变成镜像。只有对象组中单元的位置会进行镜像处理。





比较单元翻转前、后的位置。翻转的是位置而不是内容。

翻转对象或单元

1. 选择对象。

2. 在**绘图**菜单上的**排列组**中，选择“旋转”旁边的向下箭头。
3. 选择**水平翻转**，或选择**垂直翻转**。

调整对象大小

对象的大小可以使用两种方法来调整。您可以拖动它，或是指定精确的宽度与高度。

如果打开对齐网格功能，则在按比例调整大小期间，对象会与网格对齐。这会导致垂直与水平尺寸之间的比率稍有偏差。要避免此偏差，请关闭对齐网格功能。

通过拖动来调整对象的大小

1. 选择对象，然后将箭头光标指针置于手柄中央。
2. 拖动手柄以调整对象的大小。

按比例调整对象的大小

- 选择对象，然后按 SHIFT 键并拖动。

按尺寸调整对象的大小

1. 选择对象。
2. 在“属性”面板的 **W**、**H** 框中输入宽度和高度尺寸。

旋转对象

大多数对象（包括符号、文本与位图）都可以旋转。单元无法旋转。

对象可以顺时针或逆时针旋转 360 度，增量为 90 度。

在 WindowMaker 中旋转对象与在运行时或是在 WindowViewer 中动态旋转对象是两回事。在 WindowViewer 中，通过将对象链接到方向动画来旋转对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

旋转对象

1. 选择对象。

2. 在绘图菜单上的排列组中，选择旋转旁边的向下箭头。
3. 选择顺时针，或选择逆时针。

改文本外观

您可以在创建文本之前，通过预先设置缺省值来设置字体的外观；也可以在创建文本之后，再更改字体的外观。如需有关设置字体缺省值的详细信息，请参阅设置字体缺省值。

对于显示动态值的文本对象而言，文本对齐方式属性设置非常重要。对齐方式确定各个变长字段在运行时如何显示。

例如，如果在居中对齐或右对齐的文本字符串的末尾显示一个数值，则每当显示的数字位数发生改变时，整个文本字符串（包括数值）都会居中对齐或右对齐。

通过使用菜单命令或按钮，可以按多种方式配置文本。

作用	选择	按钮
更改字体、样式、颜色或文本大小	字体	
使文本变为粗体	粗体	
使文本变为斜体	斜体	
给文本添加下划线	下划线	
减少或增加字体大小	字体缩小或字体放大	
更改对齐方式	左对齐、居中或右对齐	

要配置文本外观

1. 选择文本对象。
2. 在绘图菜单上的格式组中，选择相应的文本命令。

更改线条与轮廓

您可以更改轮廓化对象周围的线条或轮廓的颜色，以及它们的线型或宽度。轮廓化对象包括填充图案（如椭圆、长方形、多边形）、位图以及其它导入的图像。

更宽的线条在运行时需要更长的时间来绘制。虚线与点划线的宽度只能是一个像素。

设置线条外观的缺省设置

1. 选择窗口的空白区域。
2. 在绘图菜单上的格式组中，选择线条宽度或线型。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 突出显示**线条颜色**工具，然后**选择一种颜**色。

更改线条的颜色

1. **选择**一个线条、一组线条或是具有轮廓的对象。
2. 在**绘图**菜单上的**格式组**中，**选择线条颜色**工具。
3. **选择一种颜**色。

更改线条或轮廓的样式或宽度

1. **选择**对象。
2. 在**绘图**菜单上的**格式组**中，**选择线条样**式或宽度。

删除轮廓

1. **选择**对象。
2. 在**绘图**菜单上的**格式组**中，**选择无线条**。

更改填充

填充的图案是由线条包围着的图案。长方形、圆角长方形、圆、椭圆以及多边形都是填充图案的示例。

更改对象的填充颜色

1. **选择**对象。
2. 在**绘图**菜单上的**格式组**中，**选择填充颜色**工具。
3. **选择一种颜**色。

设置填充图案的缺省颜色

1. **选择**窗口的空白区域。
2. 在**绘图**菜单上的**格式组**中，**选择填充颜色**工具。
3. **选择一种颜**色。

删除对象

您可以删除一个或多个对象。

删除对象

- 执行以下任一操作：
  - 右**键单击**对象，然后**选择清除**。
  - **选择**对象，然后按 Delete 键。

撤消更改

WindowMaker 会记录对每个窗口所作的编辑与格式更改。缺省条件下，WindowMaker 支持 10 级撤消/恢复，其中每级代表一个动作。您可以将 WindowMaker 设置为保留多达 25 级动作。您也可以通过将撤消/恢复的级数设置为零来关闭撤消/恢复功能。

如果关闭窗口，则会清除记录的所有操作。

撤消命令

- 在快速访问工具栏上，**选择撤消**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

恢复命令

- 在快速访问工具栏上，选择恢复。

设置撤消/恢复级数。

1. 在文件菜单上，选择配置，然后选择 **WindowMaker**。

   此时出现 WindowMaker 配置屏幕。
2. 在撤消级别框中，输入级数。

所有对象的特殊操作

您可以修改与操纵简单的对象。您可以：

- 剪切、复制以及粘贴对象。
- 剪切、复制以及粘贴对象链接。
- 创建对象的副本。

剪切、复制以及粘贴对象

WindowMaker 中的剪切、复制以及粘贴操作同其它基于 Windows 的应用程序中的操作大体相同，但也有一些需要注意的重大区别。

剪切、复制或粘贴对象时，同时也会剪切、复制或粘贴与对象相关的属性与动画链接。

粘贴的所有对象在粘贴之后仍保持选中状态，您可以通过移动来调整它们的位置。

剪切对象

- 右键单击对象，然后选择剪切。

复制对象

- 右键单击对象，然后选择复制。

粘贴对象

1. 使用鼠标右键单击窗口中的空白区域，然后选择粘贴。此时光标变为弯头符号。
2. 按住鼠标左键。此时光标变为虚线长方形，大小等于所复制的对象。
3. 拖动该长方形以指定对象的位置。
4. 释放鼠标按钮。

剪切、复制以及粘贴对象链接。

“剪贴板”是剪切或复制的链接的临时储存区。

- “剪贴板”只存储最近的剪切或复制操作对应的链接。
- 您可以将链接粘贴到支持“剪贴板”中的链接的任何对象或符号。
- 如果对象不支持所粘贴的链接（例如，文本对象上的线条颜色链接），则不会粘贴该链接。
- 如果选择多个对象进行粘贴，则这些链接会粘贴到所有的对象。

剪切、复制、粘贴以及清除链接

- 使用鼠标右键单击对象，指向**链接**，然后选择适当的命令。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建对象的副本

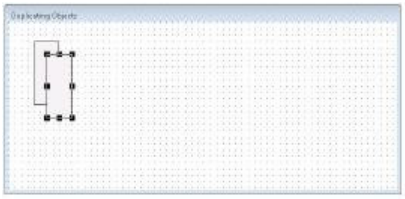
给对象创建副本同复制对象及其动画链接类似，此外也有一个优点，即多次给对象创建副本时，偏移距离与方向同样也会重复。

如果在没有取消选择的情况下移动通过创建副本得到的对象，则再次创建副本时，第三个副本的偏置距离和方向与前两个副本之间的相同。

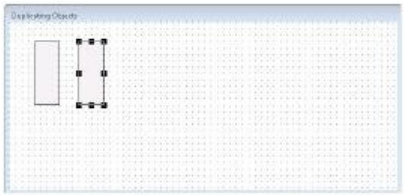
您可以根据需要重复这个操作程序任意多次。

创建对象的副本

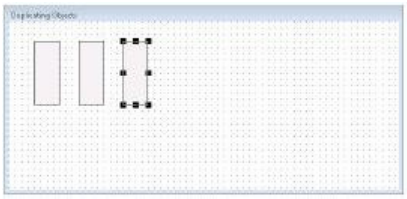
1. 右键单击该对象，然后选择重复。此时会复制对象并将它粘贴到与原始对象存在一定偏移的位置。



2. 保持选择所重复的对象，然后将它拖到一个不同的位置。



3. 同样，在没有取消选择重复的对象的情况下，使用鼠标右键单击该对象，然后选择重复。此时出现对象的第三个副本，它的相对位置与前两个副本的相同。



特殊对象的特殊操作

以下对象类型包含一些可以编辑的独特属性：

- 折线与多边形
- 位图容器
- 位图透明度
- 圆角长方形
- 对象文本

调整折线与多边形对象的形状

您可以调整折线与多边形的形状。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

调整折线或多边形的形状

1. 选择对象。
2. 执行以下操作之一。
   - 在动画菜单上的对象组中，选择调整形状以查看所有调整形状选项。
   - 使用鼠标右键单击该对象，然后单击改变对象外形。

   此时每个形状定义顶点都变成一个手柄。
3. 拖动手柄以调整形状。

向多边形添加顶点

1. 选择对象。
2. 执行以下操作之一。
   - 在动画菜单上的对象组中，选择箭头显示隐藏的命令，然后选择添加顶点。
   - 右键单击对象，然后选择添加顶点。
3. 选择多边形的边缘，然后拖动顶点以更改多边形的形状。

删除多边形的顶点

1. 选择对象。
2. 执行以下操作之一。
   - 在动画菜单上的对象组中，单击箭头显示隐藏的命令，然后选择删除顶点。
   - 右键单击对象，然后单击删除顶点。
3. 选择多边形的顶点，此时会删除该顶点，多边形的形状也随之发生更改。

使用位图容器

位图容器是用于将图形对象（如图片、屏幕截图以及图画）导入应用程序的对象。

支持的位图容器文件类型有 .bmp、.jpeg、.jpg、.pcx 以及 .tga。

导入位图时，位图自动填充位图容器；但也可以将它调整为原始的大小与比例。

位图可以按 90 度的增量旋转。

位图可以包含在单元，但无法包含在符号中。

通过使用 WindowMaker，可以将比 WindowViewer 所能加载的更多位图放入窗口。如果需要将大量的位图放入窗口，在发行应用程序之前，请务必在 WindowViewer 中测试该窗口。

导入位图图像

1. 在绘图菜单上的形状组中，选择位图。

   此时光标变成十字形。
2. 拖动光标以绘制位图容器。
3. 在动画菜单上的位图组中，选择导入。

   此时出现选择图像文件对话框。
4. 选择图像文件名，然后选择打开。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使位图变为原始大小

1. 选择图像。
2. 在动画菜单上的位图组中，选择原始大小。

粘贴位图图像

1. 将图形复制到 Windows 的“剪贴板”。
2. 选择位图工具，并在窗口中绘制一个位图容器。
3. 在动画菜单上的位图组中，选择粘贴。

   或者，右键单击该位图容器，然后选择粘贴位图。

编辑位图

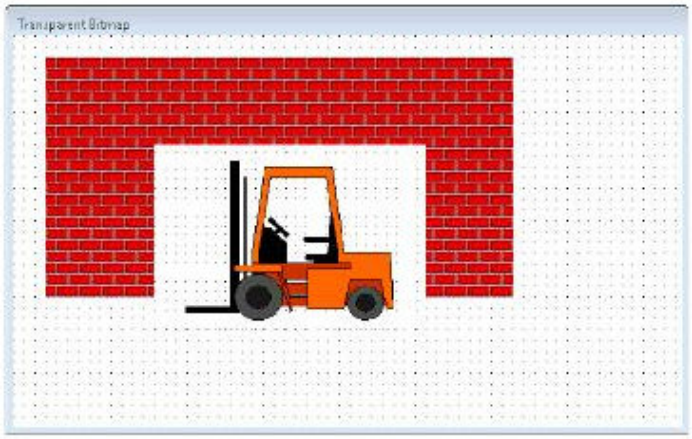
1. 选择位图。
2. 在动画选项卡上的位图组中，选择编辑。

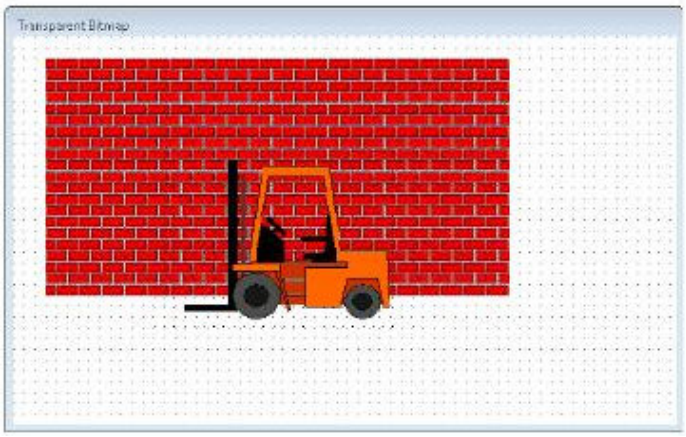
   Microsoft 的“画图”会打开并显示该位图。
3. 在 MS 的“画图”中编辑位图。
4. 保存并关闭 MS 的“画图”。

定义位图透明度

在位图中定义一种透明色时，可以透过使用透明色的所有位置看到窗口背景或位图后面的任何对象。每幅图像只能定义一种透明色。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。





创建透明位图

1. 选择位图，在绘图菜单上的格式组中选择透明色按钮，以打开透明色调色板。
2. 使用鼠标右键单击自定义调色板中的一个颜色方块。此时出现添加颜色对话框。
3. 选择吸管工具。
4. 选择位图中要变为透明色的颜色。此时该颜色会复制到调色板中所选的颜色方块。
5. 选择该颜色方块以便将透明颜色应用于位图。图像中使用该颜色的所有像素都变为透明的。

更改圆角长方形的圆角半径

您可以增加与/或减少“圆角长方形”的圆角半径。

增加或减少圆角对象的半径

1. 选择对象。
2. 在动画菜单上的对象组中，显示隐藏的命令，然后选择放大半径或缩小半径。

替换对象文本

您可以编辑包含文本的对象（如符号、单元或按钮）的文本。

更改文本字符串时，它会保留原始的所有属性，包括字体、样式以及颜色等。文本格式也适用于数值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

更改对象中的文本

1. 选择包含文本的对象或按钮。执行以下任一操作：
   - 在动画菜单上的替换组中，选择字符串。
   - 使用鼠标右键单击文本对象，指向替换，然后选择替换字符串。
2. 在新字符串框中，输入新的字符串，然后选择确定。

更改一系列文本对象中的部分文本

1. 选择所有的文本对象。
2. 在动画菜单上的替换组中，选择字符串。
3. 选择替换。

   此时出现替换文本对话框。
4. 在旧文本框中，输入要替换的部分文本。
5. 在新文本框中，输入新的文本字符串。
6. 选择确定。

   此时新的文本字符串替换掉所选的所有对象中的旧文本字符串。

打印有关 WindowMaker 对象的窗口信息

对于放置在应用程序窗口中的 InTouch 图形对象，您可以打印其相关信息。InTouch 图形对象包括简单的对象（如线条、按钮和文本），以及复杂的对象（如单元、ActiveX 控件、SmartSymbol 和工业图形）。

您可以通过打印机打印图形对象的详细信息，或将其打印到 .html 文件。系统会为每个窗口创建一个 .html 文件。该 .html 文件包含：

- 窗口图片，作为引用的 .png 文件。
- 有关窗口中每个图形对象的详细信息的列表。

如果打印多个窗口中图形对象的详细信息，则会创建一个 .html 摘要文件，该文件包含一些链接，指向具体窗口的 .html 文件。

打印有关 WindowMaker 对象的信息

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在快速访问工具栏上，选择打印。

   此时出现 **WindowMaker** 打印件对话框。
3. 选中窗口复选框。
4. 指定要打印的窗口：
   - 全部打印应用程序中所有窗口的图像/图形。

   备注：如果窗口包含许多图形，或者包含超过托管 WindowMaker 的计算机上可用内存的大图形，那么将应用程序的所有窗口打印到 XPS 文件时，WindowMaker 可能会停止响应。您应该一次打印其中一个窗口，以避免因内存不足而导致的问题。

   - 所选的仅打印特定窗口的图像/图形。此时会出现要打印的窗口对话框。选择应用程序中要打印的窗口，然后选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 批仅打印在 .csv 文件中指定的窗口的信息。

如需有关 .csv 格式的详细信息，请参阅用于打印窗口的 .CSV 格式。

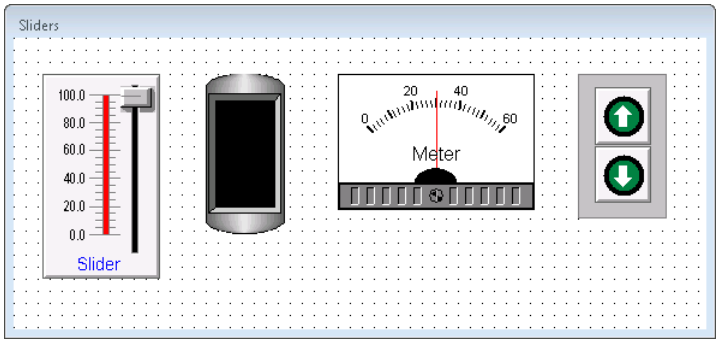
5. 选择图像/图形复选框。
6. 选择下一步。此时出现选择输出目标对话框。
7. 执行以下操作之一：
   - 选择发送输出到打印机以打印信息。
   - 选择发送输出到 HTML 文件以创建一个 .html 摘要文件，以及针对您指定的每个窗口的单个 .html 文件和 .png 文件。如果 .html 与 .png 文件已经存在，则系统会自动覆盖这些文件。
8. 选择打印。

动画对象

通过使用动画链接，可以给对象或符号设置动画效果。动画链接将标记或表达式的值连接到对象或符号。

例如，可以：

- 创建显示罐中液位的游标或罐符号。
- 创建仪表，显示一定范围的值。
- 创建可供操作员控制的触摸屏符号。



两种类型的动画链接

动画链接有两种基本类型：显示链接与触动链接。

- 显示链接用于向操作员显示信息。显示链接的例子有：更改颜色，更改填充级别、水平或垂直移动，以及使对象闪烁。
- 触动链接可供操作员向系统进行输入。触动链接的例子有：游标，或用于响应操作员输入的按钮。

您可以为对象或符号定义多个链接。通过组合各种链接，您可以创建几乎任何屏幕动画效果。

数据显示动画

数据显示动画只用于向操作员显示信息。这些动画不支持操作员进行输入。

创建值显示

使用值显示文本对象显示标记的值。这可用于显示填充级别、开/关状态或报警消息等对象。

您可以使用三种类型*值显示链接*中的任何一种来显示运行时的消息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

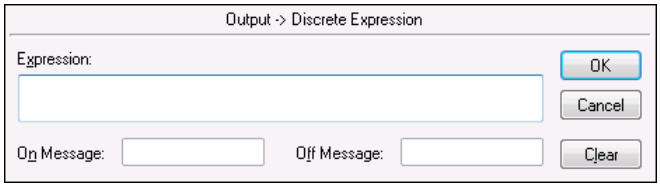
值显示类型	显示
离散	离散值，如开或关。
模拟	模拟表达式的值，如填充级别或速度。
字符串型	字符串表达式的值，如“Fill Level = 100”。

在表达式中，您可以使用长达 1023 个字符。如果需要更长的表达式，请创建一个 QuickFunction，然后在表达式中调用它。

消息会出现在原始文本对象所在位置，使用为该对象设置的字体、大小、颜色、对齐方式及链接的属性。字段的原始内容不会影响运行时的消息。

创建离散值显示链接

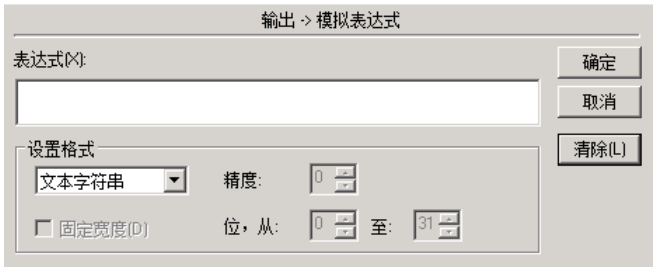
1. 使用鼠标右键单击文本对象，然后选择动画链接。此时出现动画链接对话框。
2. 在值显示区域中，选择离散。此时出现输出 -> 离散表达式对话框。



3. 在表达式框中，输入离散标记名或等于离散值的表达式。例如：
```
Cooling_Pump
```
4. 在打开消息框中，输入在表达式的值等于 1、true、on 或 yes 时要出现的消息。例如：
```
Pump is ON
```
5. 在关闭消息框中，输入在表达式的值等于 0、false、off 或 no 时要显示的消息。例如：
```
Pump is OFF
```
6. 选择确定。

创建模拟值显示链接

1. 使用鼠标右键单击文本对象，然后选择动画链接。此时出现动画链接对话框。
2. 在值显示区域中，选择模拟。此时出现输出 -> 模拟表达式对话框。



3. 在表达式框中，输入模拟（整型或实型）标记名或等于模拟值的表达式。例如：
```
Tank_CV*0.06
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

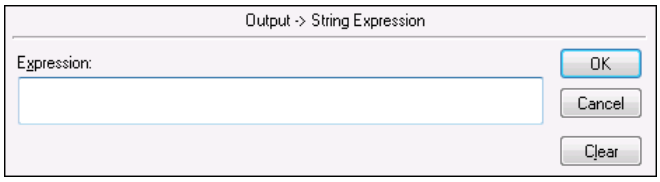
4. 在设置格式区域中，在列表中选择要为其配置运行时高级格式的每个数据类型。固定宽度复选框、精度框以及开始位和结束位框会基于选择的数据类型成为可用状态。如需有关配置这些选项的信息，请参阅文本的高级格式。

   注意：运行时期间，可通过使用指针和鼠标单击和拖拽的方式来调整模拟值输入链接字段的大小。

5. 选择确定。

创建字符串值显示链接

1. 使用鼠标右键单击文本对象，然后选择动画链接。此时出现动画链接对话框。
2. 在值显示区域中，选择字符串。此时出现输出 -> 字符串表达式对话框。



3. 在表达式框中，输入消息标记名或使用消息标记的表达式。例如：

```
"The Tank Level is:" + Text(TankLevel,"#")
```

4. 选择确定。

   注意：运行时期间，可通过使用指针和鼠标单击和拖拽的方式来调整字符串值输入链接字段的大小。

创建移动

通过使用位置链接，可以使对象在运行时移动。您可以根据模拟标记或表达式值的变化，让对象沿水平、垂直或同时沿两个方向移动。例如，随着贮料罐液位的升高或降低，指示器上下移动。

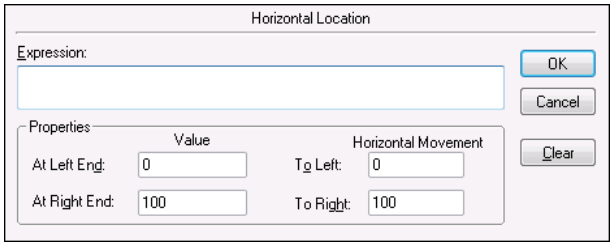
创建水平移动

1. 将对象放到屏幕上的开始位置。
2. 在绘图菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接对话框。
3. 在位置部分中，选择水平。

   此时出现水平位置对话框。



4. 在表达式框中，输入模拟标记名或等于模拟值的表达式。
5. 在属性部分中，配置对象移动的距离。执行以下操作：
   a. 在在左端框中，输入对象位于最左端时的模拟标记的值。
   b. 在在右端框中，输入对象位于最右端时的模拟标记的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

c. 在向左框中，输入对象应向起始位置左侧移动的像素数。

d. 在向右框中，输入对象应向起始位置右侧移动的像素数。

6. 选择确定。

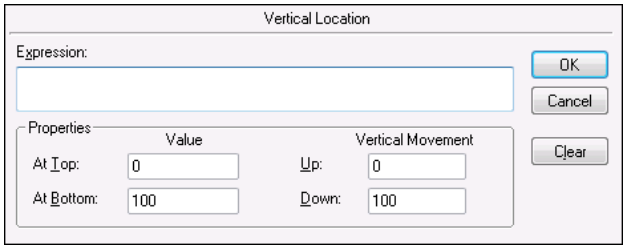
创建垂直移动

1. 将对象放到屏幕上的开始位置。
2. 在绘图菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接对话框。
3. 在位置部分中，选择垂直。

   此时出现垂直位置对话框。



4. 在表达式框中，输入模拟标记名或等于模拟值的表达式。
5. 在属性部分中，执行以下操作：

   a. 在在顶端框中，输入对象位于最顶端时的模拟标记的值。

   b. 在在底端框中，输入对象位于最底端时的模拟标记的值。

   c. 在向上框中，输入对象应从起始位置向上移动的像素数。

   d. 在向下框中，输入对象应从起始位置向下移动的像素数。
6. 选择确定。

创建旋转

通过使用方向链接，可以根据模拟标记值的变化，让对象绕中心点旋转。例如，随着压强的升高或降低，指针可以绕刻度盘旋转。

方向链接使用对象或符号的中心作为旋转的缺省中心。您可以偏移旋转的中心。

工业图形不支持方向链接。

提示：从对象的中心到旋转中心点绘制一个临时的长方形。现在，您可以从状态栏的 W、H 框中读取 X、Y 偏移大小（以像素计）。

创建方向链接

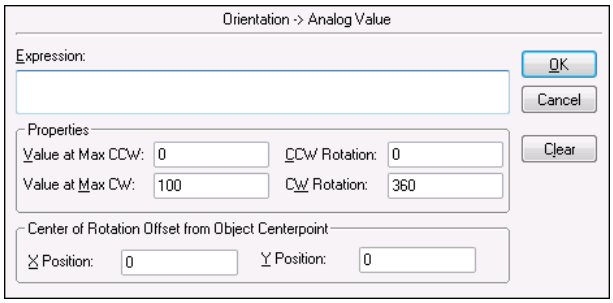
1. 在绘图菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接对话框。
2. 在其它部分中，选择方向。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现方向 -> 模拟值对话框。



3. 在表达式框中，输入模拟标记名或等于模拟值的表达式。
4. 在属性部分中，执行以下操作：
   a. 在最大逆时针旋转时的值框中，输入对象要旋转到最大逆时针位置的表达式值。
   b. 在最大顺时针旋转时的值框中，输入对象要旋转到最大顺时针位置的表达式值。
   c. 在逆时针旋转框中，输入到达最大逆时针旋转时的值时，对象逆时针旋转的度数。
   d. 在顺时针旋转框中，输入到达最大顺时针旋转时的值时，对象顺时针旋转的度数。
5. 在旋转中心相对于对象中点的偏移量部分中，执行以下操作：
   a. 在 **X** 位置框中，输入旋转中心点的水平偏移量。输入距对象中心点的偏移量（以像素计）。
   b. 在 **Y** 位置框中，输入旋转中心点的垂直偏移量。输入距对象中心点的偏移量（以像素计）。
6. 选择确定。

设置大小动画效果

通过使用对象大小链接，可以根据模拟标记或表达式的值来改变对象的高度与/或宽度。

例如，压力指示器的值随着压力的升高而增大，或者传送带上的对象通过变得更大而呈现为向观察者移动。

通过使用动画的参考位置，对象大小链接不仅可以控制对象的大小，而且可以控制对象大小变化的方向。

创建对象大小高度链接

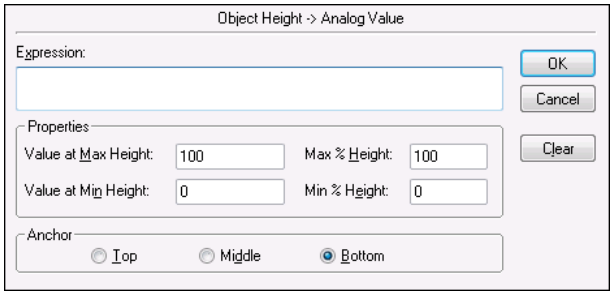
1. 在绘图菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接对话框。
2. 在对象大小部分中，选择高度。

   此时出现对象高度 -> 模拟值对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**表达式**框中，**输**入模**拟标记**名或等于模**拟值**的表达式。
4. 在**属性**部分中，**执**行以下操作：
   a. 在**最大高度值**框中，**输**入**对**象达到最大高度**时**的**标记**或表达式的**值**。
   b. 在**最小高度值**框中，**输**入**对**象达到最小高度**时**的**标记**或表达式的**值**。
   c. 在**最大百分比高度**框中，**输**入在**标记**名或表达式达到**最大高度值**框中**设**置的**值时**，**对**象占原始高度的百分比。百分比数字以**对**象的**绘**制大小的百分比来表示。**绘**制大小始**终**是 100%。
   d. 在**最小百分比高度**框中，**输**入在**标记**名或表达式达到**最小高度值**框中**设**置的**值时**，**对**象占原始高度的百分比。百分比数字以**对**象的**绘**制大小的百分比来表示。**绘**制大小始**终**是 100%。
5. **选择**对象**扩**展的**参考位置**。
   - **选择顶端**以使**对**象从**顶**端往下**扩**展。
   - **选择中间**以使**对**象从中心点向两**边扩**展。
   - **选择底端**以使**对**象从底端往上**扩**展。
6. **选择确定**。

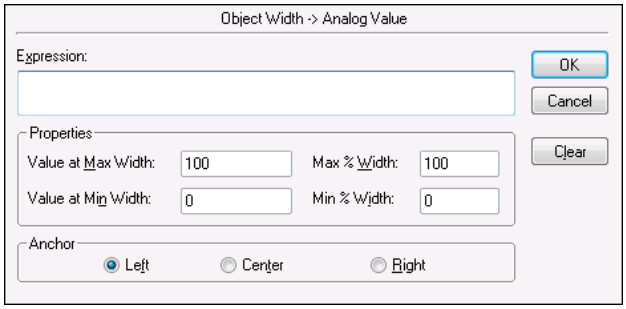
创建对象大小宽度链接

1. 在**绘图**菜单上的**模式组**中，**选择设置动画效果**。

   或者，右**键单击该对**象，然后**选择动画链接**。

   此**时**出**现动画链接对话**框。
2. 在**对象大小**部分中，**选择宽度**。

   此**时**出**现对象宽度 -> 模拟值对话**框。



3. 在**表达式**框中，**输**入模**拟标记**名或等于模**拟值**的表达式。
4. 在**属性**部分中，**执**行以下操作：
   a. 在**最大宽度值**框中，**输**入**对**象达到最大宽度**时**的**标记**或表达式的**值**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

b. 在最小宽度值框中，输入对象达到最小宽度时的标记或表达式的值。

c. 在最大百分比宽度框中，输入在标记名或表达式达到最大宽度值框中设置的值时，对象占原始宽度的百分比。百分比数字以对象的绘制大小的百分比来表示。绘制大小始终是 100%。

d. 在最小百分比宽度框中，输入在标记名或表达式达到最小宽度值框中设置的值时，对象占原始宽度的百分比。百分比数字以对象的绘制大小的百分比来表示。绘制大小始终是 100%。

5. 选择对象宽度扩展的参考位置。
   - 选择左端以使对象从左端扩展。
   - 选择中间以使对象从中心点向两边扩展。
   - 选择右端以使对象从右端扩展。
6. 选择确定。

设置颜色动画效果

通过使用颜色链接，可以给任何对象设置颜色变化动画效果。变化可以基于模拟或离散标记的值、模拟或离散表达式的值，或是离散或模拟报警状态。

您可以使用三种颜色链接来设置对象的动画效果。

- 线条颜色
- 填充颜色
- 文本颜色

对于这三种颜色链接而言，有四种类型的表达式可以控制颜色变化。

表达式类型	更改颜色的根据
离散	离散标记或表达式的值。
模拟	模拟标记或表达式的值。您可以定义十种颜色来表示不同的值。
离散报警	标记、“报警组”或“组变量”的报警状态。
模拟报警	模拟标记、“报警组”或“组变量”的报警状态。您可以定义五种颜色来表示五种报警条件。

警告！使用模拟报警链接时，如果该链接是来自 InTouch 7.11 以前的版本所创建且未转换的应用程序的远程标记，则对象不会进入报警状态。

所有离散颜色链接都是按照相同的方法创建的。以下操作程序介绍如何创建填充颜色链接。

创建离散填充颜色链接

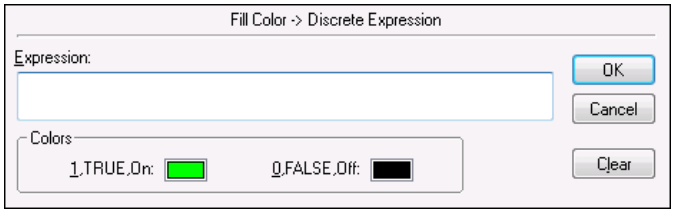
1. 在绘图菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接对话框。
2. 在填充颜色部分中，选择离散。

   此时出现填充颜色 -> 离散表达式对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在表达式框中，输入离散标记名或等于 true 或 false 值的离散表达式。

   离散表达式可以包含模拟标记。例如，TankLevel > 75。在本例中，"TankLevel" 变量的值大于或等于 "75" 时，对象的填充颜色会改变。

4. 在颜色部分中，选择每个颜色框以打开调色板。选择要给每个状态使用的颜色。
5. 选择确定。

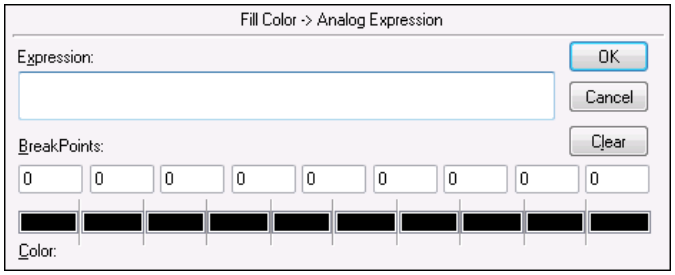
创建模拟表达式颜色链接

1. 使用鼠标右键单击对象，然后选择动画链接。

   此时出现动画链接对话框。

2. 在填充颜色部分中，选择模拟。

   此时出现填充颜色 -> 模拟表达式对话框。



3. 在表达式框中，输入模拟标记名或等于模拟值的表达式。
4. 在断点部分中，执行以下操作：
   - 指定对象将在该处改变颜色的断点值。

   提示：您不必使用十个不同的颜色。例如，如果只希望对象更改颜色三次，便只需输入三个值，然后给其余的值使用相同的颜色。如果需要更多样化的范围，请回顾工业图形的模拟填充功能。

   - 在颜色部分中，选择用于每个断点的颜色。

5. 选择确定。

创建离散报警状态颜色链接

1. 在绘图菜单上的模式组中，选择设置动画效果。

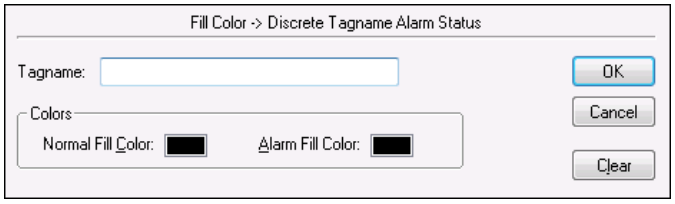
   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接对话框。

2. 在填充颜色部分中，选择离散报警。

   此时出现填充颜色 -> 离散标记名报警状态对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**标记**名框中，输入要与对象关联的离散标记名。
4. 在**颜**色部分中，给每种报警状态选择一种颜色。
5. 选择确定。

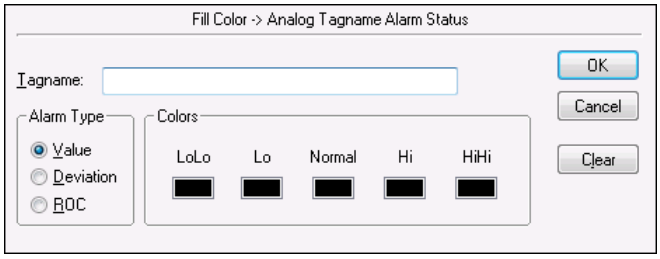
创建模拟报警状态颜色链接

1. 在**绘图**菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接对话框。
2. 在填充颜色部分中，选择模拟报警。

   此时出现填充颜色 -> 模拟标记名报警状态对话框。



3. 在**标记**名框中，输入要与对象关联的模拟标记名。
4. 在**报警类型**部分中，从三种报警类型之中选择要与该对象关联的类型。

报警类型	最多可用
值	五种颜色显示值报警的状态。
偏差	三种颜色显示偏差报警的状态。
ROC（变化率）	两种颜色显示变化率报警的状态。

5. 在**颜**色部分中，给每种报警状态选择一种颜色。
6. 选择确定。

设置填充级别动画效果

您可以使用填充百分比链接来改变对象的填充级别。填充百分比基于模拟标记或表达式的值。您可以创建水平填充、垂直填充或同时创建这两种填充。

例如，您可以使用垂直填充链接显示罐中的液位，或使用水平填充链接显示过程的进度。

水平填充百分比与垂直填充百分比链接的创建方法相同。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建填充百分比链接

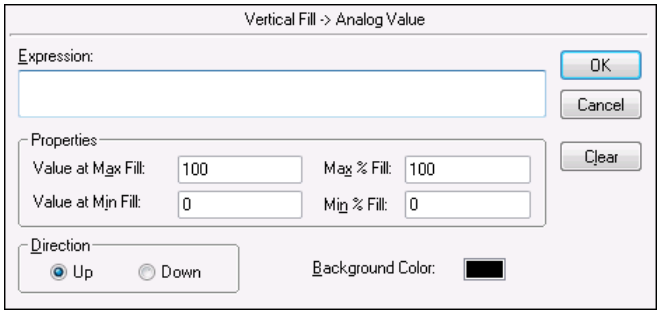
1. 在**绘图**菜单上的**模式组**中，选择**设置动画效果**。

   或者，右键单击该对象，然后选择**动画链接**。

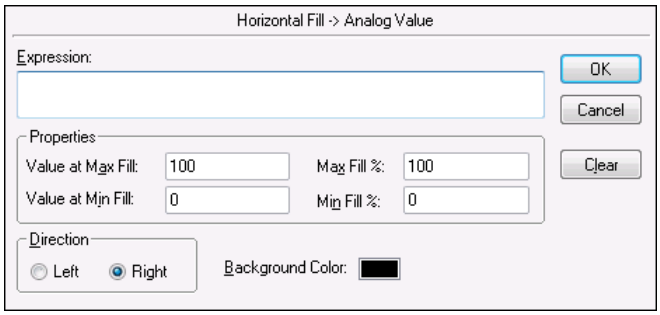
   此时出现**动画链接**对话框。

2. 在**填充百分比**部分中，执行以下操作之一：

   - 选择**垂直**，此时出现**垂直填充 -> 模拟量**对话框。



   - 选择**水平**，此时出现**水平填充 -> 模拟量**对话框。



3. 在**表达式**框中，输入模拟标记名或等于模拟值的表达式。
4. 在**属性**部分中，执行以下操作：
   a. 在**最大填充值**框中，输入将使对象填充到最大级别的表达式的值。
   b. 在**最小填充值**框中，输入将使对象填充到最小级别的表达式的值。
   c. 在**最大填充百分比**框中，输入在表达式达到**最大填充值**框中设置的级别时，对象填充的百分比 (0-100)。
   d. 在**最小填充百分比**框中，输入在表达式达到**最小填充值**框中设置的级别时，对象填充的百分比 (0-100)。
5. 在**方向**部分中，选择填充的方向。
6. 在**背景颜色**框中，选择对象未填充部分的颜色。
   - 实际填充颜色是绘制对象时选择的颜色。
   - 如果将垂直填充百分比与水平填充百分比链接同时链接到相同的对象，则最后选择的颜色是背景颜色。
7. 选择**确定**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使对象闪烁

通过使用闪烁链接，可以创建根据标记的值进行闪烁的动画对象。例如，您可以创建一个对象，在某种设备打开时，或达到报警设定点时，它呈红色闪烁。

闪烁动画同步功能可让所有打开窗口上的所有活动闪烁动画根据它们共同的闪烁速度（慢速、中速或快速）一齐闪烁。

提示：离散表达式可以包含模拟标记名。例如，TankLevel > 75。在本例中，当 TankLevel 标记的值大于 75 时，对象开始闪烁。

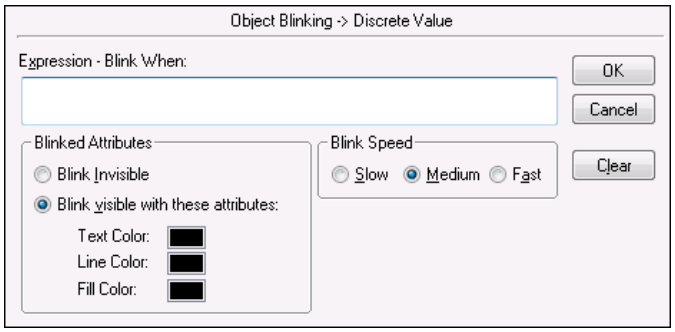
创建闪烁链接

1. 选择要应用动画的对象。
2. 在**绘图**菜单上的模式组中，选择**设置动画效果**。

   或者，右键单击该对象，然后选择**动画链接**。

   此时出现**动画链接**窗口。
3. 在其它部分中，选择**闪烁**。

   此时出现**对象闪烁信号 -> 离散值**对话框。



4. 在**表达式 - 闪烁时机**框中，输入离散标记名或等于离散值的表达式。
5. 在**闪烁属性**部分中，执行以下操作：
   - 选择**闪烁不可见**，将对象的闪烁方式设置为在窗口中消失然后重新出现。
   - 选择**带这些属性的闪烁可见**，将对象设置为保持可见，而在激活时改变颜色。
   - 选择**文本颜色**、**线条颜色**或**填充颜色**框，以选择用于对象的这些部分的颜色。此时出现调色板。

   注意：如果选择的填充闪烁颜色与对象的填充颜色相同，则对象看起来不会闪烁。
6. 在**闪烁速度**部分中，设置对象的闪烁速度。选择**慢速**、**中速**或**快速**中的一个。
7. 选择**确定**。

设置 WindowMaker 的闪烁频率

1. 打开 WindowMaker。
2. 在**文件**菜单上，选择**配置**，然后选择 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。
3. 在首**选项**选项卡上的**闪烁频率（毫秒）**部分中，输入用于三种速度的毫秒数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：对这些设置所作的任何更改都是全局性的，因此会影响应用程序中所有的闪烁速度。

4. 选择确定。

启用可见性

您可以使用可见性链接创建根据不同标记的值来隐藏对象的链接。通过使用可见性链接，您可以：

- 对象朝错误的方向移动时，通过隐藏它们，可以创造出对象只朝一个方向移动的印象。
- 创建正在移动的对象已经停止的印象。
- 使对象（如报警或错误消息）仅在激活时变得可见。

提示：离散表达式可以包含模拟标记，例如 TankLevel = 75。在本例中，当标记 TankLevel 的值大于或等于 75 时，便可以在窗口中看到对象。

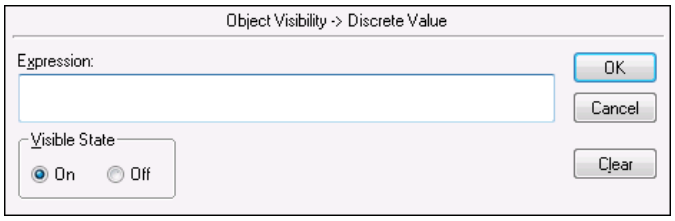
创建可见性链接

1. 选择要应用动画的对象。
2. 在**绘图**菜单上的模式组中，选择**设置动画效果**。

   或者，右键单击该对象，然后选择**动画链接**。

   此时出现**动画链接**窗口。
3. 在**其它**部分中，选择**可见性**。

   此时出现**对象可见性 -> 离散值**对话框。



4. 在**表达式**框中，输入离散标记名或等于离散值的表达式。
5. 选择对象的**可见状态**。如果选择**关闭**，则表达式的值为真时对象不可见。如果选择**打开**，则表达式的值为真时对象可见。
6. 选择确定。

使对象失效

您可以使用失效链接在应用程序上施加另一层安全机制。例如，您可以根据操作员访问级别或姓名来使触控对象失效。或者，如果无人登录，则可以禁用按钮，以防外人擅自操作。

“打开”的失效状态表示只要表达式为真，该对象或按钮的触动功能便会关闭且处于不活动状态。

提示：离散表达式可以包含模拟标记名。例如，TankLevel > 75。在本例中，当变量“TankLevel”的值大于或等于 75 时，对象失效。

创建失效链接

1. 选择要应用动画的对象。
2. 在**绘图**菜单上的模式组中，选择**设置动画效果**。

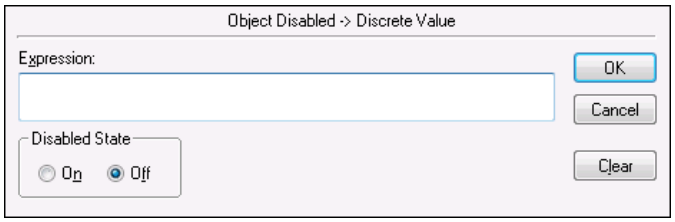
   或者，右键单击该对象，然后选择**动画链接**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现动画链接窗口。

3. 在其它部分中，选择禁用。

   此时出现对象失效信号 -> 离散值对话框。



4. 在表达式框中，输入离散标记名或等于离散值的表达式。
5. 在失效状态部分中，执行以下操作之一：
   - 选择打开，以便将失效状态设置为在离散标记或表达式为真期间对象无法激活。
   - 选择关闭以便移除失效状态，这样，在离散标记或表达式为真期间便允许对象运行。
6. 选择确定。

配置工具提示

通过使用工具提示链接，可以创建“工具提示”，向用户提供屏幕上对象的有关信息。在指针移到对象上时，工具提示出现；而指针移开时，工具提示则消失。工具提示出现的持续时间以及出现的位置都由操作系统决定。

您可以为工具提示设置表达式或静态文本。

- 创建静态工具提示，以便在工具提示每次出现时显示相同的消息。
- 创建表达式工具提示，以便每次出现工具提示时，对表达式进行求值并显示为工具提示文本。

例如，在下面的示例表达式中，文本显示为 msgTooltipTag01 消息标记的当前值。

```
msgTooltipTag01
```

在本例中，字符串之后跟有 iTemp 标记的当前值，结果作为工具提示文本出现：

```
"Current temp. is " + StringFromIntg (iTemp,10)
```

在 Windows 7 和更新版本的 Windows 中，可以更改工具提示窗口的宽度。您可以编辑 InTouch.INI 文件，添加工具提示宽度条目。

示例：

- Tooltipwidth=200
- 值 -1 (Tooltipwidth=-1) 将工具提示文本显示在一行上，没有换行符
- 如果不指定工具提示宽度，那么工具提示窗口的缺省宽度为 88。

创建工具提示链接

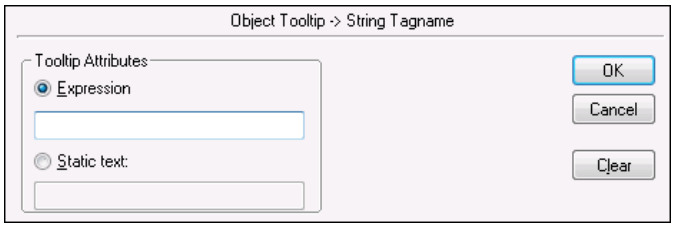
1. 选择要应用动画的对象。
2. 在绘图菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接窗口。
3. 在其它部分中，选择工具提示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现对象工具提示 -> 字符串标记名对话框。



4. 在工具提示属性部分中，选择表达式或静态文本。
   - 如果选择表达式，请输入等于消息值的表达式。这可以是一个简单的消息标记名，也可以是一个较复杂的表达式。
   - 如果选择静态文本，请输入用作工具提示文本的静态消息，最多 131 个字符。
5. 选择确定。

放置触控窗口

您可以使窗口在运行时出现在相对于触控对象的精确位置上。例如，操作员可以选择对象以查看状态、名称或其它与该对象链接的数据。在操作员通过单击或悬停鼠标选择对象时，可以使窗口出现在指定的位置。

通过使用带系统只读标记 $ObjHor 与 $ObjVer 的脚本函数 ShowAt() 或 ShowTopLeftAt()，可以相对于对象来放置窗口。您还可以在这些函数中使用固定的位置。

如果 Windows 的显示属性设置为 **Windows XP** 主题，则此功能在特定的情况下不稳定。

语法与以下类似：

```
ShowTopLeftAt (windowname, $ObjHor, $ObjVer);
```

其中

windowname：要打开的窗口的名称。

$ObjHor：所选对象中心的水平位置。

$ObjVer：所选对象中心的垂直位置。

新窗口的左上角出现在所选对象的中心。

类似的脚本函数将打开窗口，使其中心位于所选对象的中心位置。语法与以下类似：

```
ShowAt (windowname, $ObjHor, $ObjVer);
```

在所选对象的位置打开窗口

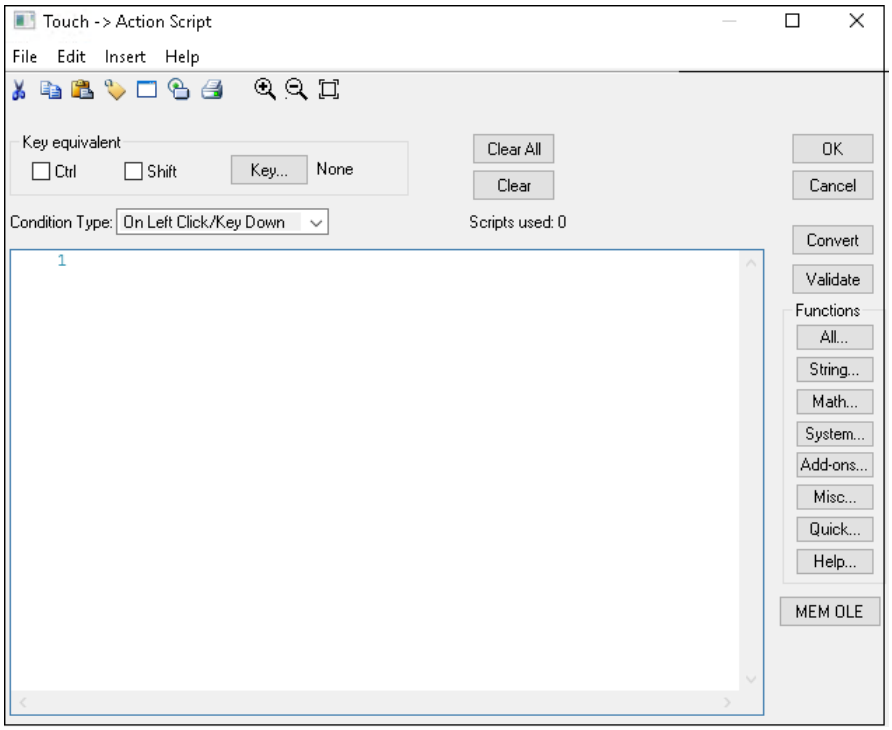
1. 设计、命名并创建要出现的窗口。
2. 在**绘图**菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接窗口。
3. 在触动按钮部分中，选择动作。

   此时出现触动 -> 动作脚本对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 根据之前定义的语法输入以下脚本之一：

```
ShowTopLeftAt (windowname, $ObjHor, $ObjVer);
```

或

```
ShowAt (windowname, $ObjHor, $ObjVer);
```

5. 在条件类型框中，单击鼠标动作以打开窗口。
6. 选择确定。

$ObjHor 系统标记

包含对象（拥有焦点）中心的水平像素位置。

类别

系统

用法

```
$ObjHor
```

数据类型

整型（只读）

另请参阅

$ObjVer

$ObjVer 系统标记

包含对象（拥有焦点）的中心的垂直像素位置。

类别

系统

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用法

```
$ObjVer
```

数据类型

整型（只读）

另请参阅

$ObjHor

数据输入动画

要创建可供操作员交互操作的对象，请使用触动链接。触动链接可供操作员在运行中应用程序中输入数据。例如，操作员可以使用键盘登录，打开或关闭阀门，输入新的报警设定点，或是开始或停止某个过程。

触控对象拥有焦点时，它周围出现一个框。用户将光标移到该对象上，或是按 TAB 或方向键将焦点移到该对象时，对象便获得焦点。

如果期望用户使用 TAB 键来选择触控对象，请尝试将它们按水平模式排列。按 Tab 键从窗口顶端开始，从左到右，然后向下，将焦点从一个对象移到另一个对象。

如果触动链接对象包含顶端重叠放置的几个文本对象，则只显示顶端的文本对象。

要激活触控对象，操作员可以单击它、按指定的等价键、在对象框架出现时按 **Enter** 键，或是实际触摸它（如果使用触摸屏显示设备）。

您可以创建九种类型的用户输入触动链接：

触动链接	动作
用户输入	• 离散 • 模拟 • 字符串型
游标	• 垂直 • 水平
按钮	• 离散值 • 动作 • 显示窗口 • 隐藏窗口

文本字段用于输入时，文本会在按键时出现在屏幕上。

如果不希望在输入时出现文本，请选择该链接的配置面板中的**仅输入选项**。

启用离散输入

您可以设计操作员输入触动链接，将离散标记的值从一个状态更改为另一个状态。例如，要打开或关闭某个泵，请使用离散链接。

创建离散输入链接

1. 选择要应用动画的对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

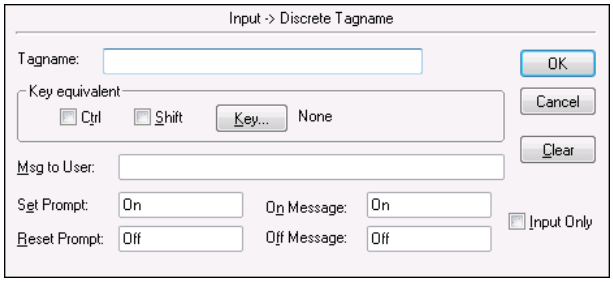
2. 在**绘图**菜单上的**模式组**中，选择**设置动画效果**。

   或者，右键单击该对象，然后选择**动画链接**。

   此时出现**动画链接**窗口。

3. 在**触动链接**区域中的**用户输入**下，选择**离散**。

   此时出现**输入 -> 离散标记名**对话框。



4. 在**标记名**框中，输入离散标记的名称。
5. 作为可选项，在**等价键**区域中，指定链接的等价键。如需详细信息，请参阅创建键盘快捷键。
6. 配置离散输入的详细信息。执行以下操作：
   - 在**给用户的消息**框中，输入要出现在**输入对话**框中的消息。
   - 在**置位提示**和**复位提示**框中，输入要出现在按钮上的消息，操作员可以选择这些按钮来打开或关闭离散值。
   - 在**打开消息**与**关闭消息**框中，输入在值打开或关闭时要出现在与该对象关联的文本字段中的消息。
7. 选中**仅输入复选**框，以避免输入内容出现在与对象相关联的文本字段中。
8. 选择**确定**。

启用模拟输入

通过使用模拟输入链接，您可以创建操作员用于输入实数值（如报警设定点或传送带速度）的对象。如果在 WindowViewer 属性下选择**仅允许十进制表示法输入选项**，则无法在输入字段中输入 1E2 等非数字值。如需有关运行时设置的详细信息，请参阅《*AVEVA™ InTouch HMI 为 InTouch HMI 组件创建标准*》文档中的配置常规 WindowViewer 属性。

创建模拟输入链接

1. 选择要应用动画的对象。
2. 在**绘图**菜单上的**模式组**中，选择**设置动画效果**。

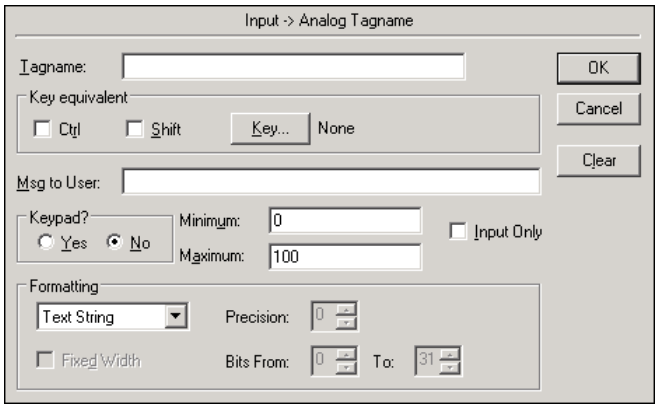
   或者，右键单击该对象，然后选择**动画链接**。

   此时出现**动画链接**窗口。

3. 在**触动链接**区域中的**用户输入**下，选择**模拟**。

   此时出现**输入 -> 模拟标记名**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**标记**名框中，输入模拟标记的名称。
5. 在**等价键**区域中，指定链接的等价键。如需详细信息，请参阅创建键盘快捷键。
6. 配置模拟输入的详细信息。执行以下操作：
   - 如果希望在屏幕上显示数字小键盘来输入值，请在**数字小键盘？**区域中选择是。在**给用户的消息**框中，输入希望出现在数字小键盘中的提示消息。
   - 在**最小值**与**最大值**框中，输入标记的最小值与最大值。这些设置限制运行时针对动画的用户输入。支持常数模拟值（整型或实型）以及引用。缺省值分别为 1 与 100。最大值必须大于最小值。

     如果您使用引用，请参阅“将引用用于模拟输入动画的最小值和最大值限制”一节以获取详细信息。
7. 选中**仅输入复选**框，以避免输入内容出现在与对象相关联的文本字段中。
8. 在**设置格式**区域中，在列表中选择要为其配置运行时高级格式的每个数据类型。**固定宽度复选**框、**精度**框以及**开始位**和**结束位**框会基于选择的数据类型成为可用状态。如需有关配置这些选项的信息，请参阅“文本的高级格式”一节。
9. 选择确定。

将引用用于模拟输入动画的最小值和最大值限制

对于模拟用户输入动画的最小值和最大值限制，支持常数模拟值和引用。可使用**输入 -> 模拟标记**名对话框配置这些限制。

引用可以是远程模拟标记或本地模拟标记（如内存标记或 I/O 标记）。

- 在将 InTouch I/O 标记用于最小值和最大值时，系统只会对 InTouch IO 标记进行赋值。如果 I/O 标记是指已赋值为字符串值的远程标记引用，那么 I/O 标记将会赋值为 0。
- 如果将远程标记引用用于最小值和最大值，那么引用会被内部赋值为字符串。如果使用无法转换成模拟值的字符串值对远程标记进行赋值，那么会在记录器中出现一条警告消息。**最小值**和**最大值**框会根据配置用户输入所用的引用的数据类型显示相应值。
- 如果**最小值**或**最大值**字段中的用户输入包含字母“e”，那么该输入将为字符串或指数值。如果用户输入包含后跟字母“e”加数字（格式为 ±N e ±N，其中 N 为数字）的数字，那么用户输入将视为指数值。如果用户输入包含字母 "e" 和其他任意字母（甚至可以仍是“e”），那么用户输入将视为引用名。

如果在运行时引用的质量下降，则会发生以下情况。

对于**最小值**框中的引用：

- 如果动画的主引用是 InTouch 标记，那么系统会从标记数据库中检索该标记配置的最小值，然后将其用作最小值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 如果动画的主引用是外部引用，则系统会使用数据类型的最小值。如果配置的数据点是字符串，则上一个已知值将用于确定数据类型。
  - 如果上一个已知值包含小数点，则数据类型为实型。
  - 如果上一个已知值不含小数点，则数据类型为整型。
  - 如果没有上一个已知值，则系统会使用整型。

对于最大值框中的引用：

- 如果动画的主引用是 InTouch 标记，那么系统会从标记数据库中检索该标记配置的最大值，然后将其用作最大值。
- 如果动画的主引用是外部引用，则系统会使用数据类型的最大值。如果配置的数据点是字符串，则上一个已知值将用于确定数据类型。
  - 如果上一个已知值包含小数点，则数据类型为实型。
  - 如果上一个已知值不含小数点，则数据类型为整型。
  - 如果没有上一个已知值，则系统会使用整型。

启用字符串输入

通过使用字符串输入链接，您可以创建供用户输入字符串（如批次名、操作员 ID 或密码）的对象。

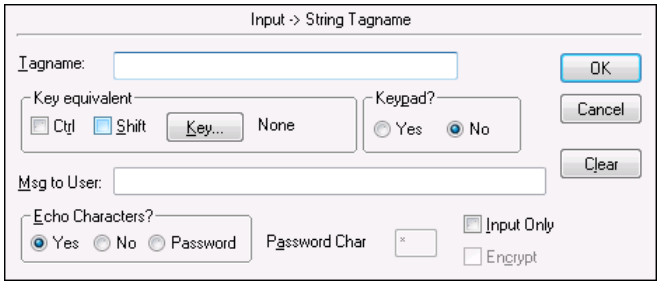
创建字符串输入链接

1. 选择要应用动画的对象。
2. 在**绘图**菜单上的模式组中，选择**设置动画效果**。

   或者，右键单击该对象，然后选择**动画链接**。

   此时出现**动画链接**窗口。
3. 在**触动链接**区域中的**用户输入**下，选择**字符串**。

   此时出现**输入 -> 字符串标记名**对话框。



4. 在**标记名**框中，输入消息标记的名称。
5. 作为可选项，配置等价键与/或键盘。
   - 在**等价键**区域中，指定链接的等价键。如需详细信息，请参阅创建键盘快捷键。
   - 如果希望显示键盘来输入新的值，请在**数字小键盘？**区域中选择**是**。在**给用户的消息**框中，输入要出现在键盘中的消息。
6. 在**回显字符？**区域中，选择在用户输入字符串时输入框中是否出现字符。
   - 选择**是**以便在输入框中显示输入的文本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 选择否以便不显示输入的文本。
- 选择密码以使用“掩码”字符代替输入的文本。在密码字符框中，输入掩码字符。选中加密复选框以加密口令。

  要点：密码加密仅在 InTouch HMI 环境中有效。如果希望将字符串传递给外部安全系统（例如操作系统或 SQL Server 数据库），请不要给字符串加密。外部安全系统无法读取加密的密码字符串，因此访问会失败。

7. 选中仅输入复选框，以避免输入内容出现在与对象相关联的文本字段中。
8. 选择确定。

启用游标

通过使用游标触动链接，可以创建供用户来回拖动的对象。用户移动对象时，它会改变与之链接的标记的值。您可以将对象链接到水平或垂直游标。

在将对象链接到游标上时，请设置参考位置（即对象上的一点，光标用它来锁定对象）。

您可以在一个对象上同时使用水平与垂直链接，这样这两个模拟标记的值便会同时改变。

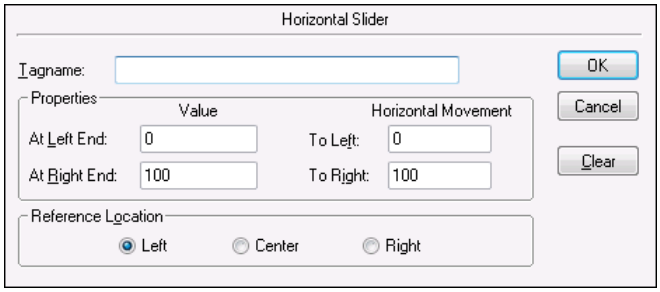
创建水平游标链接

1. 选择要应用动画的对象。
2. 在绘图菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接窗口。
3. 在游标区域中，选择水平。

   此时出现水平游标对话框。

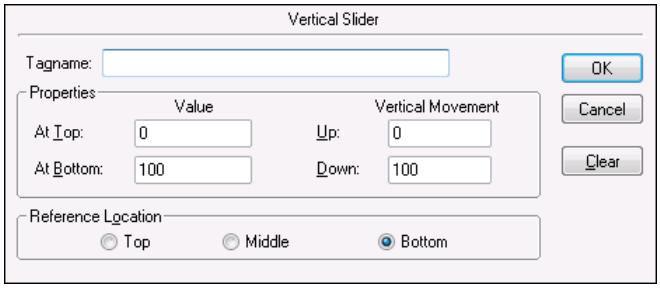


4. 在标记名框中，输入模拟标记的名称。
5. 在属性区域中，执行以下操作：
   a. 在在左端框中，输入游标位于最左端时的标记的值。
   b. 在在右端框中，输入游标位于最右端时的标记的值。
   c. 在向左框中，输入游标可向左移动的像素数。
   d. 在向右框中，输入游标可向右移动的像素数。
6. 在参考位置区域中，选择对象上光标将锁定的位置。
7. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建垂直游标链接

1. 选择要应用动画的对象。
2. 在**绘图**菜单上的模式组中，选择**设置动画效果**。
   或者，右键单击该对象，然后选择**动画链接**。
   此时出现**动画链接**窗口。
3. 在**游标**区域中，选择**垂直**。
   此时出现**垂直游标**对话框。



4. 在**标记名**框中，输入模拟标记的名称。
5. 在**属性**区域中，执行以下操作：
   a. 在**在顶端**框中，输入游标位于最顶端时的标记的值。
   b. 在**在底端**框中，输入游标位于最底端时的标记的值。
   c. 在**向上**框中，输入游标可向上移动的像素数。
   d. 在**向下**框中，输入游标可向下移动的像素数。
6. 在**参考位置**区域中，选择对象上光标将锁定的位置。
7. 选择**确定**。

启用按钮

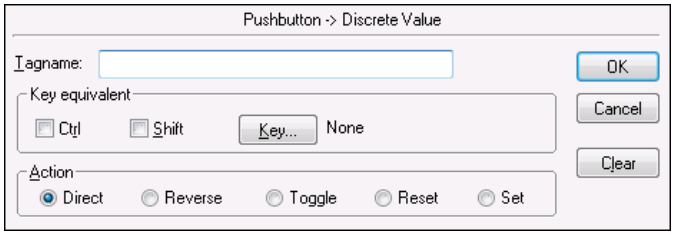
通过使用触动按钮链接，您可以创建启动动作脚本的触控对象。

动作脚本可以将标记设置为指定的值、启动与控制其它应用程序、执行函数等。

创建离散值触动链接

1. 选择要应用动画的对象。
2. 在**绘图**菜单上的模式组中，选择**设置动画效果**。
   或者，右键单击该对象，然后选择**动画链接**。
   此时出现**动画链接**窗口。
3. 在**触动按钮**区域中，选择**离散值**。
   此时出现**按钮 -> 离散值**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**标记名**框中，**输**入离散**标记**名。
5. **选择键**以指定**链**接的等价**键**。
6. 在**动作**区域中，**选择**以下**类**型之一：
   - **选择直接**以便只要按下或按住按**钮**，便将**值设**置为 1。**释**放按**钮时**，**该值**自**动复**位成 0。
   - **选择取反**以便在按下或按住按**钮时**，将**值设**置为 0。**释**放按**钮时**，**该值**自**动复**位成 1。
   - **选择切换**以反**转**离散**标记**的状**态**。
   - **选择复位**以便将**该值设**置为 0。
   - **选择置位**以便将**该值设**置为 1。
7. **选择确定**。

要**创建动**作脚本**链**接

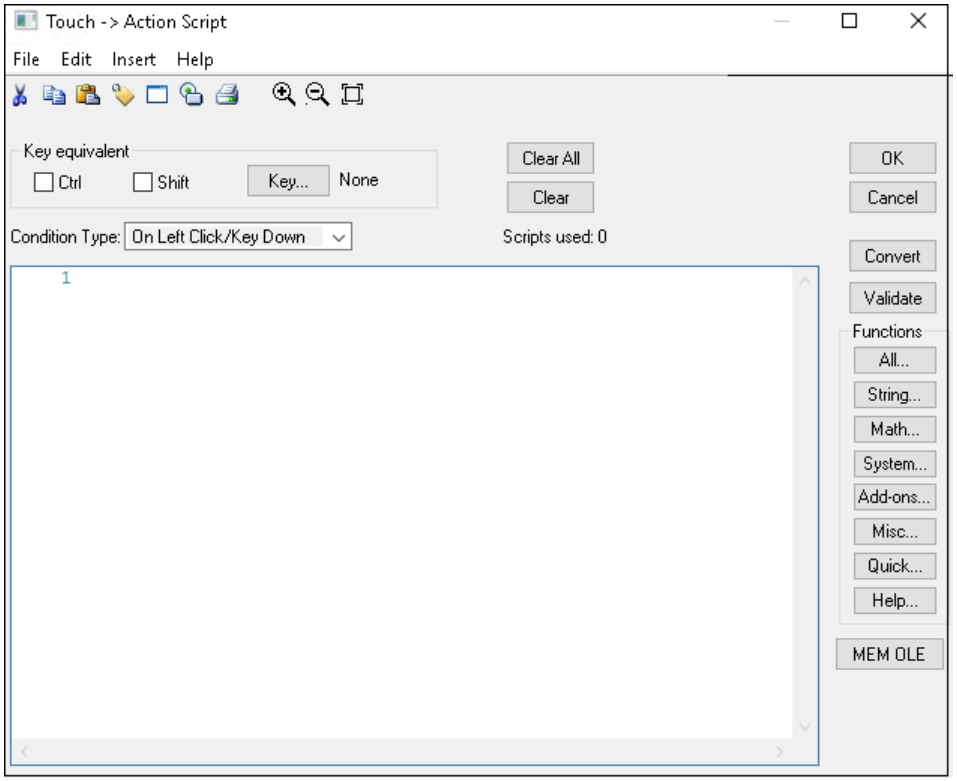
1. **选择**要**应**用**动**画的**对**象。
2. 在**绘图**菜**单**上的**模式组**中，**选择设置动**画效果。

   或者，右**键单击该对**象，然后**选择动画链**接。

   此**时**出**现动画链**接窗口。
3. 在**触动按钮**区域中，**选择动**作。

   此**时**出**现触动 -> 动**作脚本对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**条件类型**列表中，**选择一种**脚本**类**型。如需有**关动**作脚本**类**型的**详细**信息，**请**参**阅**脚本触发器。

   注意：如果将等价**键链**接指定**给动**作按**钮**和用于某个**键**脚本的同一个**键**，**则**所指定的等价**键链**接的**优**先**级**高于**该键**脚本。

5. 在**脚本编辑器**窗口中，**输**入对象激活**时**要**执**行的脚本。
6. **选择确定**。

打开与关闭窗口

通**过**使用**显**示窗口与**隐藏**窗口**链**接，可以**创**建打**开**与**关闭**其它 InTouch 窗口的触**动链**接。

您可以**给对**象**编**程以便一次打**开**多个窗口。不**过**如果将**链**接**编**程**为**打**开**多个窗口，**请**注意任何窗口交叉与各窗口**类**型。

如果其中一个打**开**的窗口是替**换类**型的窗口，并与另一个打**开**的窗口交叉，**则**另一个窗口会在打**开**之前**关闭**。

创建显示（或隐藏）窗口链接

1. **选择**要应用**动**画的对象。
2. 在**绘图**菜**单**上的**模式组**中，**选择设置动**画效果。

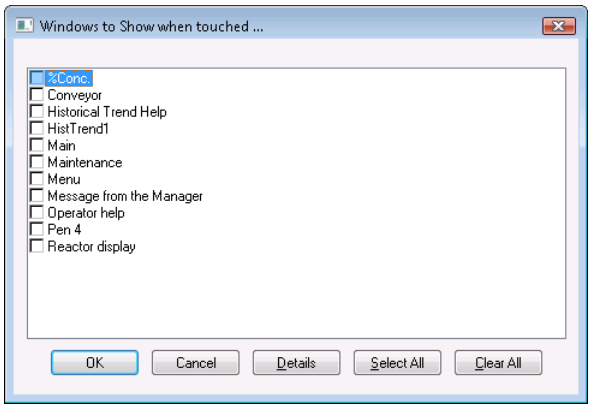
   或者，右**键单击该**对象，然后**选择动画链**接。

   此**时**出**现动画链**接窗口。

3. 在**触动按钮**区域中，**选择显**示窗口或**隐藏窗口**。

   此**时**出**现在触动时显**示的窗口或**在触动时隐藏**的窗口对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 选择要打开或隐藏的窗口。
5. 选择确定。

配置屏幕键盘

在计算机没有连接键盘的情况下，屏幕键盘可供操作员进行输入。

您可以指定三种屏幕键盘之一。

- 标准 InTouch 键盘或数字小键盘。这是缺省键盘。
- Windows 系统键盘。Windows 键盘是功能齐全的 QWERTY 型键盘，带有功能键、打印屏幕键、数字锁定键、方向键等。
- 可调整大小的键盘或数字小键盘。这种键盘的大小可以在运行时调整。

通过在脚本中使用 DialogStringEntry() 与 DialogValueEntry() 函数，您可以打开键盘。如需详细信息，请参阅 DialogStringEntry() 函数和 DialogValueEntry() 函数。

配置屏幕键盘的类型

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。
3. 在首选项选项卡上的键盘区域中，选择需要的键盘类型。
4. 如果选择可调整大小的键盘，请选择选项以选择键盘的字体、位置、尺寸等属性。
5. 选择确定。

显示屏幕键盘

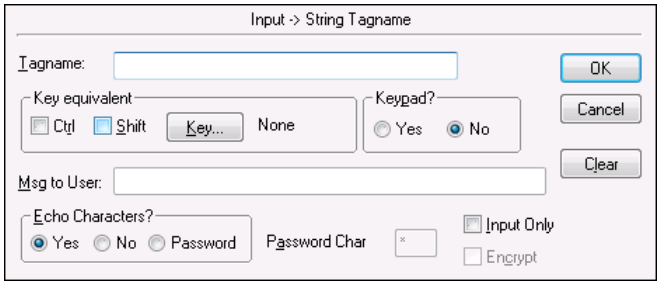
1. 配置屏幕键盘的类型。
2. 选择要应用动画的对象。
3. 在绘图菜单上的模式组中，选择设置动画效果。

   或者，右键单击该对象，然后选择动画链接。

   此时出现动画链接窗口。
4. 在触动链接区域中的用户输入下，选择字符串。

   此时出现输入 -> 字符串标记名对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 在数字小键盘？区域中，选择是。
6. 选择确定。

DialogStringEntry() 函数

在屏幕上显示字母数字键盘，供操作员在“标记名字典”中更改消息标记的当前字符串值。

类别

其它

语法

```
[Result=]DialogStringEntry(MessageTag_Text, UserPrompt_Text);
```

参数

MessageTag_Text
要修改的消息标记的名称。此值是一个字符串值。指定带英文引号括起的标记名，或使用不带英文引号的 .Name 点域。您还可以将消息标记用作指针。

UserPrompt_Text
要在键盘顶部显示的用户消息。

返回值

返回以下整数值之一：

0 = 已按过“取消”。

1 = 已按过“确定”。

-1 = 内部错误。

-2 = 无法启动。

-3 = 未定义标记名。

-4 = 标记名并非“消息”型。

-5 = 无法写入。

附注

此函数主要用在包含触摸屏的应用程序中。

示例

```
Errmsg=DialogStringEntry(MyMessageTag.Name, "Enter a new string...");
Errmsg=DialogStringEntry("MyMessageTag","Enter a new string...");
```

例如，以下脚本打开一个字母数字键盘，在键盘顶部显示消息 "Enter a new string..." 期间，允许修改 MyMessageTag：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
MessageTagX="MyMessageTag"; {将字符串 MyMessageTag（ 实际要修改的标记名）分配给内存消息标记名 MessageTagX}
MessageDisplay="Enter a new string..."; {将新消息字符串分配给内存消息标记名 MessageDisplay}
Errmsg=DialogStringEntry(MessageTagX, MessageDisplay); {不需要引号，因为 MessageTagX 被定义为消息标记名}
```

另请参阅

DialogValueEntry()

DialogValueEntry() 函数

在屏幕上显示数字小键盘，供用户更改离散、整型或实型标记名的当前值。

类别

其它

语法

```
[Result=] DialogValueEntry(ValueTag_Text, LowLimit, HighLimit, UserPrompt_Text);
```

参数

ValueTag_Text
要修改的离散、整型或实型标记的名称。此值是一个字符串值。指定带英文引号括起的标记名，或使用不带英文引号的 .Name 点域。您还可以将消息标记用作指针。

LowLimit
标记的最小值。（这应该是大于等于标记名的最小值、最小原始值或最小工程单位的定义，具体视适用情况而定）。

HighLimit
标记的最大值。（这应该是小于等于标记名的最大值、最大原始值或最大工程单位的定义，具体视适用情况而定）。

UserPrompt_Text
要在数字键盘顶部显示的用户消息。

返回值

返回以下整数值之一：

0 = 已按过“取消”。

1 = 已按过“确定”。

-1 = Highlimit<=Lowlimit.

-2 = 无法启动。

-3 = 未定义标记名。

-4 = 标记名不是“离散”、“整型”或“实型”。

-5 = 写入失败。

附注

此函数主要用在包含触摸屏的应用程序中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

```
Errmsg=DialogValueEntry(MyIntegerTag.Name, MyIntegerTag.MinEU, MyIntegerTag.MaxEU, "Enter
a new value...");
Errmsg=DialogValueEntry("MyIntegerTag", -100, 100, "Enter a new value...");
```

例如，以下脚本弹出一个数字键盘，在数字键盘顶部显示消息 "Enter a new value..." 期间，允许修改 MyIntegerTag，其最大与最小极限值分别是 -100 与 100：

```
TagnameX="MyIntegerTag"; {将字符串 MyIntegerTag（ 实际要修改的标记名）分配给内存消息标记名
TagnameX}
Min=-100; {将标记名允许的最小值指定给内存实型/整型标记名 Min}
Max=100; {将标记名允许的最小值指定给内存实型/整型标记名 Max}
MessageDisplay="Enter a new value..."; {将新消息字符串指定给内存消息标记名 MessageDisplay}
Errmsg=DialogValueEntry(TagnameX, Min, Max, MessageDisplay); {不需要引号，因为 TagnameX 被定义
为消息标记名。By assigning a Discrete, Integer or Real tagname to TagnameX, the function
will modify that assigned tagname}
```

另请参阅

DialogStringEntry()

常见动画任务

常见动画任务包括选择标记、创建键盘快捷键、更改标记名引用以及替换占位符标记。

选择标记或属性

使用**选择标记**对话框（也称为标记浏览器）来选择：

- 本地或远程 InTouch 应用程序中定义的标记。
- 使用 Galaxy 浏览器的 Application Server 对象属性。

双击任何需要输入标记名的文本框，以打开**选择标记**对话框。

选择 InTouch 标记

您可以选择在本地或远程 InTouch 应用程序中定义的标记。在支持“标记名字典”界面的任何标记源中，都可以创建对标记的引用。

例如，远程标记引用可供您的应用程序从“I/O 服务器”访问数据，而不必在本地“标记名字典”中创建标记。

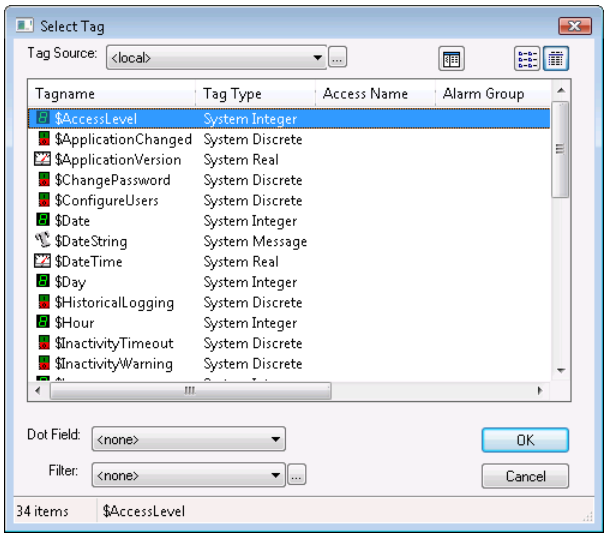
您可以为选择的每个 InTouch 标记设置点域。点域可以访问、监视及修改标记属性。如果不选择点域，则使用 .Value 点域。如需有关使用点域的详细信息，请参阅使用标记点域来查看或更改标记属性。

选择 InTouch 标记

1. 双击任何需要标记名才能输入的文本框。

   此时出现**选择标记**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**标记源**列表中，**选择**标记源的名称，或是**单击浏览**按**钮**以定义要使用的新标记源。

   如需有关定义标记源的详细信息，请参阅使用 I/O 进行数据访问。

3. 在**过滤器**列表中，**选择**过滤器以减少在窗口中显示的标记名数。要定义过滤器，请选择省略号按钮。如需详细信息，请参阅创建标记过滤器。

4. 在窗口中选择一个标记名。

   您可以更改**选择标记**对话框中标记名的显示方式。如需详细信息，请参阅更改“选择标记”对话框中的视图。

5. 在**点域**列表中，**选择**一个点域以将它附加到所选的标记名上。

   点域可以访问、监视及修改标记属性。如果不选择点域，则使用 .Value 点域。

6. 选择确定。

选择 Application Server 对象属性

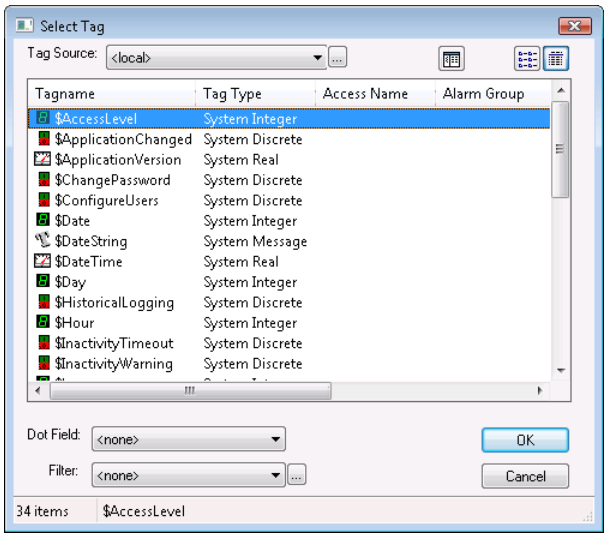
您可以选择与 Application Server 对象关联的属性。为此，必须将包含对象的 Galaxy 添加为 InTouch 应用程序的数据源。如需有关数据源的详细信息，请参阅使用 I/O 进行数据访问。

选择对象属性

1. 双击任何需要标记名才能输入的文本框。

   此时出现**选择标记**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**标记源**列表中，**选择**使用 Galaxy 的**标记**源的名称，或是**选择浏览**按**钮**以定**义**要使用的新**标记**源。

   此时出现 **Galaxy 浏览器**对话框。

3. 使用 **Galaxy 浏览器**来**查**找与**选择对**象属性。如需**详细**信息，**请**参**阅** Application Server 文档。

4. **选择确定**以**关闭 Galaxy 浏览器**。

   属性引用出**现**在要求使用**标记**名的文本框中。

   注意：要从 **Galaxy 浏览器**返回到**选择标记**对话框，**请选择 Galaxy 浏览器**右下角的**返回**按**钮**。

创建标记过滤器

如果“**标记**名字典”中包含有大量的**标记**，**则查**找**动**画的**标记**会非常麻**烦**。例如，您可能只希望**查**看指定**给**特定“**访问**名”或“**报警组**”的**标记**。您可以配置一个**过滤**器，**仅**使一部分**标记显**示在**选择标记**对话框中。

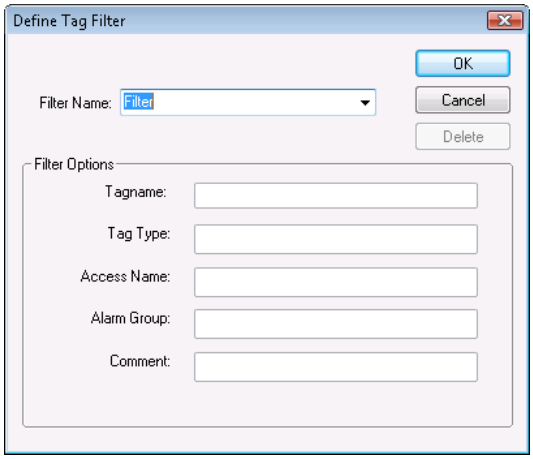
过滤器中可以使用以下通配符表达式。

- 多字符通配符是星号 (*)。例如，“Asyn*”将搜索以字符“Asyn”**开头**的所有**标记**名。
- **单**字符通配符是**问**号 (?)。例如，“Tag?”**过滤**器搜索以“Tag”**开头**的所有四个字符的**标记**名。“Tag*”**过滤**器搜索以“Tag”**开头**的所有**标记**名。
- **过滤**器可以使用有效的**标记**名字符与两个通配符的任意**组**合。有效的**标记**名字符是：A-Z、a-z、0-9、!、@、-、#、$、%、_ 及 &。

定义搜索过滤器

1. 在**选择**标记对话框中，**选择过滤器**列表旁边的省略号按**钮**。此**时**出**现定义标记过滤器**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**过滤器名**框中，输入过滤器名。
3. 在**过滤器选项**区域中，配置过滤器准则。执行以下任意操作：
   - 在**标记名**框中，输入标记名。
   - 在**标记类型**框中，输入标记类型。
   - 在**访问名**框中，输入本地访问名。
   - 在**报警组**框中，输入报警组名。
   - 在**注释**框中，输入注释表达式。
4. 选择确定。此时**过滤器名**出现在**选择标记**对话框的**过滤器**列表中。您可以选择过滤器以显示符合该过滤器准则的标记。

删除搜索过滤器

1. 在**过滤器名**框中，选择过滤器。
2. 选择**删除**。

更改“选择标记”对话框中的视图

选择标记对话框有三种不同的视图：列表视图、详细视图及目录树视图。

要查看	选择	描述
列表视图	列表视图按钮	根据每个标记名的类型，不同的小图标会出现在标记名的旁边。
详细信息视图	详细信息视图按钮	您会看到相同的图标与标记名，还有“标记类型”、“访问名”、“报警组”以及“注释”。通过选择列的标题给列表排序。
目录树视图	目录树视图图标	目录树视图以两种视图显示标记名。 您可以访问任何 SuperTag 模板中的成员标记名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建键盘快捷键

通过使用等价键链接，可以指定使用键盘上某个特定的键来激活某些动画链接。仅当包含此链接的对象可见或被选定时，等价键才可操作。如果该对象包含可见性或失效链接，则在该对象失效时，等价键不会激活。

您可以在多个窗口中定义相同的键。不过，只有最近打开的窗口中定义的键才会是活动的。在窗口重叠的情况下，顶部窗口的键才是活动的。

注意：如果将活动窗口中的任何对象或动作按钮指定给用于键动作脚本的同一个键，则活动窗口中该键的等价键链接的执行优先级高于键动作脚本。

支持等价键的动画链接出现在链接对话框中的等价键区域。

键链接只支持功能键 1-16。如果使用包含 16 个以上功能键的自定义键盘，则需要从厂商处获取设备驱动程序，才能访问系统上的扩展功能键。

将键指定给链接

1. 根据正在配置的链接类型，打开相应的动画链接对话框。
2. 如果希望操作员在按下等价键的同时按住 Ctrl 与 Shift 之一或同时按住这两者，请选择 **Ctrl** 与/或 **Shift**。
3. 选择 **键**。此时出现**选择键**对话框。
4. 选择要指定给链接的键。

   此时再次出现**动画链接**对话框，并在**键**按钮旁边显示所选键的名称。
5. 选择确定。

更改标记名引用

给对象创建副本时，所得副本是原件的完全拷贝，包括链接、动画、脚本，等等。不过，如果要在副本对象上使用不同的标记，则必须替换标记。

您可以选择与替换任何对象的标记，也可以选择多个对象并同时替换它们的标记。

如果系统许可证支持有限数目的标记名，则将本地标记转换为远程标记引用，以减少本地“标记名字典”中定义的标记数。

替换标记对全部标记与引用都有效。

用一个标记替换另一个标记

1. 选择与要更改为另一个标记的标记关联的对象。
2. 在**动画**菜单上的**替换**组中，选择**标记**。

   此时出现**替换标记名**对话框。
3. 在**新名**框中，输入新的标记名。
   - 如果双击**新名**框中的标记，则在“标记名字典”中出现它的定义。
   - 如果清除该标记名，然后双击空白框，则出现**选择标记**对话框。
4. 选择确定。

   与该对象关联的标记会自动改变。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

转换占位符标记

如果从当前应用程序导入或导出一个窗口或 QuickScript 时，所有与该窗口或 QuickScript 关联的标记都会随窗口一起转移。

本地标记不会自动添加到应用程序数据库中。相反，它们会自动标为占位符标记。

远程标记引用不会受到影响，它们不会标为占位符标记。

您必须将占位符标记转换为现有的本地标记，在新的应用程序标记名字典中定义它们，或使它们成为远程标记引用。

请注意标记之前的占位符：?d:、?i:、?m: 及 ?r:。它们指明了标记原先定义的类型。

占位符符号	标记名类型
d	离散
i	整型
m	消息
r	实型

注意：远程标记引用不会显示为占位符，而是显示为远程标记引用，例如：PLC2:Temperature。

在替换标记名对话框中，您可以使用多种方法将占位符标记转换为本地标记。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序运行时》文档中的导出与导入 InTouch 组件。

提示：如果对标记名进行了手工转换，不再需要原始的“标记名字典”中定义的原始标记，则可以更新标记名使用计数，然后删除未使用的标记。

通过从另一个应用程序导入窗口或 QuickScript，并将与动画链接或 QuickScript 关联的所有标记名转换为远程标记名引用，则不必在本地“标记名字典”定义单个标记名，即可创建一个应用程序，立刻接收来自数以百计的远程标记名的数据。

文本的高级格式

您可以使用 InTouch WindowMaker 配置供 InTouch WindowViewer 在运行时使用的智能或高级格式选项。只有当您在模拟值显示或模拟用户输入动画上使用高级格式功能时，这些选项才适用。如果您未配置高级格式选项，则 InTouch WindowViewer 会保留现有值。

高级格式选项会出现在**输入 -> 模拟标记名对话**框和**输出 -> 模拟表达式对话**框中。

数值显示的高级格式为您提供了一种让您根据数据类型和数据质量设置数值数据的格式的本机方式。通过高级格式，您可以在 InTouch WindowViewer 中在运行时期间设置模拟值的格式。

您可以以单个动画为基础使用文本字符串或启用高级格式选项。当您从较早版本的 InTouch 中迁移动画，或首次创建动画时，缺省设置为文本字符串。

高级格式配置选项有：

- 文本字符串，它使用现有的 InTouch 格式。
- 实型，它采用实数值的格式，以运行时值的大小为依据。全局小数精度配置用于确定小数位数。
- 固定小数点，它的格式取决于值的类型：
  - 实数值：该值会通过用户使用精度控制配置的指定小数位数来设置格式。
  - 整数值：该值的格式为不带小数点的整数，它会在右侧缩进以针对本应存在的小数位进行调整。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 离散值：该值的格式为不带小数点的 0 或 1，它会在右侧缩进以针对本应存在的小数位进行调整。
- 整数，它的格式始终为不带小数点的整数。
- 指数，它的格式始终为考虑到固定小数点的指数。
- 十六进制，它会以基于已配置布尔范围的十六进制格式显示。
- 二进制，它的格式始终为考虑到已配置布尔范围的二进制字符串。

在固定宽度框中，您可以增加在设计时指定的文本大小。如果已针对格式式样选择文本字符串格式，则该选项不可用。

在设计时，如果您已在动画配置中选择了固定宽度复选框，那么在运行时显示的值就不会超过文本字段的长度。如果该值超过文本字段的长度，那么全局配置的“对于固定字段宽度过大的字符”就会采用文本字段的长度。带有固定宽度的格式会阻止文本元素在运行时为了适应值的大小而水平展开。

如果您在设计时未对文本字段指定任何文本，那么系统会忽略固定宽度设置并显示完整的值。例如，如果您在设计时输入了 ""（无文本），那么对于文本字段而言，它的宽度等于 0。如果已启用固定宽度选项，那么将无法显示任何文本。在这种情况下，运行时固定字段宽度的缺省设置为“无”。

如果您选择清除，那么高级格式选项会重置为以下状态：

- 已对格式选项选择文本字符串。
- 由于在已选择文本字符串格式的情况下无法使用固定字段宽度选项，因此系统会在清除该选项后将其禁用。

缺省条件下，系统会设置以下值：

- “定点十进制”精度控制设置为 0。
- “指数”精度控制设置为 0。
- “十六进制”位范围控制设置为 0 至 31。
- “二进制”位范围控制设置为 0 至 31。

格式控件：

- “精度”控件可接受整数值，且仅在您已选择 固定小数点或指数选项时启用。标记点或 **Exponential** 选项。该控件的有效整数值范围为 0 至 8，其中 0 是缺省值。位范围包含两个调节控件，这两个控件都会将整数作为输入。只有当您已选择十六进制或二进制选项时，这些控件才可用。对于这两个控件，您需要指定从 0 到 31 的位值。您可以按正向或反向的顺序指定位范围。您还可以指定单个位，并对范围使用相同的位规格。位规格的缺省数值范围为 0 至 31。该范围是基于零的，以便与 InTouch 中已指定位的其他数位保持一致。

对于包含已配置模拟用户输入动画或模拟值显示动画的 3.5 版之前的 InTouch 版本，系统会将迁移动画的格式设置为文本字符串且关闭高级格式。

对于较新的版本，高级格式选项也会向前迁移。

将文本对象格式化为数字

文本对象可用于显示静态或动态数字值。通过将“值显示 - 模拟”动画链接与文本对象关联，可以显示模拟标记的整数值或实数值。

要指定通过设置文本对象的动画效果来显示的模拟标记值的格式，需使用以下字符：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符	描述
#	强制模拟值显示为整数。
0	表示在实数的每个指定位置有一位数。对数字的整数部分强制添加前导零。
、	在实数的指定位置插入逗号。
。	在实数的指定位置放置小数点。

其它文本格式属性（如字体、大小和颜色）应用于字符串中显示的数字值。

将文本对象格式化为数字的示例

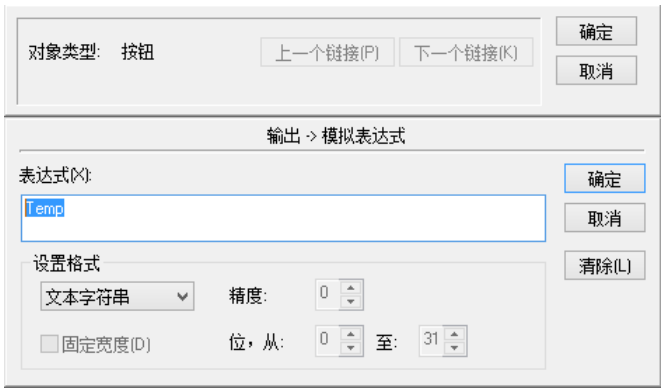
下表给出了在文本对象中使用动画格式字符时显示的模拟值的示例。在这些示例中，温度 I/O 标记的当前值为 123.45 Cº。使用“模拟”动画链接将此标记指定为文本对象的引用值。

格式化的文本对象	文本对象的显示值
Temp = # C	124 C 显示整数形式的温度。在文本对象中只需输入一个 #。
Temp = #.# C	123.5 C 利用 # 字符，温度的分数部分四舍五入为单个数字。温度的整数部分不变。
Temp = 00000 C	00124 C 向数字的整数部分添加前导零。
Temp = 000.0 C	123.5 C 由于格式中只有一个零，因此温度的分数部分四舍五入为一位。
Temp = 00.00 C	123.45 C 原始温度值保持不变。
Temp = 0,000.#	0,123.5 C 在温度参考值的整数部分强制添加一个前导零和一个逗号。利用 # 字符，分数部分四舍五入为单个数字。

将文本对象的格式指定为数字

1. 在窗口中创建一个文本对象并插入一个或多个字符以格式化与该文本对象关联的参考值。
2. 双击文本对象以显示**动画链接选择对话**框。
3. 在**值显示**区域中，选择**模拟**。此时出现**输出 -> 模拟表达式对话**框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在表达式字段中，输入模拟标记名或表达式。
5. 选择确定。
6. 保存对窗口所做的更改。
7. 运行应用程序以验证文本对象中是否正确显示数字。

使用工业图形编辑器 (IGE)

“工业图形编辑器”是用于创建工业图形的工具。如需有关使用该编辑器的操作说明，请参阅《工业图形编辑器用户》文档。下面几节将列出要用于工业图形编辑器的任何 InTouch HMI 特定配置或设置。

管理工业图形

工业图形是一些用户可创建的图形，用于在 HMI/SCADA 系统中显示数据。

可以使用“工业图形编辑器”，根据一些基本元素（如长方形、线条和文本元素）来创建工业图形。还可以使用“工业图形编辑器”，通过符号的图形工具箱库，嵌入并配置工业图形。

“工业图形编辑器”支持将可缩放矢量图形 (SVG) 导入为工业图形。您可以执行以下操作：

- 将 SVG 导入到“工业图形编辑器”中。图形元素将自动转换为工业图形，其中可以包含许多基元。
- 在使用“工业图形编辑器”工具面板中的图像图标时插入 SVG。
- 将 SVG 用于按钮上的 **UpImage** 和 **DownImage**。
- 将 SVG 用于样式配置下的质量和状态显示图标。

如需导入 SVG 和 SVG 限制的详细信息，请参阅“工业图形编辑器”帮助的“将 SVG 导入为工业图形”一节。

创建工业图形之后，可以将它嵌入另一个图形或 HMI 系统窗口中并在运行时使用它。

可以将工业图形嵌入到对象的模板或实例中，从而提供多种方法来快速轻松地显示对象特定的信息。通过在模板中嵌入图形，可以更新一个图形，并在整个应用程序中层层传递更改。

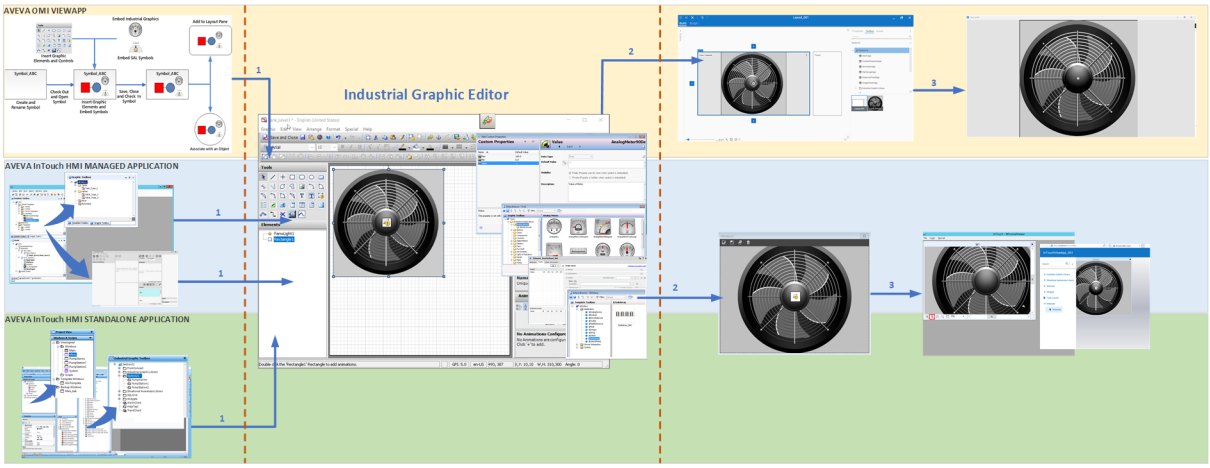
根据您的开发环境，您可以选择用于存储工业图形的位置和方法。

- 创建图形并将图形存储为可复用的标准集，例如常规阀门图形。
- 如果要在运行时在多个实例中使用图形，请将图形存储为模板。
- 存储图形以便在特定应用程序中使用。

用于 HMI/SCADA 应用程序的工业图形编辑器工作流程

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

“工业图形编辑器”是跨各种 HMI/SCADA 应用程序使用的标准图形编辑器。它可以用来创建图形，这些图形可内嵌到 HMI/SCADA 应用程序中，以代表工厂车间的资产。整体工作流程在下面几节中介绍：



托管的 InTouch 应用程序

向 InTouch HMI 托管应用程序添加图形的工作流程从 IDE 开始。

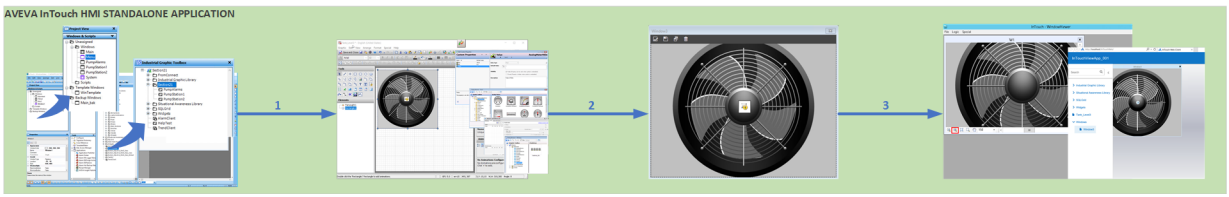
1. 创建或编辑图形，可以是
   a. 从“可视化”文件夹：
      - 创建新图形或选择现有图形。
      - 打开该图形以启动工业图形编辑器。

   - 或者 -

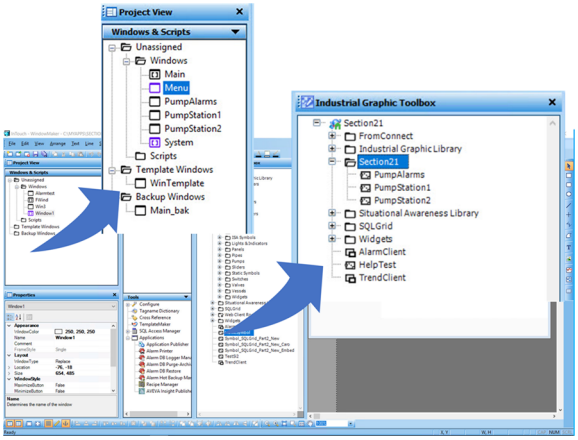
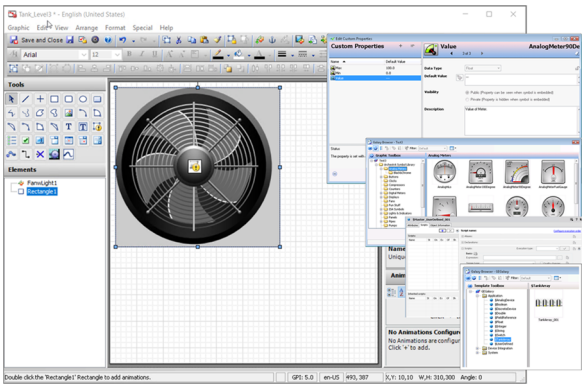
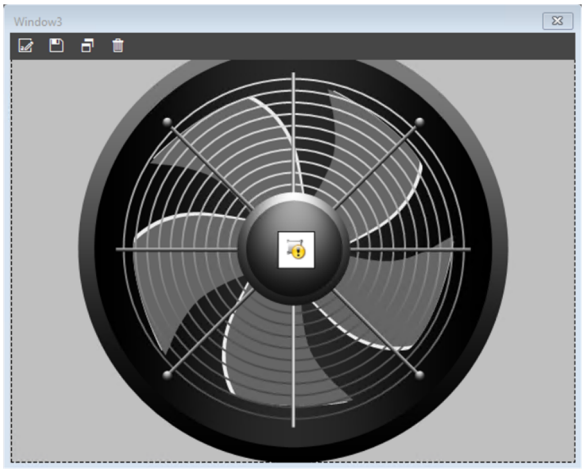
   a. 在您希望与图形关联的对象的对象编辑器中：
      - 在“内容”窗格中选择 + 按钮，以向该对象添加图形。
      - 选择编辑内容按钮，以启动“工业图形编辑器”。
2. 在工业图形编辑器中编辑图形。根据需要修改图形元素。
   - 插入图形元素并嵌入图形。
   - 添加/修改脚本、引用、自定义属性和动画，以使图形能够代表资产，并在运行时对命令和数据更改作出响应。
3. 在窗口中嵌入图形。
4. 要在运行时查看图形，请启动 WindowViewer 或 InTouch Web 客户端。

InTouch 独立应用程序

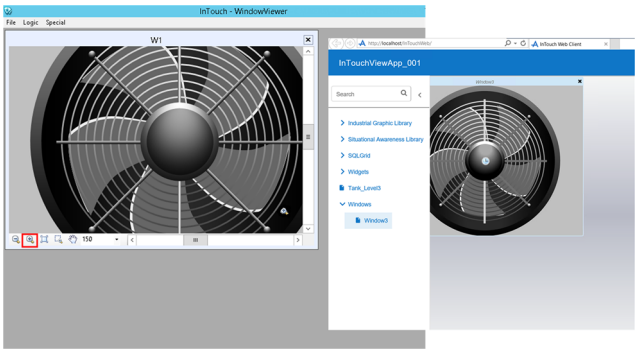
InTouch HMI 独立应用程序的工作流程从 WindowMaker 开始。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

第 1 步  	在 InTouch HMI WindowMaker 中，使用“工业图形编辑器”创建符号。 编辑图形以启动“工业图形编辑器”。
步骤 2 	在“工业图形编辑器”中，使用脚本、自定义属性、标记和对象引用、样式、工具、动画和格式化来构建图形。
步骤 3  	在“窗口”中嵌入图形。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

步骤 4	要在运行时查看图形，请启动 WindowViewer 或 Web 客户端。
	

打开图形进行编辑

您可以从图形启动“工业图形编辑器”：

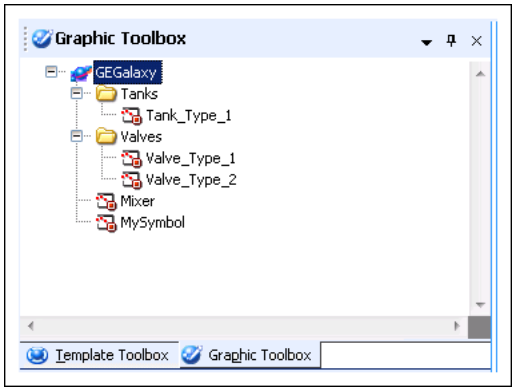
- 包含在图形工具箱中。
- 包含在“自动化对象”模板或实例中。
- 嵌入在 InTouch 窗口中。

在打开图形以使用“工业图形编辑器”进行编辑时，便签出了该图形。在您将图形签出后，其他用户便无法编辑该图形。

您可以同时打开“工业图形编辑器”的多个实例。但无法在“工业图形编辑器”的多个实例中编辑同一个图形。

在图形工具箱中编辑图形

1. 打开图形工具箱。
2. 浏览到要编辑的图形。



3. 双击该图形。此时出现“工业图形编辑器”。
4. 编辑图形。
5. 选择保存并**关闭**。“工业图形编辑器”将会关闭，并签入更新的符号。

编辑“自动化对象”中包含的符号

1. 打开“自动化对象”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 选择图形选项卡。
3. 选择要编辑的符号并选择打开。此时出现“工业图形编辑器”。
4. 编辑符号。
5. 选择保存并关闭 。“工业图形编辑器”将会关闭，并签入更新的图形。根据具体的“自动化对象”，还可能会显示一条确认消息。选择是以保存。

编辑嵌入到 InTouch 窗口中的图形

1. 在 WindowMaker 中，打开包含嵌入图形的 InTouch 窗口。
2. 右键单击要编辑的嵌入图形，指向工业图形“图形名”，然后选择编辑符号。此时出现“工业图形编辑器”。
3. 编辑图形。
4. 选择保存并关闭 。“工业图形编辑器”将会关闭，并签入更新的图形。

注意：要使图形保持签出状态，请在“工业图形编辑器”中选择保持签出。这样可以确保其他用户无法签出您的图形进行编辑。

符号更改传播

对源工业图形所作的任何修改都会传播到所有内嵌的工业图形。这会影响 WindowMaker 中的工业图形与自动化对象继承的工业图形。

如果对工业图形进行修改，并在打开的 InTouch 窗口中使用它，则状态栏的右下角会出现“符号已更改”图标。

双击此图标时，会使用所作的更改来更新内嵌的工业图形。

下例显示符号更改如何进行传播。

传播符号更改

1. 按照自动创建新的 Application Server 对象实例中的示例进行操作。
2. 在 WindowMaker 中，打开显示阀符号的窗口。
3. 打开 Application Server 对象模板 $Valve1 中存放的工业图形 ValveSymbol。
4. 进行一些更改，然后选择关闭并保存。此时更改会传播到 Application Server 对象实例 Valve1_E22。在 WindowMaker 中，出现“符号已更改”图标。
5. 双击图标以接受更改。此时会更新内嵌的工业图形。

符号动态大小传播

您可以控制将源符号大小的更改传播到内嵌符号的方式。例如，大小更改为 ：

- 调整了源符号中一个元素的大小，以至符号边界发生了改变。
- 在源符号中添加或删除元素，使其符号边界发生改变。

此功能称为动态大小更改传播，它可以启用，也可以禁用。

如需有关动态大小传播的详细信息，请参阅《工业图形编辑器用户》文档。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

比较 WindowMaker 与工业图形编辑器

您可以使用“工业图形编辑器”执行在 InTouch WindowMaker 中执行的大部分任务。您还可以使用许多相同的快捷键。

“工业图形编辑器”具有 InTouch WindowMaker 中未提供的功能：

- 额外元素。
- 元素额外的增强型外观。
- 额外的增强型设计时功能。

外观

“工业图形编辑器”对 InTouch 图形配置进行了扩展。例如，您可以使用：

- 渐变（对于线条、填充和文本颜色）。
- 图案（对于线条、填充和文本颜色）。
- 纹理（对于线条、填充和文本颜色）。
- 局部透明。
- 与符号或屏幕相关的填充行为。

元素

元素是用于创建工业图形的图形对象。“工业图形编辑器”提供 InTouch WindowMaker 中未提供的元素，例如：

- 曲线与封闭曲线。
- 由两个或三个点定义的弧形、饼图与弦形。
- 根据属性数据的质量和状态，按条件显示图标的状态元素。
- 您通过将基于线条的元素连接在一起形成一个新的封闭元素而创建的图形路径。
- Windows 公共控件，例如日历控件和日期时间选择器控件。

增强的功能

“工业图形编辑器”提供了一整套增强功能来让您的生活更加轻松，为您的生产环境实现可视化。

适用性增强

“工业图形编辑器”使得选择和配置元素变得非常轻松。您可以：

- 从列表中和画布中选择元素。这样不必移动其它元素，就可以在其下面选择元素。
- 只需在画布上选择元素，就可以查看和更改元素的属性和动画（链接）。
- 不必分解组或路径图形，即可编辑组和路径图形中包含的元素。这就是所谓的内置编辑。

样式复制

使用格式刷，您只需选择一个元素，即可轻松地应用另一个元素的样式，甚至是不同类型的元素。

动画复制

使用“工业图形编辑器”，您可以将一个元素的动画复制、剪切并粘贴到另一个元素，甚至是不同类型的元素。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素定位

“工业图形编辑器”对 InTouch WindowMaker 的定位功能进行了扩展，您可以：

- 按水平或垂直方向均匀分布元素。
- 使元素的水平和/或垂直大小相同。
- 增大或减小水平或垂直空间。
- 除去元素间的水平或垂直空间。
- 锁定元素，避免意外移动或编辑。
- 在设计时围绕旋转中心，按任意角度旋转元素。
- 同时对多个元素应用调整大小和旋转。
- 将元素的 z 顺序向后或向前移动一级。
- 对齐文本框和按钮中的文本。

组功能

“工业图形编辑器”使用组的概念来代替 InTouch WindowMaker 的单元格和符号概念。您可以：

- 在组中嵌入组。
- 在组（或嵌入的组）中编辑各个元素而不必分解组。
- 从现有组中轻松删除元素或将元素添加到现有组中。

自定义属性的可扩展性

您可以将自定义属性添加到符号或嵌入的符号。您可以将自定义属性连接到“自动化对象”属性、元素属性，甚至是 InTouch 标记。您可以在设计时和运行时使用自定义属性，就像使用任何预定义的属性一样。

注意：引用名称中包含连字符的 InTouch 标记的 Application Server 自定义属性在运行时将不工作。例如，“InTouch:TAG-1”在运行时将不工作。

元素样式

元素样式定义图形元素的一个或多个填充、线条、文本、闪烁、轮廓和状态属性。可将元素样式应用于图形元素以将元素设置为该元素样式中定义的预配置属性。在元素样式中定义的本地元素属性将被禁用。

元素样式可帮助屏幕构建人员或其他创建符号的人员制定标准。

其它增强功能

使用“工业图形编辑器”，可以：

- 通过脚本访问元素的属性和符号的自定义属性。
- 设置元素的 Tab 键顺序。
- 使用线条端点样式，例如箭头。
- 从元素动态禁用特定动画而不会丢失配置信息。
- 使用图像元文件和其它图像格式。
- 使用反锯齿来改善符号的显示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

常见 WindowMaker 任务和技术的过程

在 InTouch WindowMaker 中执行的大多数配置都可以在“工业图形编辑器”中轻松执行。图形、动画和脚本之间有一些差异，也有相似之处。

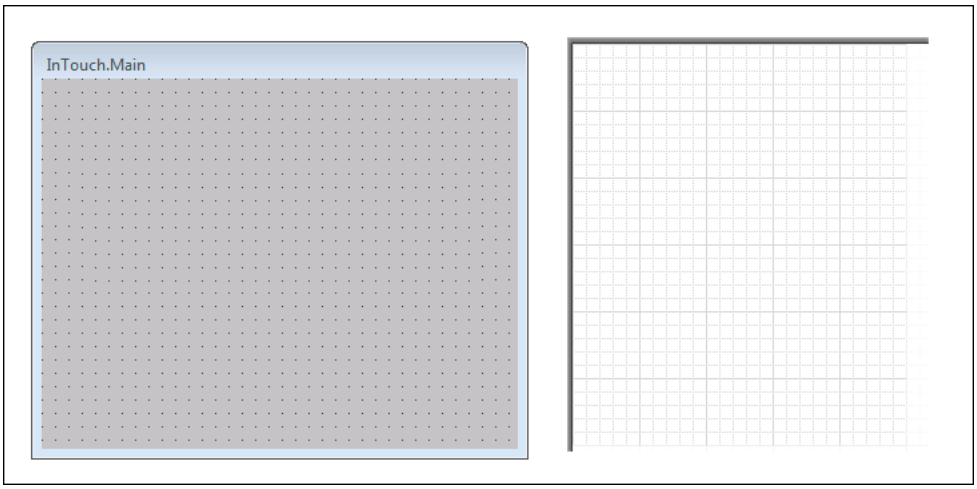
使用图形

使用“工业图形编辑器”的方式与使用 InTouch WindowMaker 基本相同。“工业图形编辑器”包括一个绘图区域，您可以在其中放置图形对象，为生产过程构造可视化表示，并提供人机交互界面。

您在 InTouch 中使用的有些对象在“工业图形编辑器”中并不存在，例如 ActiveX 控件和一些向导。其它功能更强大、可更好地集成的控件代替了它们的功能。

使用绘图区域

“工业图形编辑器”的绘图区域称为画布。可以像 InTouch 窗口一样使用它。画布的最大尺寸为 2,000*2,000 像素。



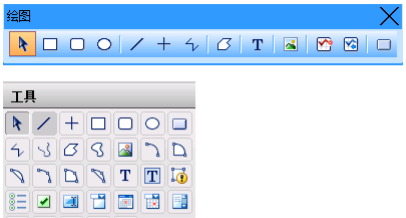
设置绘图区域颜色

在“工业图形编辑器”中无法设置绘图区域。绘图区域的颜色是透明的，并继承该符号嵌入到的环境的颜色。

如果将工业图形嵌入 InTouch 窗口，元素之间的区域将采用 InTouch 窗口的颜色。

使用基本对象

InTouch 基本对象（如长方形、椭圆形和多边形）可以在“工业图形编辑器”中使用非常相似的方法绘制。基本对象在“工业图形编辑器”中称为元素。



使用复杂对象

InTouch 对象（如 ActiveX 控件、向导、单元格和符号）在“工业图形编辑器”中并不存在。

但您可以将 SmartSymbol 导入到工业图形中。在导入 SmartSymbol 时，会转换 SmartSymbol 的元素和动画。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在“工业图形编辑器”中，您可以创建元素组。组维护各个所包含元素的属性。您可以设置组的 TreatAsIcon 属性来更改组的行为。

使用向导

您无法将 InTouch 向导导入工业图形或图形工具箱。而是可以使用：

- 工业图形库，您可以将该库导入到图形工具箱。
- 属于工具箱一部分的 Windows 控件。您可以使用：
  - 单选按钮组
  - 复选框
  - 编辑框
  - 组合框
  - 日历控件
  - 日期时间选择器
  - 列表框

使用动画

在“工业图形编辑器”中，您可以像在 InTouch WindowMaker 中一样使用动画来设置符号的运行时行为。您可以为一个元素或图形配置一个或多个动画。数据可以来自各种数据源。

配置数据源

在 InTouch WindowMaker 中，可以使用“标记名字典”来定义包含值的变量。在“工业图形编辑器”中，数据源可以是：

- “自动化对象”属性。
- 自定义属性和继承的符号属性。
- InTouch 标记本身。“工业图形编辑器”使用特殊的 InTouch 引用，可用于直接连接到 InTouch 标记。

使用数据类型

工业图形使用不同于 InTouch 数据类型的数据类型。

下表显示了两者的数据类型以及它们之间的对应关系：

InTouch	Application Server	描述
离散	布尔型	布尔值。例如：0 或 1
整型	整型	整数值。例如：-4、7 或 22
实型	浮点或双精度	具有不同精度的浮点值或双精度值。 例如：3.141, -5.332 或 1.343e+17 浮点型：32 位。IEEE 单精度浮点标准，当需要 6-7 位有效数字时使用。缺省值为 NAN。 双精度：64 位。IEEE 双精度，当需要 15-16 位有效数字时使用。缺省值为 NAN。
消息	字符串型	字符串值。例如：“Hello World”

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch	Application Server	描述
不可用	日期时间	日期时间值。例如："04/13/2006 04:03:22.222 AM"
不可用	经过时间	表示经过时间的浮点值，以秒为单位。它通常显示为以下列格式，但存储为浮点值。 <1>[</1>–<2>][DDDDDD] [HH:MM:]SS[.fffffff]</2> 值如下所示： • DDDDDD 是从 0 到 999999 • HH 是从 0 到 23 • MM 是从 0 到 59 • SS 是从 0 到 59 • fffffff 是在小数点右边的小数部分 经过时间可以是正数，也可以是负数。
不可用	国际化字符串	一种可以存储特殊字符的特殊字符串数据类型。

您可以将工业图形配置为从 Galaxy 检索数据。

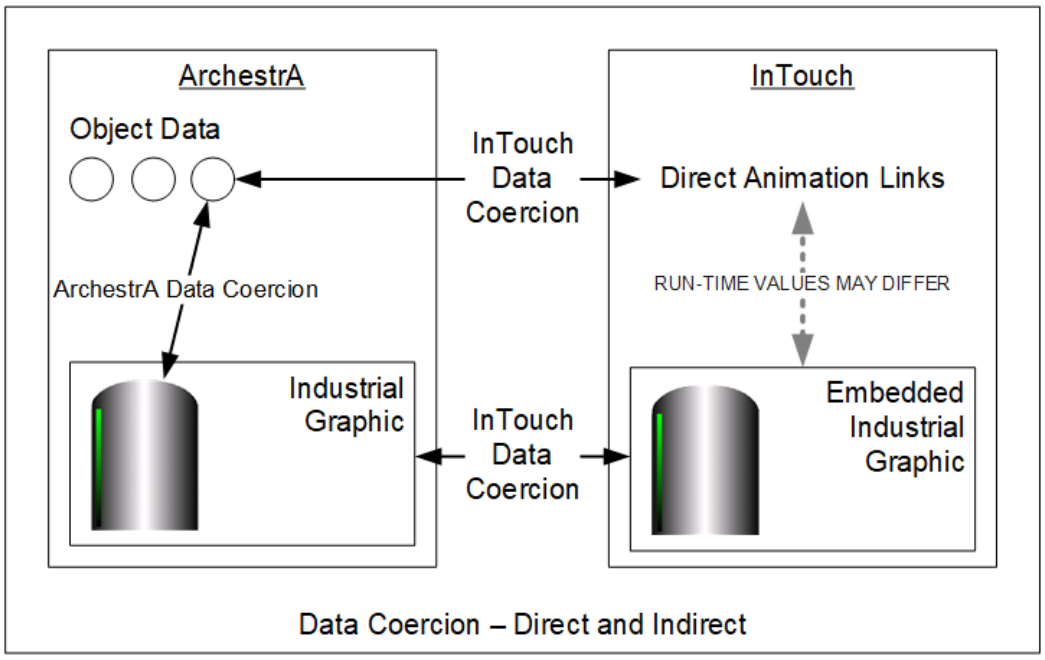
当源数据类型不同于它用于的数据类型时，就会根据 IDE 数据强制转换规则对数据进行强制转换，字符串值“-10”在动画中将强制转换为“True”。

如果将此工业图形嵌入 InTouch 窗口，将根据 InTouch 数据强制转换对动画链接的数据类型进行强制转换。嵌入的工业图形在 InTouch HMI 中显示为“True”。

但如果您直接在指向原始数据源的 InTouch HMI 中创建离散型动画显示链接，则结果值可能会不同。

在本例中，字符串值“-10”在 InTouch HMI 中显示为“False”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



使用动画

您可以使用**动画链接对话**框来配置 InTouch 动画。您可以通过双击 InTouch 对象打开此对话框。

您可以在“工业图形编辑器”中使用**编辑动画对话**框配置动画，通常可以通过双击元素打开此对话框。

部分动画类型不同，其它动画类型已分组，用于简化配置。可使用下表查找“工业图形编辑器”中对等的动画类型：

InTouch 动画	工业图形编辑器动画
用户输入 - 离散	用户输入 - 布尔型
用户输入 - 模拟	用户输入 - 模拟
用户输入 - 字符串	用户输入 - 字符串
游标 - 垂直	垂直游标
游标 - 水平	水平游标
触动按钮 - 离散值	按钮 - 布尔值
动作	动作脚本
显示窗口	（不支持）
隐藏窗口	（不支持）
线条颜色 - 离散	线条样式 - 布尔
线条颜色 - 模拟	线条样式 - 真值表

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 动画	工业图形编辑器动画
线条颜色 - 离散报警	转换为线条样式
线条颜色 - 模拟报警	转换为线条样式
填充颜色 - 离散	填充样式 - 布尔
填充颜色 - 模拟	填充样式 - 真值表
填充颜色 - 离散报警	转换为填充样式
填充颜色 - 模拟报警	转换为填充样式
文本颜色 - 离散	文本样式 - 布尔
文本颜色 - 模拟	文本样式 - 真值表
文本颜色 - 离散报警	转换为文本样式
文本颜色 - 模拟报警	转换为文本样式
对象大小 - 高度	高度
对象大小 - 宽度	宽度
位置 - 垂直	垂直位置
位置 - 水平	水平位置
填充百分比 - 垂直	垂直填充百分比
填充百分比 - 水平	水平填充百分比
其它 - 可见性	可见性
其它 - 闪烁	闪烁
其它 - 相对方向	相对方向
其它 - 禁用	禁用
其它 - 工具提示	工具提示
值显示 - 离散	值显示 - 布尔
值显示 - 模拟	值显示 - 模拟
值显示 - 字符串	值显示 - 字符串

使用脚本

在“工业图形编辑器”中配置脚本的方式与 InTouch WindowMaker 中基本相同。不过仍有些不同之处：

InTouch 脚本	工业图形编辑器脚本
应用程序脚本	（不可用）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 脚本	工业图形编辑器脚本
窗口脚本	符号预定义脚本
键脚本	具有键触发器的动作脚本动画
条件脚本	具有 OnTrue、OnFalse、WhileTrue 或 WhileFalse 触发器的符号命名脚本
数据改变脚本	具有 DataChange 触发器的符号命名脚本
快速功能	（不可用）
ActiveX 事件脚本	（不可用）
动作脚本	动作脚本动画

使用应用程序脚本

在 InTouch HMI 中， 可以按以下条件触发应用程序脚本：

- 当应用程序在 WindowViewer 中启动时触发一次。
- 当应用程序在 WindowViewer 中运行时定期触发。
- 当应用程序在 WindowViewer 中关闭时触发一次。

工业图形与 InTouch 应用程序对应，使您可以配置直接与符号关联的预定义脚本。例如：

- 显示时
- 显示期间
- 隐藏时

使用键脚本

您不能在“工业图形编辑器”中使用键脚本，但可以将元素与由组合键激活的动作脚本相关联。

使用条件脚本

您可以使用“符号脚本”功能，配置脚本在满足某个条件时运行。这样就可以定义触发器，在值或表达式处于以下情况时运行脚本：

- 已满足。（为真期间）
- 达到。（为真时）
- 不满足。（为假期间）
- 未达到。（为假时）

使用数据更改脚本

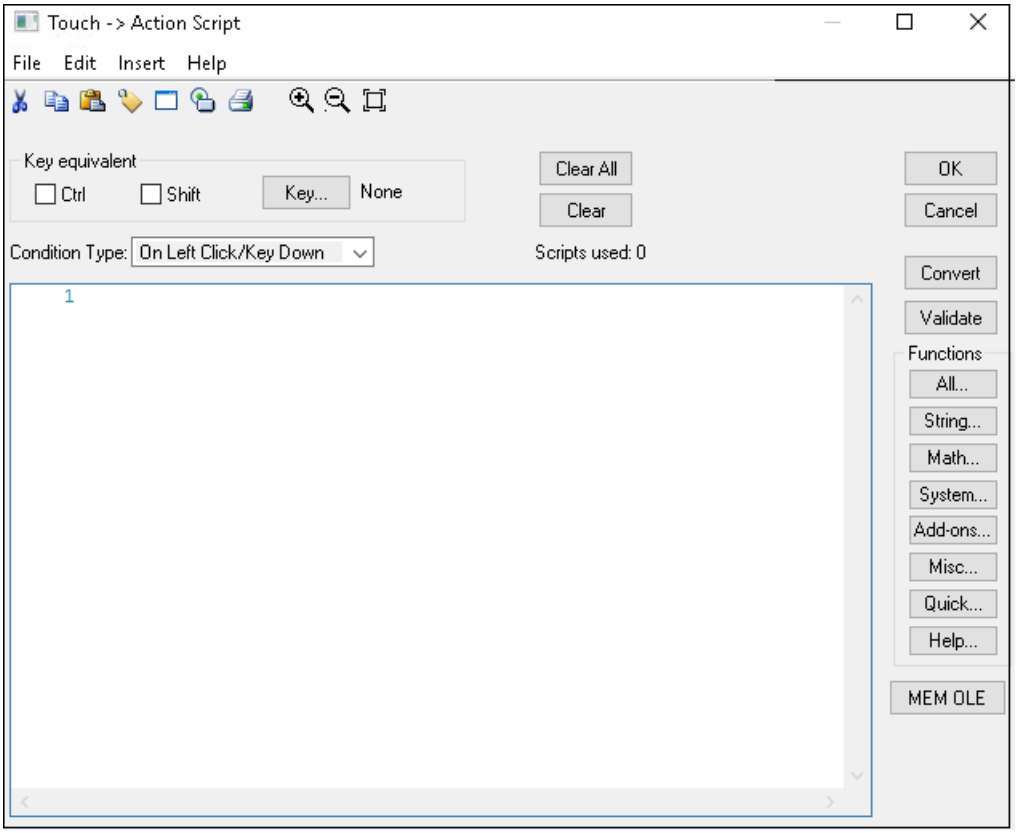
您可以使用“符号脚本”功能，配置脚本在值或表达式发生改变时运行。这样就可以定义触发器，在值或表达式发生改变时运行脚本。

使用动作脚本

在“工业图形编辑器”中配置动作脚本的方式与 InTouch WindowMaker 中基本相同。当运行时用户与元素交互时，例如使用鼠标或通过按键，就会运行动作脚本。

您可以使用 InTouch 动作脚本窗口创建动作脚本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您可以使用“工业图形编辑器”动作脚本窗口创建动作脚本。

您可以为个别元素，也可以为整个符号配置动作脚本。

您可以在“工业图形编辑器”中使用 InTouch WindowMaker 的许多预定义功能。有关可以用于工业图形的 InTouch 预定义功能的完整列表，请参阅导入操作脚本。

其它 InTouch 脚本类型（如应用程序脚本和键脚本）可以使用 Application Server“自动化对象”配置。

连接动画与 InTouch 标记

您可以将元素动画和外观与 InTouch 标记相连接。InTouch 标记在运行时提供用于控制元素动画和外观的值。

您可以通过下列方式将元素动画与 InTouch 标记相连接：

- 使用 **intouch:tag** 语法配置引用。使用工业图形的 InTouch HMI 独立应用程序中的引用标记不需要此语法。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 在 InTouch 中，使用自定义属性并配置嵌入工业图形中的自定义属性，以引用 InTouch 标记。配置 Application Server 属性引用以指向包含作为属性的 InTouch 标记的托管 InTouchViewApp 对象。InTouchViewApp 对象使用 InTouchProxy 对象的功能。
- 配置 Application Server 属性引用以指向包含作为项的 InTouch 标记的 InTouchProxy 对象。这是配置 Application Server 属性引用的特例。

要点：“历史摘要”数据类型仅对用于 AVEVA OMI ViewApps 或 InTouch HMI 托管应用程序的 Application Server 对象属性有效。请勿尝试在 InTouch HMI 新型应用程序中使用自定义属性分配的历史摘要数据类型。

使用 InTouch:Tagname 语法

在使用 **intouch:tagname** 语法时，动画会连接到使用图形的节点的 InTouch 标记。关于此语法的使用方式有一些限制：

- 与在 Application Server 中不同，不能使用 True 和 False 作为布尔值。应改为使用 1 和 0。
- 如果要引用 InTouch SuperTag，请改为使用以下语法：

  `attribute("InTouch:SuperTag\Member")`

- 带有工业图形的 InTouch 独立应用程序不需要“InTouch:”前缀。标记浏览和运行时标记绑定将在没有“InTouch:”前缀的情况下工作。
- 引用名称中包含连字符的 InTouch 标记的 Application Server 自定义属性在运行时将不工作。例如，“InTouch:TAG-1”在运行时将不工作。

设置输入模式

在某些框中，您可以输入使用静态和/或属性引用和元素属性引用的值或表达式。同时支持这两种输入模式的框中有一个**输入模式选择图标**。

选择：

- **静态模式输入图标**可指定文字静态值或表达式，如 3.141 或“测试”。
- **Reference Mode input**（引用模式输入）图标可指定对属性或元素属性的引用，如：贮料罐_001.PV。

注意：要在引用模式下使用带或不带引用的静态字符串值，可以使用双引号将其括起来，如："Description:"+Tank_001.Desc

配置元素在配置字段或脚本中引用其自己的一个属性时，可以仅使用其属性名称。对于工业图形，没有自引用关键字，例如用于“自动化对象”的“me.”。

但您可以使用“me.”关键字引用正在托管当前配置的工业图形的“自动化对象”的属性。

连接动画与 InTouchViewApp 属性

要能够浏览 InTouch 标记，必须先：

- 通过衍生“InTouch View 应用程序”模板并在 WindowMaker 中对其进行配置来创建托管的 InTouch 应用程序。
- 衍生一个“InTouch View 应用程序”衍生模板的实例。

InTouch 标记由“InTouch View 应用程序”对象实例的属性表示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用 Galaxy 浏览器 InTouch 标记浏览器选项卡

在配置工业图形动画或客户端脚本需要 InTouch 标记的引用时，可以直接从 Galaxy 浏览器中选择 InTouch 标记。

在从动画编辑器或脚本编辑器调用 Galaxy 浏览器时，它会并排显示 **InTouch 标记浏览器选项**卡以及**属性浏览器**和**元素浏览器选项**卡。

InTouch 标记浏览器选项卡的左侧窗格中列出了当前 Galaxy 的所有 InTouchViewApp 实例和模板。右侧窗格中显示所选 InTouchViewApp 的 InTouch 标记。**点域：**列表框将显示与所选标记相关的点域。

利用右侧面板下方的**点域**列表框，可以指定所选标记的点域。

InTouch 标记浏览器选项卡的行为如下：

- InTouch 标记浏览器功能仅在动画编辑器或脚本编辑器中可用。
- Galaxy 浏览器从“标记名字典”中读取 InTouch 标记。
- 无论是否安装了 InTouch HMI，都会安装“标记名字典”组件并可供 Galaxy 浏览器使用。
- 只有在关闭并重新打开 Galaxy 浏览器时，才会刷新标记。
- 在关闭并重新打开 Galaxy 浏览器之前，所有标记都会保留在内存中。
- 如果当前用户签出了 InTouchViewApp，那么 Galaxy 浏览器会读取该 InTouchViewApp 的最新“标记名字典”内容。
- 如果任何用户签入并访问了 InTouchViewApp，那么 Galaxy 浏览器会始终读取“标记名字典”的已签入版本。
- 如果由非当前用户的其它用户签出了 InTouchViewApp，那么 Galaxy 浏览器会读取该 InTouchViewApp 最新签入的“标记名字典”。
- 如果选择了 InTouchViewApp 模板，则输出引用字符串语法为 InTouch:selectedTag。如果选择了 InTouchViewApp 实例，则输出引用字符串语法为 <SelectedInTouchViewAppInstance.selectedTag>。

连接动画与 InTouch 标记

1. 双击元素。

   此时出现**编辑动画**对话框。
2. 从“动画”列表中选择动画。
3. 选择参数。
4. 选择浏览按钮。

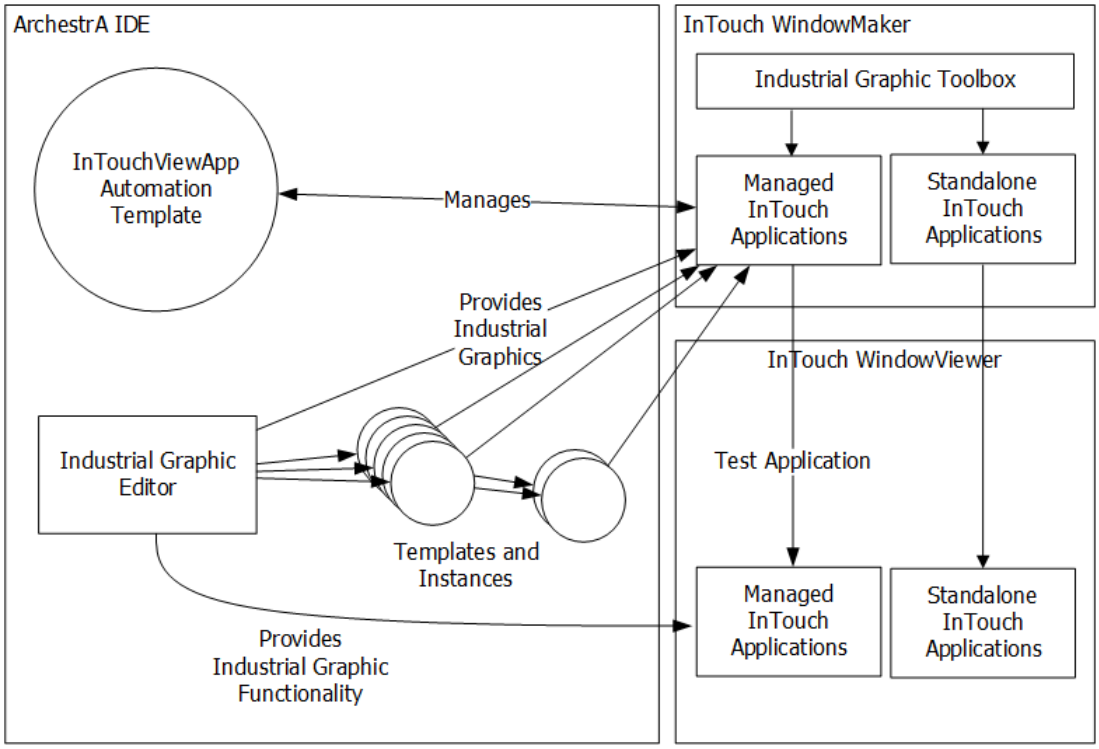
   此时出现 **Galaxy 浏览器**。
5. 选择 **InTouch 标记浏览器选项**卡以显示 **InTouch 标记浏览器**页面。
6. 选择对应托管的 InTouch 应用程序的 InTouchViewApp 对象。此时右侧面板显示 InTouch 标记。
7. 选择标记，然后选择确定。

   所选的对 InTouch 标记的引用会显示在配置框中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在 WindowMaker 中使用工业图形

在托管的或独立 InTouch 应用程序中，可以使用由“工业图形编辑器”创建的工业图形。还可以直接从 WindowMaker 的“工业图形工具箱”添加工业图形。如需有关使用工业图形和 Situational Awareness Library 符号的详细信息，请参阅*工业图形编辑器用户文档*或 WindowMaker 帮助。



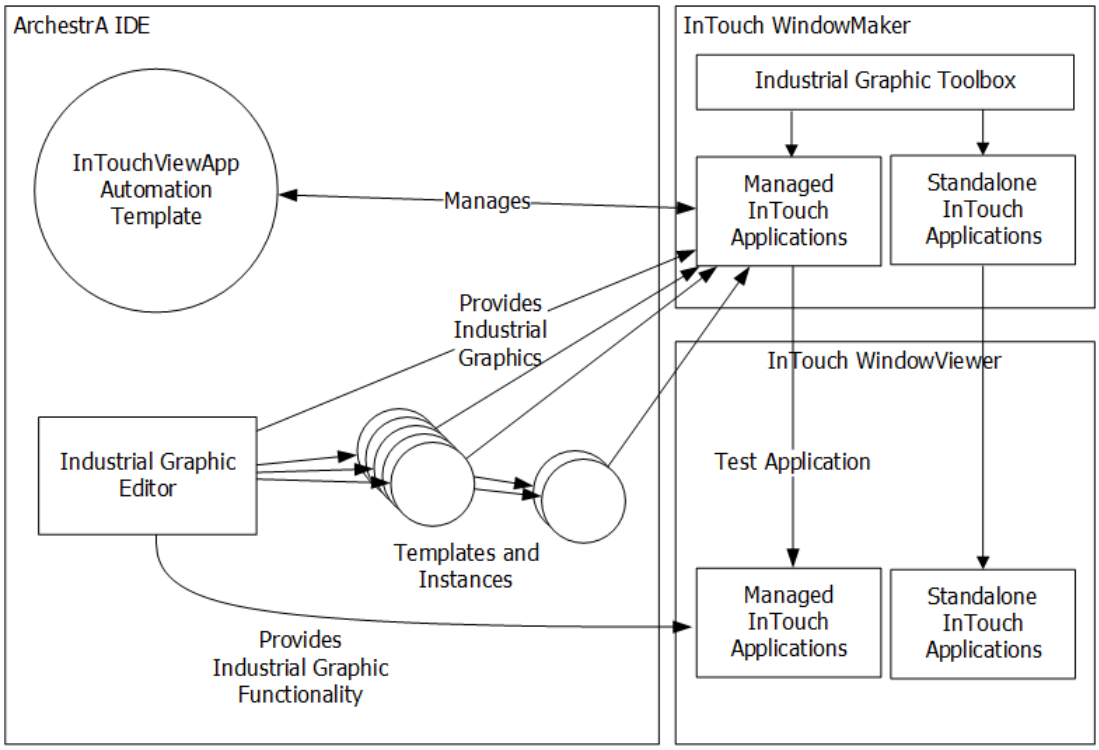
您可以：

- 将工业图形嵌入 InTouch 窗口。
- 将工业图形直接从“工业图形工具箱”拖放到 InTouch 窗口中。
- 调整嵌入的工业图形的大小。
- 将有限数目的 InTouch 动画添加到工业图形中。
- 配置嵌入的工业图形的自定义属性。
- 启动“工业图形编辑器”。
- 在 WindowViewer 中测试工业图形。
- 为存放嵌入工业图形的 AutomationObject 创建新的实例。

关于在 WindowMaker 中使用工业图形

在托管的或独立 InTouch 应用程序中，可以使用由“工业图形编辑器”创建的工业图形。还可以直接从 WindowMaker 的“工业图形工具箱”添加工业图形。如需有关使用工业图形和 Situational Awareness Library 符号的详细信息，请参阅*工业图形编辑器用户指南*或 WindowMaker 帮助。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您可以：

- 将工业图形嵌入 InTouch 窗口。
- 将工业图形直接从“工业图形工具箱”拖放到 InTouch 窗口中。
- 调整嵌入的工业图形的大小。
- 将有限数目的 InTouch 动画添加到工业图形中。
- 配置嵌入的工业图形的自定义属性。
- 启动“工业图形编辑器”。
- 在 WindowViewer 中测试工业图形。
- 为存放嵌入工业图形的 AutomationObject 创建新的实例。

将工业图形嵌入 InTouch 窗口

您可以将工业图形嵌入到托管的和独立的 InTouch 应用程序的 InTouch 窗口中。

工业图形可以是以下对象的一部分：

- 图形工具箱。
- 对象模板。
- 对象实例。

可嵌入的工业图形包含具有已应用“元素样式”的元素的符号以及使用“符号向导编辑器”创建的“符号向导”。

您可以发布托管的 InTouch 应用程序，并在发布的应用程序中包含工业图形。不过，您无法在已发布的应用程序中添加新的工业图形或编辑现有的符号。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从自动化模板中嵌入工业图形

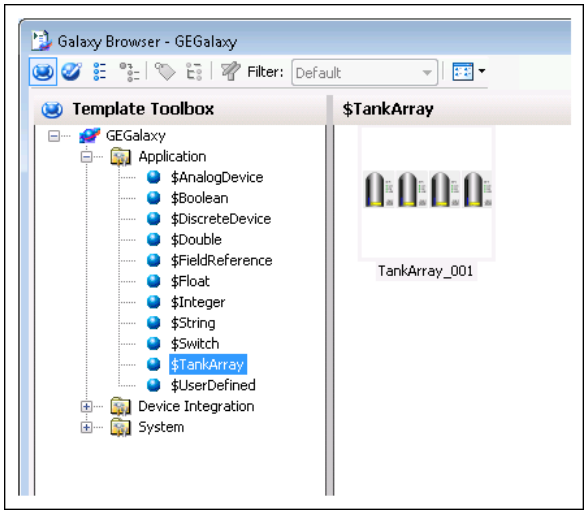
您可以从存放工业图形的自动化模板中嵌入工业图形。同时会为所选的模板创建一个新的衍生实例。

如需有关根据 InTouch 窗口上已有的工业图形创建新实例的详细信息，请参阅在 WindowViewer 中测试工业图形。

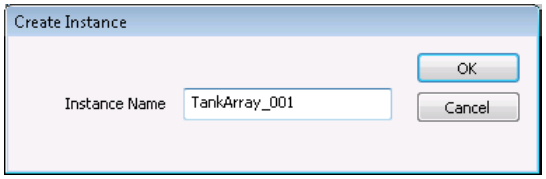
从自动化模板嵌入工业图形

1. 打开 WindowMaker。
2. 右键单击对象，然后选择嵌入工业图形。

   此时出现 **Galaxy 浏览器**对话框。
3. 选择“模板工具箱”图标。此时模板工具箱列表出现在左侧。

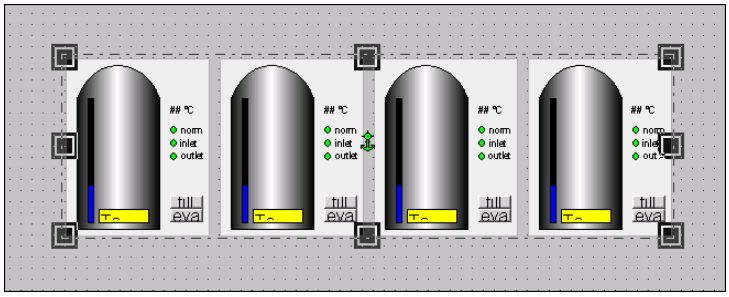


4. 选择包含要嵌入的工业图形的模板。此时所选模板中包含的工业图形出现在右侧。
5. 选择要嵌入的工业图形，然后选择确定。此时“Galaxy 浏览器”关闭，如果将光标移到 InTouch 窗口上，则会出现插入图标。
6. 选择 InTouch 窗口中要嵌入工业图形的位置。此时出现创建实例对话框。



7. 在实例名框中，输入实例的名称。
8. 选择确定。此时会从指定名称的模板中自动衍生一个实例。此时符号嵌入到 InTouch 窗口中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



从实例嵌入工业图形

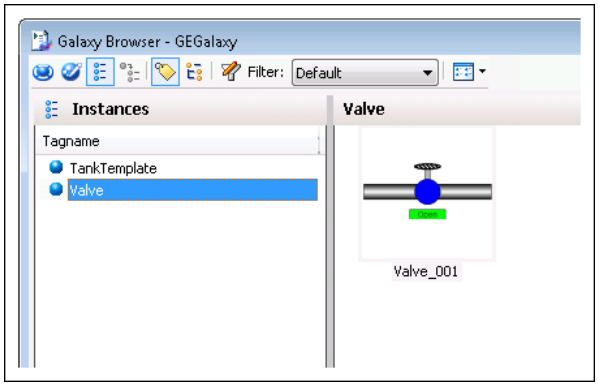
您可以从包含关联的工业图形的实例中嵌入工业图形。

从某个实例中嵌入工业图形时，符号与该实例关联。

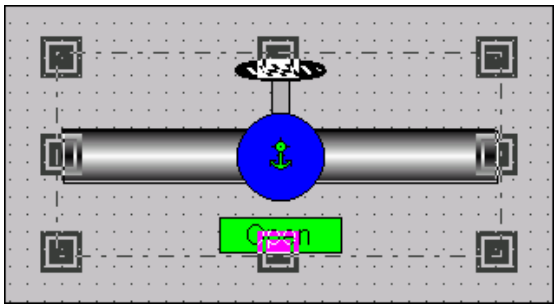
从实例嵌入工业图形

1. 打开 WindowMaker。
2. 右键单击对象，然后选择嵌入工业图形。

   此时出现 **Galaxy 浏览器**对话框。
3. 选择“实例”图标。此时实例列表出现在左侧。



4. 选择包含要嵌入的工业图形的实例。此时与所选实例关联的工业图形出现在右侧。
5. 选择要嵌入的工业图形，然后选择确定。此时“Galaxy 浏览器”关闭，如果将光标移到 InTouch 窗口上，则会出现插入图标。
6. 选择 InTouch 窗口中要嵌入工业图形的位置。此时符号嵌入到 InTouch 窗口中。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从“图形工具箱”嵌入工业图形

您可以从“图形工具箱”中嵌入工业图形。

从“图形工具箱”嵌入工业图形

选项 1

1. 打开 WindowMaker。

   注意：当工业图形正在加载或者如果您在图形工具箱中看到进度条时，请勿执行与工业图形相关的任何操作。

2. 从“工业图形”窗格中选择所需的“工业图形”文件夹。

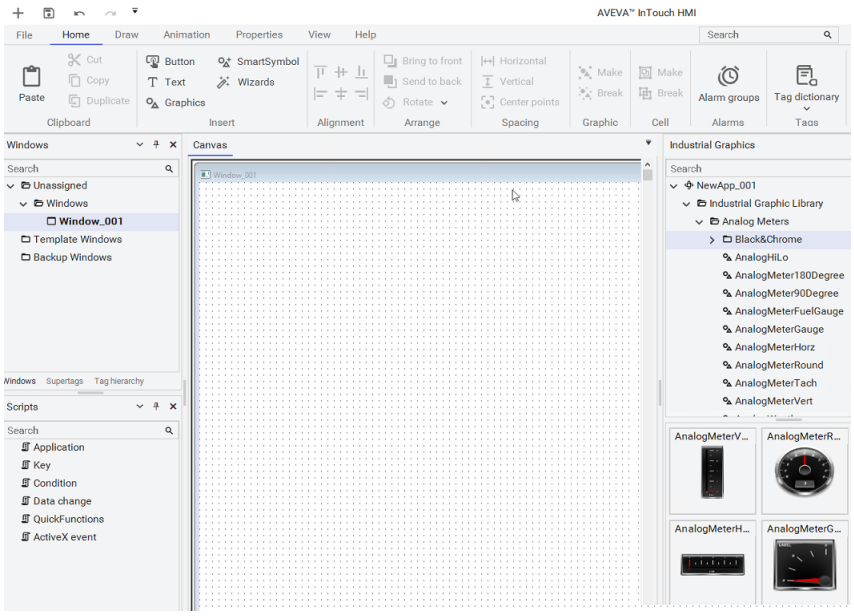
   该文件夹下的工业图形显示为缩略图。

   或

   在搜索框中搜索所需的工业图形。

3. 选择所需的图形并将其拖放到画布上。

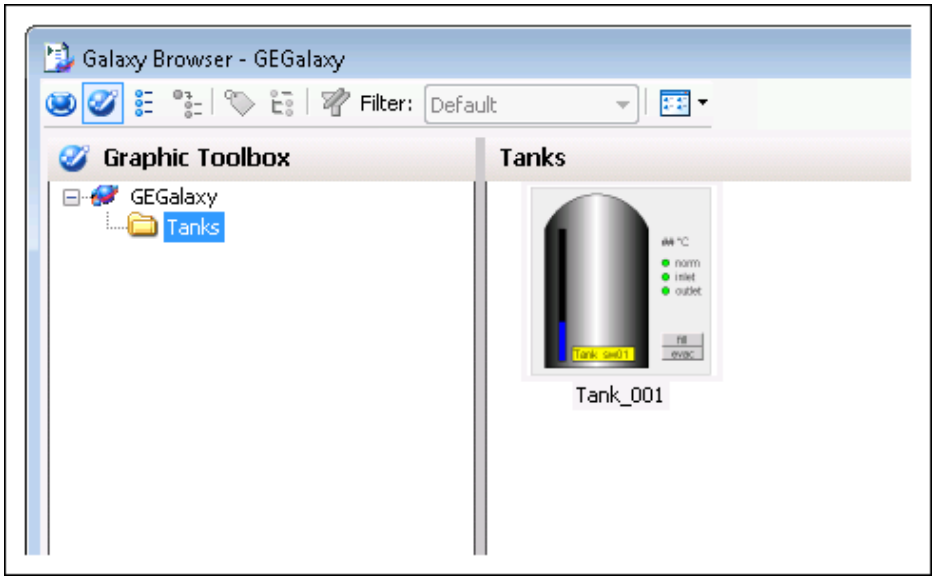
   此时将创建一个框架窗口。



选项 2

1. 打开 WindowMaker。
2. 右键单击对象，然后选择嵌入工业图形。
3. 此时出现 **Galaxy 浏览器**对话框。
4. 选择图形工具箱图标 。此时图形工具箱列表出现在左侧。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 选择要嵌入的工业图形，然后选择确定。如果将光标移到 InTouch 窗口上，则会出现插入图标。
6. 选择 InTouch 窗口中要嵌入工业图形的位置。此时符号嵌入到 InTouch 窗口中。

嵌入已应用元素样式的工业图形

您可以嵌入应用了“元素样式”的工业图形。“元素样式”定义了图形的一个或多个填充、线条、文本、闪烁和轮廓属性。“元素样式”中定义的视觉属性应用于图形。使用“元素样式”可以轻松地将一致的样式应用到多个元素。“元素样式”还为屏幕构建者和其他创建符号的人员建立了视觉标准。

应用了“元素样式”的工业图形嵌入到 InTouch 窗口的方式与任何其它工业图形都类似。

如需有关对工业图形使用“元素样式”的详细信息，请参阅*工业图形编辑器用户文档*中的“使用元素样式”。

托管的 InTouch 应用程序中的元素样式

WindowViewer 一次只能运行一个应用程序。如果在本地节点上部署平台，则 Galaxy 的已配置样式优先于任何其它独立应用程序中的任何已配置样式。

嵌入符号向导

您可以从 System Platform IDE 将“符号向导”嵌入到托管 InTouch 应用程序的窗口中。

嵌入“符号向导”

1. 打开 WindowMaker 并显示将包含嵌入符号的窗口。
2. 右键单击对象，然后选择嵌入工业图形。

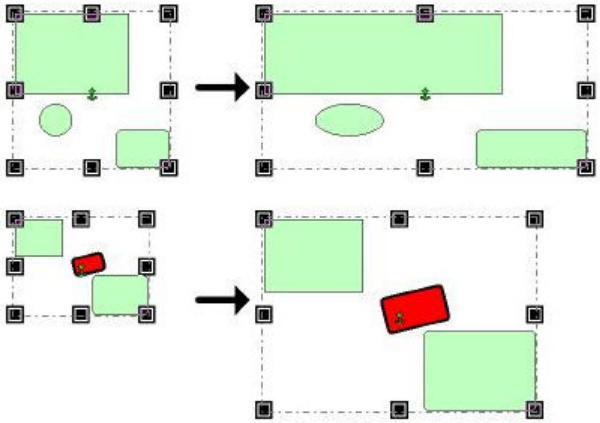
   此时出现 **Galaxy 浏览器**对话框。
3. 选择图形工具箱图标 。此时图形工具箱列表出现在左侧。
4. 选择要嵌入到应用程序中的“符号向导”，然后选择确定。
5. 选择 InTouch 窗口中要嵌入符号向导的位置。此时符号将以缺省的“符号向导”配置嵌入到 InTouch 窗口中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

调整嵌入的工业图形的大小

将工业图形嵌入到 InTouch 窗口中之后，可以使用手柄调整它的大小，或者像处理任何其它 InTouch 对象那样输入宽度与高度值。

不过，如果工业图形包含至少一个旋转元素，则仅能按比例调整工业图形的大小。



您无法将嵌入的工业图形调整到比最小尺寸还小。最小尺寸可以由所包含的元素的笔刷宽度确定。

您可以将嵌入的工业图形复位到在“工业图形编辑器”中创建它时的原始大小。

调整嵌入的工业图形的大小

1. 选择工业图形使其手柄出现。
2. 执行以下操作之一：
   - 拖动手柄之一将工业图形调整到新的大小。
   - 在状态栏上的 **W** 与 **H** 框中输入宽度与高度。

将嵌入的工业图形调整到原始大小

- 使用鼠标右键单击要更改到原始大小的嵌入的工业图形，指向**工业图形**，然后**选择符号 - 原始大小**。此时嵌入的工业图形更改为原始大小。

在 WindowMaker 中配置工业图形

您可以通过以下方式在 WindowMaker 中配置嵌入的工业图形：

- 标准的编辑操作，如复制、剪切、粘贴、创建副本、调整大小、移动及删除。
- 设置 WindowMaker 动画链接。
- 连接工业图形与 InTouch 标记。
- 选择相同父对象的替代实例。
- 选择相同实例的替代符号。
- 启用或禁用动态大小传播。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置工业图形的 WindowMaker 动画链接

像对待其它 InTouch 对象那样，您可以给嵌入的工业图形配置 WindowMaker 动画链接。您仅能配置嵌入的工业图形的外部动画链接，例如：

- 对象大小
- 对象位置
- 可见性
- 启用

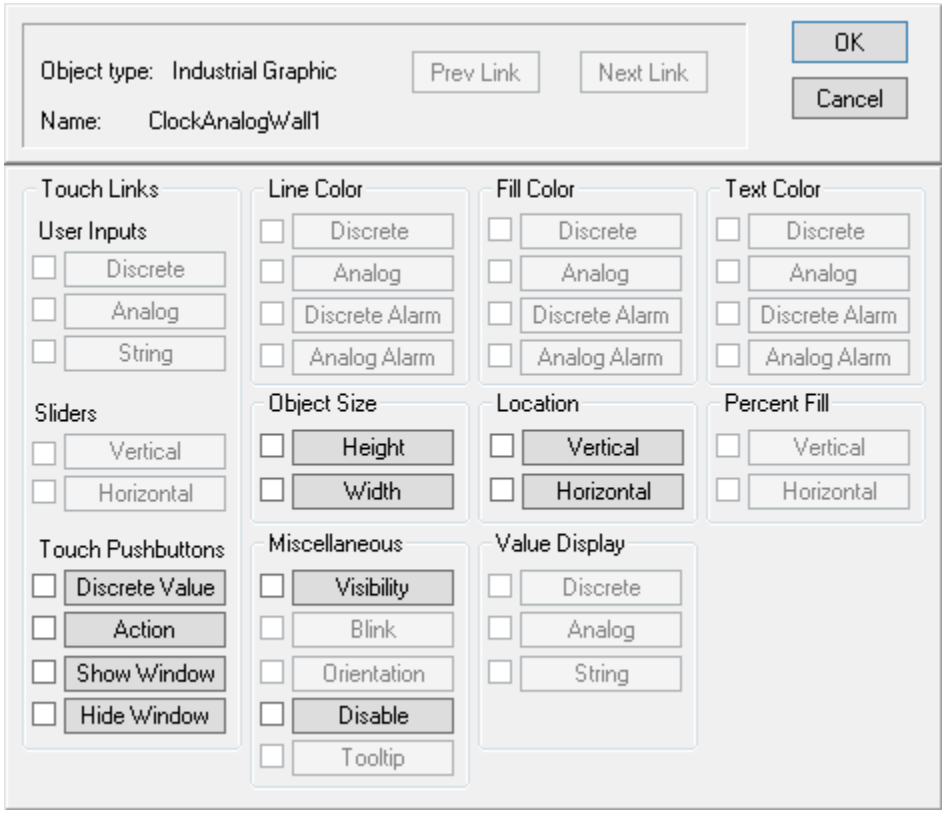
WindowMaker 中配置的动画链接独立于“工业图形编辑器”中配置的那些动画链接。它们不继承工业图形的设置，并且在 WindowViewer 中运行时具有更高的优先级。

配置嵌入工业图形的 WindowMaker 动画链接

1. 选择嵌入的工业图形。
2. 在**绘图**菜单上的模式组中，选择**设置动画效果**。

   或者，右键单击该对象，然后选择**动画链接**。

   此时出现**动画链接**窗口。

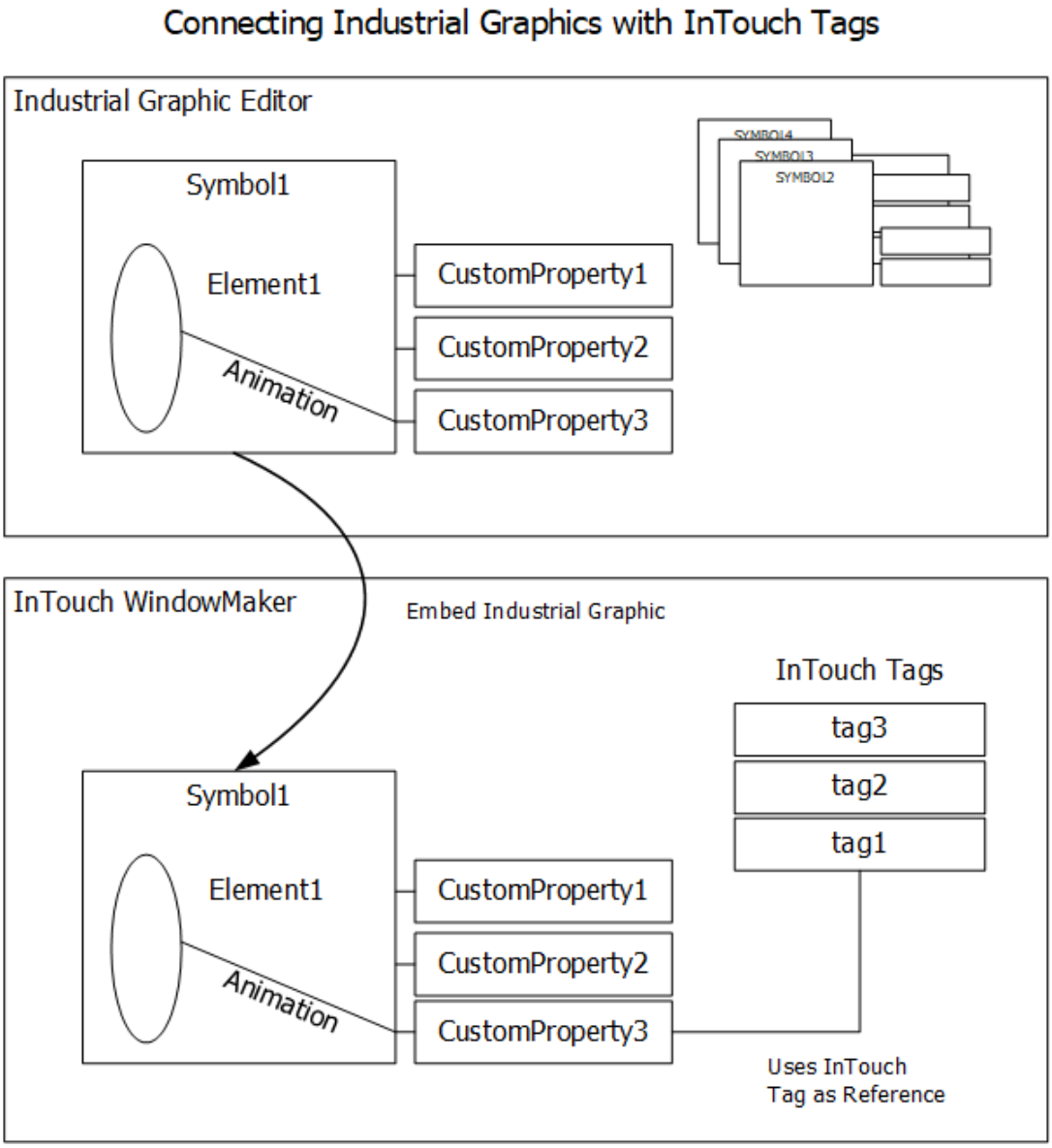


3. 像对待任何其它 InTouch 对象那样，执行任何更改。
4. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将工业图形连接到 InTouch 标记

通过覆盖嵌入的工业图形的自定义属性，可以将工业图形连接到 InTouch 标记。自定义属性向 InTouch 提供工业图形的属性。工业图形的动画可能会，也可能不会在内部使用自定义属性。



将工业图形嵌入到 InTouch 窗口时，会转换动画链接中的引用，如下表所示：

工业图形	嵌入的工业图形
Object.Extension	galaxy:Object.Extension
intouch:Tagname	Tagname

如需有关自定义属性的详细信息，请参阅*工业图形编辑器用户文档*。

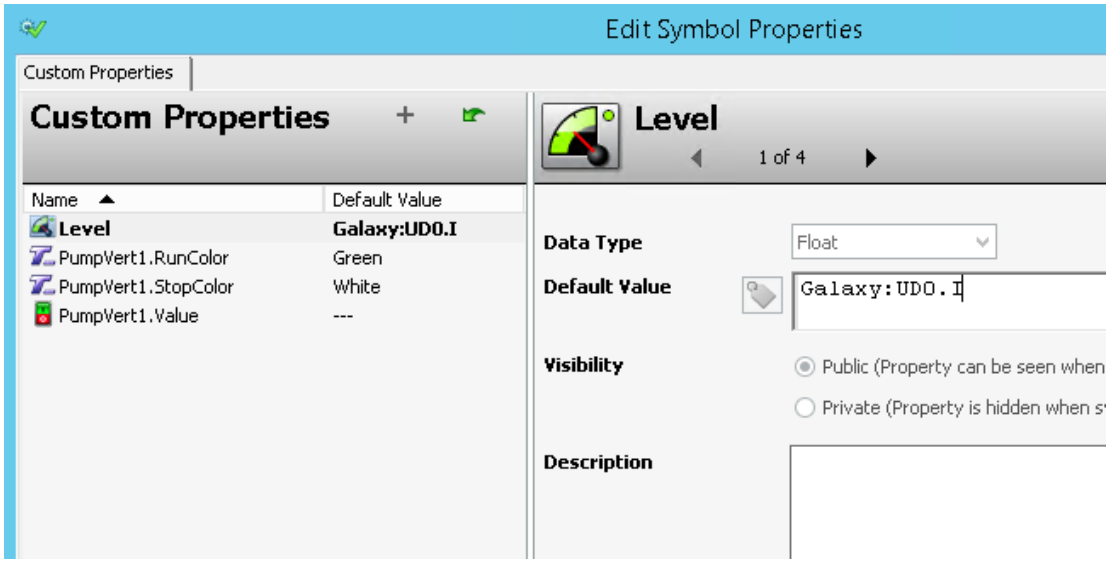
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您在动画编辑器或脚本编辑器中工作时，可以直接从“Galaxy 浏览器”选择 InTouch 标记。从上述任意编辑器调用时，Galaxy 浏览器将列出当前 Galaxy 下所有 InTouchViewApp 实例，并显示所选 InTouchViewApp 实例对应的全部 InTouch 标记及点域。

如需有关使用“Galaxy 浏览器”选择 InTouch 标记的详细信息，请参阅从 InTouch 浏览应用程序服务器对象属性。

将工业图形连接到 InTouch 标记

1. 使用鼠标右键单击 InTouch 窗口中嵌入的工业图形，指向工业图形，然后选择编辑符号属性。此时出现编辑符号属性对话框。



2. 选择要连接到 InTouch 标记的自定义属性。此时所选自定义属性的配置出现在右侧窗格中。
3. 在缺省值框中，执行以下操作之一：
   - 输入 InTouch 标记的名称。
   - 选择浏览按钮，从选择标记对话框中选择一个标记。

4. 要恢复自定义属性的原始值，请选择恢复。
5. 选择确定。现在，使用所选自定义属性配置的工业图形中的任何动画在处理期间都使用 InTouch 标记值。

将工业图形连接到 InTouch 标记的示例

本例显示如何将贮料罐符号连接到 InTouch 标记，该符号包含使用“工业图形编辑器”创建的垂直填充百分比动画。

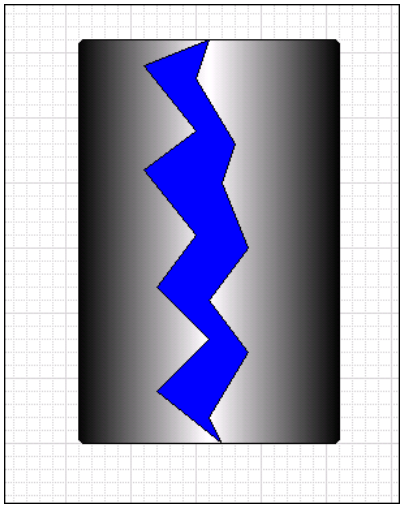
此操作分为三个主要步骤：

- 使用工业图形创建一个示例贮料罐。
- 创建 InTouch 应用程序。
- 衍生并在 WindowViewer 中查看示例贮料罐。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用工业图形创建一个示例贮料罐

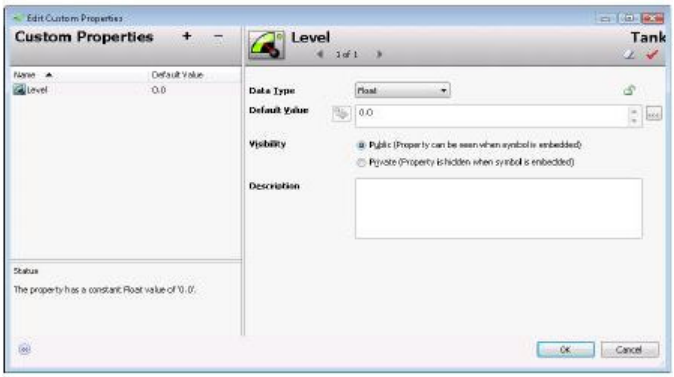
1. 在 IDE 中，创建一个名为“Tank”的新符号，并在“工业图形编辑器”中打开它。
2. 在画布上粘贴一个长方形。根据需要更改其外观。
3. 创建一个代表贮料罐截面的彩色多边形元素，以显示罐中的液位。



4. 选择画布。
5. 在属性菜单上的图形组中，选择编辑。

   或者，双击图形对象。

   此时出现编辑自定义属性对话框。
6. 添加 **Level** 自定义属性。
7. 配置属性详细信息。执行以下操作：
   - 在数据类型列表中，选择浮点型。
   - 在缺省值框中，输入 0。

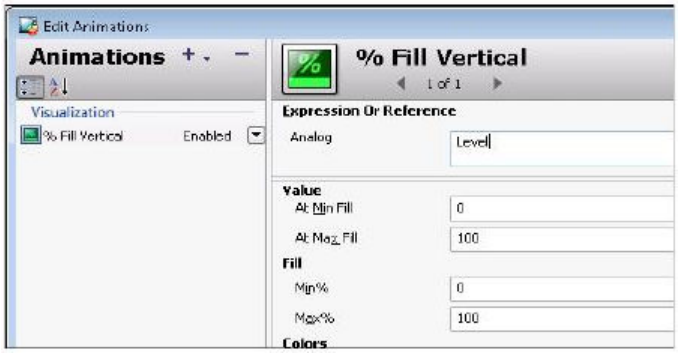


8. 选择确定。
9. 双击代表贮料罐液位的多边形元素。

   此时出现编辑动画对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

10. 添加**垂直填充百分比动**画。
11. 在右**侧**窗格的**模拟**框中，**输**入自定**义**属性的名称。在本例中，它是 Level。



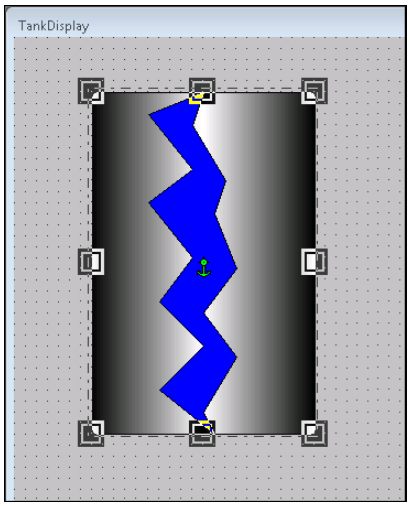
12. **选择** 确定以**关闭编辑动**画对话框。
13. **选择关闭**并保存以关闭“工业图形**编辑**器”。

创建 InTouch 应用程序

1. 在 System Platform IDE 中，**创**建一个新的托管 InTouch **应**用程序。如需**详细**信息，**请**参**阅**创建托管的 InTouch 应用程序。
2. 在 WindowMaker 中打**开**托管的 InTouch **应**用程序。
3. **创**建一个新窗口 TankDisplay。
4. 打**开**“**标记**名字典”，**创**建一个新的**实**型 InTouch **标记** TankLevel。
5. **选择**“嵌入的工**业图**形”**图标**。

   此**时**出**现 Galaxy 浏览器**对话框。
6. **选择** Tank 符号，然后**选择确定**。
7. **选择**窗口中要嵌入**该**符号的新位置。

   此**时** Tank 符号嵌入到窗口中。



8. 使用鼠**标右键单击**嵌入的工**业图**形，指向**工业图形“Tank”**，然后**选择编辑**符号**属性**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现编辑符号属性对话框。

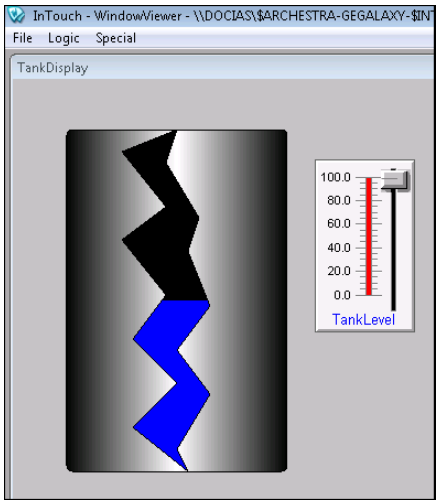
9. 选择自定义属性 Level。
10. 在缺省值框中，输入 TankLevel。您还可以选择省略号按钮，以使用选择标记对话框通过浏览找到 TankLevel。
11. 选择确定。
12. 将游标粘贴到窗口上，然后使用本地 InTouch 标记 TankLevel 来配置它。
13. 保存更改并关闭 WindowMaker。

    此时托管的 InTouch 应用程序会自动签入。

衍生与测试示例贮料罐

1. 在 System Platform IDE 中，衍生托管的 InTouch 应用程序的实例，并连同 WinPlatform 与 ViewEngine 实例一起来部署它。
2. 打开“InTouch 应用程序管理器”，启动 WindowViewer 中列出的应用程序。

   此时贮料罐与游标出现在 WindowViewer 中的窗口上。
3. 您可以移动游标以更改贮料罐液位。



从相同的父对象中选择替代实例

您可以将工业图形中的所有引用重定向到替代实例。工业图形的外观不改变，只是大小可能除外，原因在于不可能编辑继承的工业图形。

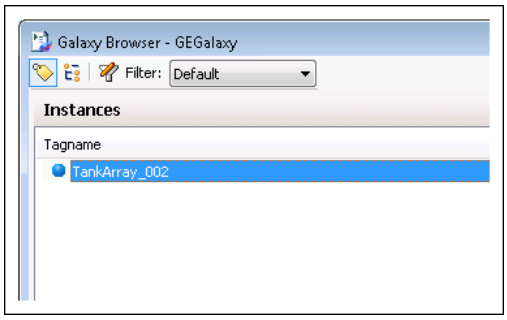
您无法将此功能用于源自“图形工具箱”的工业图形，原因在于它们不与任何对象关联。

从相同的父对象中选择替代实例

1. 使用鼠标右键单击嵌入的工业图形，指向工业图形，然后选择选择替代实例。

   此时出现 Galaxy 浏览器对话框。此时它显示具有相同父对象的其它所有实例。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 从列表中选择一个替代实例，然后选择确定。

   此时工业图形的引用更新为指向新的替代实例。

选择相同实例的替代符号

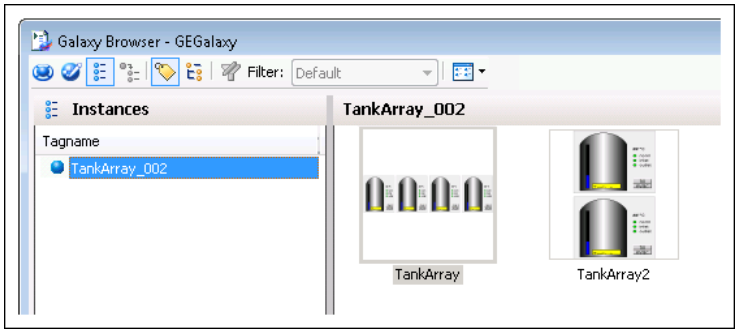
您可以使用属于相同实例的其它工业图形来替代嵌入的工业图形。

您无法将此功能用于源自“图形工具箱”的工业图形，原因在于它们不与任何对象关联。

从相同的实例中选择替代工业图形

1. 使用鼠标右键单击嵌入的工业图形，指向工业图形，然后选择选择替代符号。

   此时出现 **Galaxy 浏览器**对话框。



2. 从右侧窗格中选择一个替代符号，然后选择确定。
3. 如果替代符号与原始符号的大小不同，则会出现一条消息，提示您是否要保持当前嵌入的工业图形的大小。执行以下任意操作：
   - 选择是，保持所选工业图形符号的当前大小。
   - 选择否，将所选的工业图形的大小更新为新工业图形的大小。

   在两种情况下，嵌入的工业图形都会更新为新的替代工业图形。

替换工业图形中的字符串

您可以使用替代字符串来替换嵌入的工业图形中的所有字符串。

替换嵌入的工业图形中的所有字符串

1. 选择嵌入的工业图形。
2. 在动画菜单上的替换组中，选择字符串。

   此时出现替换字符串对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 在相应的框中输入新字符串，然后选择确定。

   此时嵌入的工业图形中的字符串替换为新的替代字符串。

替换工业图形中的引用

您可以使用替代引用来替换嵌入的工业图形中的所有引用。

替换嵌入的工业图形中的所有引用

1. 选择嵌入的工业图形。
2. 在动画菜单上的替换组中，选择标记。

   此时出现替换标记对话框。
3. 在相应的框中输入新引用，然后选择确定。

   此时嵌入的工业图形中的引用替换为新的替代引用。

启用或禁用嵌入的工业图形的动态大小更改传播

您可以启用或禁用嵌入的工业图形的动态大小更改传播。

如果启用动态大小更改传播功能，则对源符号的绝对定位点位置进行任何更改时：

- 嵌入的符号的定位点保持不变。
- 嵌入的符号的位置会相应地移动。

如果禁用动态大小更改传播功能，则对源符号的绝对定位点位置进行任何更改时：

- 嵌入的符号的定位点会相应地移动。
- 嵌入的符号的位置保持不变。

如需有关动态大小传播的详细信息，请参阅《*工业图形编辑器用户*》文档。

启用或禁用嵌入的符号的动态大小更改传播

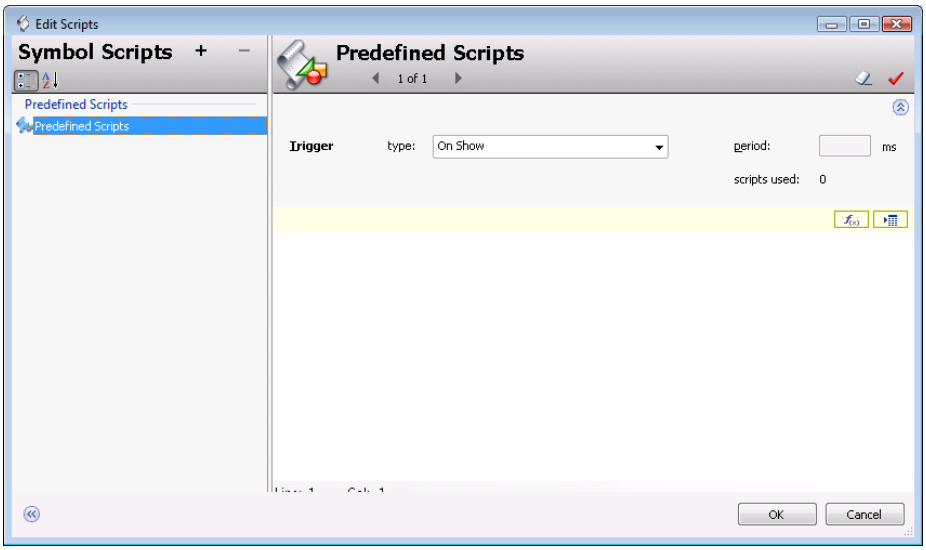
- 使用鼠标右键单击嵌入的工业图形，指向工业图形，然后选择或取消选择动态大小更改。

将脚本与工业图形关联

您可以将脚本与放入 InTouch 应用程序中的工业图形关联。脚本会在 InTouch 应用程序运行时让图形呈现动画效果或修改图形元素。

选择嵌入到 InTouch 窗口中的工业图形之后，可以选择其值会触发脚本运行的表达式或引用。您可以使用“工业图形编辑器”来选择用于触发脚本运行的触发器。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

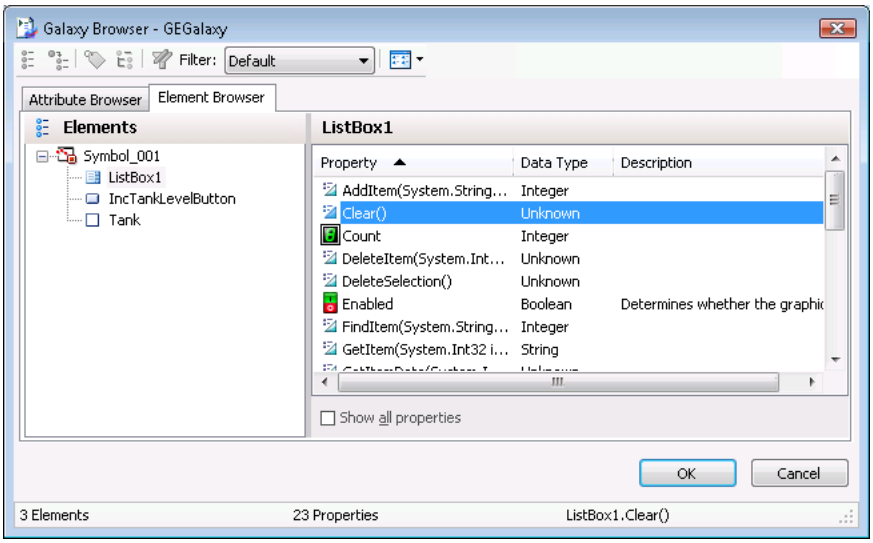


- 图形脚本可以是预定义脚本或命名脚本。预定义脚本会根据运行中应用程序内的图形状态来运行。命名脚本会在与脚本关联的表达式或引用更改状态时运行。
- 预定义脚本与 InTouch HMI 窗口脚本相似。根据脚本触发器的配置方式，预定义的符号脚本可以：
  - 在图形打开或显示之后运行一次。
  - 在图形出现在运行中应用程序内时定期运行。
  - 在图形关闭或隐藏之后运行一次。
- 根据脚本的配置方式，命名的图形脚本可以在触发器值或表达式为真、假、或在真假状态之间转换时运行。此外，命名的图形脚本也可以在与触发器表达式关联的数据值或其质量状态值发生改变时运行。

在工业图形脚本中使用方法

有些元素支持脚本方法。这些方法可以在运行时对元素本身执行各种功能。通常，您可以配置动作脚本来访问这些方法。

要查看元素所支持的属性和方法，可以打开“Galaxy 浏览器”，然后选择元素。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 您可以运行一个包含“编辑框”控件方法的脚本，以便在运行时将文本从文件加载到控件。您还可以运行一个脚本，以便在运行时将“编辑框”控件的当前内容保存到文件。
- “编辑框”控件方法按以下形式在脚本中声明：

  ```
  控件名.SaveText(文件名);
  ```

  其中 *ControlName* 是“编辑框”控件的名称；*FileName* 是文件的名称，该文件包含要加载或保存的控件内容。在上例中，SaveText 是将“编辑框”控件的内容保存到文件的方法的名称。
- 您可以使用包含“组合框”与“列表框”控件方法的脚本，以便在运行时更改列表中的内容。列表项可以添加、删除或修改。
- “组合框”与“列表框”控件方法在脚本中的声明方式类似于“编辑框”控件。

在“工业图形编辑器”中编辑工业图形

您可以使用在 System Platform IDE 中集成的“工业图形编辑器”来编辑嵌入的工业图形。操作步骤为：

1. 在“工业图形编辑器”中打开嵌入的工业图形，修改符号，然后保存它。此时模板、实例或“图形工具箱”中的工业图形会进行更新。

   如需详细信息，请参阅编辑嵌入的工业图形。
2. 在 WindowMaker 中通过选择状态栏右下角的“符号已更改”图标以接受更改。然后更改会传播到 WindowMaker。

   如需详细信息，请参阅在 WindowMaker 中接受符号更改。

编辑嵌入的工业图形

您可以在 InTouch WindowMaker 中轻松地编辑嵌入的工业图形。

如果源符号或其嵌入的符号由其它托管的 InTouch 应用程序使用，则更改会传播到嵌入的符号及 InTouch 应用程序。

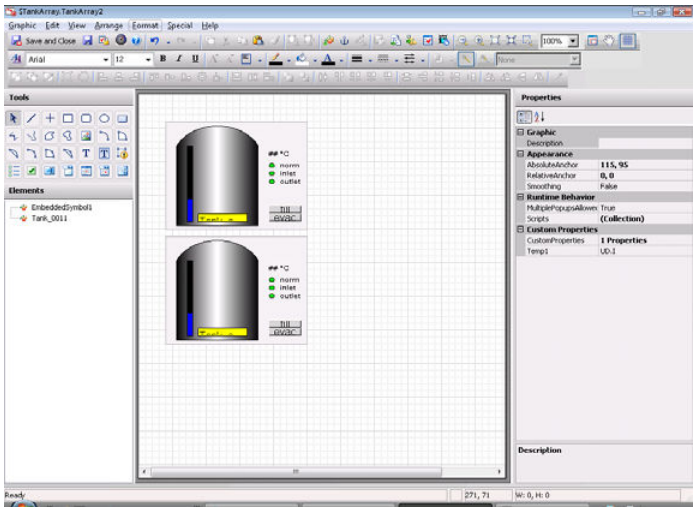
对工业图形所作的任何更改不会自动传播到嵌入的工业图形。如需详细信息，请参阅在 WindowMaker 中接受符号更改。

使用“工业图形编辑器”编辑嵌入的工业图形

1. 使用鼠标右键单击嵌入的工业图形，指向工业图形，然后选择编辑符号。

   此时出现包含该工业图形的工业图形编辑器。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 编辑工业图形。如需有关详细信息，请参阅《创建与管理工业图形用户》文档。

   编辑工业图形包括向符号应用“元素样式”。如需有关使用“元素样式”的详细信息，请参阅 *Application Server 用户文档*。

3. 选择关闭并保存。此时会保存更改并关闭“工业图形编辑器”。
4. 如果 Application Server 对象存放在实例或模板中，请在 IDE 中保存并关闭对象编辑器。

在 WindowMaker 中接受符号更改

如果工业图形发生更改，并且您当前正在 WindowMaker 中的 InTouch 窗口中使用它，则可以立即在 WindowMaker 中接受更改。

如果不立即接受更改，则关闭并再次打开窗口时，符号会在 WindowMaker 中更新。

如果切换到 WindowViewer 以测试应用程序，或者如果您在目标节点上的 WindowViewer 中打开应用程序，则符号也会更新。

在 WindowMaker 中立即接受符号更改

- 对于包含嵌入的工业图形的 InTouch 窗口，执行以下操作之一：
  - 双击状态栏右下角的“符号已更改”图标。
  - 关闭包含嵌入的工业图形的 InTouch 窗口，然后再次打开它。

  在两种情况下，对工业图形的更改会反应到 InTouch 窗口内嵌入的符号中。

在 WindowViewer 中接受符号更改

如果工业图形发生更改，并且您当前正在 WindowViewer 中测试它，则可以在 WindowViewer 中接受更改。

如需有关测试嵌入的工业图形的详细信息，请参阅在 WindowViewer 中测试工业图形。

在测试时在 WindowViewer 中接受符号更改

执行以下操作之一：

- 快速切换到 WindowMaker，然后返回到 WindowViewer。
- 关闭 InTouch 窗口，然后再次打开它。这仅在选择 **WindowViewer 属性对话**框中的**总是从磁盘加载窗口选项时**有用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在两种情况下，对工业图形的更改会反应到 InTouch 窗口内嵌入的工业图形中。

使用 InTouch 标记创建图形元素和工业图形

“工业图形编辑器”的“属性”配置窗格中的“标记”选项卡显示 InTouch 应用程序中可用的所有标记。您只需将标记拖放到画布上即可创建图形元素或工业图形。当一次创建多个符号时，将自动绑定点域属性。此方法简化了图形开发工作流程，并显著地缩短了应用程序开发时间。

注意：此标记选项卡在 CONNECT 和托管 InTouch 的“工业图形编辑器”中不可用。

使用 InTouch 标记创建图形元素

1. 在“工业图形编辑器”中，将标记从**标记**窗格拖放到画布上。

   此时显示**图形元素**和**工业图形选项**。

   注意：您可以将多个标记一起进行拖放，以创建具有相同元素类型和动画的多个图形元素。

2. 要设置缺省值
   - **选择**屏幕上的任意位置。

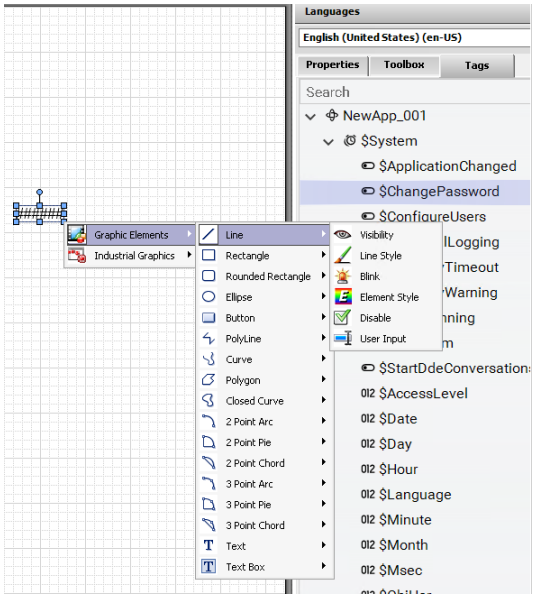
     或

   - 将鼠标悬停在**图形元素选项**上，以**选择**所需的图形元素和动画。

     将按数据类型显示图形元素列表和适用的图形动画。

3. 将鼠标悬停在所需的图形元素上。

   此时显示可用的动画。



4. **选择**所需的动画。

   将创建与标记相关联的新图形元素。

使用 InTouch 标记创建工业图形

1. 在“工业图形编辑器”中，将标记从**标记**窗格拖放到画布上。

此时显示图形元素和工业图形选项。

注意：您可以将多个标记一起进行拖放，以创建相同类型的多个工业图形。

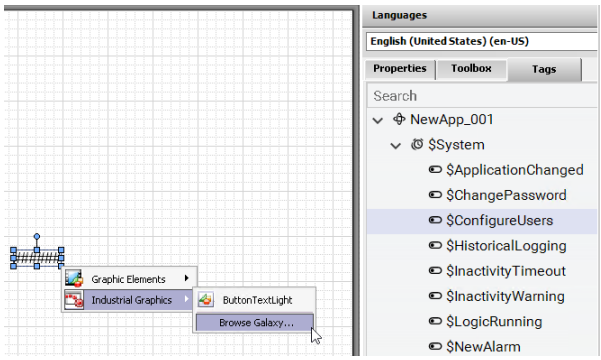
2. 要设置缺省值
   - 选择屏幕上的任意位置。

     或

   - 将鼠标悬停在工业图形选项上以选择所需的工业图形。

     此时显示浏览选项。

     注意：如果多次拖动相同类型的标记并将鼠标悬停在工业图形选项上，那么将显示先前所选的五个图形名称以及浏览 Galaxy 选项。



3. 选择先前所选图形中的任何一个图形，或选择浏览 Galaxy 来选择一个新图形。
4. 如果您选择浏览 Galaxy 选项，请浏览并选择所需的工业图形，然后选择确定。

   此时将创建一个与该标记相关联的新工业图形。

   注意：在您所选的工业图形的自定义属性中，如果可见性是公开，并且：

   - 如果 Value 属性的缺省值为空或 ---，那么它将替换为您拖放的标记的名称。

   - 如果点属性的缺省值为空或 ---，那么它将被标记名.<各自的点属性名> 替换。

   在上述两种情况下，在您选择的工业图形的自定义属性中，如果属性的可见性为私有，那么该属性在您创建的工业图形中将不可见。由于可见性是“私有”，并且它不应该对其它图形可见，因此可见性为私有的自定义属性在其它工业图形中将不可见。

从工业图形编辑器的工具箱选项卡嵌入图形

您可以使用“工业图形编辑器”属性配置窗格中的工具箱选项卡将现有工业图形嵌入到另一个图形中。工具箱选项卡显示 InTouch 库中可用的所有工业图形。将图形嵌入到另一个图形中后，您就可以像编辑图形的任何其它组件一样编辑它。您还可以使用“工业图形编辑器”中的“嵌入工业图形”图标来嵌入图形。如需详细信息，请参阅“工业图形编辑器”帮助中的“嵌入图形”主题。

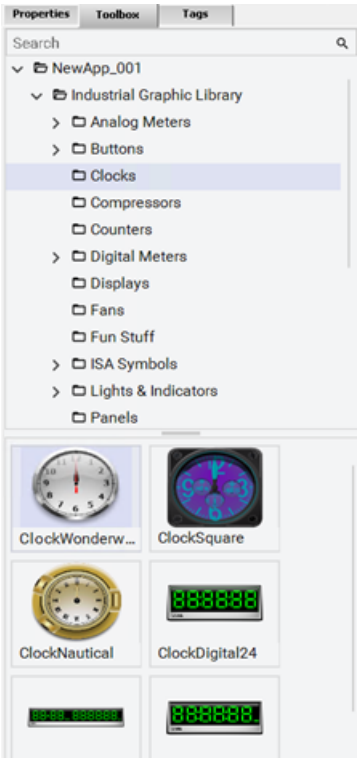
从“工业图形编辑器”的“工具箱”选项卡将现有图形嵌入到另一个图形

1. 在属性配置窗格中，选择工具箱选项卡。
2. 浏览文件夹并选择所需的工业图形，或在搜索框中搜索图形。

   您选择的文件夹或对象中的图形将显示在导航区域下方。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. **选择**要添加的**图**形并将其拖**动**到画布上的所需位置。

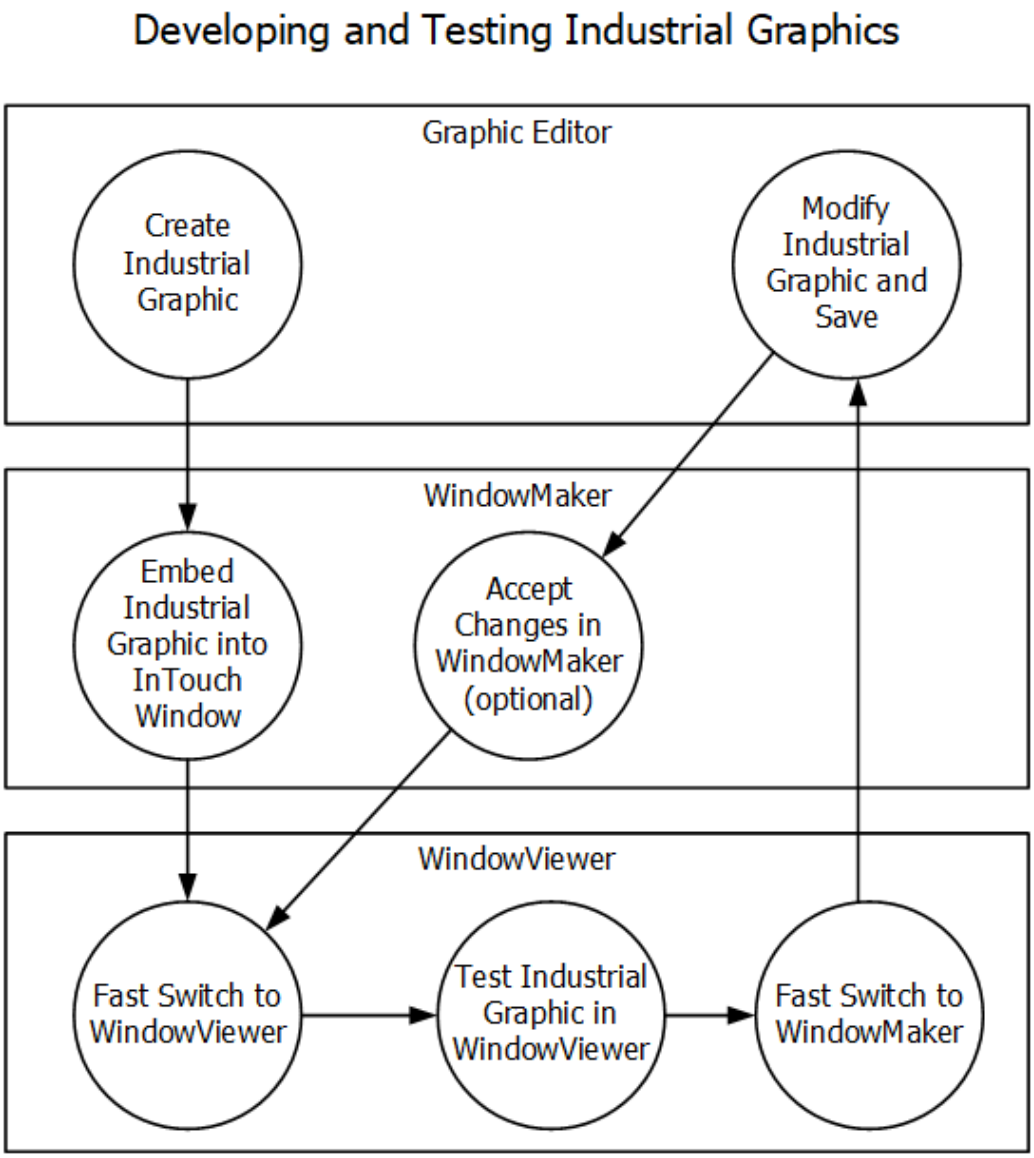


在 WindowViewer 中测试工业图形

您可以在 InTouch 窗口中**测试**嵌入的工**业图**形，而不必衍生 InTouchViewApp **实**例。如果先前**执**行**过**以下操作，**则**可以**测试**嵌入的工**业图**形：

- 在“**图**形工具箱”、自**动**化模板或自**动**化**实**例中**创**建了工**业图**形。
- **创**建了托管的 InTouch **应**用程序。
- 在托管的 InTouch **应**用程序中嵌入了工**业图**形。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



要在 WindowViewer 中测试工业图形

1. 在 WindowMaker 中，选择运行时以切换到 WindowViewer。
2. 像在常规运行时环境中那样，测试嵌入的符号的动画、行为及外观。
3. 您可以快速切换回 WindowMaker 以更改工业图形的嵌入方式。

在 WindowViewer 中更改和测试嵌入的工业图形

1. 在“工业图形编辑器”中更改工业图形。
2. 保存更改。

   如果 WindowViewer 是打开的，则一小段时间之后，WindowViewer 中会出现一条消息，提示您接受更改。选择是。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果 WindowViewer 是关闭的，则您可以从 WindowMaker 快速切换到 WindowViewer 以查看更改。同等待更改传播到打开的 WindowViewer 会话相比，关闭 WindowViewer 然后再重新打开 WindowViewer 会更快。

评估图形性能

您可以在运行时使用图形性能指数 (GPI) 评估工业图形的性能。

在运行时启动您在“工业图形编辑器”中构建的符号时，此工具可计算估计的调用时间。调用时间与用户或系统请求显示相关图形与屏幕上以实时数据显示图形之间的间隔有关。计算基于运行时在 InTouch WindowViewer 中启动的符号的内容，以及具有完整外部引用订阅情况下的最佳情况估计时间。

如需有关使用 GPI 的详细信息，请参阅*工业图形编辑器用户文档*中的“评估图形性能”。

创建新的自动化实例

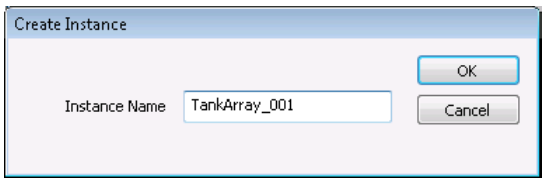
您可以快速创建一个新的 AutomationObject 实例来存放嵌入的工业图形。这样便不必切换到 System Platform IDE 并衍生实例。

新实例并未指定，所以在使用它之前，您需要在 System Platform IDE 中指定并部署它。

您仅能为模板或实例存放的工业图形创建新的自动化实例。“图形工具箱中”的工业图形没有此功能。

创建一个新的自动化实例

1. 使用鼠标右键单击嵌入的工业图形，指向**工业图形**，然后**选择新建实例**。此时出现**创建实例**对话框。



2. 在**实例名**框中，输入实例的名称。
3. **选择确定**。此时会从指定名称的模板中自动衍生一个实例。

自动创建新的 Application Server 对象实例

如果从模板内嵌工业图形，则 InTouch HMI 可以创建该对象的一个实例，并且该符号实例会引用新的实例。

下例显示如何自动创建新的 Application Server 对象实例。

自动创建新的 Application Server 对象实例

1. 创建对象模板 **$Valve1**，并在 IDE 对象编辑器中打开它。
2. 在**图形选项**卡上，添加名为 **ValveSymbol** 的工业图形。
3. 创建 InTouchViewApp 对象的衍生模板，并在 WindowMaker 中打开它。
4. 创建新的 InTouch 窗口，并从对象模板 $Valve1 嵌入工业图形 **ValveSymbol**。WindowMaker 会提示您输入实例名。
5. 输入名称，例如 `Valve1_E122`，然后**选择确定**。此时该工业图形粘贴到 InTouch 窗口上，并且在 Galaxy 中创建了自动化对象实例 Valve1_E122。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将 InTouch 窗口转换为工业图形

您可以将托管的 InTouch 应用程序的窗口转换为工业图形。转换后的工业图形会出现在 WindowMaker 的“图形工具箱”和“IDE 图形工具箱”中。除了窗口中显示的图形之外，InTouch 脚本还会转换为 Application Server 脚本。

准备转换窗口

在转换 InTouch 窗口之前：

- 只有 InTouch 独立和托管应用程序中的窗口能转换为工业图形。
- 必须在 WindowMaker 中关闭要转换的窗口。

窗口转换规则和例外

只能转换窗口的符号和脚本。窗口的颜色、类型、框架、标题栏、大小控件和关闭按钮将从转换后的符号中排除。

根据 InTouch 图形类型，窗口图形的转换方式如下：

- 所有 InTouch 图形基元均转换为相应的工业图形基元。
- InTouch SmartSymbol 转换为工业内嵌图形。
- 窗口内的工业图形转换为内嵌符号。不会为内嵌符号创建任何新符号。
- InTouch 符号转换为属性 TreatAsIcon 为 True 的组。
- InTouch 单元转换为属性 TreatAsIcon 为 False 的组。
- InTouch 窗口控件转换为 ArchestrA 窗口控件。
- 某些 Intouch 图形组件无法转换为工业图形：
  - InTouch 实时和历史趋势无法转换。
  - ActiveX 控件和“分布式报警显示”无法转换。

转换动画脚本

窗口中嵌入的所有 InTouch 动画链接均会在 Application Server 中转换为相应的动画。

转换期间不会记录任何验证警告或错误消息。您应该使用工业图形脚本验证过程来验证转换后的脚本，以查找任何不支持的脚本语法。

转换 InTouch 动画脚本时，会出现以下例外：

- “线条颜色”和“填充颜色”动画链接的“离散报警”和“模拟报警”转换为“线条样式”和“填充风格”动画的“布尔值”和“真值表”。
- ShowWindow 动画链接转换为带 ShowGraphic 脚本函数的动作脚本。HideWindow 动画链接转换为带 HideGraphic 脚本函数的动作脚本。
- 动画链接表达式中配置的所有 Intouch 标记的标记名称都添加前缀“InTouch”。例如，Tag1 转换为 InTouch:Tag1。
- 动画链接中配置的 Application Server 属性引用的“galaxy:”前缀将被删除。例如, galaxy:UD001.Value 转换为 UD001.Value。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 动作脚本中配置的所有 InTouch 脚本函数都不会转换。除了 InTouch 标记和 Application Server 属性引用处理之外，所有脚本都将被复制。

窗口转换的已知限制

将 InTouch 窗口转换为工业图形并不总是完全准确。<1>在某些情况下</1><2>, </2><3>窗口</3><4>组件无法转换为符号</4><5>。</5> 该节介绍了将 InTouch 窗口转换为工业图形时的已知限制以及任何替代解决方案。

- 转换包含垂直鼠标放开游标 SmartSymbol 的窗口

  <1>垂直鼠标放开游标 SmartSymbol 包含两种类型的动画</1><2>。</2> <1>符号使用填充动画来显示刻度和可移动游标旋钮的当前测量值来设置值</1><2>。</2> 当包含嵌入的垂直鼠标放开游标 SmartSymbol 的窗口转换为工业图形时，不会保留可移动游标动画。

- 转换包含 Symbol Factory 符号的窗口

  并非 Symbol Factory 符号中包含的所有类型的动画都能转换为工业图形。<1>无法转换下列类型的 Symbol Factory 符号动画</1><2>：</2>

  - 填充百分比
  - 线条颜色
  - 水平移动
  - 垂直移动

- 转换包含 InTouch ActiveX 控件、InTouch OCX DLL 或 InTouch DLL 对象的窗口

  “窗口转换”不支持 InTouch ActiveX 控件、InTouch OCX DLL 和 InTouch DLL 对象。这些组件将不包含在新符号中。

- 将 InTouch 翻译字符串迁移到工业图形

  InTouch 翻译字符串存储在应用程序文件夹内的 XML 文件中。每种语言的翻译字符串放置在单独的 XML 文件中。导入本地化字符串后，可以从应用程序目录中的 XML 文件获得所有窗口和 SmartSymbol 字符串的翻译。

  以下步骤说明了如何使用语言助理工具将 InTouch 窗口本地化字符串迁移到工业图形。

  a. 将 InTouch 窗口转换为工业图形。
  b. 将 InTouch 语言 XML 文件的内容导入“语言助理”以创建全局短语字典。
  c. 将工业图形导出到 XML 文件。
  d. 将符号 XML 文件导入“语言助理”，后者会自动将全局字典翻译应用到工业图形短语。
  e. 从“语言助理”发布工业图形 XML 文件。
  f. 将 XML 文件导入包含翻译文本字符串的“工业图形工具箱”。

- 将 InTouch 脚本迁移到工业图形

  包含 InTouch 或 QuickScript 函数的脚本无法转换为工业图形。

- 将 InTouch 历史对象迁移到工业图形

  包含 InTouch 历史趋势的 InTouch 窗口不会完全转换为工业图形。<1>只有某些部分的历史趋势可能出现在转换后的符号中</1><2>。</2> 指示器条之类的趋势组件可能不会出现在转换后的工业图形中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

转换窗口之后

转换后，符号将添加到 System Platform IDE 工具箱和 WindowMaker 工业图形工具箱。原始的转换后的 InTouch 窗口将备份。InTouch 应用程序中将创建一个新的 InTouch 窗口，其中嵌入了新创建的工业图形。

如果只转换一部分窗口，则用户必须手动验证转换后的窗口是否与未转换的窗口一起工作。

转换后的窗口的名称缺省指定为新工业图形的名称。如果 InTouch 窗口名包含不支持的字符，将用下划线 (_) 替换每个不支持的字符。如果符号已存在，将向转换后的符号的名称附加数字后缀。例如，Main_001。

美元符号 ($)、井号 (#) 和下划线 (_) 这几个特殊字符是唯一的例外。

如果迁移某个应用程序，那么在编辑窗口或修改任何窗口属性时，将会用下划线 (_) 替换窗口名中的任何特殊字符。这适用于所有类型的窗口，包括应用程序、框架和模板。

对于所有转换的符号，将创建一个新工具集，并为其指定 InTouch 应用程序名称。窗口转换后，该工具集将保持 InTouch 文件夹层次结构。

- 转换自未指定的 InTouch 窗口的符号将添加到被指定 InTouch 应用程序名称的工具集中。
- 如果窗口属于已指定的区域，将在两个工具箱中均创建原始文件夹结构，并指定 InTouch 应用程序的名称为顶级根文件夹名称。

窗口转换过程

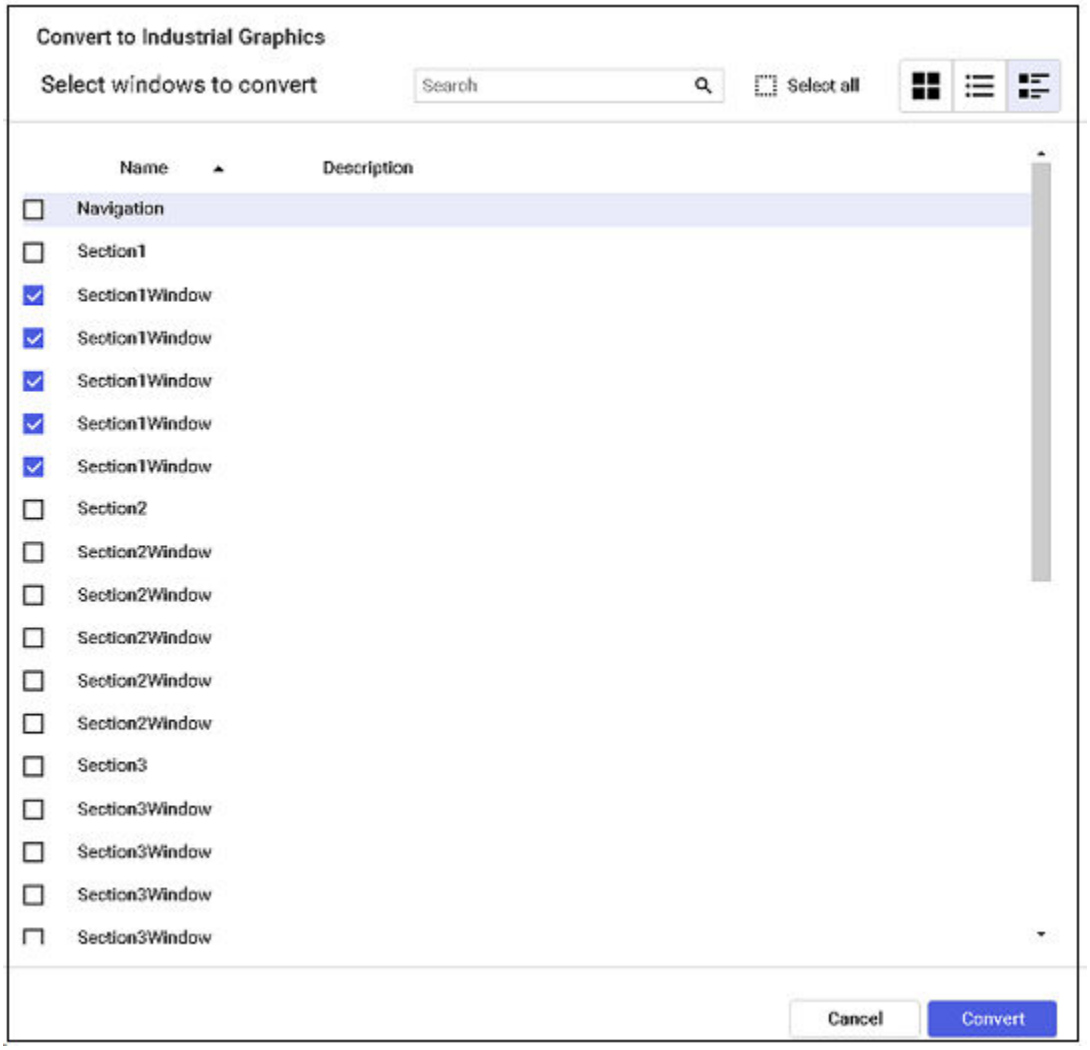
将窗口转换为工业图形

1. 关闭所有将要转换为符号的 InTouch 窗口。
2. 从窗口窗格中选择要转换的窗口。
3. 使用鼠标右键单击以显示快捷菜单，然后选择**转换为工业图形...**。

   此时出现**选择要转换的窗口**屏幕。
4. 选择要转换的窗口。

   缺省条件下，会检查在前一个屏幕中选择的窗口。现在可以选择其它窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 选择**转换**。

   随后出现一条消息，指示正在连续转换窗口。窗口转换完毕后，将显示一连串签入对话框，可在这些对话框中为每个转换的窗口输入可选注释。

6. 观察 WindowMaker 工业图形工具箱和 System Platform IDE 图形工具箱。

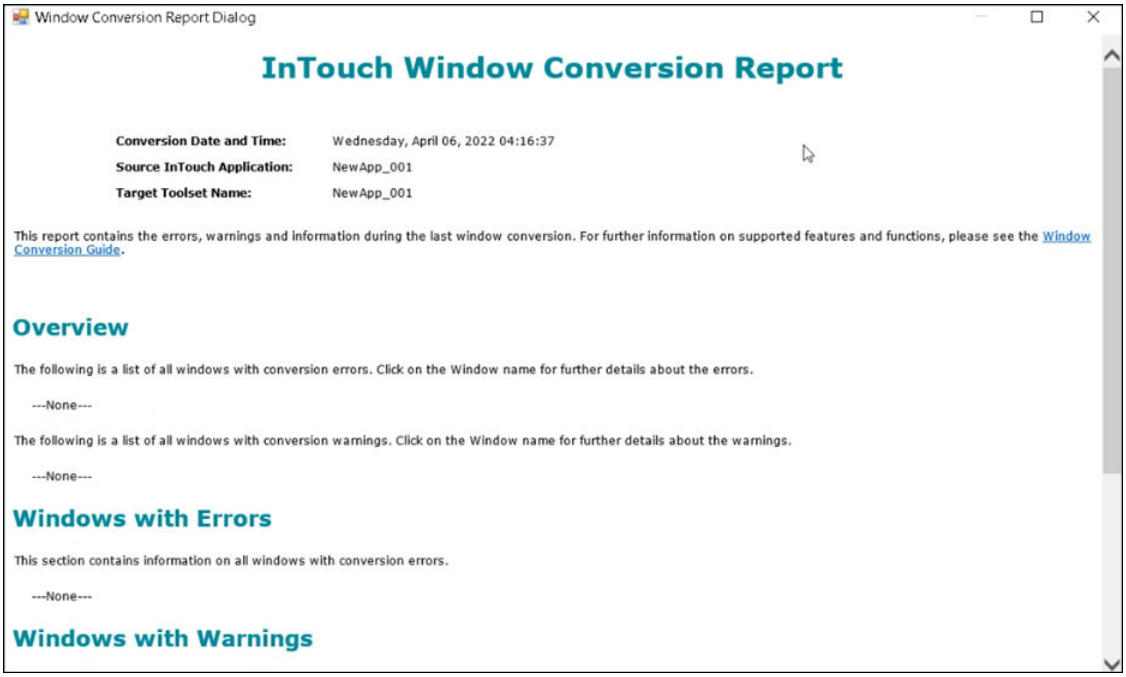
   转换后的窗口应显示为两个工具箱中的工业图形。

诊断窗口转换错误

转换期间，一个进度条将显示在窗口转换期间出现的警告或错误消息。转换窗口时，将创建一个自定义日志 标帜来包含每个元素、动画或脚本的有用诊断信息。

转换后，在转换进度条中选择窗口转换报告以将所有消息导出到 HTML 转换报告，可在消息窗口中查看该报告。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

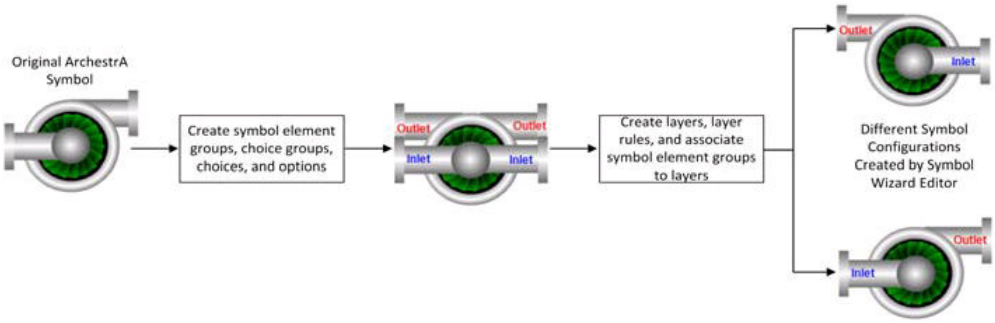


转换进度对话框和转换报告均包含以下与窗口转换有关的信息：

- 窗口名
- 用于指示窗口转换成功还是发生错误的转换状态。
- 转换期间出现的警告消息。
- 窗口转换期间出现的错误消息。

使用符号向导编辑器创建符号向导

“工业图形编辑器”包含“符号向导编辑器”，可用于创建符号的不同视觉和功能配置。这些多配置符号称为“符号向导”。创建的“符号向导”的一个示例是泵符号，该符号显示了中央叶片外壳左侧或右侧的出口管。使用“符号向导编辑器”可将两个配置包含在一个“符号向导”中。



“符号向导”不与任何特定的 Application Server 对象模板或 Application Server 对象实例关联。除了能够选择特定的符号配置之外，“符号向导”的行为与标准工业图形的行为类似。

创建具有不同配置的“符号向导”的能力可减少需要为应用程序创建的符号数量。同时，符号减少也会使维护和设计时间问题变少。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建符号向导

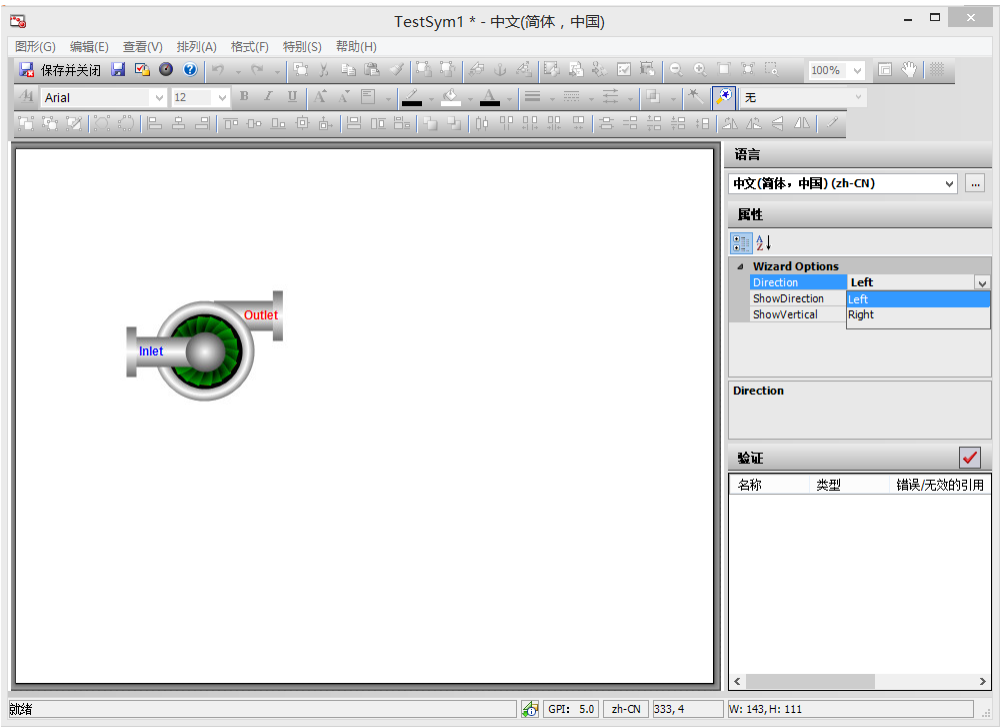
创建和实施“符号向导”的过程有两个工作流程，即“设计者”工作流程和“使用者”工作流程：

- 设计者使用“符号向导编辑器”创建多个符号配置。
- 使用者选择“符号向导”的适当配置并将该符号嵌入托管的 InTouch 应用程序中。

符号向导设计者工作流程

设计者使用“符号向导编辑器”定义各种符号配置并将图形元素、自定义属性和脚本以层的形式指定给符号配置。为规则指定了“选择组”、“选择”和“选项”，以确定在符号配置中包括层的时间。设计者选择将成为在托管的 InTouch 应用程序中嵌入符号时显示的缺省符号的配置。

创建所有符号配置后，设计者会验证符号的各个配置将如何提交到使用“符号向导预览”的使用者。设计者使用向导选项验证各个配置是否按照设计基于为符号设置的层规则进行显示。



完成“符号向导”后，设计者将其保存到 Galaxy 库，使其可嵌入在托管的 InTouch 应用程序中。

如需有关创建“符号向导”的详细信息，请参阅《工业图形编辑器用户》文档。

符号向导使用者工作流程

当使用者将“符号向导”的实例嵌入托管的 InTouch 应用程序时，将选择该符号的缺省配置。使用者可通过从“符号向导”的向导选项视图中选择选项来更改配置。根据所选配置，使用者可选择额外的与配置相关的属性。

当 InTouch 应用程序正在运行时，“符号向导”显示为使用者选择的配置。运行时不能更改符号的配置。

如需有关如何将“符号向导”嵌入到托管的 InTouch 应用程序中的详细信息，请参阅嵌入符号向导。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

了解趋势笔历史数据检索

当包含趋势笔的应用程序开始运行时，历史查询会检索整个趋势笔时间段的数据。在检索历史数据的时间段内，会将实时数据绘制在趋势线上。在检索历史数据之后，会将其添加到实时数据中以填回整个时间段的趋势线。

以下过程显示了配置趋势笔的步骤。配置趋势笔通常包括几个步骤，以将趋势笔放置在仪表图形旁边以便直观地指示“趋势笔”图表显示图形的过程值经过一段时间的更改情况。

要配置趋势笔

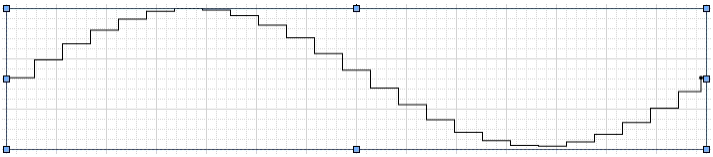
1. 从“可视化”文件夹中选择趋势笔。

   将光标置于“工业图形编辑器”画布区域上方时，光标会变成十字形。

2. 将光标置于画布上您要设置“趋势笔”图表水平边界的位置上方。

3. 在画布上，通过拖动为趋势笔控件选择长方形。

   趋势笔长方形的水平和垂直边界表示在运行时期间绘制“趋势笔”图表的区域。图形长方形的水平轴表示趋势的时间段。垂直轴表示趋势可能值的范围。

   此时出现趋势笔对话框。也可以通过双击趋势笔图形或从特别菜单中选择编辑动画 来显示趋势笔对话框。

4. 在引用字段中输入引用。

   引用是显示为趋势所显示的值的数据源，它可以是诸如对象的属性、模拟标记值或自定义属性等外部引用。但不允许使用常量和表达式。

5. 选择 **Historian** 或 **InTouch** 日志历史/**LGH** 作为历史来源。如果选择 Historian，则继续步骤 7。

6. 如果选择 **InTouch** 日志历史/**LGH**，则可以使用对应 **UNC** 路径的图标将该字段的输入模式切换为表达式或静态文本。

   - 如果选择表达式模式，可以选择省略号按钮来启动 HMI 的属性/标记浏览器并选择自定义属性、特性或标记。
   - 如果选择静态文本模式，可以选择省略号按钮来启动文件浏览器，在其中可以指定 LGH 文件名。

7. 对于识别历史位置的方法，请选择自动检测或表达式。

   - 自动检测：Historian 服务器是从运行引用属性的 AppEngine 上自动检测的。例如，如果将引用字段设置为 UDO.UDA1, 那么自动检测字段会设置为针对运行 UDO 的 AppEngine 所配置的 Historian 服务器名称。自动检测仅对“Application Server 引用”有效。
   - 表达式：当在服务器名称字段中输入表达式或引用时，趋势笔会连接到指定的历史服务器。

   服务器名字段左侧的图标会将该字段的输入模式切换为表达式或静态文本。

   如果在表达式模式下将服务器名称字段留空，则趋势笔将只显示实时数据。

8. 选择移动或固定作为趋势时间段的类型。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- **移动**：趋势时间段的开始时间是当前时间，结束时间是从开始时间算起的该时间段的持续时间。下一个时间段的开始时间被设为上一个趋势时间段的结束时间。
- **固定**：在固定趋势时间段中，StartTime 和 EndTime 属性不会自动更改。趋势时间段的开始时间最初为当前时间。StartTime 属性可由脚本更改。

趋势时间段（移动和固定）的 EndTime 属性为只读。趋势时间段的结束时间通过指定的开始时间和时间段的持续时间进行计算。

9. 在**持续时间（分钟）**字段中设置趋势线的 X 轴时间段。

   可以将趋势时间段指定为常量、外部引用、表达式或自定义属性。如果输入的是浮点数，则会将时间段四舍五入为最接近的分钟值。

   最小趋势时间段为 1 分钟，最大时间段为 10080 分钟（1 周）。

10. 对于在趋势线上放置过程值的缩放方法，请选择**自动范围** 或**剪切超出范围的值**。

    如果选择了**自动范围**，则会禁用**最小范围**和**最大范围**字段。趋势线的 Y 轴会自动调整以在趋势笔图形的上下边界之间显示完整的趋势值范围。

    如果选择了**剪切超出范围的值**，则会启用**最小范围**和**最大范围**字段。**最小范围**和**最大范围**设置趋势 Y 轴值范围的上下限。这两个字段都可以设置为常量、外部引用或自定义属性。

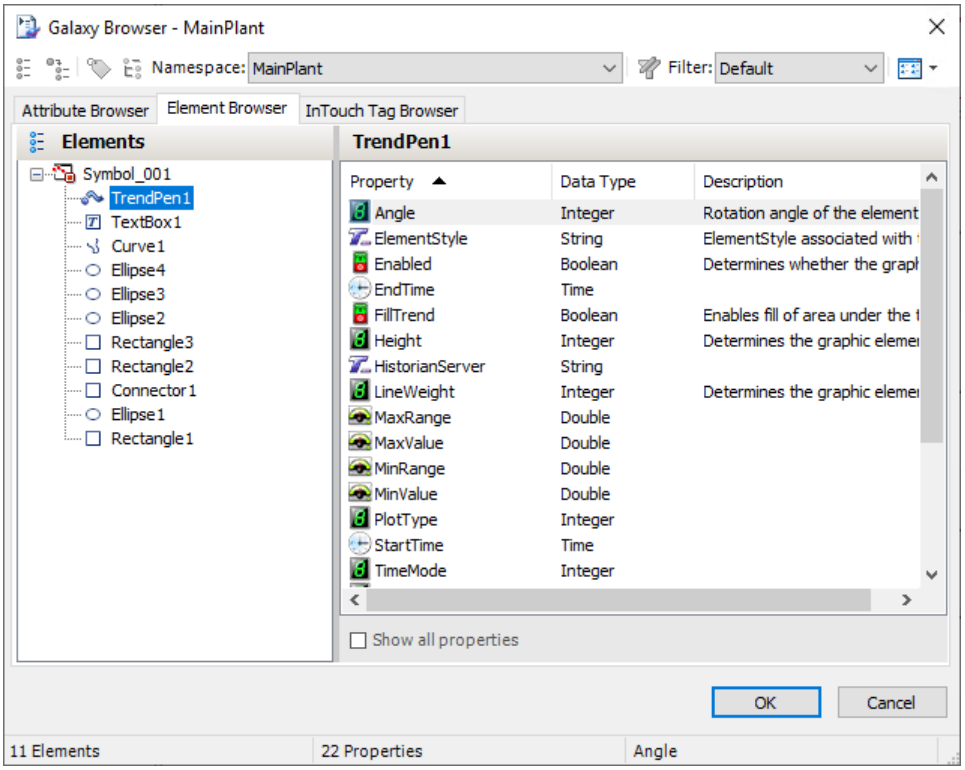
    当使用**剪切超出范围的值**并且值超出趋势的最小或最大限制时，趋势线会在值范围限制的位置被截断，并将过程值超出趋势值范围的时间段显示为水平线。

11. 从**绘图类型**中，选择**阶梯线**或**直线**作为趋势图的类型。
    - **阶梯线** 图表会沿趋势的 X 轴方向从一个趋势数据点到下一个数据点的时间之间画一条水平线，然后再画一条到该数据点的垂直线。
    - **直线**图表会在趋势时间段内的每个连续点之间画一条直接相连的线。

在运行时期间更改趋势笔属性

“趋势笔”包含一组属性，可在运行时期间对这些属性的值进行修改以更改趋势的可视特征和功能特征。下图是 HMI 软件中通常提供的浏览器或资源管理器示例，用于搜索其值可在运行时更改的属性或标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



下面的副主题描述了在运行时期间通常会进行修改，以更改趋势笔的可视特征和功能特征的属性。

MinValue 属性

在运行时期间，可以修改 MinValue 属性的值以更改趋势所显示的最小度量值。在运行时期间，根据在设计时期间为趋势笔的 **Y 轴范围选项**所指定的值，MinValue 既可是读/写属性，也可是只读属性。

- 当在设计时期间将 **Y 轴范围**设置为自动范围时

  趋势所显示的最小度量值会设置为在趋势时间段内从“历史数据”或从当前数据接收到的数据的最小值。MinValue 为只读。

- 当在设计时期间将 **Y 轴范围**设置为剪切超出范围的值时。

  趋势所显示的最小度量值会设置为在设计时期间来自**最小范围选项**的最小限制集。

  MinValue 为读/写属性，可在运行时期间进行更改。当在运行时期间更改 MinValue 时，会根据指定给 MinValue 和 MaxValue 属性的值来重新绘制趋势线。

  如果指定给 MinValue 和 MaxValue 属性的值相同，则趋势的 Y 轴范围会自动更改为“自动范围”。

MaxValue 属性

在运行时期间，可以修改 MaxValue 属性的值以更改趋势所显示的最大度量值。在运行时期间，根据在设计时期间为趋势笔的 Y 轴范围选项所指定的值，MaxValue 既可是读/写属性，也可是只读属性。

- 当在设计时期间将 **Y 轴范围**设置为自动范围时

  趋势所显示的最大度量值会设置为在趋势时间段内从“历史数据”或从当前数据接收到的数据的最大值。MaxValue 为只读。

- 当在设计时期间将 **Y 轴范围**设置为剪切超出范围的值时

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

趋势所显示的最大度量值会设置为在设计时期间来自最大范围选项的最大限制集。

MaxValue 为读/写属性，可在运行时期间进行更改。当在运行时期间更改 MaxValue 时，会根据指定给 MaxValue 和 MaxValue 属性的值来重新绘制趋势线。

如果指定给 MinValue 和 MaxValue 属性的值相同，则趋势的 Y 轴范围会自动更改为“自动范围”。

StartTime 属性

在运行时期间，可以修改 StartTime 属性的值以根据在设计时期间为趋势笔时间段设置的值来更改趋势时间段的开始时间。

- 当在设计时期间将时间段设置为固定时

  指定给 StartTime 属性的缺省值是趋势笔第一次在 WindowViewer 中出现的时间，开始时间也会随时间的经过而发生变化。StartTime 为读/写属性，可在运行时期间进行更改。当 StartTime 的值更改时，趋势笔会使用新的 StartTime 值重新绘制趋势。

- 当在设计时期间将时间段设置为移动时

  为 StartTime 属性设定的值是系统的当前日期和时间。StartTime 为只读。

EndTime 属性

在运行时期间，可以修改 EndTime 属性的值以根据在设计时期间为趋势笔时间段设置的值来更改趋势时间段的结束时间。

- 当在设计时期间将时间段设置为固定时

  指定给 EndTime 属性的缺省值是在设计时期间设置的结束时间。EndTime 为读/写属性，可在运行时期间进行更改。当 EndTime 的值更改时，趋势笔会使用新的 EndTime 值重新绘制趋势。

- 当在设计时期间将时间段设置为移动时

  为 EndTime 属性设定的值是系统的当前日期和时间。EndTime 为只读。

PlotType 属性

在运行时期间，可以修改 **PlotType** 属性的值以更改趋势图的类型。

- 当 **PlotType** 为 0 时，“趋势笔”图表类型为“阶梯线”。缺省值。
- 当 **PlotType** 为 1 时，“趋势笔”图表类型为“直线”。

如果 **PlotType** 的值既不是 0 也不是 1，则将忽略该值。

当 **PlotType** 的值在运行时更改时，将在绘制趋势之前再次检索趋势数据。

TimeMode 属性

在运行时期间，可以修改 **TimeMode** 属性的值以更改趋势时间段的类型。

- 当 **TimeMode** 为 0 时，趋势时间段模式为移动。趋势的结束时间为当前时间。缺省值。
- 当 **TimeMode** 为 1 时，趋势时间段模式为固定。趋势的开始时间是趋势笔第一次在 WindowViewer 中出现的时间，开始时间也会随时间的经过而发生变化。

如果 **TimeMode** 的值既不是 0 也不是 1，则将忽略该值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

当时间段在运行时期间从“移动”更改为“固定”时，趋势的开始时间和结束时间会在切换之前保持不变，并且数据也会保留。当时间段在运行时期间从“固定”更改为“移动”时，将在绘制趋势之前再次检索数据。“移动”模式会自动调整趋势的开始和结束时间。

FillTrend 属性

在运行时，可以使用 FillTrend 属性更改趋势笔曲线下方区域的外观。

还可以使用脚本修改 FillTrend 属性。例如 `TrendPen6.FillTrend = not TrendPen6.FillTrend;`

根据常规元素样式优先规则，填充样式动画、元素样式动画或元素样式中的 Fill 属性现在将应用于 FillTrend 属性。

如需有关填充样式动画的详细信息，请参阅设置填充样式。

如需有关元素样式动画的详细信息，请参阅使用元素样式配置动画。

注意：FillTrend 属性可以在设计期间或运行时启用。在运行时，颜色缺省为白色（如果在设计时未选择颜色）。

配置多笔趋势

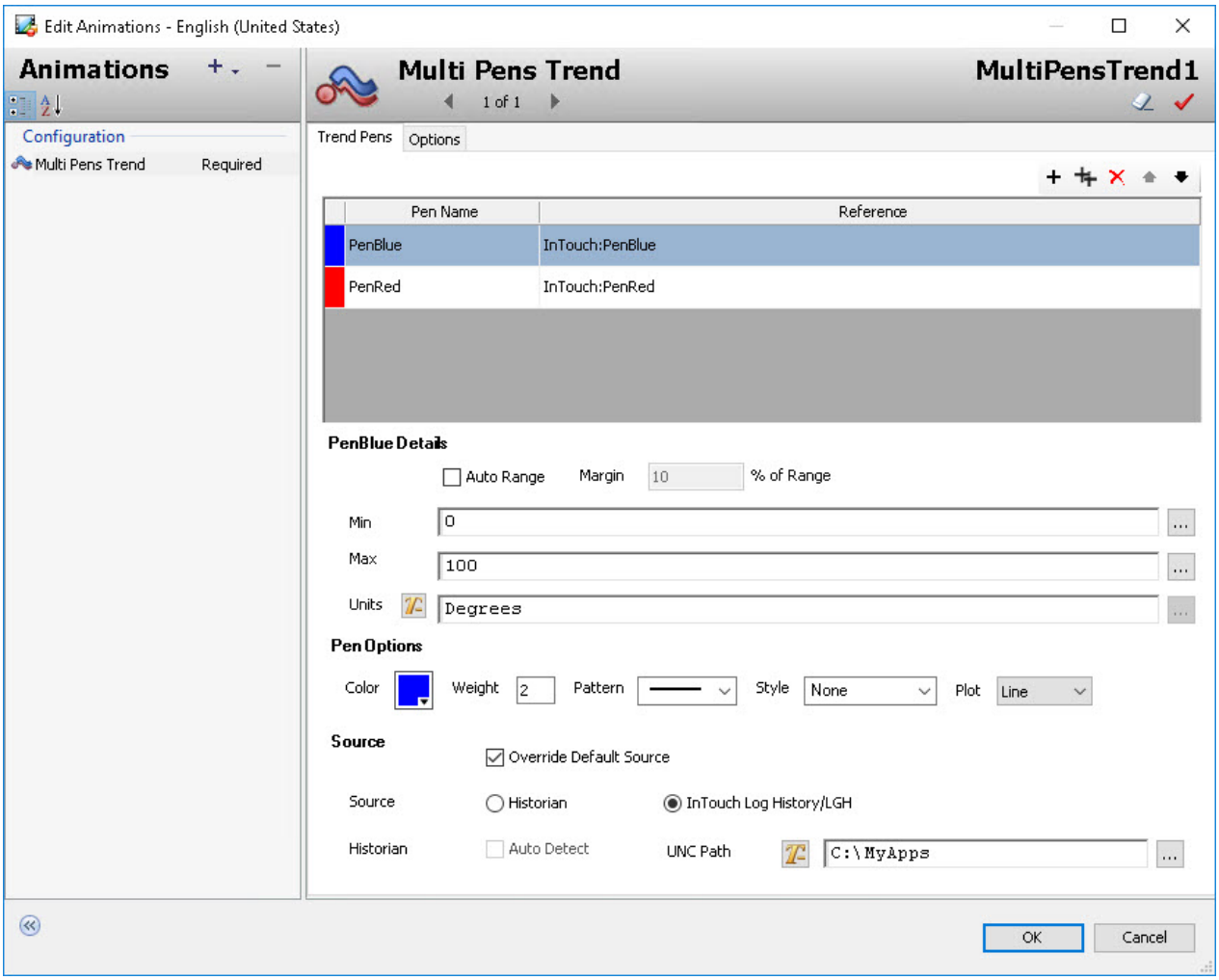
多笔趋势类似于趋势笔，其会显示一连串过程值，但会包含多个趋势线。如需有关笔趋势类型的详细信息，请参阅配置趋势笔下的相关部分。您可以使用工业图形程序 API 导出和导入嵌入了多线趋向图的符号。有关如何导出和导入图形元素的更多信息，请参阅“AVEVA 工业图形编辑器帮助”中的“以编程方式导出和导入工业图形”。

注意：
- 多笔趋势尚未在非英语操作系统上进行测试。
- GPI 中不包括多笔趋势的性能指标计算。
- AVEVA OMI ViewApp 不支持多笔趋势。
- Web 客户端不支持多笔趋势。

要配置多笔趋势：

1. 选择多笔趋势元素。
2. 单击“图形编辑器”画布。
3. 此时出现编辑动画对话框。对于每支笔，配置趋势笔选项卡上的以下部分：
   - 笔名和引用
   - 笔详细信息
   - 笔选项
   - 源

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



配置笔名和引用

在“趋势笔”选项卡上，对于每支笔：

1. 在笔名字段中，输入笔的名称。

   笔名字段必须满足以下规则：

   - 只允许静态文本字符串。
   - 必须包含一个字母。
   - 最多 32 个字符。
   - 有效字符为字母数字和特殊字符（$、#、_）。
   - 两支趋势笔不能同名。
   - 笔名不区分大小写。

   工具提示会显示所配置的笔名。

2. 在引用字段中，输入引用。使用该字段旁边的省略号按钮可选择引用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

引用是显示为趋势所显示的值的数据源，它可以是诸如对象的属性、InTouch 模拟标记值或自定义属性等外部引用。但不允许使用常量和表达式。如需详细信息，请参阅设置输入模式 ()。

3. 缺省条件下，会在网格中创建两个空行，最多可以创建 8 支笔。
4. 使用 + 图标可添加更多趋势笔。使用 ✕ 图标可删除任何趋势笔。使用 ▲ 和 ▼ 箭头可更改笔的显示顺序。使用 ✚ 图标可复制现有笔配置。

配置笔详细信息

在“笔详细信息”部分下，可以配置每支笔的范围和单位。

1. 选择自动范围缩放方法，将过程值放置在趋势线上。

   选择自动范围会禁用最小值和最大值字段。趋势线的 Y 轴会自动调整，以显示趋势上下边界内的趋势值的完整范围。

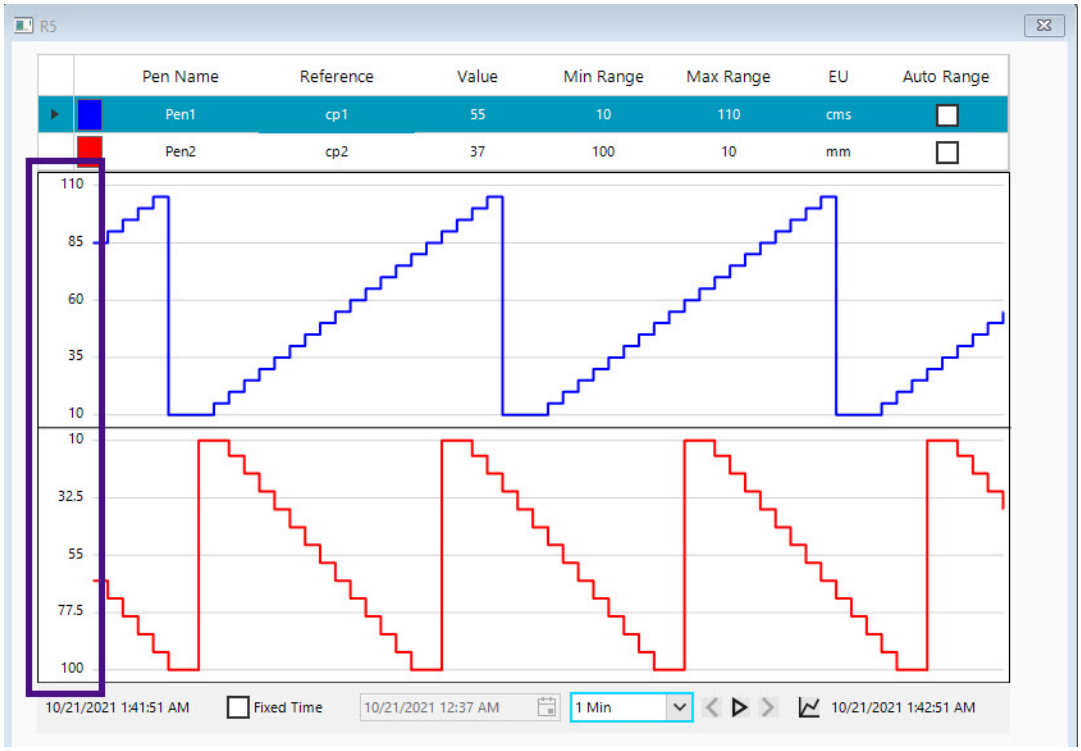
   a. 指定边距百分比值。在运行时，趋势区域的边距将基于趋势值和指定的边距百分比动态显示。这仅用于显示目的，而不会影响轴的值。

      例如：如果最小值为 10，最大值为 20，范围为 10 (20-10)。
      如果指定 20% 的边距，则边距的计算方法为：范围*边距百分比；10*20%=2。
      Y 轴的最大值将为最大值 + 边距；20+2=22。Y 轴的最小值将为最小值 - 边距；10-2=8。

2. 如果未选择自动范围，请为所选笔的 Y 轴指定最小值和最大值。

   您可以直接指定值，或单击 ...。此时显示“Galaxy 浏览器”。

   如果“最小值”大于“最大值”，则在运行时，趋势笔将以不同方式进行呈现，并且比例与“最大值”大于“最小值”时相反。



3. 浏览并选择一个属性或标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 输入单位。使用“切换”按钮输入静态文本，或从“Galaxy 浏览器”中选择引用值。例如：度数。
5. 单击确定。

配置笔选项

在“笔选项”下，可以配置每支笔的样式。

1. 颜色：选择笔的显示颜色。
   a. 您可以使用颜色选择器从屏幕的一部分选择一种颜色。
   b. 还可以单击加载调色板... 以使用“加载自定义调色板”。
   c. 还可以按 + 向自定义调色板添加颜色，按“删除”按钮来删除颜色。
   d. 开发自定义调色板后，可以将其保存以供将来使用。单击保存调色板。
2. 磅数：输入笔的磅数。磅数决定了趋势笔线条的粗细。
3. 图案：从预定义选项列表中选择趋势笔线条的显示图案。
4. 样式:您可以从预定义样式选项列表中进行选择。样式在 IDE 中的“元素样式”下进行配置。如需详细信息，请参阅更改元素样式的视觉属性。
5. 绘图：要显示的趋势线的类型。选择“阶梯线”或“直线”。
6. 单击确定。

配置源

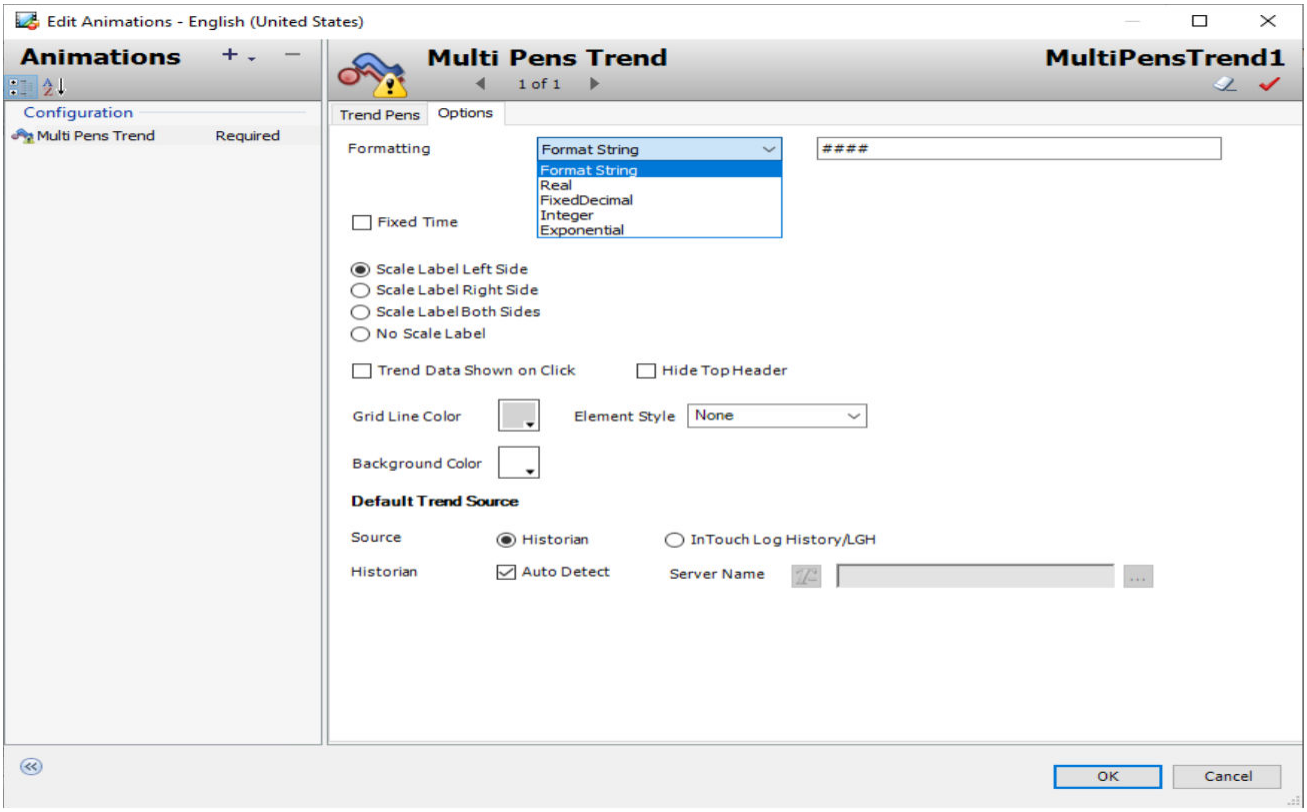
您可以覆盖选项选项卡上配置的历史数据来源。如需详细信息，请参阅自定义多笔趋势 ()。

- 选择覆盖缺省源以更改历史数据来源。

自定义多笔趋势

您可以使用选项选项卡来自定义运行时多笔趋势的显示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



1. 使用**格式设置选项**可配置以下**值**的数字格式：
   - **值**
   - 最小范围
   - 最大范围
   - 光标读出中的**值**

   默认选择是**格式**字符串。用户可以使用 0、#、逗号和小数点来设置上述**值**的格式。

   其他**选择**（包括 Real、FixedDecimal、Integer 和 Exponential）可通过 **IDE** > **Galaxy** >**配置** > **样式**进行配置。上述**值**将按照所**选样**式设置格式。
2. **固定时间**设置可以在“固定时间模式”（选中时）与“移动模式”（未选中时）之间切换。选中“固定时间”复选框时，趋势区域会显示特定持续时间的趋势数据，而不会移动。
3. 从下拉列表中的预定义选项列表中选择“时间间隔”。在此处选择的时间间隔将设置为运行时的初始值。
4. 选择应该在趋势区域的何处显示比例标签；左侧、右侧、左右两侧，或不显示比例标签。
5. 单击**单击时显示的趋势数据**以自定义“光标读出”对话框在运行时的行为。

   缺省条件下，当光标悬停在趋势区域上时，将显示“光标读出”对话框。选择此选项时，可以单击趋势区域来显示与隐藏“光标读出”对话框。
6. 要隐藏运行时中的顶部标题，请选择**隐藏顶部标题**。将仅显示趋势区域和工具栏。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

7. 要根据当前值呈现并绘制任何数据质量的趋势，请选中呈现任何质量的数据复选框。对于质量非良好的数据，将用虚线表示趋势。如果未选中此复选框，将按照最后一个质量为良好的值将趋势绘制为一条直线。
8. 使用网格线颜色选项选择所需的网格线颜色。
9. 从元素样式下拉列表中选择预定义的样式。
10. 还可以使用背景颜色选项来选择背景的颜色。
11. 在缺省趋势源下的源字段中，选择“Historian”或“InTouch 日志历史/LGH”。

    缺省条件下，会选择“Historian”源和“自动检测”选项。选中自动检测会禁用“服务器名称”。

    在“服务器名称”字段中输入表达式或引用时，多笔趋势会连接到指定的 Historian 服务器。服务器名称字段左侧的按钮可将该字段的输入模式切换为表达式或静态文本模式。

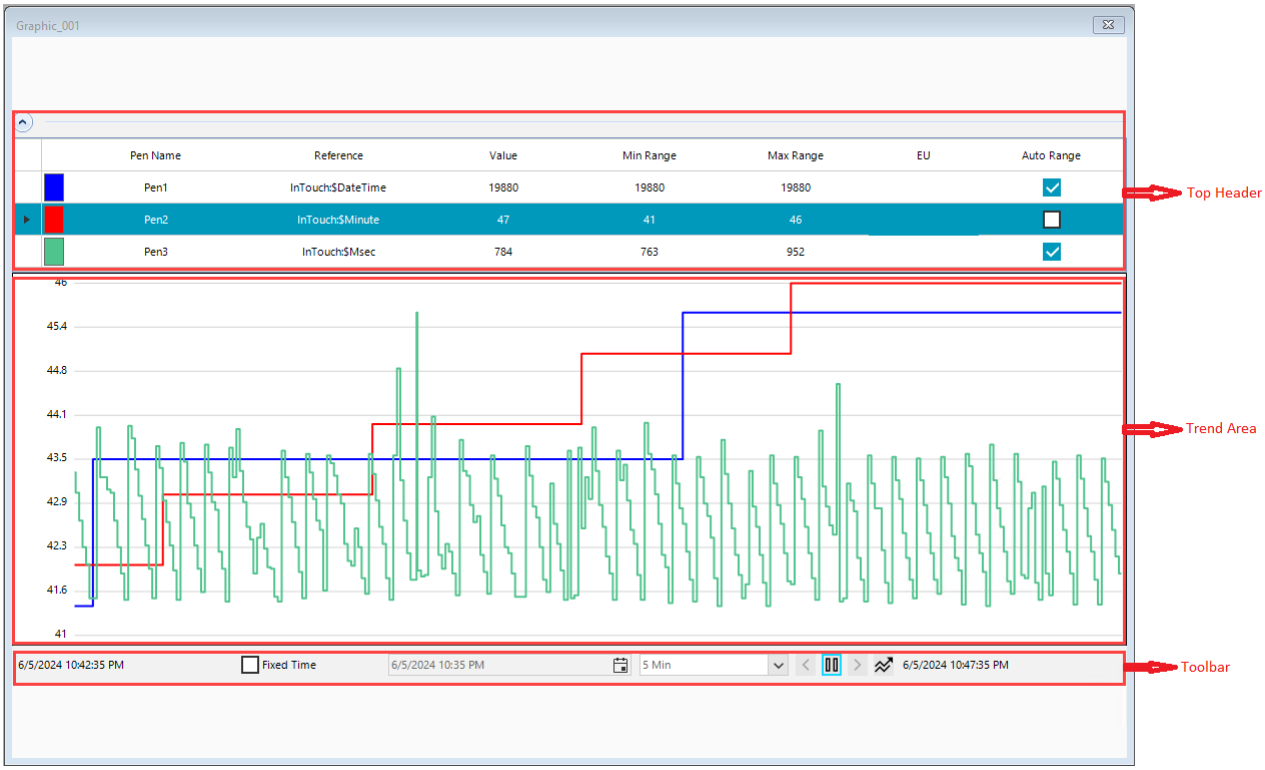
    Historian 服务器是从运行引用属性的 AppEngine 上自动检测的。例如，如果将“引用”字段设置为“UDO.UDA1”，那么“自动检测”字段会设置为针对运行 UDO 的 AppEngine 所配置的 Historian 服务器名称。“自动检测”仅对 Application Server 引用有效。如果在表达式模式下将“服务器名称”字段留空，则趋势笔将仅显示实时数据。
12. 如果选择 **InTouch** 日志历史**/LGH**，请使用 **UNC** 路径字段旁边的按钮将该字段的输入模式切换为表达式或静态文本。
    a. 如果选择表达式模式，可以单击省略号按钮来启动 HMI 的属性/标记浏览器并选择自定义属性、特性或标记。
    b. 如果选择静态文本模式，可以单击省略号按钮来启动文件浏览器，在其中可以指定 LGH 文件位置。

在运行时使用多笔趋势

您可以在 WindowViewer 中更改多笔趋势在运行时的行为。例如，添加笔，删除笔或播放历史趋势数据。多笔趋势可以按 3 个区域进行组织：

- 顶部标题：以表格式列出笔详细信息
- 趋势区域：此区域以直观方式呈现笔
- 工具栏：包含播放和堆叠选项

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



光标读出对话框

当光标位于笔上的特定点时，“光标读出”对话框会显示以下字段：

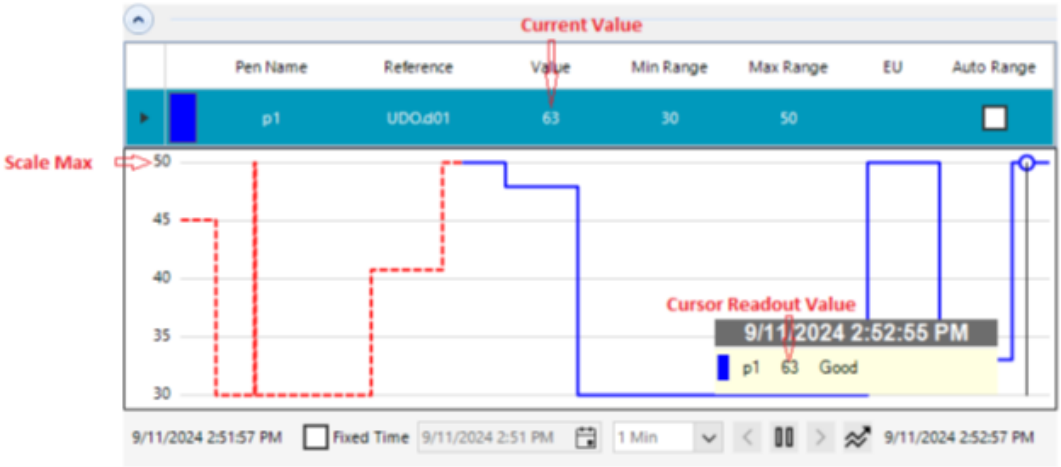
- 笔名
- 值
- 质量
- 工程单位

要查看光标读出对话框，请将光标悬停在趋势区域上，光标读出将显示上述指定时间的值。

取消选择自动范围并且笔引用值超出最小/最大值范围时，光标读出将显示实际值，但趋势绘图中将显示一条平直线。

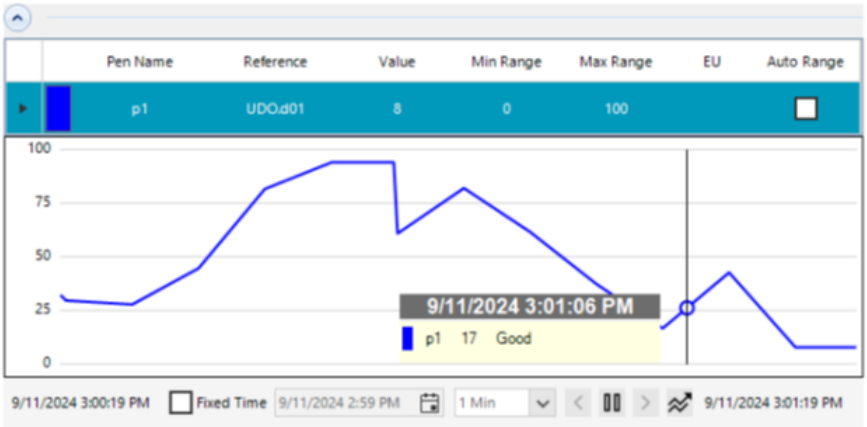
例如，在下面的屏幕截图中，当前 p1 值为 63。最新的光标读出值也显示 63，但笔绘制基于“最大范围”设置显示 50。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



当绘图模式为直线模式时，光标读出将显示上一个数据点值，虽然光标位于斜率的中间。

例如，在下面的屏幕截图中，光标位于两个数据点（17 和 45）之间。虽然光标基于刻度显示 26，但光标读出将显示 17 作为值，而这是上一个数据点。



显示数据质量

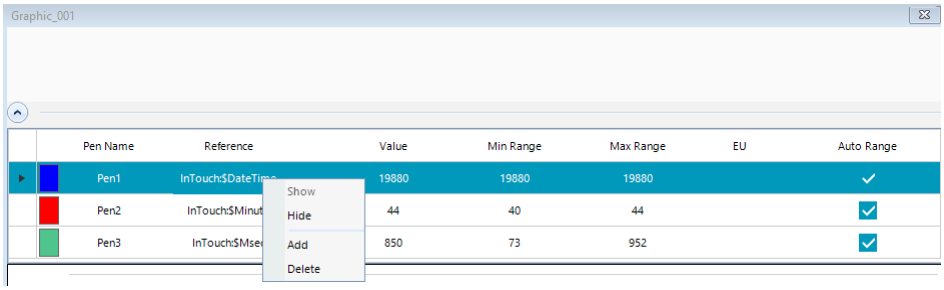
当引用没有优质数据时，趋势笔的样式会更改为“质量不佳”样式。所显示的趋势值将取决于呈现所有质量的数据复选框的设置。如果引用数据的质量不是“良好”，将应用错误的样式。

顶部标题选项

您可以使用“顶部标题”区域在运行时编辑笔详细信息。您也可以：

- 在运行时展开并折叠顶部标题
- 隐藏与显示单独的笔
- 添加笔
- 删除笔

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



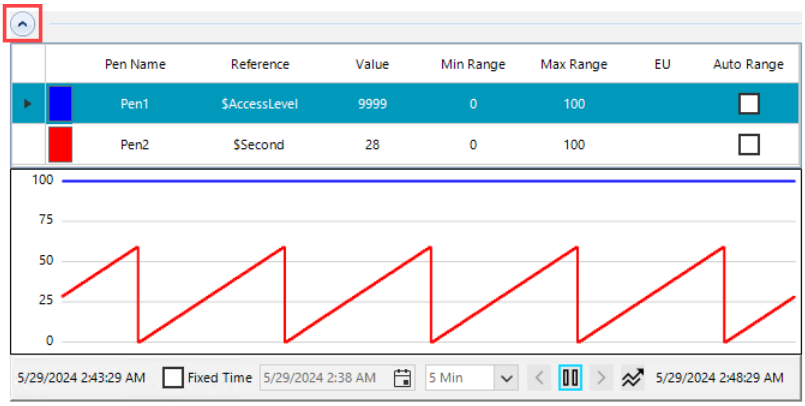
在运行时展开并折叠顶部标题

如果笔的数目更多，将很难阅读和分析趋势，因为趋势图区域将更小。“展开和折叠顶部标题”选项允许您折叠顶部标题并增加图表区域，以便您能够轻松地分析趋势。此外，您还可以在需要时展开顶部标题，以查看笔配置。

折叠顶部标题

1. 在运行时，选择左上角的向上箭头。

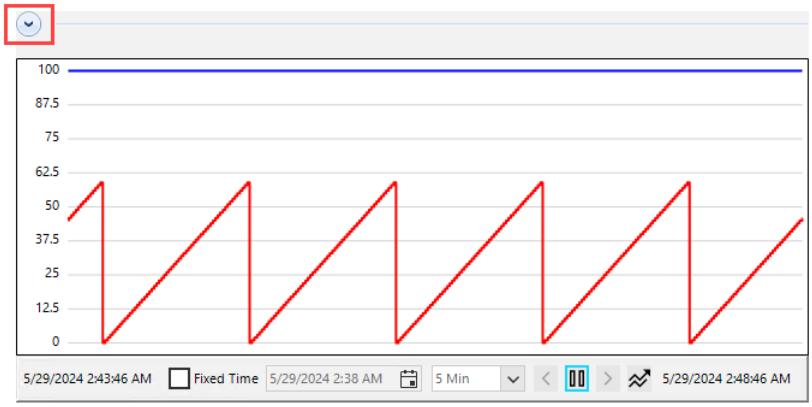
   顶层标题将折叠。



展开顶部标题

1. 在运行时，选择左上角的向下箭头。

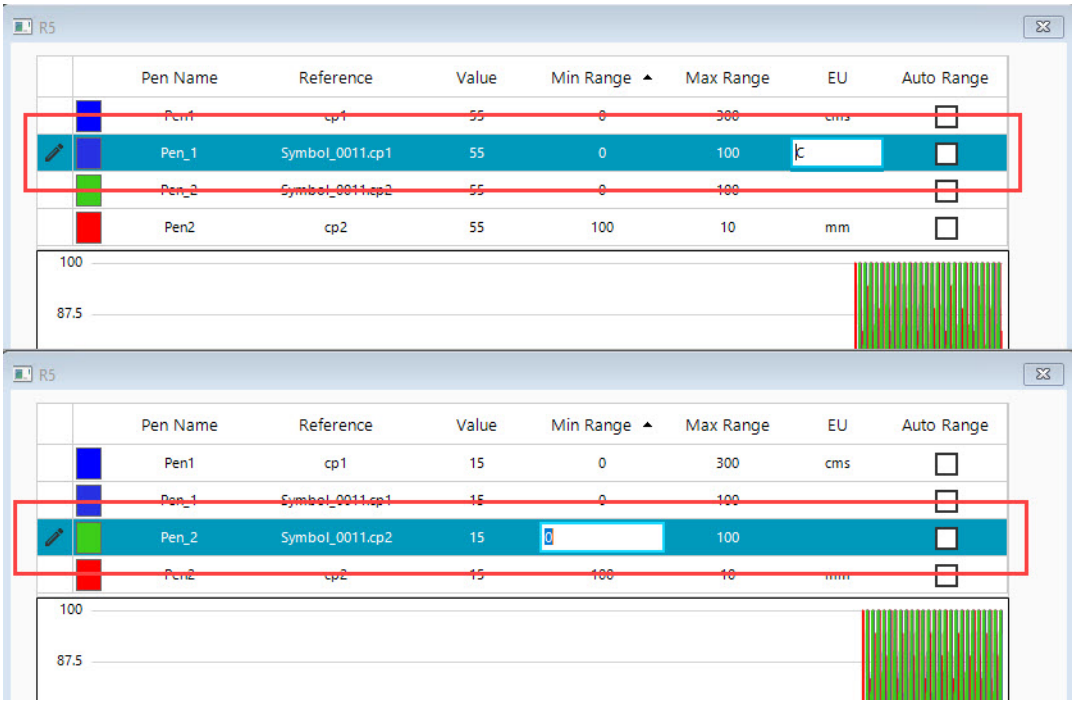
   顶部标题将展开。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在运行时编辑笔详细信息

1. 双击字段以进行编辑。
2. 更新值。
3. 单击行的外部以保存新值。



隐藏与显示单独的笔

1. 使用鼠标右键单击某行。
2. 从上下文相关菜单中选择隐藏。

   笔将在顶部标题区域中隐藏，并且不会呈现在趋势区域中。
3. 要显示笔，请使用鼠标右键单击顶部标题区域。
4. 从上下文相关菜单中，选择显示，然后选择笔名。

   笔将显示在顶部标题和趋势区域中。

添加笔

1. 使用鼠标右键单击顶部标题区域，然后选择添加。

   顶部标题区域中将包含一个新行。
2. 提供笔详细信息，包括名称和引用。
3. 按 Enter 键。

   如果数据可用，趋势区域将开始呈现笔。

删除笔

- 使用鼠标右键单击顶部标题区域，然后选择删除。或者，按 Delete 键。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

笔将从多笔趋势控件中被删除。

对顶部标题区域排序

1. 单击标题行中的标题名称以对笔详细信息进行排序。

   对于以下标题名称，顶部标题将按升序和降序进行排序：笔名、引用、EU。

   所有其他标题名称将按从小到大的顺序进行排序。

   除了颜色之外的所有标题都可以用于对顶部标题区域进行排序。

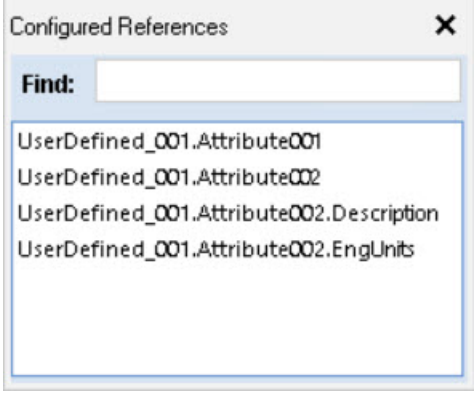
2. 再次单击标题名称可更改排序的方向。
3. 第三次单击标题名称可恢复到排序前的原始状态。

已配置的引用对话框

在“已配置的引用”对话框中，可以查看图形元素上配置的属性。如果在未选择任何元素的情况下右键单击开放区域，将在脚本级别显示可用的属性。如果右键单击某个元素，将显示该元素的相应属性。您可以调整该对话框的大小以及其在窗口上的位置。WindowViewer 在重新启动后不会改变该对话框的大小与位置。

要启动“已配置的引用”对话框：

- 使用鼠标右键单击窗口。此时出现“已配置的引用”对话框。



要查找属性：

- 在“查找”字段中输入属性名的前几个字符，然后按 Enter 键。此时会显示与输入文本相匹配的属性的列表。

要复制属性名：

- 在“已配置的引用”对话框中，使用鼠标右键单击属性，然后单击复制。

  这会将属性的名称保存到操作系统剪贴板中。

使用拖放添加笔

您可以选择引用并使用拖放以在运行时添加趋势笔。

1. 使用鼠标右键单击图形或元素，此时出现“已配置的引用”对话框，其中列出该图形的所有可用属性。
2. 选择引用并将其拖放到多笔趋势的顶部标题上。这将创建新的笔，并且趋势区域将开始呈现所选笔的数据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

趋势区域比例

趋势区域的比例将在运行时计算和显示，具体取决于所选趋势笔的值。显示的网格线数取决于趋势区域的高度。 如果多笔趋势的高度发生改变，比例也将相应地改变。时间间隔数将为 2、4 或 8，具体取决于高度。

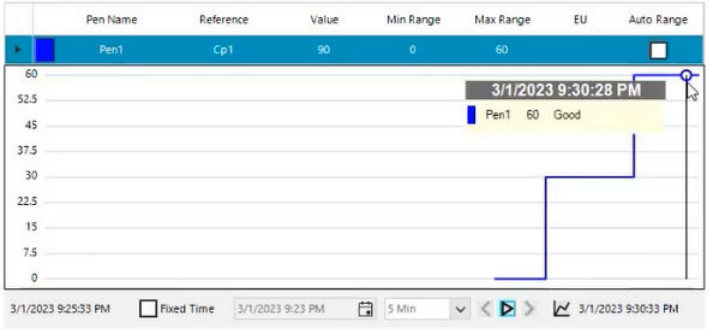
更改最小和最大范围

如果未选中“自动范围”复选框，您可以更改最小和最大范围。趋势区域的比例将会更新，新的比例将会用于绘制趋势。

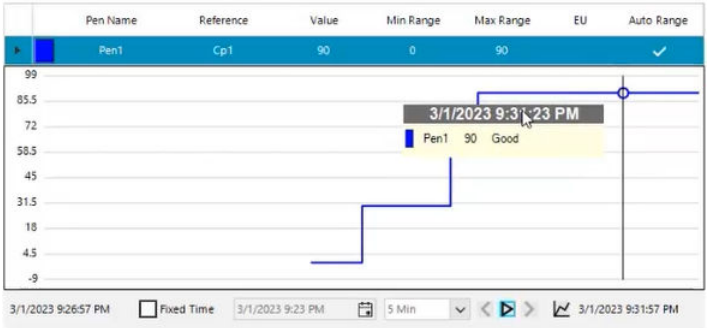
- 对于所选行，单击顶部标题中的“最小范围”和“最大范围”字段，然后输入新值。

趋势区域中笔的比例将会更新。

当标记值超出最小和最大范围时，趋势线将被截断为最小和最大范围值。光标读出框始终显示实际值。在下图中，即使标记值为 58，由于最大范围是 46，因此趋势会被截断为 46，并且光标读出也显示 58。



要查看无截断的趋势线并查看光标读出中的实际值，请选中“自动范围”复选框。如果为笔指定的最小范围值和最大范围值相等，那么会将其视为“自动范围”条件，并且它将忽略为笔指定的值。在下图中，由于选中了“自动范围”复选框，因此趋势线不会被截断，并且光标读出显示的是实际值。



如需有关自动范围的详细信息，请参阅启用自动范围主题。

启用自动范围

“自动范围”会根据趋势笔值来动态更新趋势区域的比例。您为笔指定的最小和最大范围值会被忽略。

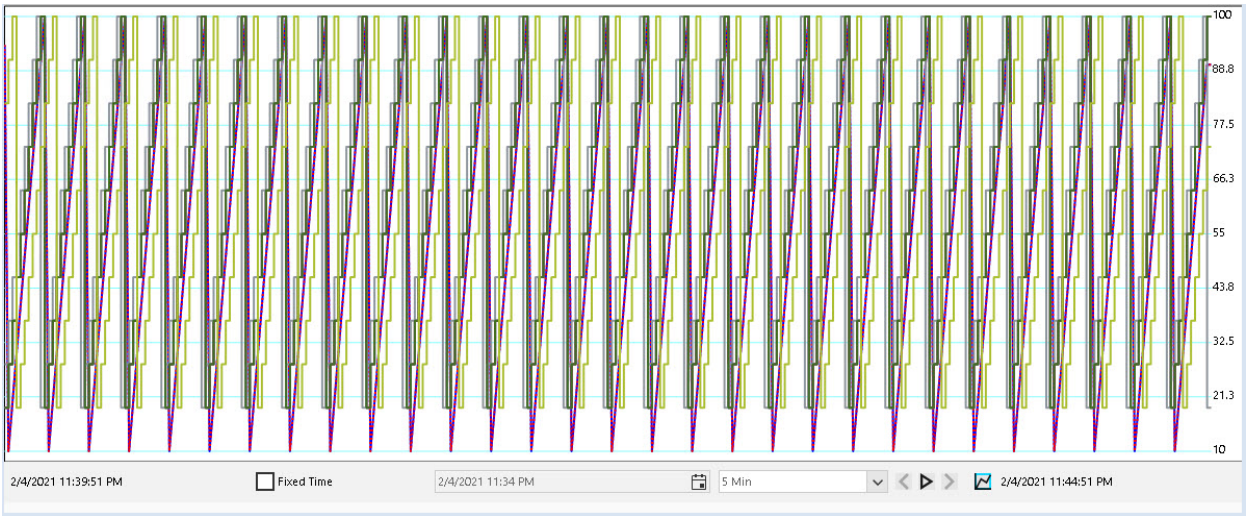
- 针对所需的笔，选中自动范围复选框。

趋势区域中笔的比例将会在您选择笔所在的行时更新。

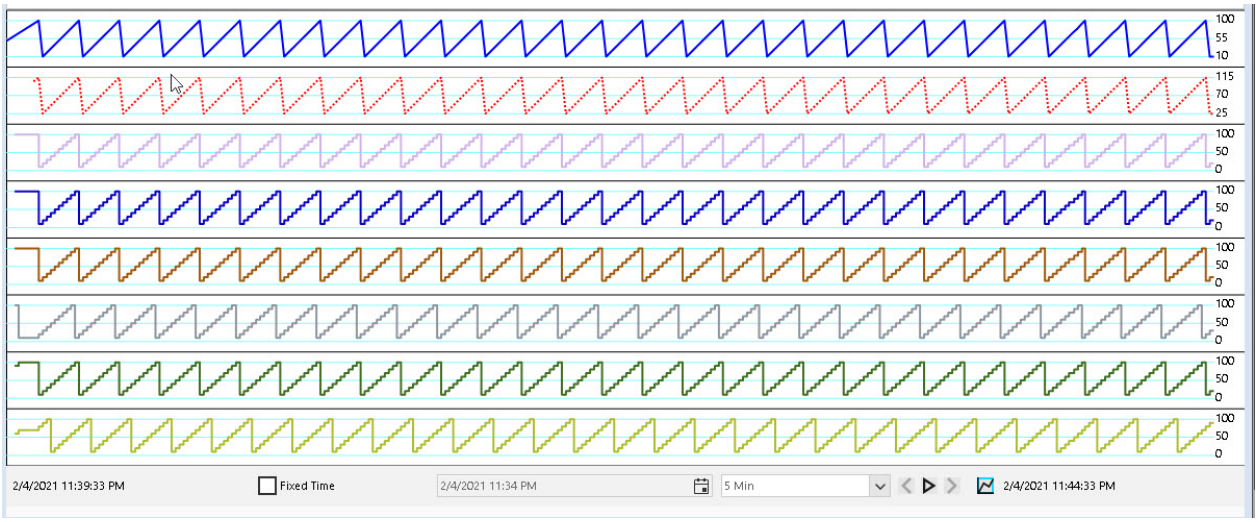
堆叠和取消堆叠趋势

在运行时，可以使用“堆叠/取消堆叠”选项来查看单独的笔。缺省条件下，笔会显示在一起。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



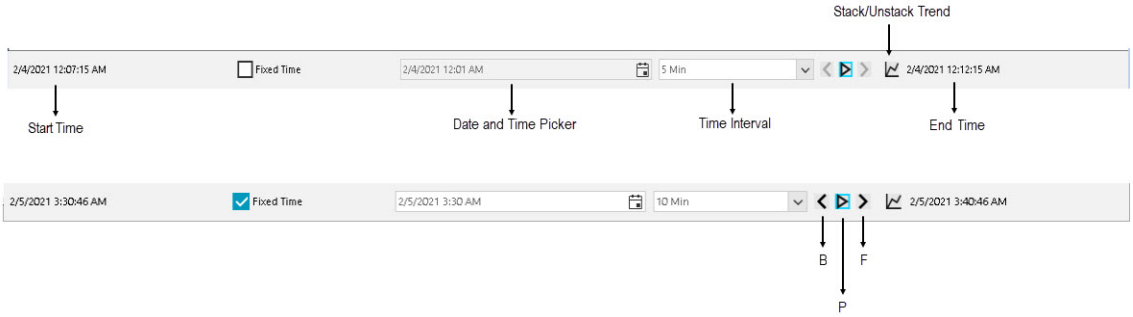
要单独显示笔，请单击工具栏中的 堆叠趋势按钮。笔将以彼此堆叠的方式单独显示。再次选择 可恢复到取消堆叠的视图。



注意：如果在运行时添加笔，且趋势区域较小，则各个比例会重叠。要更清晰地查看结果，可以隐藏一些笔。

播放趋势数据

使用工具栏中的播放选项，可以遍历在指定开始和结束时间之间的趋势数据。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

移动模式

缺省条件下，趋势数据会显示“时间间隔”字段中指定的值的当前趋势数据。开始时间和结束时间的关系是：结束时间 = 开始时间 + 时间间隔。在移动模式下，结束时间始终为当前时间，并将根据输入而变化。在固定模式下，开始时间和结束时间会被冻结，直到使用播放控件。

您可以从预定义值列表中选择时间间隔。趋势区域会刷新趋势笔数据，并仅显示该时间段的数据。

固定模式

1. 如果想要查看过去或将来特定时间段的趋势笔，请选中固定时间复选框。
2. 选择时间间隔。

   趋势区域将反映开始时间和结束时间之间所选时间间隔的笔数据。
3. 使用向后移动趋势 (B)、暂停趋势/恢复趋势 (P) 和向前移动趋势 (F)。

   “暂停”按钮会冻结所选时间间隔的趋势区域。

< 和 > 将在开始时间和结束时间之间的时间块内将趋势区域移动半个时间间隔。

例如：时间间隔 = 10 分钟

模式	开始时间	结束时间
实时	2/8/2021 3:15:16 AM	2/8/2021 3:25:16 AM
固定：向后移动	2/8/2021 3:10:16 AM	2/8/2021 3:20:16 AM
固定：向前移动	2/8/2021 3:20:16 AM	2/8/2021 3:30:16 AM

使用日期和时间选择器

您可以使用日期和时间选择器来选择特定的开始日期和时间。这仅适用于选中“固定时间”复选框的情况。

开始时间将显示所选的日期和时间。结束时间将显示所选的日期和时间 + 时间间隔。

趋势区域将显示所选日期和时间 + 时间间隔的数据。

关于 SmartSymbol

SmartSymbol 是转换成可复用模板的 InTouch 图形。对 SmartSymbol 模板所作的更改会传播到整个应用程序中该 SmartSymbol 的所有实例。对于创建、修改、验证以及重新验证整个应用程序中反复使用的图形，这可以减少重复劳动。

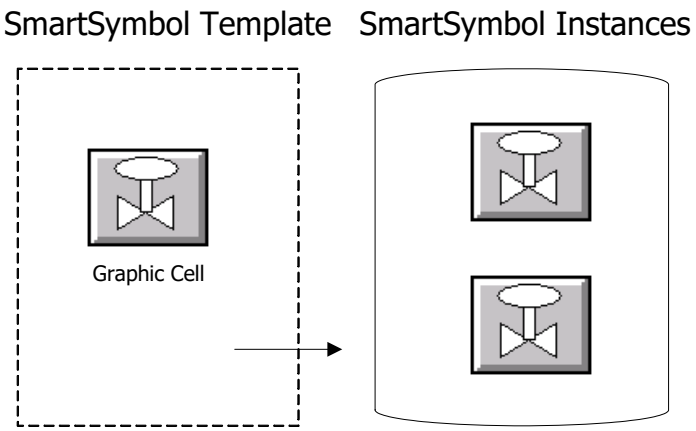
SmartSymbol 可以引用 InTouch 标记和 Application Server 属性。未引用任何标记或仅引用 InTouch 标记的 SmartSymbol，称为 InTouch SmartSymbol。

引用一个或多个自动化对象或其实例的属性的 SmartSymbol，称为工业图形 SmartSymbol。工业图形 SmartSymbol 也可以引用 InTouch 标记。

InTouch SmartSymbol 和工业图形 SmartSymbol 与工业图形不同。工业图形通过 System Platform 集成开发环境 (IDE) 进行设计与管理。

如果在使用 InTouch 第 10 版或更新的版本来创建新的图形，则对于您的应用程序，工业图形或许比 SmartSymbol 更适合。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



SmartSymbol 管理器与库

SmartSymbol 库包含 InTouch 应用程序的 SmartSymbol。“SmartSymbol 管理器”可以导入、导出以及整理 SmartSymbol 库的内容。

SmartSymbol 存储在用户 InTouch 应用程序所在文件夹下的 \Symbols 文件夹中。

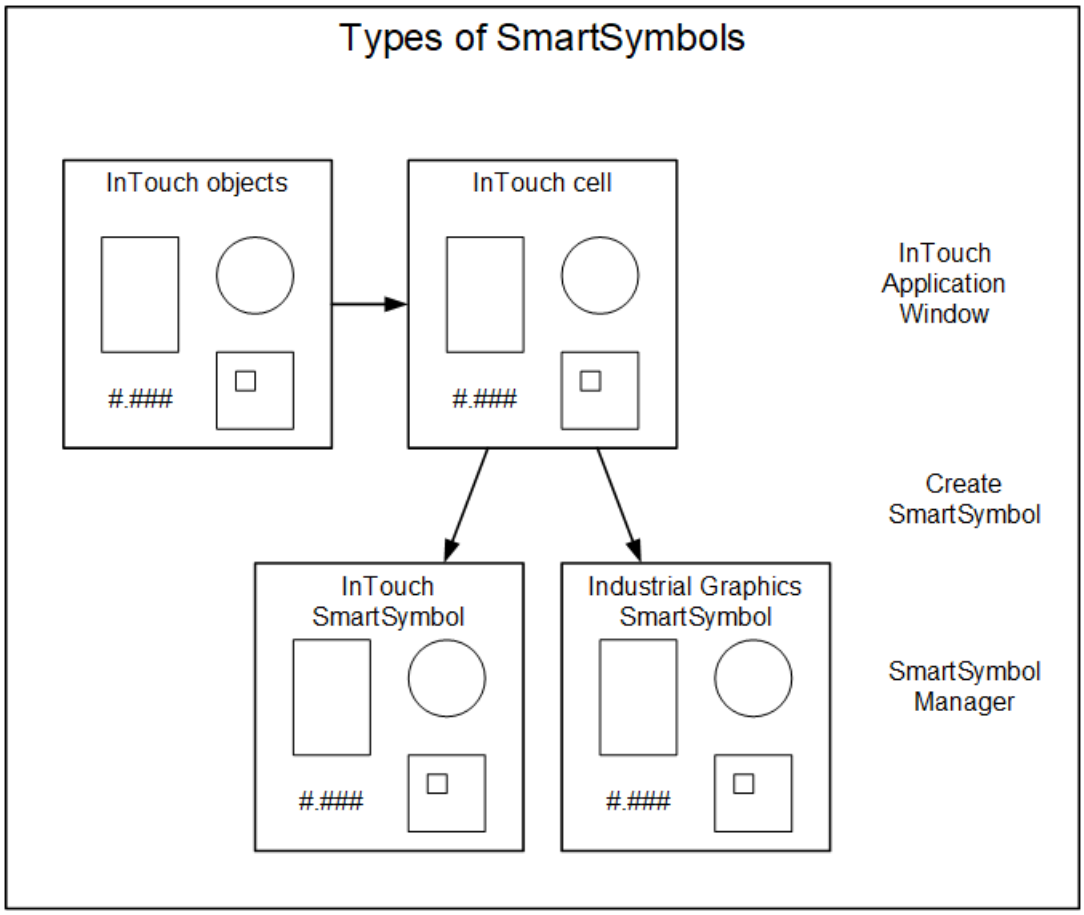
关于库中 SmartSymbol 的引用信息存储在一个 XML 文件中。此 XML 文件不应编辑。

InTouch SmartSymbol 与工业图形 SmartSymbol

InTouch SmartSymbol 与工业图形 SmartSymbol 并不相同。工业图形接替了工业图形 SmartSymbol，是使用 System Platform IDE 开发的。InTouch SmartSymbol 引用 InTouch 标记数据。工业图形 SmartSymbol 引用 Galaxy 对象或模板属性。

下图显示各种 SmartSymbol 的类型。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

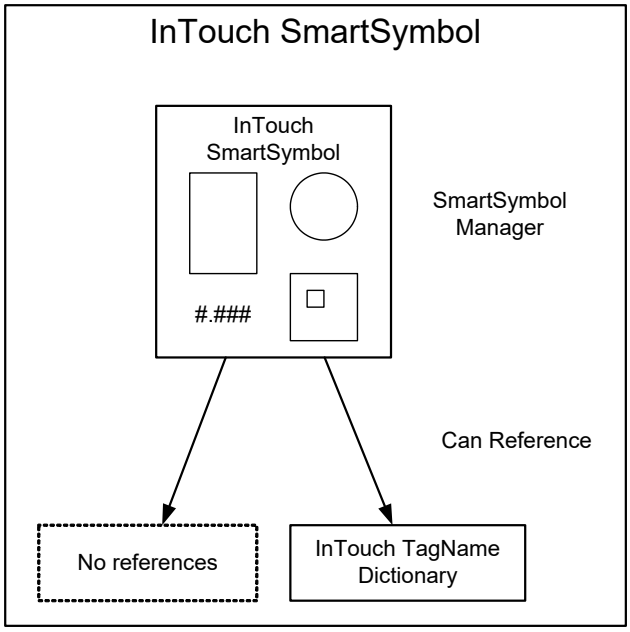


InTouch SmartSymbol

InTouch SmartSymbol 存储在“SmartSymbol 管理器”中的“InTouch 符号”文件夹。

您可以使用本地与远程 InTouch 标记引用给 InTouch SmartSymbol 中的图形元素配置动画。如需有关详细信息，请参阅创建 SmartSymbol 模板与实例。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



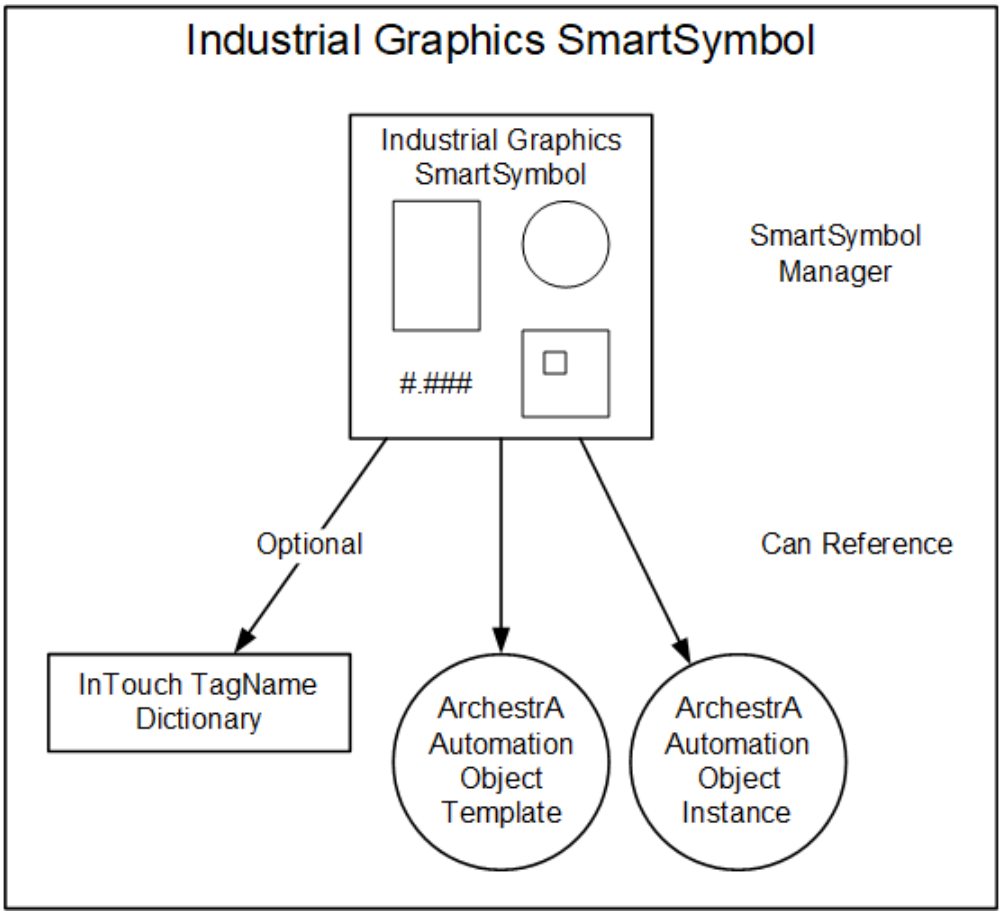
工业图形 SmartSymbol

工业图形 SmartSymbol 存储在“SmartSymbol 管理器”中的“工业图形”文件夹。

备注：尽管文件夹的标签是“工业图形”，但该文件夹包含的是工业图形 SmartSymbol，而不是“工业图形”。

要创建工业图形 SmartSymbol，请选择一个或多个 Galaxy 对象模板来给各种图形元素定义动画引用。如需详细信息，请参阅创建 SmartSymbol 模板与实例。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



SmartSymbol 的限制

以下是 SmartSymbol 的已知限制。

- SmartSymbol 不能包含趋势对象。如果试图创建包含趋势对象（历史或实时）的 SmartSymbol，则会出现错误消息。
- SmartSymbol 不能包含“分布式报警显示”控件、窗口控件、InTouch ActiveX 控件（如 AlarmViewer）或是 InTouch 应用程序中配置的第三方 ActiveX 控件。
- 无法浏览 Galaxy 中通过 Application Server 1.5 版创建的实例。要浏览这些实例，请安装 Application Server V2.0 或更高版本。
- 不支持从“SPC 图表”向导生成 SmartSymbol。
- SmartSymbol 不能引用局部脚本变量。
- 属性浏览器不显示衍生的对象实例。要解决此问题，请在“SmartSymbol 管理器”中创建衍生的模板，或是在浏览器中创建自定义的过滤器。

创建 SmartSymbol 模板与实例

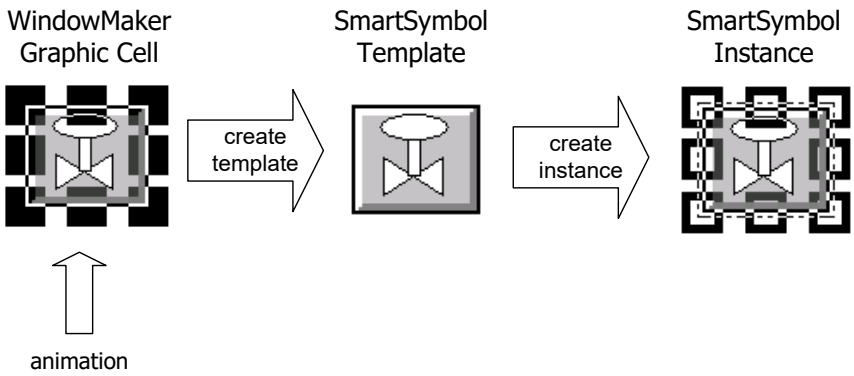
使用 WindowMaker 可以创建 SmartSymbol 模板与实例。

要创建 SmartSymbol 模板，请在 WindowMaker 中绘制一个或多个图形，将它们合并成一个单元，然后将该单元转换成 SmartSymbol。您不必将模板连接或链接到 InTouch 标记或 Application Server 对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建 SmartSymbol 模板后，可以在应用程序窗口中创建该 SmartSymbol 的实例。

也可以根据现有工业图形 SmartSymbol 实例来创建 Application Server 对象实例，以避免在 InTouch WindowMaker 与 System Platform IDE 之间切换。



创建 SmartSymbol 模板以使用 InTouch 数据

InTouch SmartSymbol 从单元进行创建，可以包含图形元素、动画以及对 InTouch 标记的引用。

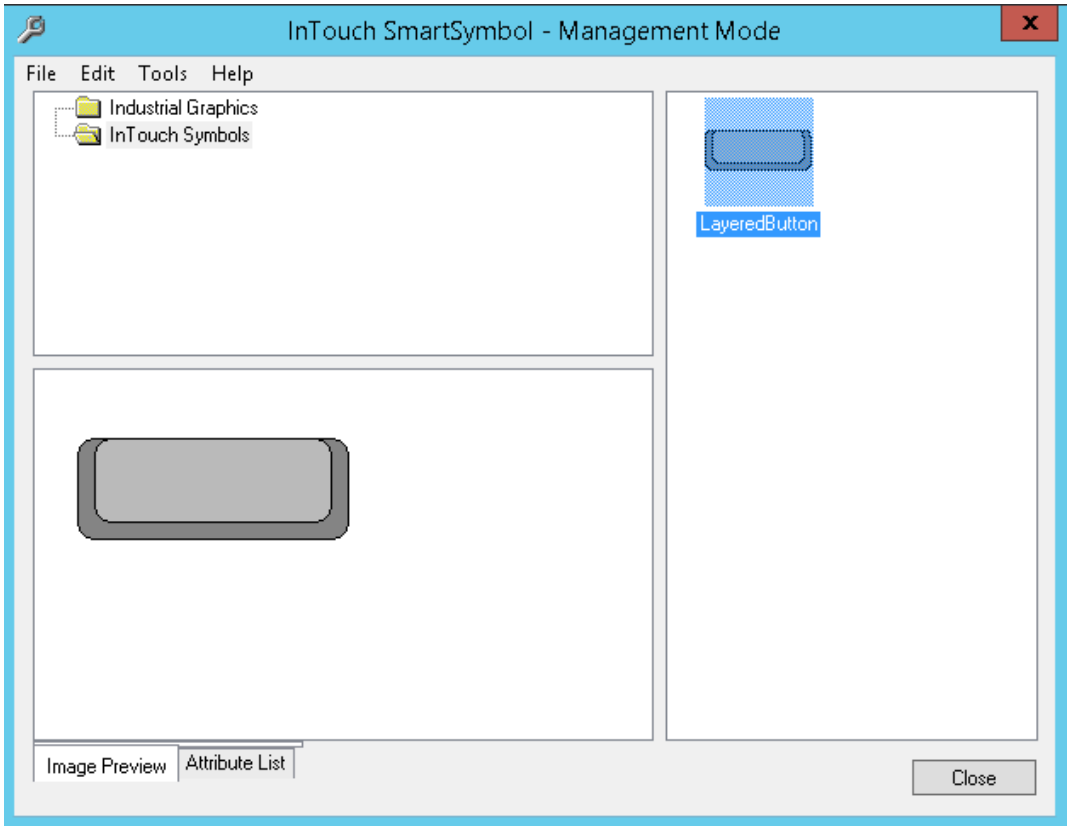
以下操作程序指定创建新的窗口、图形以及单元，但您也可以使用现有的窗口、图形或单元来创建 SmartSymbol。

创建新的 InTouch SmartSymbol 模板

1. 在 WindowMaker 中创建一个新窗口。
2. 使用绘图工具与/或<1>“</1>向导<2>”</2>来创建要包含在 SmartSymbol 中的一个或一组图形<3>。</3>
3. 为图形配置动画链接。
   如需操作说明，请参阅动画对象。
4. 选择要包含在 SmartSymbol 模板中的所有对象。
5. 在动画菜单上的单元组中，选择制作单元。
6. 选择刚刚制作的单元。
7. 在绘图菜单上的 SmartSymbol 组中，选择生成。

   此时出现 InTouch SmartSymbol - 管理模式对话框，新的 SmartSymbol 会突出显示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



缺省条件下，新的 SmartSymbol 会放置在“InTouch 符号”顶层文件夹中。程序自动给符号指定缺省名称<1>（</1>例如<2>“</2>新建符号 1<3>”）。</3>

8. 输入新的名称或是接受缺省值。您随时可以更改 SmartSymbol 的名称。如需有关重命名 SmartSymbol 的详细信息，请参阅重命名 SmartSymbol 模板。
9. 选择关闭。此时出现一则消息，提示您使用新的 SmartSymbol 替换图形单元。选择 是 或“否。如果选择是，则图形单元替换成 SmartSymbol。如果选择否，则图形单元保持不变。不论是哪一种情况，新的 SmartSymbol 都会存储在 SmartSymbol 库中供将来使用。

创建工业图形 SmartSymbol 模板

根据 InTouch 图形单元创建 SmartSymbol 时，如果该单元包含至少一个对自动化对象模板或实例的引用，则最终会生成一个工业图形 SmartSymbol。

对自动化模板的引用包含一个“$”符号。

您可以生成与对象模板和/或实例关联的 SmartSymbol 模板。

在 InTouch 窗口上创建 SmartSymbol 实例时，可以实例化其引用的对象模板。

生成新的工业图形 SmartSymbol 模板

1. 在 WindowMaker 中创建一个新窗口。
2. 使用绘图工具与/或<1>“</1>向导<2>”</2>来创建要制作到 SmartSymbol 中的图形<3>。</3>
3. 为图形配置动画链接。

   如需有关配置 Galaxy 数据源名的操作说明，请参阅从 InTouch 访问 Application Server 数据。如需有关配置 Application Server 属性链接的操作说明，请参阅动画对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 选择要制作到单元中的图形。
5. 在动画菜单上的单元组中，选择制作单元。
6. 在绘图菜单上的 **SmartSymbol** 组中，选择生成。

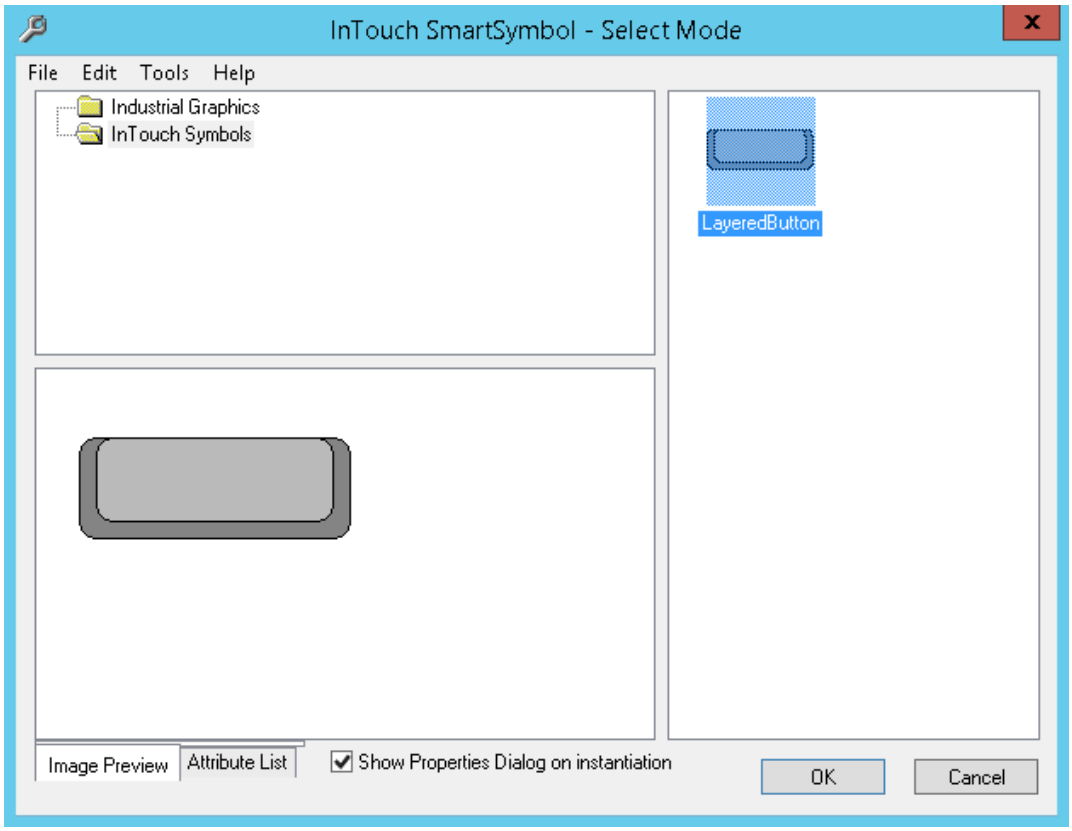
   此时“SmartSymbol 管理器”为 Galaxy 数据创建新的 SmartSymbol。
7. 输入新的名称，或是接受缺省名称并在以后更改。
8. 选择关闭。此时出现一则消息，提示您使用新的 SmartSymbol 替换图形单元。选择 是 或“否。如果选择是，则图形单元替换成 SmartSymbol。如果选择否，则图形单元保持不变。不论是哪一种情况，新的 SmartSymbol 都会存储在 SmartSymbol 库中供将来使用。

从 InTouch SmartSymbol 模板创建 SmartSymbol 实例

从一个 SmartSymbol 模板可以创建多个 SmartSymbol 实例。每个实例都继承所有的引用与文本标签。在将实例放到 InTouch 窗口上之前，可以更改引用与文本标签。

从 InTouch SmartSymbol 模板创建 SmartSymbol

1. 打开 WindowMaker，然后打开要使用 SmartSymbol 的窗口。
2. 在主页菜单上的插入组中，选择 **SmartSymbol** 图标。
3. 选择 WindowMaker 窗口中要用于放置该符号的位置。此时出现 **InTouch SmartSymbol - 选择**模式对话框。

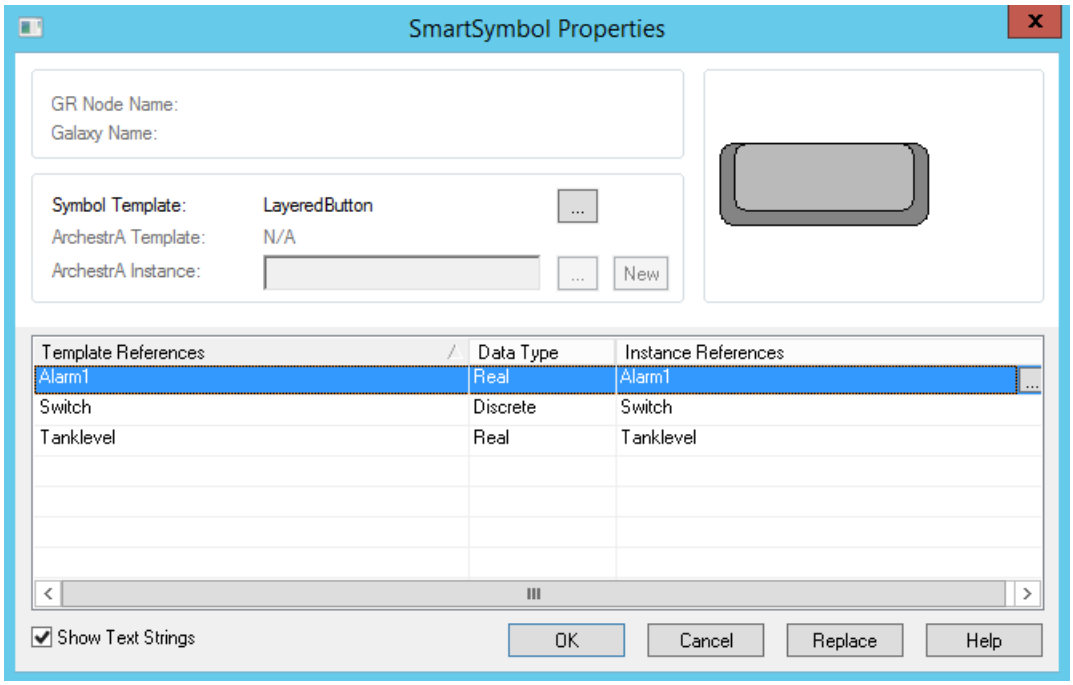


注意：缺省条件下，在实例化时显示属性对话框复选框处于选中状态。如果不希望更改新 SmartSymbol 实例的任何引用或文本标签，请清除该复选框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 在 **InTouch 符号**文件夹中，双击 SmartSymbol。新的符号将显示在应用程序窗口中。

   如果在上一步中选择了在**实例化时显**示属性对话框复选框，则出现 **SmartSymbol** 属性对话框。



5. 在**实例引用**列中，单击省略号按钮。此时出现**选择标记**或**标记名字典**对话框。
6. 选择要链接到 SmartSymbol 的标记。关闭窗口，随后出现 **SmartSymbol** 属性对话框。

   注意：如果输入尚未定义的新标记名，则出现**标记名未定义**对话框，此时选择确定，并从“标记名字典”中定义一个新标记。

7. 选择确定。新的符号将显示在应用程序窗口中。

从 ArchestrA SmartSymbol 模板创建 SmartSymbol 实例

从一个 ArchestrA SmartSymbol 模板可以创建多个 ArchestrA SmartSymbol 实例。

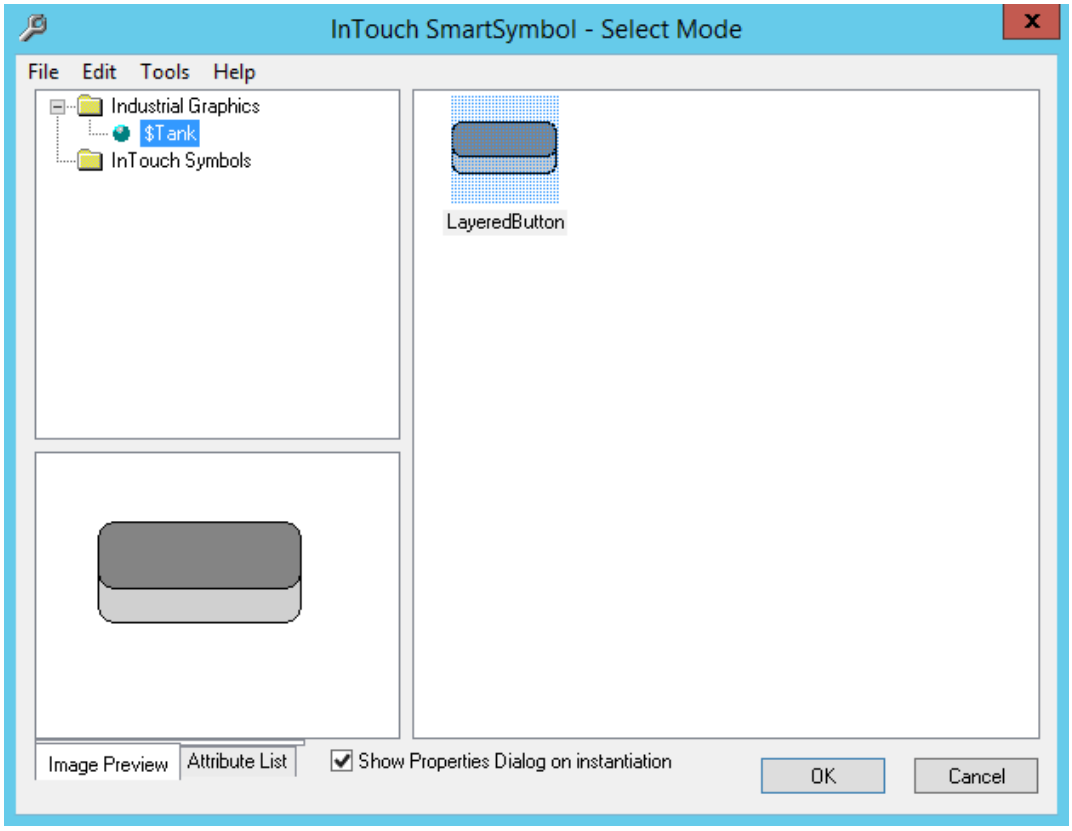
从 ArchestrA SmartSymbol 模板创建 SmartSymbol

1. 选择 **SmartSymbol** 图标，然后选择 WindowMaker 窗口中要用于放置符号的位置。此时出现 **InTouch SmartSymbol - 选择**模式对话框。

注意：缺省条件下，在**实例化时显**示属性对话框复选框处于选中状态。

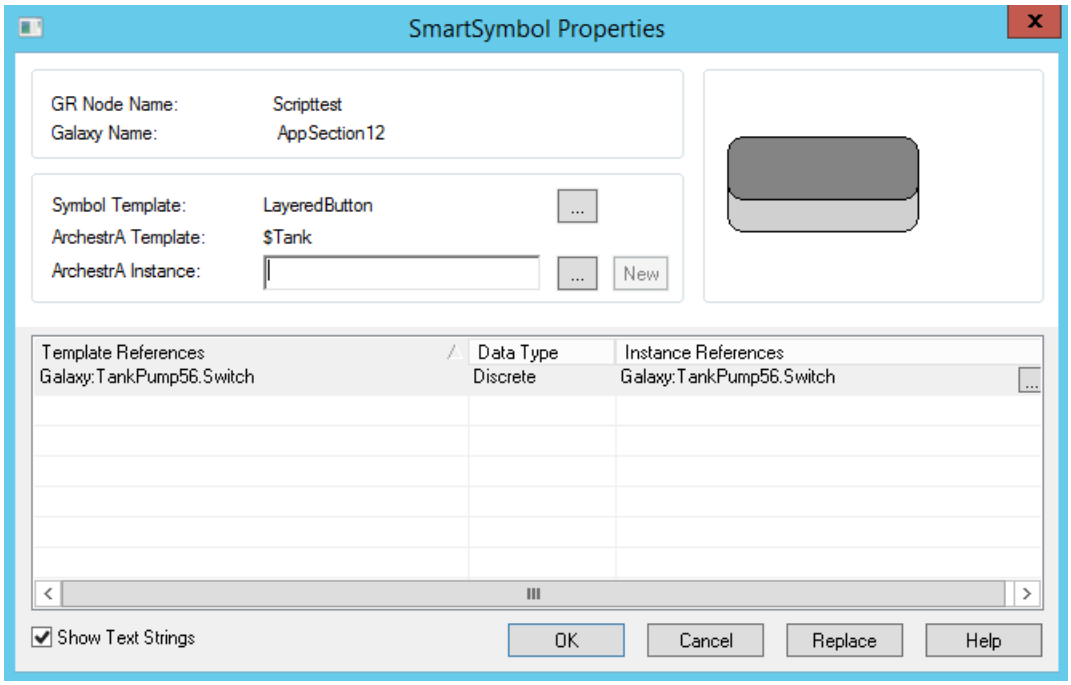
1. 选择工业图形文件夹。此时工业图形出现在右侧窗格中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. **选择该** SmartSymbol，然后**选择确定**。新的符号将**显**示在**应**用程序窗口中。

   如果在上一**步**中**选择**了**在实例化时显示属性对话**框**复选**框，则出现 **SmartSymbol 属性对话**框。



3. 在 **ArchestrA 实例**文本框中，可以**执**行以下两**种**操作之一：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 浏览并选择 ArchestrA 对象。
- 创建从关联的对象模板衍生而来的新 ArchestrA 对象实例。输入实例的名称，然后选择新建。

此时实例属性引用出现在实例引用列中。

注意：如果尚未连接 Galaxy，则出现一个对话框，提示输入节点名与 Galaxy 名。

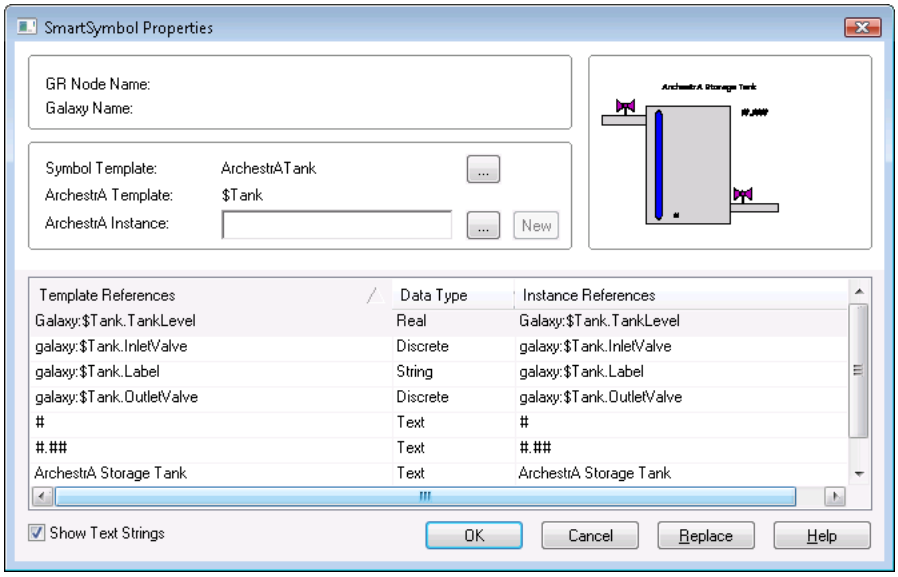
4. 在实例引用列中，根据需要更改引用。您可以手工输入语法正确的引用，或是选择省略号按钮以使用属性浏览器。
5. 选择确定。此时新的符号出现在窗口中。

从 ArchestrA SmartSymbol 实例创建 ArchestrA 对象实例

从现有的 ArchestrA SmartSymbol 实例可以创建新的 ArchestrA 对象实例。这样做时，不需要在 WindowMaker 与 IDE 之间切换。

创建新的 ArchestrA 对象实例

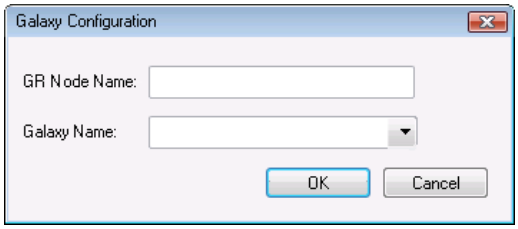
1. 在 WindowMaker 中，打开 SmartSymbol 实例所在的窗口。
2. 双击应用程序窗口中的 SmartSymbol 实例。此时出现 **SmartSymbol 属性对话**框。



3. 在 **ArchestrA 实例**框中，为新的“自动化”对象输入有效的名称。

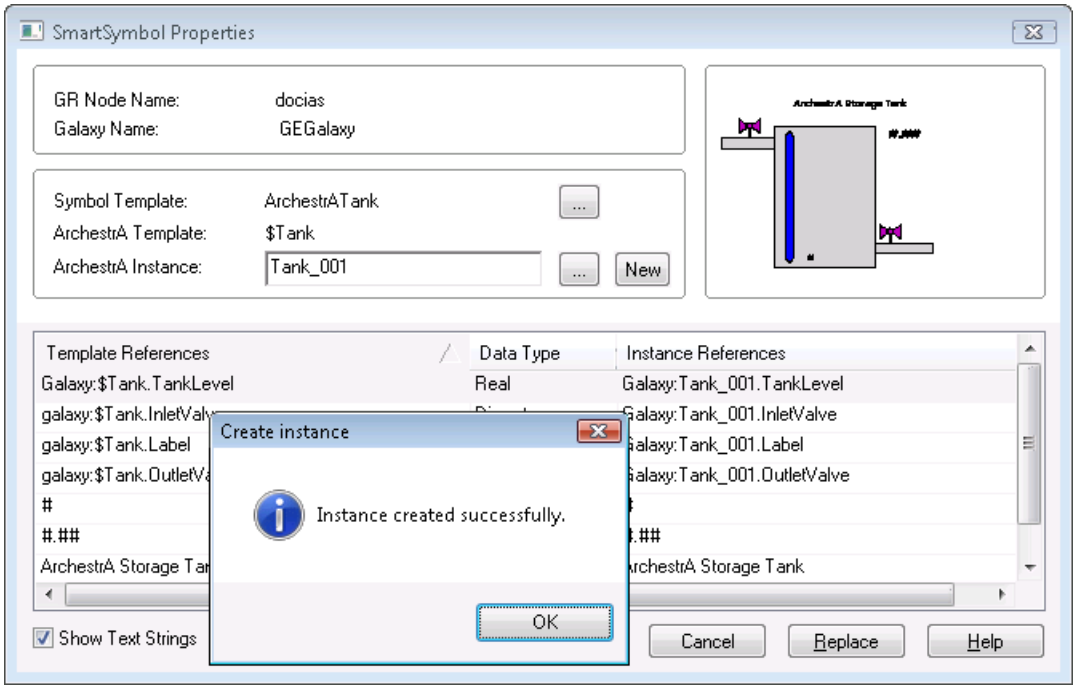
注意：如果这是第一次指定对象，程序会提示您进行登录。请提供有效的用户名、口令以及域名。如果 Application Server 安全性设置为“无”之外的其它模式，则仅在基于“操作系统用户”或“操作系统用户组”安全模式下，才需要输入域名。

4. 选择新建。出现消息提示选择有效的 Galaxy 以便在其中创建新对象时，选择确定。此时出现 **Galaxy 配置对话**框。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

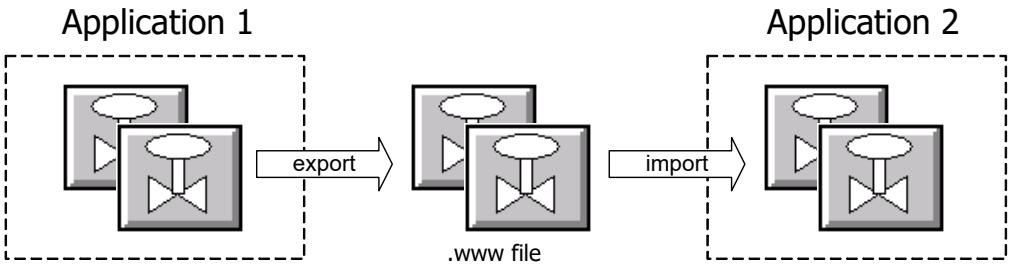
5. 指定 Galaxy。执行以下操作：
   a. 在 **GR 节点名**框中，输入正在运行 Galaxy 的计算机的名称。
   b. 在 **Galaxy 名**列表中，选择 Galaxy。
   c. 选择确定。此时创建 ArchestrA 对象实例，并且实例引用指向新的实例。



   a. 再次选择确定以关闭创建实例对话框。
6. 选择确定以关闭 **SmartSymbol 属性**对话框。此时新的 SmartSymbol 实例出现在应用程序窗口中。

管理 SmartSymbol

通过使用“SmartSymbol 管理器”，可以在多个 InTouch 应用程序以及不同的物理系统之间导入与导出 SmartSymbol。导出与导入 SmartSymbol 是在 InTouch 应用程序之间移动 SmartSymbol 的最佳方式。



您还可以导入带 SmartSymbol 的窗口，此时将导入图形而不是模板信息，结果会产生孤立的 SmartSymbol 实例。如需详细信息，请参阅恢复 SmartSymbol。

借助<1>“</1>SmartSymbol 管理器<2>”，</2>可以重命名<3>、</3>复制<4>、</4>删除及保存 SmartSymbol 模板<5>。</5>

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

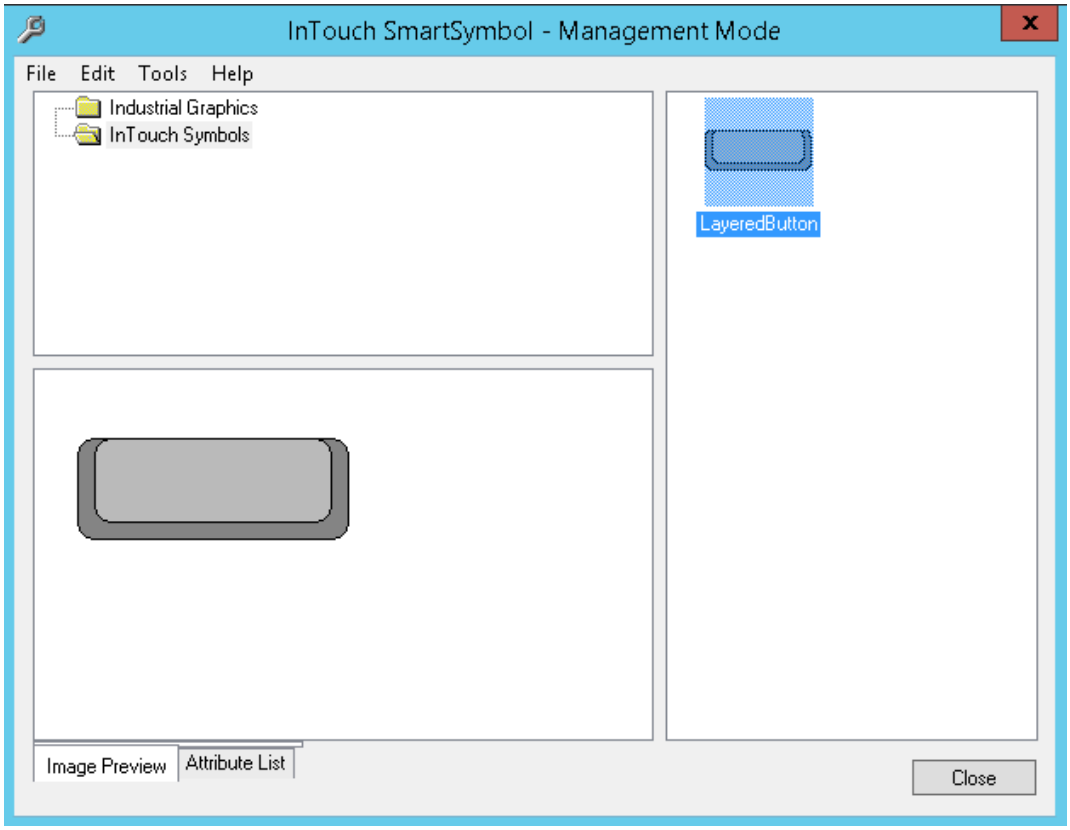
导入 SmartSymbol

您可以从其它 InTouch 应用程序将 SmartSymbol 导入自己应用程序的 SmartSymbol 库。通过从其它应用程序导入符号，可以实现符号的重复利用，而不必重新去创建那些符号。

将 SmartSymbol 导入 SmartSymbol 库

1. 关闭所有的应用程序窗口。
2. 在**绘图**菜单上的 **SmartSymbol** 组中，选择管理。

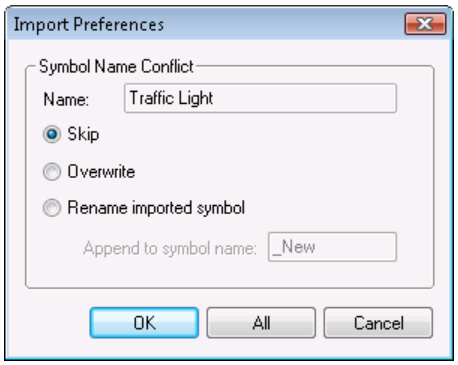
   此时出现 **InTouch SmartSymbol - 管理模式**窗口。



3. 在文件菜单上，选择导入。此时出现导入符号对话框。
4. 浏览到包含要导入的 SmartSymbol 的文件。符号导出文件的文件扩展名是 .www。
5. 选择文件，然后选择确定。此时该文件中的 SmartSymbol 出现在<1>“</1>SmartSymbol 管理模式<2>”</2>窗口中<3>。</3>

   如果存在名称冲突，则出现导入首选项对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



6. 执行以下一项或多项操作：
   - 要跳过此符号的导入操作，**请选择跳过**。如果导入多个符号，仍会导入其余的符号。
   - 要使用新的符号覆盖现有的符号，**请单击改写**。
   - 要使用未使用过的名称重命名新的符号，**请单击重新命名已导入的符号**。在**将此附加到符号名**框中，**输**入名称。
7. 执行以下操作之一：
   - **选择确定**将所选的**选项应**用于 SmartSymbol。
   - 如果**选择**了**重新命名已导入的符号**, 则**选择全部**可以把将此**附加到符号名**框中的文本**应**用于数据包文件中存在名称冲突的所有 SmartSymbol。

   此时导入的 SmartSymbol 出现在 InTouch **SmartSymbol 管理模式**窗口中。

导出 SmartSymbol

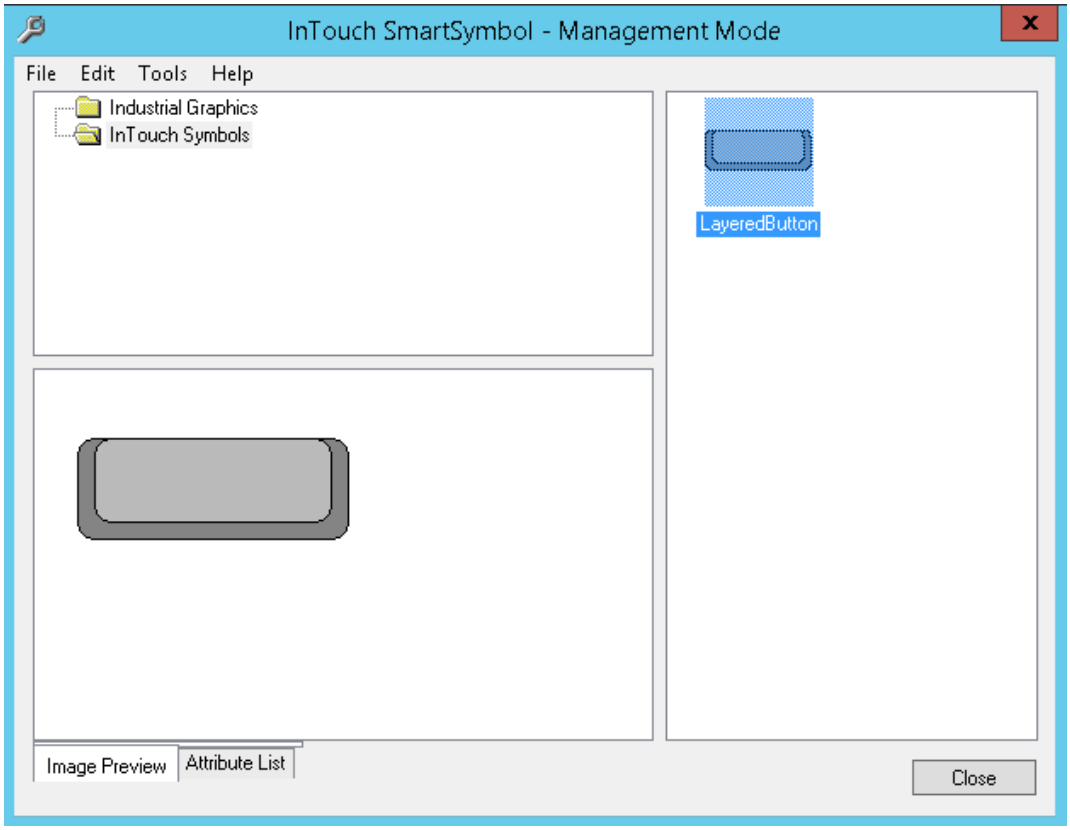
在应用程序 SmartSymbol 库中创建或导入 SmartSymbol 之后，可以将一个或多个 SmartSymbol 模板导出到其它 InTouch 应用程序。建议使用导出 SmartSymbol 模板的方式在 InTouch 应用程序之间移动 SmartSymbol。

导出 SmartSymbol

1. 在**绘图**菜单上的 **SmartSymbol** 组中，**选择管理**。

   此时出现 **InTouch SmartSymbol - 管理模式**窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 从 SmartSymbol 和文件夹的列表中，选择要导出的 SmartSymbol 或文件夹。
3. 在文件菜单上，选择导出。此时出现导出符号对话框。
4. 浏览到要将符号导出到其中的文件夹。
5. 输入带扩展名 .www 的文件名，然后选择保存。此时 SmartSymbol 与/或文件夹导出到所指定的文件夹。

重命名 SmartSymbol 模板

通过使用<1>“</1>SmartSymbol 管理器<2>”，</2>可以重命名 SmartSymbol 模板<3>。</3> 重命名 SmartSymbol 模板不影响任何 SmartSymbol 实例。

重命名 SmartSymbol 模板

1. 在<1>“</1>SmartSymbol 管理器<2>”</2>中<3>，</3>选择要重命名的 SmartSymbol 模板<4>。</4>
2. 在编辑菜单上，选择重命名。
3. 给符号输入新的名称，然后按 **Enter**。此时出现的 SmartSymbol 模板已经使用新的名称。

创建 SmartSymbol 模板的副本

创建 SmartSymbol 模板之后，可以创建它的副本。例如，您可以创建模板的副本，然后修改并编辑它，使之成为具有类似功能的新模板。

如需有关编辑 SmartSymbol 模板的详细信息，请参阅“编辑 SmartSymbol”。

创建 SmartSymbol 模板的副本

1. 在 SmartSymbol 管理器中，选择要创建副本的 SmartSymbol。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 在编辑菜单上，选择复制。
3. 选择新 SmartSymbol 的文件夹。
4. 在编辑菜单上，选择粘贴。此时出现新的 SmartSymbol。如果放在与原件相同的文件夹，则新的 SmartSymbol 会命名为“<原始名称> 的副本”。

删除 SmartSymbol 模板

如果删除 SmartSymbol 模板，则不再能打开、编辑或查看基于该模板的 SmartSymbol 实例的属性。这些 SmartSymbol 实例的运行时状态不受删除操作的影响。

您可以从实例中恢复已删除的 SmartSymbol。如需详细信息，请参阅恢复 SmartSymbol。

删除 SmartSymbol 模板

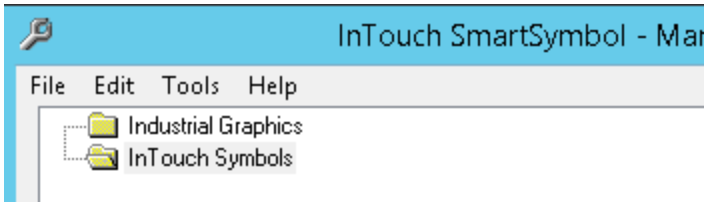
1. 在<1>“</1>SmartSymbol 管理器<2>”</2>中<3>，</3>选择要删除的 SmartSymbol<4>。</4>
2. 在文件菜单上，选择删除。出现消息时，选择是。

   此时 SmartSymbol 模板从 SmartSymbol 库中删除。此 SmartSymbol 的所有实例都成为孤立的实例。

在文件夹层次结构中保存 SmartSymbol

SmartSymbol 按标准的层次化文件夹结构存储在 SmartSymbol 库中。您可以看到其中包含两个标准文件夹，用于简化在 SmartSymbol 库中整理 SmartSymbol 的工作。

- 一个用于工业图形 SmartSymbol 模板的顶层文件夹
- 一个用于 InTouch SmartSymbol 模板的顶层文件夹



您可以使用“SmartSymbol 管理器”给模板创建子文件夹。创建工业图形 SmartSymbol 模板时，将它们存储在应该与之关联的模板文件夹中。例如，如果创建要与 $Valve 对象配合使用的 SmartSymbol，请将符号模板存储在 "$Valve" 模板文件夹中。

工业图形 SmartSymbol 无法拖到“InTouch 符号”文件夹中，InTouch SmartSymbol 也无法拖到“工业图形符号”文件夹中。

将 SmartSymbol 移到不同的文件夹

1. 选择要移动的 SmartSymbol。
2. 将 SmartSymbol 拖到新的文件夹中。

对 SmartSymbol 与语言切换的支持

如果应用程序中存在 SmartSymbol 模板，则语言切换功能可以对 SmartSymbol 起作用。

如果 SmartSymbol 包含可以翻译的文本对象，则导出字典时会生成一个单独的 XML，例如 SSD_<符号名>_<语言标识>_<标识>.xml。此 XML 文件包含该 SmartSymbol 中包含的所有可翻译字符串。您可以在 Excel 中打开它，然后翻译文本字符串，就像处理任何 InTouch 应用程序那样。

导入 InTouch 应用程序的译文时，也会导入每个 SmartSymbol 的译文。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在 WindowViewer 中切换语言时，包含可以翻译的字符串的任何 SmartSymbol 都会按照其翻译显示出译文。

导出有字典文件的 SmartSymbol 时，字典文件随 .www 文件一起导出。如需有关语言切换的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序运行时*》中的在运行时切换语言。

恢复 SmartSymbol

从库中删除 SmartSymbol 模板时，该 SmartSymbol 的所有实例都被视作“孤立的”实例。您可以从孤立的实例中恢复已删除的 SmartSymbol。如果应用程序窗口中不存在孤立的实例，则无法恢复 SmartSymbol。

在删除 SmartSymbol 模板之后，如果试图打开其实例的属性，则出现一条警告消息，提示库中已经不存在该 SmartSymbol。

如果导入一个包含 SmartSymbol 的窗口，则也可能会产生孤立的实例。您必须从孤立的实例中恢复 SmartSymbol，然后重命名该 SmartSymbol。

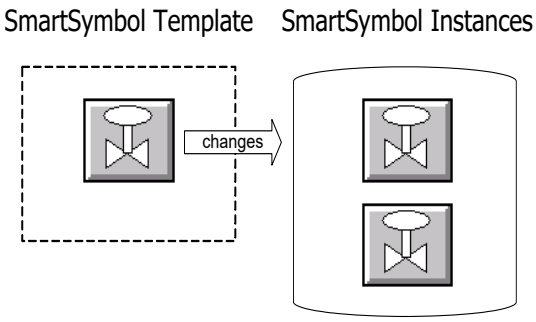
恢复已删除的 SmartSymbol

1. 在 InTouch HMI 应用程序窗口中，选择已删除的 SmartSymbol 的孤立实例。
2. 在**绘图**菜单上的 **SmartSymbol** 组中，选择**恢复**。

   此时 SmartSymbol 出现在 **SmartSymbol 管理模式**窗口中，名称为新建符号。
3. 根据需要重命名 SmartSymbol。

编辑 SmartSymbol

创建 SmartSymbol 之后，可以通过更改与修改 SmartSymbol 的模板或实例来编辑它。



更改 SmartSymbol 模板

要编辑 SmartSymbol，请分解单元，然后使用绘图工具进行更改。您还可以更改与 SmartSymbol 关联的动画。更改模板会影响 SmartSymbol 的所有实例。

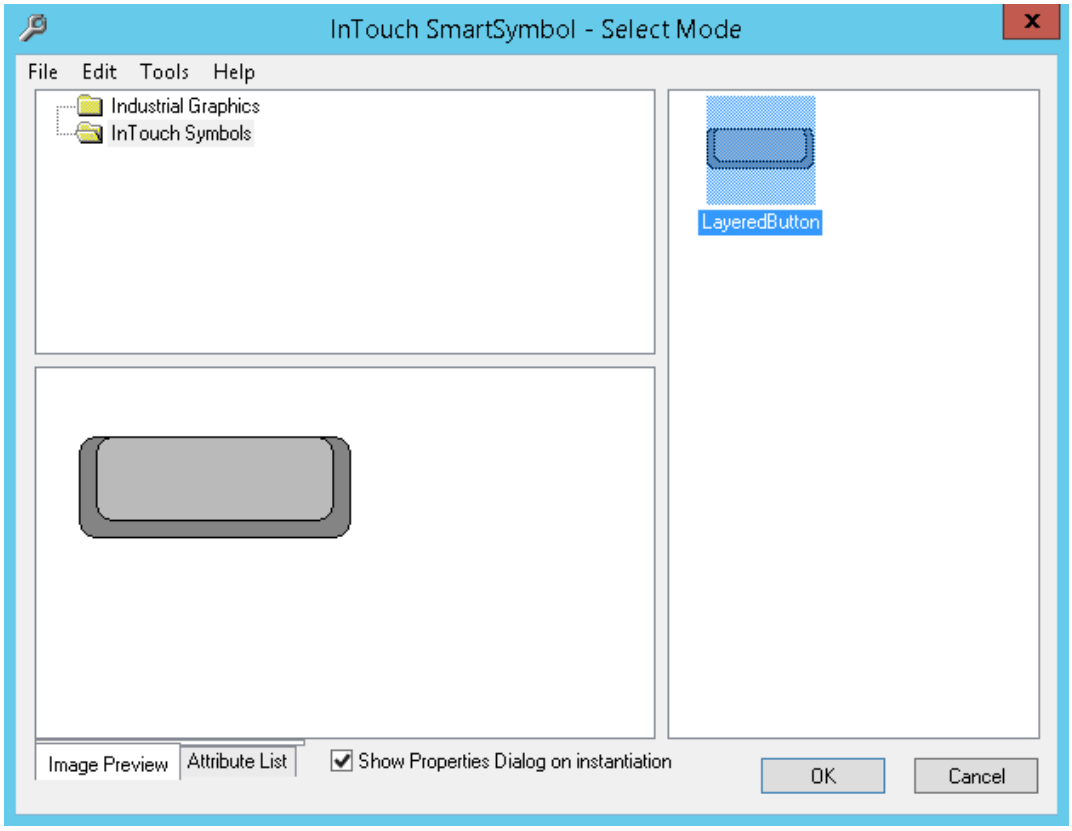
注意：在临时窗口而不是应用程序窗口中编辑 SmartSymbol。

编辑现有的 SmartSymbol 模板

1. 在**绘图**菜单上的 **SmartSymbol** 组中，选择**开始编辑**。
2. 选择窗口中要编辑 SmartSymbol 的位置。

   此时出现 InTouch **SmartSymbol - 选择**模式窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. **选择**要**编辑**的 SmartSymbol，然后**选择确定**。

   此**时该** SmartSymbol 的一个**实**例放入**应**用程序窗口中。

4. 在**动画**菜**单**上的**单元组**中，**选择分解**。

   此**时**，符号分解成各个**组**成元素。

5. **现**在可以**编辑**一个或多个元素。

   注意：如果将元素添加到属于某个 SmartSymbol 一部分的**单**元，**这**会**导**致**该单**元的空**间变**大。将更改**传**播到 SmartSymbol **实**例**时**，可以**选择**是否**传**播大小更改。

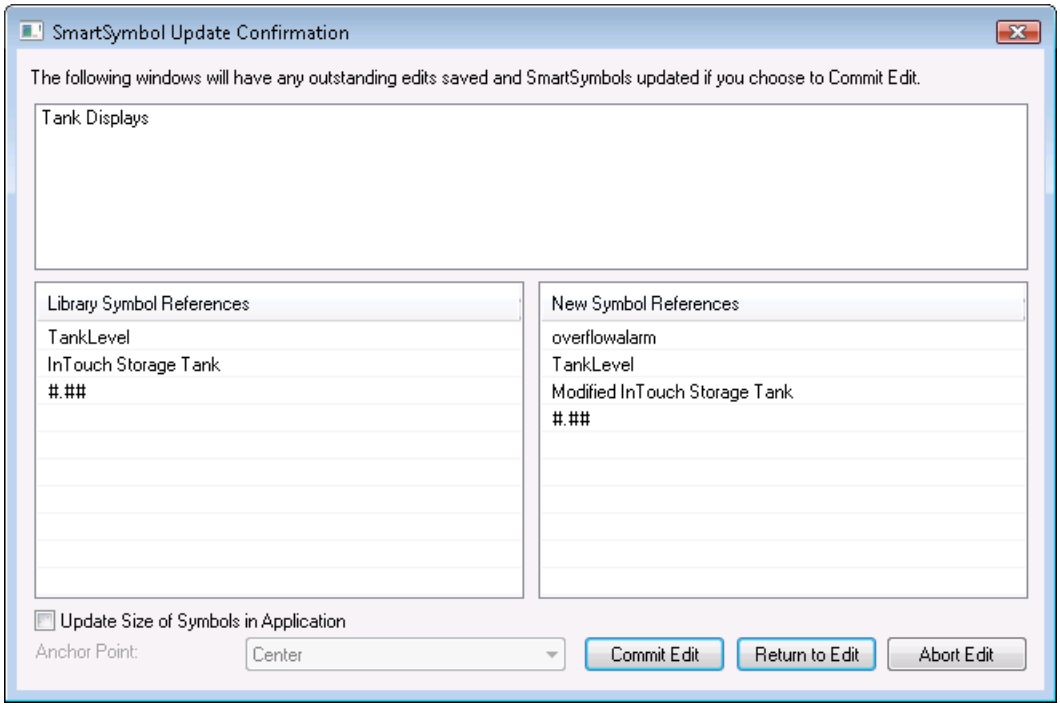
6. 完成**编辑时**，**选择该**符号的所有元素。

7. 在**动画**菜**单**上的**单元组**中，**选择制作单元**。

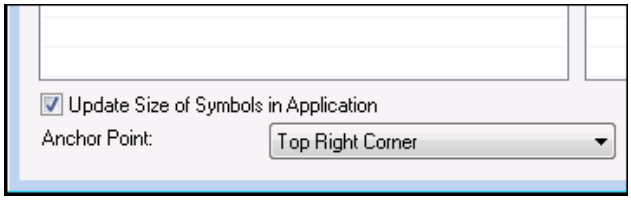
8. 在**绘图**菜**单**上的 **SmartSymbol 组**中，**选择结束编辑**。

   此时出**现 SmartSymbol 更新确认**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



9. 如果编辑后的 SmartSymbol 大小发生了更改，则可以配置大小传播方式。执行以下操作之一：
   - 要不影响现有 SmartSymbol 实例的大小，请清除更新应用程序中的符号大小复选框。
   - 要将模板大小的更改传播到 SmartSymbol 实例，请选中更新应用程序中的符号大小复选框，并在定位点列表中，选择在完成大小调整时，SmartSymbol 实例的哪个部分要“固定”在屏幕上。



10. 执行以下操作之一：
    - 要应用所作的更改，请选择确认编辑。此时<1>“</1>SmartSymbol 管理器<2>”</2>更新 SmartSymbol 模板及其所有实例<3>。</3>
    - 要继续编辑 SmartSymbol，请选择返回到编辑。此时再次出现应用程序窗口以便进一步编辑。

更改 SmartSymbol 实例

SmartSymbol 实例中的任何引用与静态文本都可以更改。您可以搜索并替换实例中的静态文本。

为 SmartSymbol 实例选择不同的引用

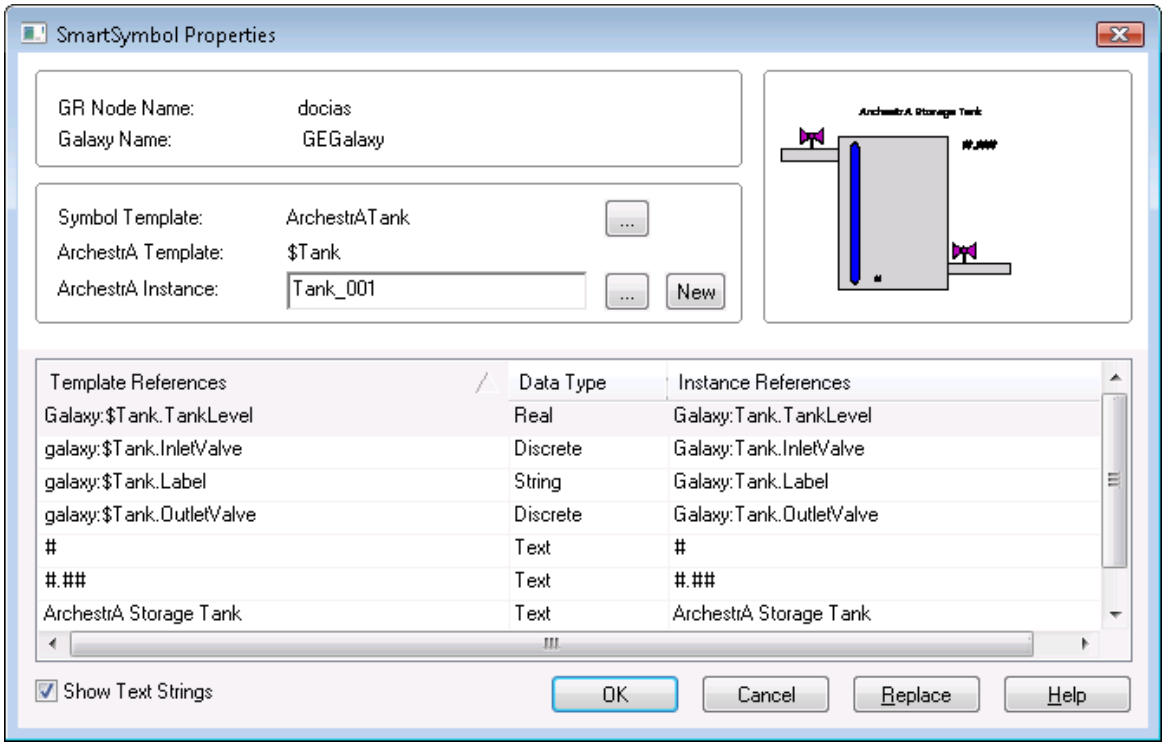
将 SmartSymbol 作为实例放入窗口之后，可以更改它的引用以指向别处，如另一个对象或不同的标记。此时 SmartSymbol 模板不受影响。

在运行时，可以通过使用 IOSetRemoteReferences() 脚本函数来更改 SmartSymbol 实例所引用的标记。如需详细信息，请参阅在运行时重定向远程引用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

编辑 SmartSymbol 实例中的引用

1. 双击 SmartSymbol 实例。此时出现 **SmartSymbol 属性**窗口。



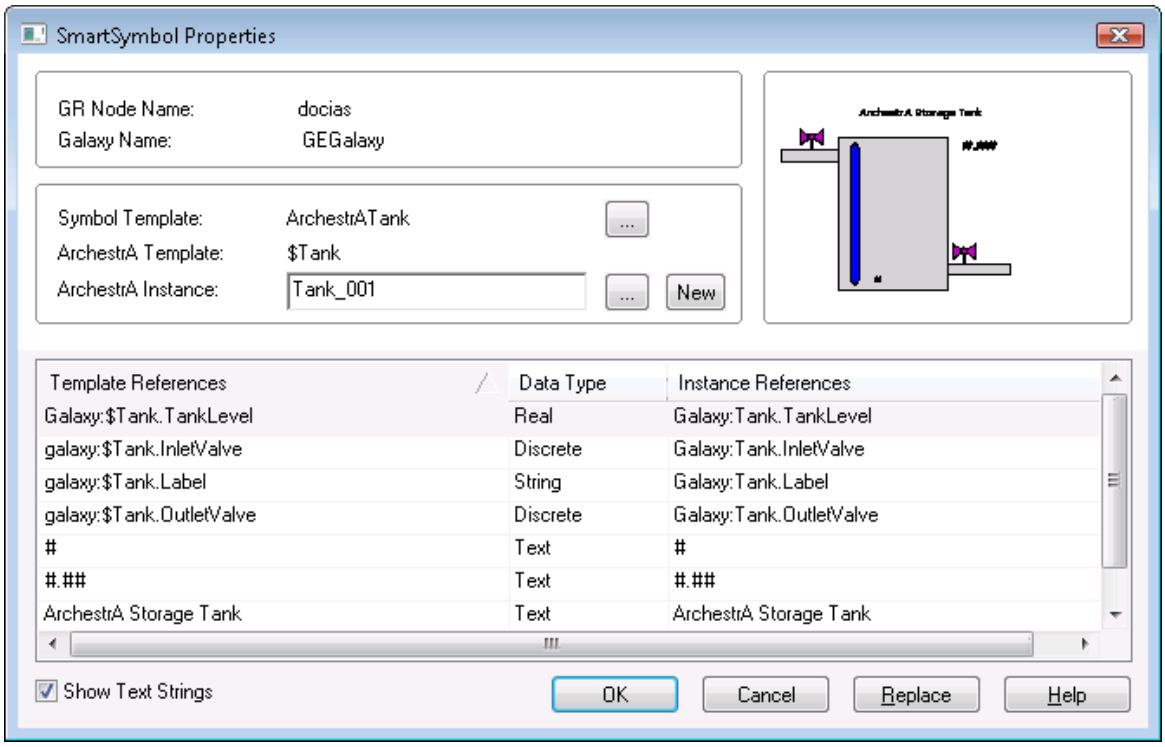
2. 执行以下任意操作：
   - 选择符号模板旁边的省略号按钮，以选择新的 SmartSymbol 模板。
   - 选择 **ArchestrA 实例**文本框旁边的省略号按钮，以通过浏览查找 ArchestrA 对象实例。
3. 选择要映射到 SmartSymbol 的不同对象实例，然后选择确定。

手工编辑 SmartSymbol 实例的文本与引用

在应用程序窗口中创建 SmartSymbol 实例之后，可以更改实例中的静态文本。

更改 SmartSymbol 实例中的静态文本

1. 双击 SmartSymbol 实例。此时出现 **SmartSymbol 属性**窗口。



2. 在**实例引用**列中，**选择**文本框中并修改文本。
3. **选择确定**。

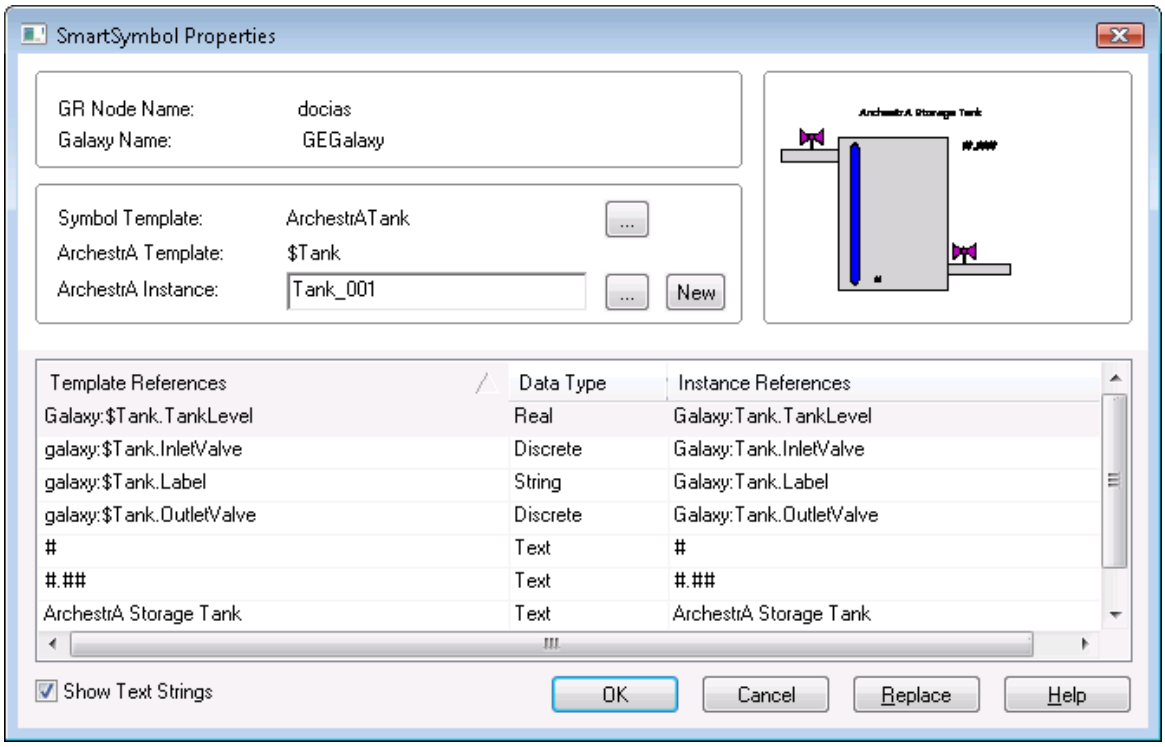
替换 SmartSymbol 实例的标记名与文本字符串

如果 SmartSymbol 实例中有多个引用与文本字符串需要进行相同的更改，则可以使用替换功能。

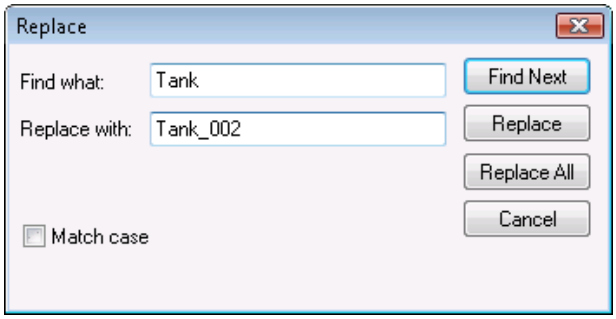
替换 SmartSymbol 实例引用

1. 双击 SmartSymbol 实例。此时出现 **SmartSymbol 属性**窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 选择替换。此时出现替换对话框。



3. 配置替换文本字符串。执行以下操作：
   a. 在查找内容框中，输入要替换的文本。选中区分大小写复选框时，搜索会区分大小写。
   b. 在替换为框中，输入替换文本。替换文本总是完全按照输入时的形式使用。
4. 执行以下操作之一：
   - 要替换所有文本，请选择全部替换。
   - 要一次一个实例地查找与替换文本，请选择查找下一个，然后选择替换以替换该实例。
5. 选择确定。应用程序窗口中的 SmartSymbol 实例出现时，其标记与文本字符串已发生更改。

迁移 InTouch SmartSymbol

通过导入（迁移）InTouch SmartSymbol，您可以在工业图形中使用它们。SmartSymbol 外观和配置将导入并转换为动画配置。

导入的 SmartSymbol 可以：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 添加到画布上的现有元素。
- 替换画布上的现有元素。

通常您可以将任何 InTouch SmartSymbol 导入到工业图形中。

注意：SmartSymbol 可能包含一些无法导入或可以导入但存在功能限制的对象。如需这些对象的完整列表，请参阅 SmartSymbol 导入的限制。

将 InTouch SmartSymbol 导入到工业图形中

将 InTouch SmartSymbol 导入到工业图形中

1. 打开“工业图形编辑器”。
2. 在特别菜单上，单击导入 **InTouch SmartSymbol**。此时出现查找 **InTouch** 应用程序向导。
3. 如果 InTouch 应用程序位于非缺省文件夹中，请单击浏览按钮以浏览 InTouch 应用程序的路径。
4. 选择查找应用程序。搜索根目录框将显示要搜索所有应用程序的路径。
5. 如果 InTouch 应用程序不在指定的搜索根目录路径下，请通过输入或浏览新的开始文件夹进行搜索来更改搜索根目录路径。
6. 选择查找。指定搜索根目录文件夹中的所有 InTouch 应用程序都已找到并列出。
7. 选择要从其中导入 SmartSymbol 的应用程序，然后单击下一步。此时出现选择 **InTouch SmartSymbol** 对话框。
8. 在 SmartSymbol 层次结构中浏览到 SmartSymbol 的位置，选择要导入的 SmartSymbol，然后单击确定。

   如果在画布上已具有元素，此时将出现对话框，提示您是否希望替换现有元素。

   - 如果希望删除现有元素并在空画布上导入 SmartSymbol，则选择是。
   - 如果希望保持现有元素并导入 SmartSymbol，则选择否。
9. 如果 SmartSymbol 包含操作系统上目前未安装的字体，则将出现编辑字体映射对话框。

   您可以单击继续来接受建议的字体映射，或针对每个单独字体更改字体映射。要完成这项工作，请执行以下操作：

   a. 在映射的字体列中选择字体名称。此时出现浏览按钮。
   b. 选择浏览按钮。此时出现选择支持的字体对话框。
   c. 从列表中选择一个字体，然后单击确定。选择的字体将在映射的字体列中出现。
   d. 针对要映射到另一个字体的其它任何字体重复上述步骤。

   注意：如果希望保存映射以便下一次导入 SmartSymbol 时使用，请选中保存映射。
10. SmartSymbol 将导入并在画布上出现。

SmartSymbol 导入的限制

导入 InTouch 智能符号时，将导入以下配置：

- InTouch 图形
- 图形动画
- 脚本
- 引用

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

导入 InTouch 图形

下表显示的 InTouch 图形：

- 可以导入且没有任何问题。
- 可以导入但其功能有所更改或在导入过程中丢失某些功能。
- 无法导入。

以下 InTouch 图形可以导入且没有任何问题：

InTouch 图形	工业图形元素	备注
长方形	长方形	
圆角长方形	圆角长方形	
椭圆	椭圆	
线条	线条	
水平/垂直线条	线条	SmartSymbol 将水平/垂直线条转换为“线条”。因此，ArchestrA 仅可以生成线条。
折线	折线	
多边形	多边形	
文本	文本	
位图	位图	
单元	组	ArchestrA 属性“Treat as Icon”= False。
符号	组	ArchestrA 属性“Treat as Icon”= True。
按钮	按钮	

以下 InTouch 图形可以导入，但其功能有所更改或在导入过程中丢失某些功能：

InTouch 图形	工业图形元素	备注
向导	元素	在 SmartSymbol 中分组时，它将显示为一组元素。
SmartSymbol	元素	在另一个 SmartSymbol 中分组时，它将细分为单元，丢失其 SmartSymbol 属性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

无法导入以下 InTouch 图形，因为无法将其添加到 SmartSymbol：

InTouch 图形	工业图形元素	备注
实时趋势	不可用	无法添加到 SmartSymbol。
历史趋势	不可用	无法添加到 SmartSymbol。
ActiveX 控件	不可用	无法添加到 SmartSymbol。这将包括所有 ActiveX 报警控件（Alarm DB View、Alarm Viewer 等）

导入图形动画

导入 InTouch 智能符号时，InTouch 动画中配置的所有数据都会导入 ArchestrA 动画中。InTouch 动画与 ArchestrA 动画通常使用不同的名称，但执行相同的功能。

下表显示哪些动画互相对应：

InTouch 动画链接	ArchestrA 动画
用户输入 - 离散	用户输入
用户输入 - 模拟	用户输入
用户输入 - 字符串	用户输入
游标 - 垂直游标	垂直游标
游标 - 水平游标	水平游标
触动按钮 - 离散值	按钮
触动按钮 - 动作	动作脚本
触动按钮 - 显示窗口	不支持
触动按钮 - 隐藏窗口	不支持
线条颜色 - 离散	线条样式
线条颜色 - 模拟	线条样式
线条颜色 - 离散报警	线条样式
线条颜色 - 模拟报警	线条样式
填充颜色 - 离散	填充样式
填充颜色 - 模拟	填充样式
填充颜色 - 离散报警	填充样式

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 动画链接	ArchestrA 动画
填充颜色 - 模拟报警	填充样式
文本颜色 - 离散	文本样式
文本颜色 - 模拟	文本样式
文本颜色 - 离散报警	文本样式
文本颜色 - 模拟报警	文本样式
对象大小 - 高度	高度
对象大小 - 宽度	宽度
位置 - 垂直	垂直位置
位置 - 水平	水平位置
填充百分比 - 垂直	垂直填充百分比
填充百分比 - 水平	水平填充百分比
其它 - 可见性	可见性
其它 - 闪烁	闪烁
其它 - 相对方向	旋转
其它 - 禁用	禁用
其它 - 工具提示	工具提示
值显示 - 离散值	值显示
值显示 - 模拟值	值显示
值显示 - 字符串值	值显示

导入操作脚本

导入 SmartSymbol 时，也会导入与 SmartSymbol 中对象关联的所有动作脚本。SmartSymbol 中的动作脚本将成为工业图形中的脚本动画。

将导入大多数预定义的 InTouch 函数 (QuickScript)。

数学函数

“工业图形编辑器”支持 InTouch WindowMaker 中的以下数学函数：

```
Abs、ArcCos、ArcSin、ArcTan、Cos、Exp、Int、Log、LogN、Pi、Round、Sgn、Sin、Sqrt、Tan、Trunc
```

字符串函数

“工业图形编辑器”支持 InTouch WindowMaker 中的以下字符串函数：

```
Dtext、StringASCII、StringChar、StringCompare、StringCompareNoCase、StringFromGMTTimeToLocal、
StringFromIntg、StringFromReal、StringFromTime、StringFromTimeLocal、StringInString、
StringLeft、StringLen、StringLower、StringMid、StringReplace、StringRight、StringSpace、
StringTest、StringToIntg、StringToReal、StringTrim、StringUpper、Text、wwStringFromTime
```

系统函数

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

“工业图形编辑器”支持 InTouch WindowMaker 中的以下系统函数：

```
ActivateApp
```

其它函数

“工业图形编辑器”支持 InTouch WindowMaker 中的以下其他函数：

```
DateTimeGMT、LogMessage、SendKeys、WWControl
```

导入引用

导入 SmartSymbol 时，会对标记和引用进行以下更改：

InTouch SmartSymbol	工业图形	示例
本地标记	以“InTouch:”关键字为前缀	实数内存标记“TankLevel1”转换为“InTouch:TankLevel1”
包含点域的本地标记	以“InTouch:”关键字为前缀	离散内存标记“TankLevel1.InAlarm”转换为“InTouch:TankLevel1.InAlarm”
SuperTag	以“InTouch:”关键字为前缀。您需要使用以下语法手动将表达式括起来： attribute("..."); 	实数“SuperTag”成员“Reactor1\Level”转换为“InTouch:Reactor1\Level”。您需要按如下所示手动更改表达式： attribute("InTouch:Reactor1\Level");
I/O 引用	以“InTouch:”关键字为前缀	整数 I/O 标记“Testprot:i00”转换为“InTouch:Testprot:i00”
Galaxy 引用	“Galaxy:”前缀已删除	Galaxy 引用“galaxy:Pump1.Valve1”转换为“Pump1.Valve1”

以下项目将导入，但功能有所更改：

InTouch SmartSymbol	工业图形	示例
Galaxy:ObjectTagname.Property.#VString	“Galaxy:”前缀已删除，但 #VString 不受支持。同样适用于 #VString1、#Vstring2、#VString3 和 #VString4	“Galaxy:Tank.PV.#VString4”转换为“Tank.PV”
Galaxy:ObjectTagname.Property.#ReadSts	“Galaxy:”前缀已删除，但 #ReadSts 不受支持	“Galaxy:Tank.PV.#ReadSts”转换为“Tank.PV”

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

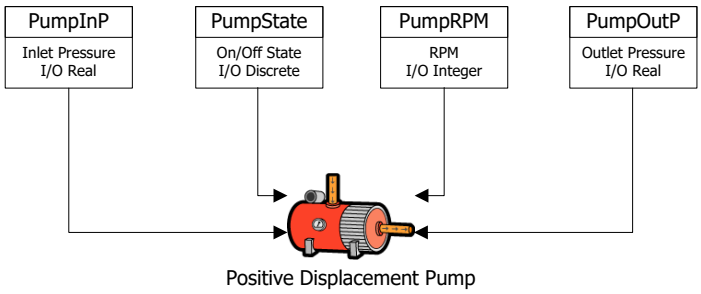
InTouch SmartSymbol	工业图形	示例
Galaxy:ObjectTagname.Property.#WriteSts	“Galaxy:”前缀已删除，但 #WriteSts 不受支持	“Galaxy:Tank.PV.#WriteSts”转换为“Tank.PV”
Galaxy:ObjectTagname.Property.#EnumOrdinal	“Galaxy:”前缀已删除，但 #EnumOrdinal 不受支持	“Galaxy:Selection.Sel1.#EnumOrdinal”转换为“Selection.Sel1”

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 9 标记

InTouch 人机界面 (HMI) 应用程序是生产环境中各种组件的图形化表示。工厂操作员使用这种图形界面来监视和管理生产过程。

下图显示一个泵的示例，它是生产过程中的一个组件。该泵有一些属性及关联的值。压力、RPM 及状态都是泵的属性，这些属性的值通过 HMI 进行监控。



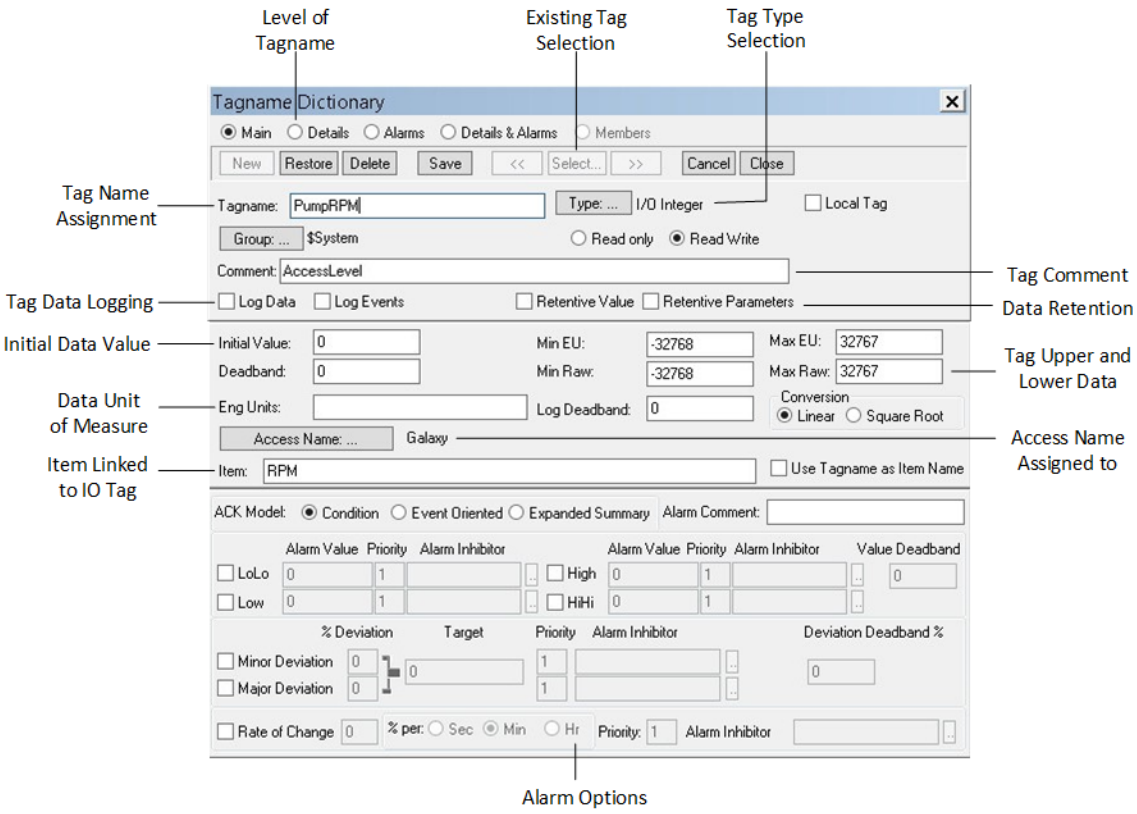
在 InTouch HMI 应用程序中，*标记*代表数据项。通过使用标记，可以将特定的组件属性当作数据项从生产环境中访问。在上图中，PumpState 标记指出泵是打开还是关闭的。对于生产环境中的各种组件，如果要在 InTouch 应用程序中监视或控制其属性，则可以为它们创建标记。

对于从生产组件中采集的不同类型的数据，可以使用不同类型的标记。例如，PumpState 标记返回 On/Off 布尔值，以指出泵是正在运行还是已经停止。对于要成为应用程序一部分的数据类型，需要指定适当类型的 InTouch 标记。

使用标记名字典管理标记

通过使用“标记名字典”，可以为 InTouch 应用程序创建标记。下图显示**标记名字典对话**框，它包含用于定义 I/O 标记属性的所有选项。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



如需有关每个字段的详细信息，请参阅内存标记属性和 I/O 标记属性。

规划标记的使用

在最初的规划阶段，可以通过确定应用程序标记的关键需求来缩短开发时间。全面的规划可以缩短创建 InTouch 应用程序所需的时间。

在创建标记之前：

- 确定要在 InTouch 应用程序中呈现的过程的所有物理组件。

  创建要在应用程序中代表数据源的组件属性的列表。

- 确定与每个组件属性关联的数据类型。

  根据与组件属性关联的数据给每个标记指定一种标记类型。如需有关给标记指定数据类型的详细信息，请参阅创建新标记。

- 确定要合并到 InTouch 应用程序中的数据的特征。

  评估每个标记的以下数据特征：

  - 期望的数据值范围
  - 指定给数据值的度量单位
  - 初始数据值
  - 死区值，用于设置将数据值视为已发生改变的阈值
  - 要在标记值状态改变时显示的消息

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如需有关定义标记数据特征的详细信息，请参阅理解标记属性。

- 制订标记命名惯例与标准。

  复杂的应用程序通常要求使用许多标记。制订标准化的命名惯例，就应用程序中标记的组织结构提出相应的建议。如需有关标记命名惯例的详细信息，请参阅标记名惯例。

- 确定要保存的过程数据。

  所选数据保存到日志文件。您可以使用记录的数据来创建历史趋势，以显示标记值随时间变化的情况。如需有关设置标记记录功能的详细信息，请参阅标记记录。

创建新标记

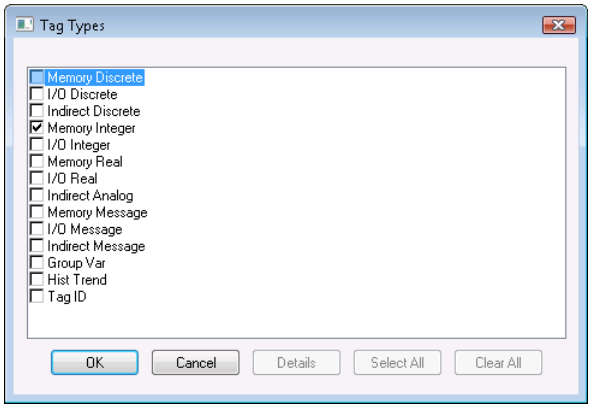
可以使用 WindowMaker **标记名字典**创建标记。在开始之前，请分析工厂流程以确定需要在 InTouch 应用程序中创建的标记。

创建新标记

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在主页菜单上的**标记**组中，选择**标记字典**。

   第一次打开“标记名字典”时，**标记名**框中出现 **$AccessLevel** 系统标记的定义。在保存新标记之后，**标记名字典**显示最近保存的标记定义。

3. 执行以下操作：
   a. 选择**新建**。此时清除**标记名**框。
   b. 输入新标记的名称。如需有关标记命名要求的详细信息，请参阅标记名惯例。
   c. 作为可选项，在**注释**框中输入关于新标记的注释。
4. 选择**类型**此时出现**标记类型**对话框，其中列出所支持的 InTouch 标记类型。



5. 从列表中选择标记类型，然后选择**确定**。此时再次出现<1>“</1>标记名字典<2>”，</2>并显示所选的标记类型<3>。</3>
6. 如果需要，选择**详细信息**以查看所选标记类型的附加“标记名字典”选项。
7. 在**标记名字典**对话框中，进一步指定各个标记选项。

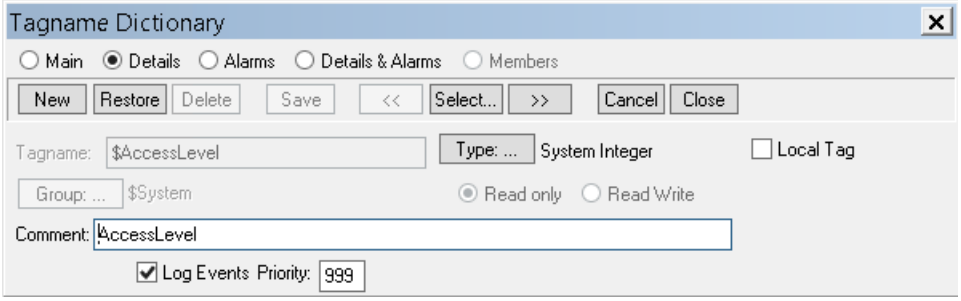
   如需有关指定标记属性的详细信息，请参阅配置标记属性。

8. 选择**保存**。选择**关闭**以关闭**标记名字典**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置标记属性

通过使用标记名字典对话框，可以指定每个标记定义中都会有的公共标记属性。您必须给每个标记指定一个名称。您可以添加可选的注释。标记名与注释都是所有标记的公共属性。



每种类型的 InTouch 标记都有唯一的数据属性。在选择标记类型之后，标记名字典对话框展开，根据所选的标记类型显示一组选项。

公共标记属性

您必须给每个标记指定唯一的名称。可选的注释可以是标记定义的一部分。给定义的每个标记指定适当的注释是一种很好的做法。

所有标记都属于某个报警组，报警组是另一个公共的标记属性。缺省条件下，所有标记都属于 $System 报警组。如需有关将标记指定给其它报警组的详细信息，请参阅配置报警。

标记名惯例

给标记命名时，如果需要许多包含相似属性的标记，请使用一致的命名惯例。

遵循以下这些 InTouch 标记名的命名惯例：

- 标记名不超过 128 个字符。
- 使用字母数字（**A-Z**、**a-z**、**0-9**）作为标记名的第一个字符。

  标记名最好仅使用字母数字字符。
- 标记名至少使用一个英文字符。

  （可选）使用以下特殊字符：

- 连字符	! 感叹号	# 井号
$ 美元号	% 百分号	& 与号
© 问号	@ At 号	_ 下划线

  如果可能，请避免在标记名中使用特殊字符，除非应用程序确实需要。
- 避免在标记名中使用连字符 (-)。

  连字符是 InTouch 标记名的有效字符。但是在逻辑或算术表达式中，InTouch 将连字符用作负号或减号运算符。例如，表达式 A=B-C 可以解释为 A = B 减 C，但也可以说是将名称为 B-C 的标记指定给标记 A。

  以数字开头的标记名不允许使用断字符或减号 (-)。
- 请勿在标记名中使用空格。
- 请勿在标记名中使用可以解释为指数的数字。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

例如，不能将标记命名为 125E4，因为它可以解释成底数的四次方。

- 请勿在标记名中使用可以解释为十六进制数的数字。

  例如，不能将标记命名为 0x123B，因为它可以解释成十六进制数。

自动给标记命名

在“标记名字典”中给标记命名时，InTouch HMI 跟踪所用的命名惯例。如果将标记命名为 Pump01、Pump02，则 InTouch HMI 会建议将下一个标记命名为 Pump03。您可以接受或拒绝此名称。此命名帮助功能称为“设置索引”。

“设置索引”基于标记名中的最后一个连续数字。例如，如果标记名是 PumpInP04LotB99A，则 InTouch HMI 建议将下一个标记命名为 PumpInP04LotB100A，而不是 PumpInP05LotB99A。

标记注释

创建标记时，可以在标记名字典的注释框中，输入最多 160 个字符的可选注释。

第一次访问“标记名字典”时，**$AccessLevel** 系统标记的缺省注释出现在注释框中。删除此注释，以防止它与创建的任何标记关联。

理解标记属性

在指定公共标记属性之后，必须定义正在创建的标记类型所特有的其它属性。下表按照标记类型显示内存标记的基本属性。

标记类型	唯一属性
离散	初始值、打开消息、关闭消息、注释
整型	初始值、最小值、死区、工程单位、最大值、记录死区、注释
实型	初始值、最小值、死区、工程单位、最大值、记录死区、注释
消息	最大长度、初始值、注释

I/O 标记又一些额外的属性，可用于建立网络通讯以及将网络设备上的原始数据转换成 InTouch 应用程序所用的规格化值。如需有关定义 I/O 标记的详细信息，请参阅配置 I/O 标记的属性。

值范围、度量单位及初始值

在 WindowViewer 中启动 InTouch 应用程序时，离散、整型、实型以及消息标记都会指定一个初始值。如果是离散标记，初始值是可能的二值状态之一。对于整型与实型标记，初始值是应用程序启动时与该标记关联的数字。在“标记名字典”中指定初始值。

您可以将初始标记值指定为应用程序在 WindowViewer 中停止运行时标记的最后一个值。通过从“标记名字典”中选择保留值选项，可以将标记的最后一个活动值指定为应用程序再次启动时的初始值。

整型与实型标记包含一些属性，这些属性分别设置指定给标记的可能数值范围的上下限。整型与实型标记均包含定义范围上下限的最小值与最大值属性。

整型与实型标记还包含工程单位属性，可以指定一个工程单位标签来描述标记值的度量单位。例如，您可以将 PSI 指定为与泵压关联的整型标记的工程单位属性。

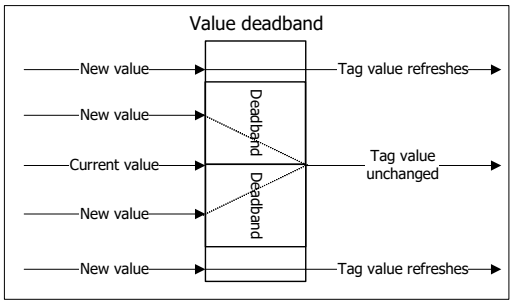
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

标记死区

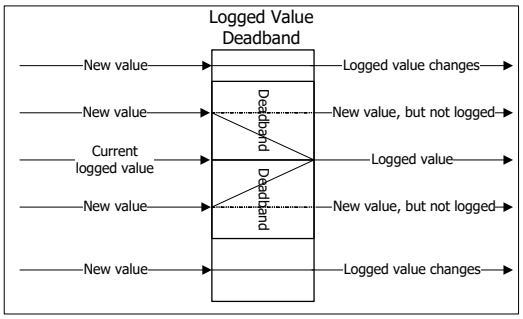
死区是标记值的灵敏度设置。死区通常与其值持续改变的 I/O 标记关联。死区过滤掉标记值中细小的瞬间改变，可以减少 InTouch 的数据处理量。

对于同整型与实型数据关联的标记，“标记名字典”包含两个死区属性。

- 值死区：值死区属性设置一个阈值；必须超过这个阈值，WindowViewer 才会刷新运行时内存中的标记值。下图显示当前标记值周围的绝对死区范围。



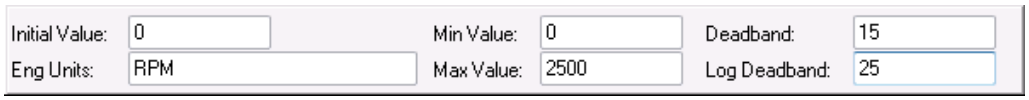
- 记录死区：记录死区设置一个阈值；必须超过这个阈值，才会将标记值写入日志文件。下图显示所记录的标记的当前值周围的死区。



只有死区以外的新标记值才会写入日志文件。死区范围内值的微小改变将会被忽略，而不会记录下来。

设置标记死区

1. 打开标记名字典对话框。
2. 单击选择。此时出现选择标记对话框。当前为应用程序定义的标记会列出来。
3. 从列表中选择整型或实型标记。
4. 选择确定。选择实型或整型标记时，标记名字典对话框的详细资料部分显示一些附加的选项。



5. 根据所选的标记类型，通过在死区框中输入整数或实数来设置值死区。

   值死区设置的是以工程单位为单位的一个绝对阈值水平。
6. 根据所选的标记类型，通过在记录死区框中输入整数或实数来设置记录死区。

   同值死区类似，记录死区设置的是以工程单位为单位的一个绝对阈值。
7. 选择保存以保存对死区所作的更改。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

8. 选择**关闭**以关闭“标记名字典”对话框。

标记值保留

标记名字典的详细资料部分包含两个属性，可以保留**标记值**和操作员对报警限所作的更改。

所有**标记类**型都包含一个**保留值**属性。**选择保留值**可以保留应用程序停止时标记的当前值。再次启动应用程序时，WindowViewer 将保留下来的值用作该标记的初始值。

WindowViewer 再次启动应用程序时，不会将保留的值写入 I/O 设备。在“I/O 服务器”初次扫描提供数据的设备之后，I/O 值便会更新。

如果 WindowViewer 在运行，则无法为新的标记或现有的标记选择或清除**保留值选项**。选择此选项时，标记的初始值会不断更新，以反映它的当前值。WindowViewer 停止时，初始值设置为最后一个保留值。如果随后清除了此选项，则该标记的初始值设置为最后一个保留值。

整型与实型标记包含**保留参数**属性。**选择保留参数**可以保留应用程序运行时操作员对标记的报警限所作的任何更改。重新启动应用程序时，WindowViewer 将修改后的报警限用作报警限的初始值。

I/O 连接

所有类型的 I/O 标记都必须指定外部数据源的<1>“</1>访问名<2>”</2>与<3>“</3>项目名<4>”。</4> 如需有关为 I/O 标记指定“访问名”与“项目名”的详细信息，请参阅设置 I/O 访问参数。

标记记录

在运行时，每次标记值的改变超过指定的记录死区时，WindowViewer 便可以将一个项目写入历史日志文件。WindowViewer 也可以按照固定的间隔将一些项目写入日志，而不论当前标记值如何。缺省条件下，这个固定间隔是一小时。

注意：如果需要更加可控、功能更多样的记录功能，可以考虑使用 Historian 来存储 InTouch 历史数据。

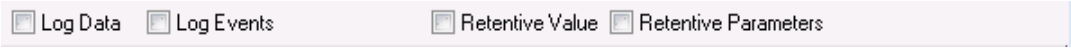
标记名字典对话框包含一些单独的选项，可以将数据与事件记录到日志文件中。您可以设置一些**值记录选项**。如需有关设置事件记录功能的信息，请参阅配置报警。

对于要写入历史日志文件的标记值，必须启用历史记录。如需有关设置常规记录属性的详细信息，请参阅配置历史记录。

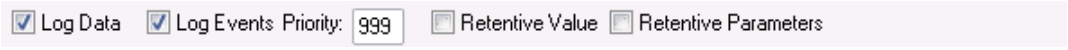
对于整型与实型标记，可以在它们各自的详细信息对话框中设置“记录死区”。**记录死区选项**指定标记的值必须改变多少个工程单位，才会写入一个日志项。

为标记配置记录功能

1. 打开**标记名字典**。
2. 选择要将其数据保存到日志文件的标记。
3. 选择**记录数据**。



4. 如果希望记录由操作员、I/O、QuickScript 或操作系统对标记值所作的改变，请选择**记录事件**。在选择**记录事件**之后，会出现**优先级**框。



优先级值确定标记的事件优先级。有效值是 1 到 999；其中 1 是最高优先级，999 是最低优先级。

5. 选择**保存**，然后关闭“标记名字典”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建离散标记

您可以指定离散标记，以显示在 InTouch 应用程序中运行的内部过程的二进制状态。离散标记必须指定一个初始值，即打开或关闭。此外，也可以指定过程与进入或脱离某种报警状态的标记转换关联时，在报警事件窗口中出现的消息。

以下步骤显示如何定义内存离散标记。I/O 离散标记指出可编程控制器、过程计算机的所有输入与输出以及网络节点数据的二进制状态。

如需有关设置 I/O 离散标记属性的详细信息，请参阅指定离散 I/O 标记。

定义内存离散标记的初始值与消息

1. 在标记类型对话框中，选择内存离散作为标记的类型。
2. 选择标记名字典对话框顶部的详细，以显示详细选项。此时出现标记名字典对话框的详细信息部分。

3. 选择打开或关闭作为与标记关联的初始值。应用程序启动时，标记设置为这个初始值。
4. 在打开消息与关闭消息框中输入标记进入或脱离某种报警状态时出现的消息。

   这些消息可用在任何动画链接或脚本中，而不论该标记是否配置了报警。

   - 如果定义在标记值等于 1 (On, True) 时处于活动状态的离散报警，则在打开消息框中输入的消息会出现在 ActiveX 报警显示的值与报警限列中。

     标记的报警状态恢复正常时，在关闭消息框中输入的消息会出现在值列中，“打开消息”保持在报警限列中。

   - 如果定义在标记值等于 0 (Off, False) 时处于活动状态的离散报警，则在关闭消息框中输入的消息会出现在 ActiveX 报警显示的值与报警限列中。

     标记的报警状态恢复正常时，在打开消息框中输入的消息会出现在值列中，“关闭消息”保持在报警限列中。

5. 将更改保存到标记中。

创建整型与实型标记

您可以指定整型与实型标记以显示 InTouch 应用程序中运行的过程的数值。

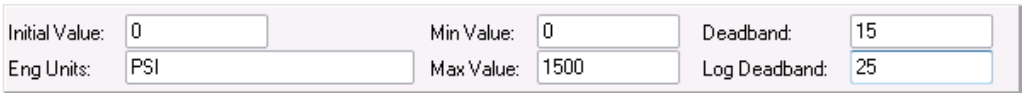
以下步骤显示如何定义内存与 I/O 整型和实型标记。内存整型与实型标记必须指定一个初始值。您还必须设置最小与最大数据范围。

如需有关设置 I/O 整型与实型标记的 I/O 属性的信息，请参阅指定整型与实型 I/O 标记。

要点：InTouch 实数的精度限制为最多八位。要避免可能发生的舍入错误，指定实型标记的属性时，不要超过八位精度。如需详细信息，请参阅 IEEE 十进制单位。

定义内存整型与实型标记值

1. 在标记类型对话框中，将内存整型或内存实型指定为标记的类型。此时出现标记名字典对话框的详细信息部分。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 设置整型与实型标记的属性。执行以下操作：
   - 在初始值框中，输入应用程序启动时与标记关联的整数或实数。
   - 在最小值框中，输入标记的最小整数或最小实数。

     最小值设置与内存整型与实型标记关联的数字可能的最小值。
   - 在最大值框中，输入标记的最大整数或最大实数。

     最大值设置与内存整型与实型标记关联的数字可能的最大值。
   - 在工程单位框中，输入要用作标记工程单位的标签。
3. 将更改保存到标记中。

如需有关设置标记的死区与记录死区属性的详细信息，请参阅标记死区。

创建消息标记

您可以为内部与外部过程指定消息标记。这些标记包含一些属性，可以指定 WindowViewer 启动应用程序时出现的初始消息，以及可以从 Alarm Viewer 读取的注释。

如需有关定义 I/O 消息标记的详细信息，请参阅指定消息型 I/O 标记。

定义内存消息标记

1. 在标记类型对话框中，将内存消息指定为标记的类型。
2. 如果需要，选择详细以显示标记名字典对话框的详细资料部分。

3. 设置内存消息标记的属性。执行以下操作：
   - 在最大长度框中，输入一个整数，它是标记的消息中可以出现的最大字符数。或者，接受缺省长度 131 个字符，即消息的最大长度。
   - 在初始值框中，输入要在 WindowViewer 启动应用程序时指定给标记的消息文本。
   - 如果选择了消息标记的记录事件选项，则在报警注释框中，输入可以在 AlarmViewer 控件中读取的消息。
4. 将更改保存到标记中。

创建 I/O 标记

I/O 标记包含一组公共属性，可以指定 InTouch 应用程序与外部过程之间的网络连接。标记名字典对话框的详细资料部分包含一些选项，可用于设置 I/O 标记的外部属性。

如需有关设置 I/O 属性的详细信息，请参阅使用 I/O 进行数据访问。

修改标记

修改标记同创建标记类似。您从<1>“</1>标记名字典<2>”</2>中选择要修改的标记<3>。</3> 然后，执行与创建标记相同的步骤来更改标记的属性。

要点：在 InTouch 应用程序中使用了标记之后，要再修改它的类型并不容易。可选择的标记类型可能要局限于类似的数据类型。定义标记时，应该仔细选择正确的数据类型。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

修改标记

1. 打开标记名字典对话框。
2. 单击选择。此时出现选择标记对话框。当前为应用程序定义的一列标记会显示出来。
3. 从列表中选择要修改的标记。
4. 选择确定。此时标记名字典对话框显示为所选标记指定的值。
5. 更改标记的属性。
6. 选择保存以使用所作的更改来更新标记。
7. 选择 **关闭** 以关闭标记名字典。

从 OPC UA 服务器创建 InTouch 标记

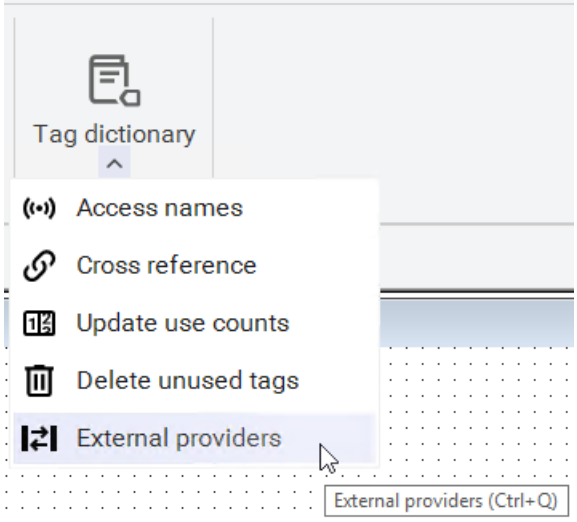
标记字典下的外部供应器选项列出了在同一台计算机上的 Gateway Communication Driver 中配置的所有 OPC UA 服务器连接。使用此选项，可以避免多个手动步骤，轻松地为 OPC UA 项目引用创建访问名和 InTouch 标记。

从 Gateway Communication Driver 中配置的 OPC UA 服务器创建 InTouch 标记：

前提条件：

- 在 Gateway Communication Driver 中配置 OPC UA 服务器连接。如需详细信息，请参阅 Gateway Communication Driver 帮助。
- Gateway Communication Driver 和 InTouch WindowMaker 应位于同一台计算机上。

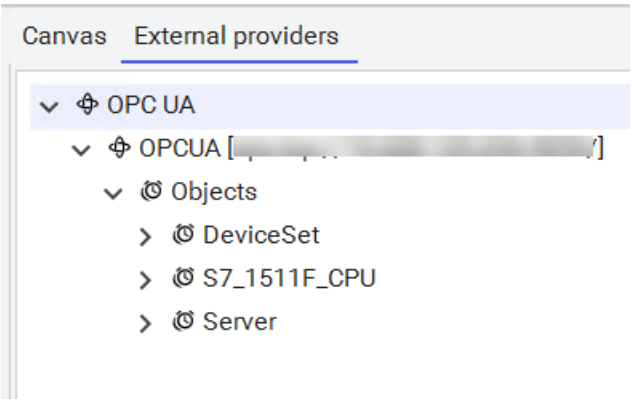
1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在主页菜单的标记组中，使用标记字典下面的下拉箭头选择外部供应器。或者，可以使用键盘快捷键 **Ctrl + Q**。



此时会显示在 Gateway Communication Driver 中配置的所有 OPC UA 服务器连接。

3. 您可以展开 OPC UA 树来查看和浏览项目引用。

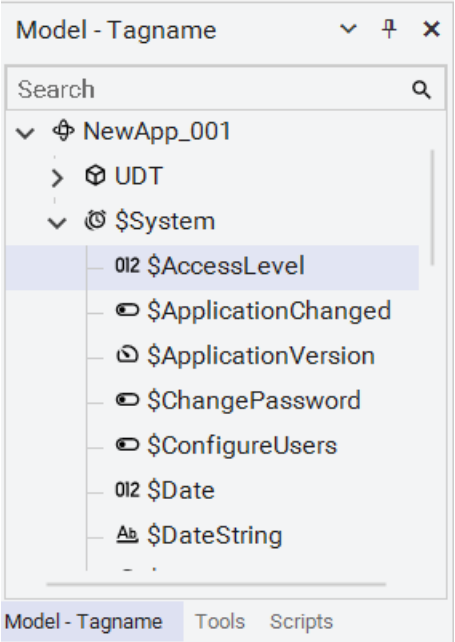
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



注意：
不会显示远程 Gateway Communication Driver 中的 OPC UA 服务器。
在显示 OPC UA 项目后，如果您配置另一个 OPC UA 服务器，请选择外部供应器窗口底部的刷新按钮，以查看新添加的 OPC UA 服务器项目。

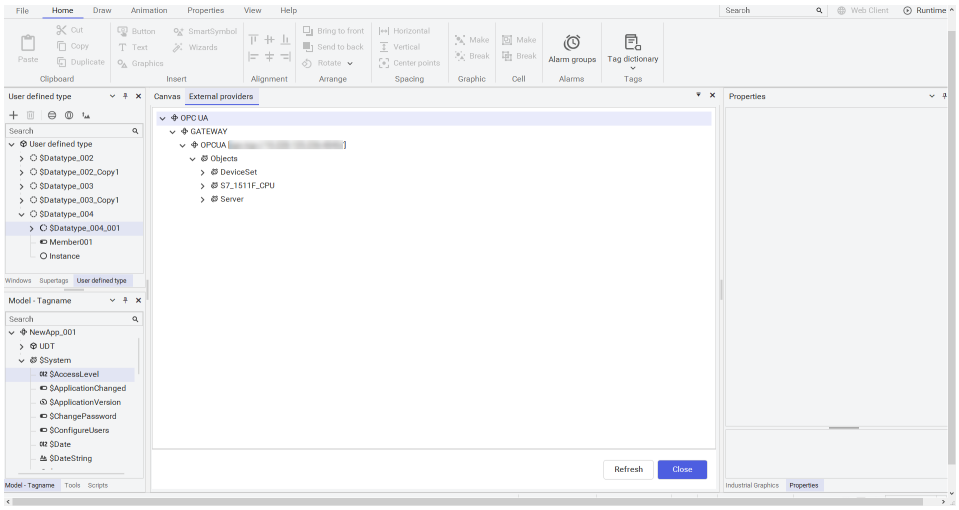
4. 打开模型 - 标记名窗格。

   模型 - 标记名窗格显示 InTouch 应用程序的所有报警组和标记。如需有关“模型 - 标记名”窗格的详细信息，请参阅模型 - 标记名主题。



5. 将项目从外部供应器窗口拖放到“模型 - 标记名”中。您可以通过在选择项目时按住 **Ctrl** 键来选择多个项目。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



- 您可以拖放到所需的**报警组**节点，以在特定**报警组**下添加 OPC UA 标记。
- 如果随意进行拖放，将会在 **$System** 节点下创建标记。

  拖放项目后，即会创建 OPC UA 标记，您可以在**模型 - 标记名**窗格中看到新创建的 OPC UA 标记。

1. 要关闭外部供应器窗口，请选择**关闭**。

附加信息：

- 对于不同的 OPC UA 连接，会添加不同的访问名。标记的访问名将成为 Gateway Communication Driver 中 OPC UA 连接的节点名。

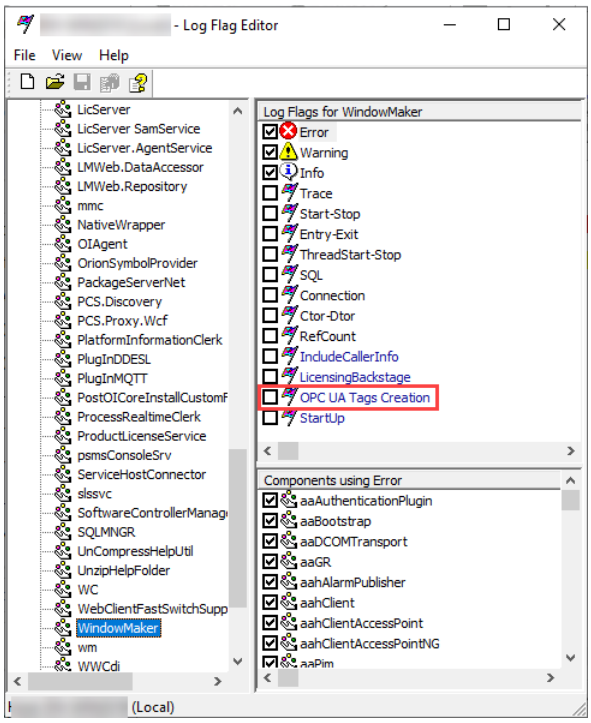
  注意：在添加标记后，如果在 Gateway Communication Driver 中重命名 OPC UA 节点，那么必须手动更改访问名，它不会自动更改。

- 重复的标记名会附加索引号。例如：如果有两个名称为“Input”的标记，那么第一个标记将命名为“Input”，而第二个标记将命名为“Input_1”。

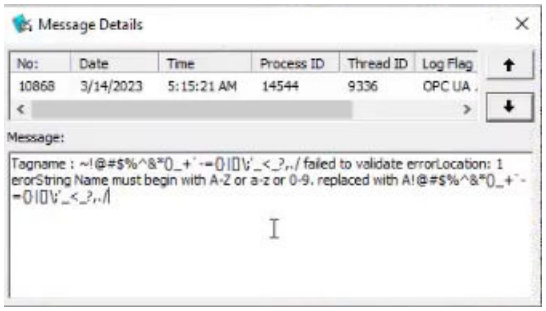
  注意：如果标记名长度超过最大标记长度 128，将删除标记名末尾的字符，以使其长度为 128。当标记名长度超过 128 个字符，并且存在重复的标记名时，将删除标记名末尾的字符数，以便标记名 +_{index} 长度为 128。

- 如果 OPC UA 标记的数据类型未知，将会创建相应的 InTouch 标记作为 I/O 消息类型。
- 您还可以通过在“应用程序管理器”中将 InTouch 配置为 OPC UA 服务器来将其用作 OPC UA 服务器。如需详细信息，请参阅配置和使用 InTouch OPC UA 服务器主题。
- 在日志标帜编辑器中的 WindowMaker 下，如果选中 **OPC UA 标记创建复选**框，那么可以在 Log Viewer 中查看与 OPC UA 标记相关的日志消息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



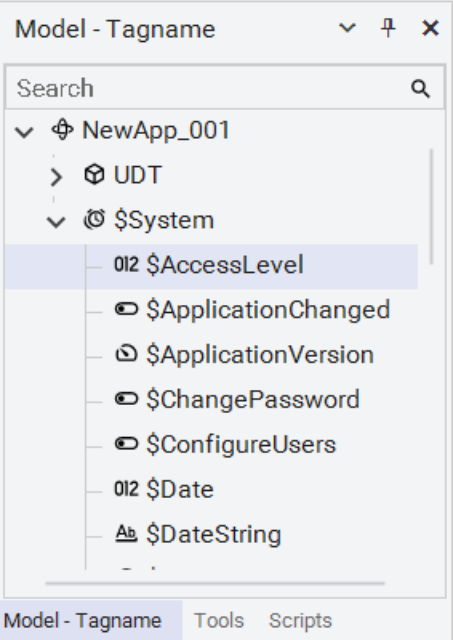
- 如果 OPC UA 项目名含有不支持的特殊字符，在创建相应的 InTouch 标记时，标记名中不支持的特殊字符将替换为“A”或“_”。如果不支持的字符位于标记的开头，它将替换为“A”，在其它位置时将替换为“_”。您可以在 Log Viewer 的消息详细信息窗口中查看有关被替换字符的详细信息。



模型 - 标记名

模型 - 标记名选项卡与“窗口”和 Supertag 选项卡一起位于 WindowMaker 的左侧。选择模型 - 标记名选项卡以查看“模型 - 标记名”窗格。它显示 InTouch 应用程序的所有报警组和标记。您可以使用搜索框来搜索所需的标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



打开标记

1. 在模型 - 标记名窗格中，使用鼠标右键单击要打开的标记。
2. 选择打开。

   该标记将在标记名字典窗口中打开。

将标记从一个报警组移动到另一个报警组

1. 在模型 - 标记名窗格中，选择要移动的所需标记。
2. 拖放到所需的报警组。

   所选标记将会移动到所需的报警组。

删除标记

1. 在模型 - 标记名窗格中，使用鼠标右键单击要删除的标记。

   您可以选择多个标记。
2. 选择删除。
3. 在确认删除标记的消息中，选择是。

   此时将删除所选标记。

删除标记

系统会为给 InTouch 应用程序定义的所有标记维护一个计数。删除包含动画链接或脚本的窗口时，标记计数不会自动减少。与删除的窗口关联的标记被视作仍在使用，且无法删除。

要删除不再使用的标记，必须关闭 WindowViewer，并更新本地与远程标记的使用计数。通过使用 InTouch 的“交叉引用”实用程序，可以确定标记的使用位置。如需有关使用“交叉引用实用程序”与更新标记计数的详细信息，请参阅减少标记使用。

在更新使用计数之后，可以删除标记。如需有关详细信息，请参阅删除未使用标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

打印标记列表与使用信息

在 WindowMaker 中，可以使用“WindowMaker 打印输出”实用程序打印 InTouch 应用程序数据库、窗口及脚本的内容。

如需有关详细信息，请参阅保存与打印标记交叉引用列表。

使用标记点域来查看或更改标记属性

每种类型的 InTouch 标记都有一组独特的属性，这些属性描述与标记关联的数据或可能的条件。点域确定标记属性。<1>“</1>标记名字典<2>”</2>中显示的差不多每个标记属性都有点域<3>。</3> 某些点域是每种类型的 InTouch 标记所共有的。例如，.Name 点域总是与标记的名称关联。另一些点域则仅适用于特定标记类型的独特属性。

通过在脚本、表达式或用户输入中使用点域，可以在应用程序运行期间监视与修改标记的属性。下例显示在脚本或表达式中使用点域来访问标记属性的语法。

tag_name.property_dotfield

例如，要让操作员能够在应用程序运行期间更改 HiHi 报警限，可以创建一个“模拟 - 用户输入”触动链接。然后，将该链接应用于使用 Analog_Tag.HiHiLimit 点域表达式所定义的按钮。在运行期间，操作员可以选择该按钮，并为标记输入新的 HiHi 报警限值。

通过使用点域，可以输入和输出与标记关联的数据；此外，通过使用历史点域，可以修改正在运行的应用程序当前显示的历史趋势。例如，您可以在脚本中使用点域，让操作员能够修改历史趋势滚动方式、锁定或调整趋势上指示器的位置，或是将笔重新指定给新的标记。

标记类型可用的点域

每种类型的 InTouch 标记都有一组与它的独特属性关联的点域。下表按字母顺序列出所有标记类型的点域。

	标记类型													
	内存				I/O				间接			其它		
点域	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件
.Ack	•	•	•		•	•	•		•	•		•		
.AckDev		•	•			•	•			•		•		
.AckDsc	•				•				•			•		
.AckROC	•	•	•			•	•			•		•		
.AckValue		•	•			•	•			•		•		

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	
.Alarm	•	•	•		•	•	•		•	•		•			
.AlarmAccess														•	
.AlarmAckModel	•	•	•		•	•	•		•	•					
.AlarmClass														•	
.AlarmComment	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
.AlarmDate														•	
.AlarmDev		•	•			•	•			•		•			
.AlarmDevCount		•	•			•	•			•		•			
.AlarmDevDeadband		•	•			•				•					
.AlarmDevUnAckCount		•	•			•	•			•		•			
.AlarmDisabled	•	•	•		•	•	•		•	•		•			
.AlarmDsc	•				•				•			•			
.AlarmDscCount	•				•				•			•			
.AlarmDscDisabled	•				•				•			•			
.AlarmDscEnabled	•				•				•			•			
.AlarmDscInhibitor	•				•				•			•			
.AlarmDscUnAckCount	•				•				•			•			
.AlarmEnabled	•	•	•		•	•	•		•	•		•			
.AlarmGroup														•	
.AlarmGroupSel														•	
.AlarmHiDisabled		•	•			•	•			•					

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	
.AlarmHiEnabled		•	•			•	•			•					
.AlarmHiHiDisabled		•	•			•	•			•					
.AlarmHiHiEnabled		•	•			•	•			•					
.AlarmHiHiInhibitor		•	•			•	•			•					
.AlarmHiInhibitor		•	•			•	•			•					
.AlarmLimit														•	
.AlarmLoDisabled		•	•			•	•			•					
.AlarmLoEnabled		•	•			•	•			•					
.AlarmLoInhibitor		•	•			•	•								
.AlarmLoLoDisabled		•	•			•	•			•					
.AlarmLoLoEnabled		•	•			•	•			•					
.AlarmLoLoInhibitor		•	•			•	•			•					
.AlarmMajDevDisabled		•	•			•	•			•					
.AlarmMajDevEnabled		•	•			•	•			•					
.AlarmMajDevInhibitor		•	•			•	•			•					
.AlarmMinDevDisabled		•	•			•	•			•					
.AlarmMinDevEnabled		•	•			•	•			•					
.AlarmMinDevInhibitor		•	•			•	•			•					
.AlarmName														•	

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	
.AlarmOprName														•	
.AlarmOprNode														•	
.AlarmPri														•	
.AlarmProv														•	
.AlarmROC		•	•			•	•			•					
.AlarmROCCount		•	•			•	•			•					
.AlarmROCDisabled		•	•			•	•			•					
.AlarmROCEnabled		•	•			•	•			•					
.AlarmROCInhibitor		•	•			•	•			•					
.AlarmROCUnAckCount		•	•			•	•			•					
.AlarmState														•	
.AlarmTime														•	
.AlarmTotalCount	•	•	•		•	•	•		•	•		•			
.AlarmType														•	
.AlarmUnAckCount	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum1	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum1Set	•	•	•		•	•	•		•	•		•			
.AlarmUserDefNum2	•	•	•		•		•		•	•		•			
.AlarmUserDefNum2Set	•	•	•		•	•	•		•	•		•			

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	
.AlarmUserDefStr	•	•	•		•	•	•		•	•		•			
.AlarmUserDefStrSet	•	•	•		•	•	•		•			•			
.AlarmValDeadband		•	•			•	•			•					
.AlarmValue			•				•			•		•		•	
.AlarmValueCount		•	•			•	•			•		•			
.AlarmValueUnAckCount		•	•			•	•			•		•			
.Caption														•	
.ChartLength													•		
.ChartStart													•		
.Comment	•	•	•	•	•	•	•	•	•	•	•	•	•		•
.DevTarget		•	•			•	•			•					
.DisplayMode													•		
.Enabled														•	
.EngUnits		•	•			•	•			•					
.Freeze														•	
.HiHiLimit		•	•			•	•			•					
.HiHiSet		•	•			•	•			•					
.HiHiStatus		•	•			•	•			•					
.HiLimit		•	•			•	•			•					
.HiSet		•	•			•	•			•					
.HiStatus		•	•			•	•			•					

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	
.ListChanged														•	
.ListCount														•	
.ListIndex														•	
.LoLimit		•	•			•	•			•					
.LoLoLimit		•	•			•	•			•					
.LoLoSet		•	•			•	•			•					
.LoLoStatus		•	•			•	•			•					
.LoSet		•	•			•	•			•					
.LoStatus		•	•			•	•			•					
.MajorDevPct		•	•			•	•			•					
.MajorDevSet		•	•			•	•			•					
.MajorDevStatus		•	•			•	•			•					
.MaxEU		•	•			•	•			•					
.MaxRange													•		
.MaxRaw		•	•			•	•								
.MinEU		•	•			•	•			•					
.MinorDevPct		•	•			•	•			•					
.MinorDevSet		•	•			•	•			•					
.MinorDevStatus		•	•			•	•			•					
.MinRange													•		
.MinRaw		•	•			•	•			•					
.Name	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	
.NewIndex														•	
.NextPage														•	
.Normal	•	•	•		•	•	•		•	•		•			
.NumAlarms														•	
.OffMsg	•				•				•						
.OnMsg	•				•				•						
.PageNum														•	
.Pen1 至 .Pen8													•		
.PendingUpdates														•	
.PrevPage														•	
.PriFrom														•	
.PriTo														•	
.Quality	•	•	•	•	•	•	•	•	•	•	•				
.QualityLimit	•	•	•	•	•	•	•	•	•	•	•				
.QualityLimitString	•	•	•	•	•	•	•	•	•	•	•				
.QualityStatus	•	•	•	•	•	•	•	•	•	•	•				
.QualityStatusString	•	•	•	•	•	•	•	•	•	•	•				
.QualitySubstatus	•	•	•	•	•	•	•	•	•	•	•				
.QualitySubstatusString	•	•	•	•	•	•	•	•	•	•	•				
.QueryState														•	
.QueryType														•	
.RawValue	•	•	•		•	•	•		•	•		•			

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分析计报警对象/控件	
.ReadOnly														•	
.Reference	•	•	•	•	•	•	•	•	•	•	•				
.ReferenceComplete	•	•	•	•	•	•	•	•	•	•	•				
.ROCPct		•	•			•	•			•					
.ROCSet		•	•			•	•								
.ROCStatus		•	•			•	•			•					
.ScooterLockLeft													•		
.ScooterLockRight													•		
.ScooterPosLeft													•		
.ScooterPosRight													•		
.Successful														•	
.SuppressRetain														•	
.TagID	•	•	•		•	•	•		•	•					
.TimeDate	•	•	•	•	•	•	•	•	•	•	•				
.TimeDateString	•	•	•	•	•	•	•	•	•	•	•				
.TimeDateTime	•	•	•	•	•	•	•	•	•	•	•				
.TimeDay	•	•	•	•	•	•	•	•	•	•	•				
.TimeHour	•	•	•	•	•	•	•	•	•	•	•				
.TimeMinute	•	•	•	•	•	•	•	•	•	•	•				
.TimeMonth	•	•	•	•	•	•	•	•	•	•	•				
.TimeMsec	•	•	•	•	•	•	•	•	•	•	•				
.TimeSecond	•	•	•	•	•	•	•	•	•	•	•				

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	标记类型														
	内存				I/O				间接			其它			
	离散	整型	实型	消息	离散	整型	实型	消息	离散	模拟	消息	报警组	历史趋势	分布式报警对象/控件	
.TimeTime*	•	•	•	•	•	•	•	•	•	•	•				
.TimeTimeString	•	•	•	•	•	•	•	•	•	•	•				
.TimeYear	•	•	•	•	•	•	•	•	•	•	•				
.TopIndex														•	
.TotalPages														•	
.UnAck	•	•	•		•	•	•		•	•		•			
.UpdateCount													•		
.UpdateinProgress													•		
.UpdateTrend													•		
.Value(Tagname)	•	•	•	•	•	•	•	•	•	•	•	•			•
.Value（窗口控件）														•	
.Visible														•	

点域可以按其目标功能进行分类。如需有关点域功能类别的详细信息，请参阅以下几节：

类别	请参阅。
值与极限	更改标记的值极限。
报警参数	《AVEVA™ InTouch HMI 应用程序运行时》中的在运行时控制标记与组的报警属性。
I/O	使用 I/O 进行数据访问。
分布式报警对象	*使用分布式报警显示对象*。
趋势显示	绘制标记数据的趋势。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别	请参阅 。
窗口控件	向导

更改标记的值极限

来自“I/O 服务器”的原始输入数据会转换为适合 InTouch 应用程序的值范围。标记的值限定在由最小值与最大值确定的范围之间，它们分别由“标记名字典”的中最小原始数据与最大原始数据属性指定。

随后，这些原始值会转换成由最小工程单位与最大工程单位 选项设置的工程单位范围。您可以使用一组点域来监视与修改标记的原始值与工程单位范围。

下表列出在应用程序运行期间监视或更改标记值的点域。

点域	可读写	显示
.MinRaw	只读	从“I/O 服务器”收到的原始值的低钳位设置。
.MaxRaw	只读	从“I/O 服务器”收到的原始值的高钳位设置。
.MinEU	只读	指定给标记的最小工程单位值。
.MaxEU	只读	指定给标记的最大工程单位值。
.EngUnits	可读写	从“标记名字典”的工程单位选项中给模拟标记指定的文本值。
.RawValue	只读	标记从“I/O 服务器”收到的实际离散或模拟值（未应用缩放）。
.Value	可读写	标记的当前值。
.OnMsg	可读写	离散标记的求值结果为 True、On 或 1 时指定给离散标记的消息。
.OffMsg	可读写	离散标记的求值结果为 False、Off 或 0 时指定给离散标记的消息。
.Comment	可读写	从<1>“</1>标记名字典<2>”</2>中指定的标记注释<3>。</3>

查看原始值极限

对于来自“I/O 服务器”的原始输入标记值，在 InTouch 应用程序中使用之前，可能需要进行钳位处理。钳位功能将原始值限制到所定义的上、下限范围内。**.MinRaw** 和 **.MaxRaw** 点域显示原始输入范围的上、下边界。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.MinRaw 点域

.**MinRaw** 点域显示给标记指定的“最小原始数据”下钳位设置。.**MinRaw** 点域的值来自在“标记名字典”中给 I/O 标记指定的最小原始数据值。任何低于此设置的原始值都会调整到这个最小值。

类别

标记

用法

```
tag_name.MinRaw;
```

参数

Tag_name
任何 I/O 整型、I/O 实型、及间接模拟标记的名称。

附注

这个只读的点域显示给“最小原始数据”下钳位设置指定的值。

数据类型

实型或整型（只读）。

有效值

任何模拟值。

示例

如果与泵入口压力关联的原始值超出由标记的最小原始数据与最大原始数据属性所设置的上、下值边界，则以下脚本显示一个错误窗口。

```
IF ((PumpInP.RawValue > PumpInP.MaxRaw) OR
   (PumpInP.RawValue < PumpInP.MinRaw)) THEN
   Show "Instrument Failure Window";
ENDIF;
```

另请参阅

.EngUnits, **.MinEU**, **.MaxEU**, **.MaxRaw**, **.RawValue**

.MaxRaw 点域

.**MaxRaw** 点域显示从“标记名字典”中的最大原始数据属性指定给 I/O 标记的“最大原始数据”的上钳位设置。任何超出此设置的原始数据值均限定为这个最大原始数据值。

类别

标记

用法

```
Tag_name.MaxRaw
```

参数

Tag_name
任何 I/O 整型、I/O 实型、及间接模拟标记的名称。

附注

这个只读的点域显示指定给“最大原始数据”上钳位设置的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

实型或整型（只读）。

有效值

任何模拟值。

示例

此脚本确定标记值是否超出正常操作范围，如果发生这种情况，则显示一个窗口。

```
IF ((Temp01.RawValue > Temp01.MaxRaw) OR (Temp01.RawValue <
   Temp01.MinRaw))THEN
   Show "Instrument Failure Window";
ENDIF;
```

另请参阅

.EngUnits, **.MinEU**, **.MaxEU**, **.MinRaw**, **.RawValue**

查看标记的原始值

.**RawValue** 点域显示从“I/O 服务器”接收的所监视属性的实际离散或模拟值。原始值是在应用钳位与缩放来将值规格化为标记工程单位之前的实际输入值。

.RawValue 点域

.**RawValue** 点域显示 WindowViewer 从“I/O 服务器”接收的实际值。.RawValue 点域让您可以访问在 InTouch 应用缩放之前的 I/O 标记值。

类别

标记

用法

```
Tag_name.RawValue
```

参数

Tag_name
任何 I/O 整型、I/O 实型、I/O 离散、间接离散、及间接模拟标记的名称。

附注

这个只读点域用于显示 InTouch 应用缩放之前的实际离散或模拟 I/O 值。

数据类型

任何适合与 .RawValue 点域关联的标记类型的数据。例如，实型标记使用实数，离散标记使用离散值（只读）。

示例

原始泵入口压力低于或高于标记的最小与最大钳位极限时，以下脚本发出一则警告消息。

```
IF ((PumpInP.RawValue > PumpInP.MaxRaw) OR (PumpInP.RawValue < PumpInP.MinRaw)) THEN
   AlarmMessage = "Pump sensor is out of calibration or requires replacement.";
ENDIF;
```

另请参阅

.EngUnits, **.MinEU**, **.MaxEU**, **.MinRaw**, **.MaxRaw**

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

查看工程单位值极限

来自“I/O 服务器”的值在第一次进入 WindowViewer 时被视为原始数据。原始值可能需要缩放。InTouch HMI 在原始数据限定的输入值上执行算术转换，以便将它们缩放到标记的工程单位范围内。I/O 整型与实型标记类型包含最小工程单位与最大工程单位属性，分别显示工程单位范围的上、下边界。

.MaxEU 点域

.MaxEU 点域显示在“标记名字典”中给指定的标记所指定的最大工程单位值。

类别

标记

用法

```
Tag_name.MaxEU
```

参数

Tag_name
任何整型、实型或间接模拟标记。

附注

.MaxEU 点域用于将原始数据值缩放到给标记定义的工程单位范围内。它定义工程单位范围的上限。

数据类型

实型标记使用实型，整型标记使用整型（只读）。

有效值

依赖于指定的标记的类型。

示例

现场有一个“可编程逻辑控制器”来读取液位仪的数据。液位传送器发送 4 至 20 mA 范围内的一个信号。PLC 将这个信号转换为 0 到 4095 之间的整数值。这个值随后指定给 TankTwoLevel 标记。

显示原始数据值（0 到 4095 之间）并不能向操作员提供有用的数据。必须将此值缩放到适当的工程范围。

要完成这项工作，必须正确设置“最小工程单位”与“最大工程单位”字段。在本例中，如果原始数据值 0（现场读数是 4mA）转换为“0 加仑”，4095 这个值（现场读数是 20mA）转换为“100 加仑”，则需要进行以下设置以便在屏幕上显示正确的值：

```
TankTwoLevel.MinRaw = 0;
TankTwoLevel.MaxRaw = 4095;
TankTwoLevel.MinEU = 0;
TankTwoLevel.MaxEU = 100;
```

通过这些设置，现场的原始数据值为 4095 时，显示给操作员的值为 100。

另请参阅

.EngUnits, **.MinEU**, **.MinRaw**, **.MaxRaw**, **.RawValue**

.MinEU 点域

.MinEU 点域显示在“标记名字典”中给指定的标记所指定的最小工程单位值。

类别

标记

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用法

```
Tag_name.MinEU
```

参数

Tag_name
任何整型、实型或间接模拟标记。

附注

.MinEU 点域用于将原始数据值缩放到为标记定义的工程单位范围内。它定义工程单位范围的下限。

数据类型

实型标记使用实型，整型标记使用整型（只读）。

有效值

依赖于指定的标记的类型。

示例

本例将给 Tag1 标记定义的工程单位范围指定给 **AbsoluteTagRange** 标记。

```
AbsoluteTagRange = (Tag1.MaxEU - Tag1.MinEU);
```

另请参阅

.EngUnits, **.MaxEU**, **.MinRaw**, **.MaxRaw**, **.RawValue**

更改标记的工程单位

您可以将点域与标记关联起来，以便确定指定给标记的工程单位的文本值。

.EngUnits 点域

.**EngUnits** 点域显示指定给有工程单位属性的模拟标记的文本值。.**EngUnits** 点域将工程单位显示为文本值。

备注：写入此点域的值不可保留。

类别

标记

用法

```
Tag_name.EngUnits
```

参数

Tag_name
任何整型、实型或间接模拟标记。

数据类型

消息（可读写）。

附注

.**EngUnits** 点域不影响与标记关联的实际数据的缩放、转换或格式。

有效值

任意包含 0 到 31 个字符的字符串。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

如果标记的工程单位是摄氏度，则以下脚本调用华氏度转换函数。

```
IF Temperature.EngUnits == "Celsius" THEN
   CALL TempFConvert(Temperature);
ENDIF;
```

另请参阅

.MinEU, **.MaxEU**, **.MinRaw**, **.MaxRaw**, **.RawValue**

按工程单位的格式查看标记值

应用程序中没有明确指定点域时，给 InTouch 标记指定缺省的点域。

.Value 点域

.**Value** 点域按工程单位显示指定的标记的当前值。.**Value** 是隐式应用于所有标记的缺省 InTouch 点域。如果没有给标记指定点域，则缺省使用 .**Value** 点域。

类别

标记

用法

```
tag_name.Value
```

参数

tag_name
除“历史趋势”标记之外的任意类型的标记。

附注

很少需要使用**.Value** 点域。不过在有些情况下，它可以使得计算或参数的使用更加明确。

数据类型

与指定的标记的类型相同（可读写）。

有效值

依赖于指定的标记的类型。

示例

下面的语句将内存整型 **PumpRPM** 标记的值设为等于 100：

```
PumpRPM.Value=100;
```

这在功能上等价于：

```
PumpRPM=100;
```

查看或更改离散标记消息

.**OnMsg** 和 .**OffMsg** 点域显示从“标记名字典”中给离散标记的开或关状态指定的消息。离散标记的打开与关闭消息是最多包含 15 个字符的短字符串。

.OnMsg 点域

.**OnMsg** 点域让您可以访问从“标记名字典”中给离散标记指定的“打开消息”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

标记

用法

```
Tag_name.OnMsg
```

参数

Tag_name
任意离散标记。

数据类型

消息（可读写）。写入此点域的值不可保留。

有效值

任意包含 0 到 15 个字符的字符串。

示例

如果给间接 Ind**PumpState** 标记的“打开消息”指定的字符串值是 "Pump1 running"，则下面的语句发出一条消息。

```
IF IndPumpState.OnMsg == "Pump1 running" THEN
   TypeOfTag = "The IndPumpState tag is assigned to Pump1.";
ENDIF;
```

另请参阅

.OffMsg

.OffMsg 点域

.**OffMsg** 点域让您可以访问从“标记名字典”中给离散标记指定的“关闭消息”。

类别

标记

用法

```
Tag_name.OffMsg
```

参数

Tag_name
任意离散标记。

数据类型

消息（可读写）。写入此点域的值不可保留。

有效值

任意包含 0 到 15 个字符的字符串。

示例

下面的语句根据 **MyDiscrete** 标记的状态将适当的字符串指定给 **StateMessage** 标记。

```
StateMessage=Dtext (MyDiscrete, MyDiscrete.OnMsg, MyDiscrete.OffMsg);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

.OnMsg

查看或更改标记的注释

只有从<1>“</1>标记名字典<2>”</2>中才能对标记注释作出永久更改<3>。</3> 您可以在应用程序运行时使用 .**Comment** 点域分配其他注释。这仅更改运行时会话期间的注释。它不会在“标记名字典”中永久性更改该标记的注释。关闭并重新启动 WindowViewer 之后，便会将原始注释指定给该标记。

.Comment 点域

.**Comment** 点域显示从“标记名字典”中给标记指定的注释。标记注释可以是最多包含 160 个字符的字符串。

类别

标记

用法

```
Tag_name.Comment
```

参数

Tag_name
任意标记名。

附注

在 InTouch 应用程序运行期间，使用 .Comment 点域可以修改与标记关联的注释。在应用程序停止之后，仍然将从“标记名字典”中指定的原始注释指定给标记。

数据类型

消息

有效值

任意 1 到 160 个字符的字符串。

示例

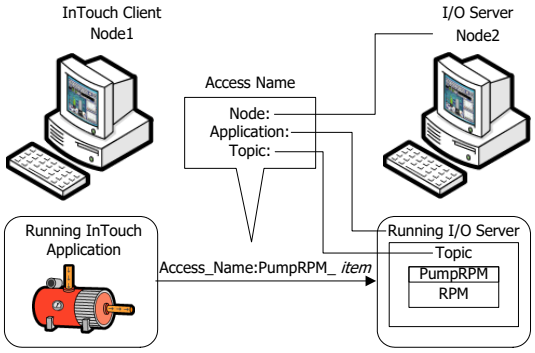
下面的语句通过将给标记指定的注释与标记的名称结合起来，而创建一条操作员消息。

```
OperatorMessage=PumpRPM.Name + " has a comment of: " + PumpRPM.Comment;
```

使用 I/O 进行数据访问

您可以开发分布式应用程序，使其中的各个 InTouch 系统功能组件位于不同的节点上。下图显示如何为另一个节点上存储的数据配置 I/O 请求。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您可以设置 InTouch 应用程序，使之能够使用第三方寻址惯例来确定另一个节点上存储的数据。此惯例包含节点、应用程序及主题名。要从远程节点获取数据，需要为 InTouch 应用程序配置指定这三个项目的“访问名”。

例如，如果希望从正在另一个节点上运行的远程“I/O 服务器”访问数据，“访问名”需要包含以下内容：

访问名选项	描述
节点名	运行“I/O 服务器”程序的计算机的节点名。
应用程序名	节点上运行的<1>“</1>I/O 服务器<2>”</2>程序的名称<3>。</3> 例如，“UA_SampleServer”可能是 OPC UA 服务器的名称。 如需有关与 Operations Integration (OI) 服务器关联的应用程序名的详细信息，请参阅 OI 服务器文档。
主题名	指定给“I/O 服务器设备组”的标签。

如果将 Excel 电子表格用作 InTouch 数据源，则可以按如下方式定义“访问名”：

访问名选项	描述
节点名	运行 Excel 程序的计算机的节点名。
应用程序名	Excel 是<1>“</1>应用程序名<2>”。</2>
主题名	包含所请求的数据的 Excel 工作簿与电子表格的名称。例如，[Book1]Sheet1。

除节点、应用程序、主题以及项目之外，还需要指定远程节点上的数据的类型。此信息确定在<1>“</1>标记名字典<2>”</2>中定义标记时给它指定的 I/O 类型<3>。</3>

使用 Gateway Communication Driver

Gateway Communication Driver 是用作门户和协议转换器的应用程序，允许两个计算机系统或程序相互通讯。您可以使用 OI Gateway 将使用不同协议（如 OPC 和 OPC UA）进行通讯的客户端和数据源关联起来。您可以通过 InTouch 访问 Gateway Communication Driver，以便连接到现场设备。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 利用 Gateway Communication Driver 通过 OPC、OPC UA 和其它协议在自动化应用程序和控制应用程序之间、现场系统和现场设备之间以及企业应用程序和办公应用程序之间进行通讯。

Gateway Communication Driver 会与 WindowViewer 捆绑在一起，并且在您安装 InTouch 时自动安装。它还可以安装为独立应用程序。

关于 Gateway Communication Driver

Gateway Communication Driver 充当通讯协议转换器。它可用于将使用不同协议进行通讯的客户端和数据源关联起来。

本用户帮助出版物仅涵盖配置和运行 Gateway Communication Driver 组件所需的信息。相关组件随附的文档提供了有关组件操作的详细信息。

您可以使用 Log Viewer（操作控制管理控制台 (OCMC) 的一个管理单元）来排解 Gateway Communication Driver 的问题。请参阅 Log Viewer 帮助文件来查找有关以下操作的信息：

- 查看错误消息。
- 确定显示哪些消息。
- 对错误消息做书签。

您还可以使用客户端应用程序（如 InTouch HMI 软件）排解问题。客户端应用程序可以使用系统设备项目确定节点的状态和某些参数的值。

Gateway Communication Driver 的基本规则包括：

- 每个节点只能运行一个 Gateway Communication Driver 实例。
- 可以使用 OI Server Manager 管理单元激活和停用 Gateway。
- 可以使用标准 COM 激活机制将 Gateway 激活为 COM 服务器（OPC 服务器）。
- Gateway 只能在 OPC 客户端内以进程外方式运行。
- Gateway 只能与 Application Server v2.0 及更高版本提供的 ArchestrA 数据源组件进行通讯。不支持更早版本的 Application Server。

Gateway Communication Driver 入门

以下步骤介绍了 Gateway Communication Driver 的入门工作流程。

1. 安装 InTouch。
   - Gateway Communication Driver 已包括在 InTouch 安装过程中。
   - 您必须另外安装 OPC 和 OPC UA 服务器。
2. 安装 OPC 或 OPC UA 服务器并将其配置为与现场设备进行通讯。
3. 根据需要配置其它协议。
4. 在 InTouch 中创建应用程序。
5. 将 Gateway Communication Driver 配置为指向已安装的 OPC 服务器、OPC UA 服务器或其它数据源。
6. 激活 Gateway Communications Driver。

   如需有关设置 Gateway Communication Driver 的详细信息，请参阅“Gateway Communications Driver 帮助”中的*第一次设置 Gateway Communication Driver*。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

支持的 InTouch 通讯协议

您可以配置 InTouch HMI 以使用 DDE 或 SuiteLink。InTouch HMI 还支持“消息交换”通讯协议。

如需有关在 InTouch 应用程序中使用“消息交换”的详细信息，请参阅从 InTouch 访问 Application Server 数据。

动态数据交换

“动态数据交换”(DDE) 通讯协议使 Windows 应用程序之间可以相互通讯。DDE 在两个当前正在运行的应用程序之间建立客户端-服务器关系。服务器应用程序提供数据，并接受由希望使用服务器数据的任何其它应用程序所发出的请求。发出请求的应用程序称为客户端。InTouch 应用程序可以既是客户端，又是服务器。

SuiteLink

SuiteLink 是专为工业应用设计且基于 TCP/IP 的协议。SuiteLink 能够提供数据完整性、高吞吐量以及一些简单的诊断过程。Microsoft Windows NT 4.0 及更高版本支持 SuiteLink 协议。

SuiteLink 并非 DDE 或 NetDDE 的替代品。客户端与服务器之间的每个连接都取决于网络要求。

SuiteLink 提供以下功能：

- “数值时间质量”（Value Time Quality，简称 VTQ）在传递给支持 VTQ 的客户端的所有数据值上都会插入一个时间标签与质量指示符。
- 通过 Microsoft Windows 操作系统的性能监视器，可以对数据吞吐量、服务器负载、计算机资源消耗以及网络传输等进行非常全面的诊断。
- 不管应用程序是在单个节点上，还是分布在很多节点上，都可以在应用程序之间始终维持很高的数据吞吐量。
- 网络传输协议是使用 Microsoft 标准 Winsock 接口的 TCP/IP。
- 您可以使用 TLS 1.2 协议安全地配置 Suitelink 节点。

以标准用户身份配置安全的 Suitelink 通讯

通过 TLS 1.2 协议启用的安全 Suitelink 连接可对客户端与服务器之间的通讯通道进行加密。如果是使用加密的 Suitelink 来访问远程数据，那么必须从配置器中配置“系统管理服务器 (SMS)”。加密是通过 SMS 提供的证书来实现的。为了让 SuiteLink 服务器和客户端使用加密的通讯，必须在 SuiteLink 3.0 安装中选择 ASB 运行时组件功能。

自从引入安全的 SuiteLink (V3) 以来，SuiteLink 一直在提供回退功能，以便 SuiteLink 服务器可以在安全（V3，已加密）或不安全（V2，未加密）模式下同时接受传入连接。自 System Platform 2023 发行版开始，缺省条件下，将禁用接受 V2 连接的回退模式。如果要启用回退模式以接受不安全的连接，那么必须在“系统管理服务器”中进行配置更改。通过检查 Log Viewer 中的 WWSLS 组件，可以验证 SuiteLink 连接的模式，无论是安全的还是不安全的。

注意：自 System Platform 2023 起，SuiteLink 连接缺省设置为使用 TLS 加密进行保护。如果未启用不安全的回退模式且不满足安全连接要求，那么 SuiteLink 连接将会失败。如需详细信息，请参阅在配置器中配置系统管理服务器。

使用注册表编辑器启用/禁用回退（不安全）Suitelink 连接

1. 打开“注册表编辑器”。
2. 导航到路径：
   [HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\ArchestrA\WebApplications\Default\SuiteLink]

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 找到条目：

   "V2Server"=dword:00000001

   "V3Server"=dword:00000001

4. 要启用备用模式，请将 V2Server 的值设置为 1。
5. 要禁用备用模式，请将 V2Server 的值设置为 0。
6. 重新启动计算机。

在配置器中配置系统管理服务器

系统管理服务器 (SMS) 是 System Platform 通用平台服务的一部分，用于实施 System Platform 2023 的重要安全措施。这些方法包括：

- 设置用于节点间通讯的端口号。
- 设置 SuiteLink 安全模式：可以将通过 SuiteLink 连接进行的通讯配置为仅使用加密（安全）通讯，或者在无法建立安全 (TLS) 连接时允许使用未加密的通讯。在 System Platform 中，SuiteLink 可用于多种不同的应用程序。如需有关配置 SuiteLink 安全性的信息，请选择“高级配置”按钮并转至“通讯”选项卡。
- 证书管理。
- 通过 OpenID 连接标准进行用户身份验证，该标准允许通过外部标识供应器进行单一登录 (SSO)。

为了实现安全性，每个 System Platform 节点都必须与系统管理服务器进行通讯。您的 System Platform 拓扑中只能有一台系统管理服务器，否则可能发生通讯中断。系统管理服务器会存储共享的安全证书，并在各台机器之间建立信任关系。您可以将一个附加节点配置为冗余 SSO 服务器，在无法接通系统管理服务器时，该服务器会充当单一登录的备份。

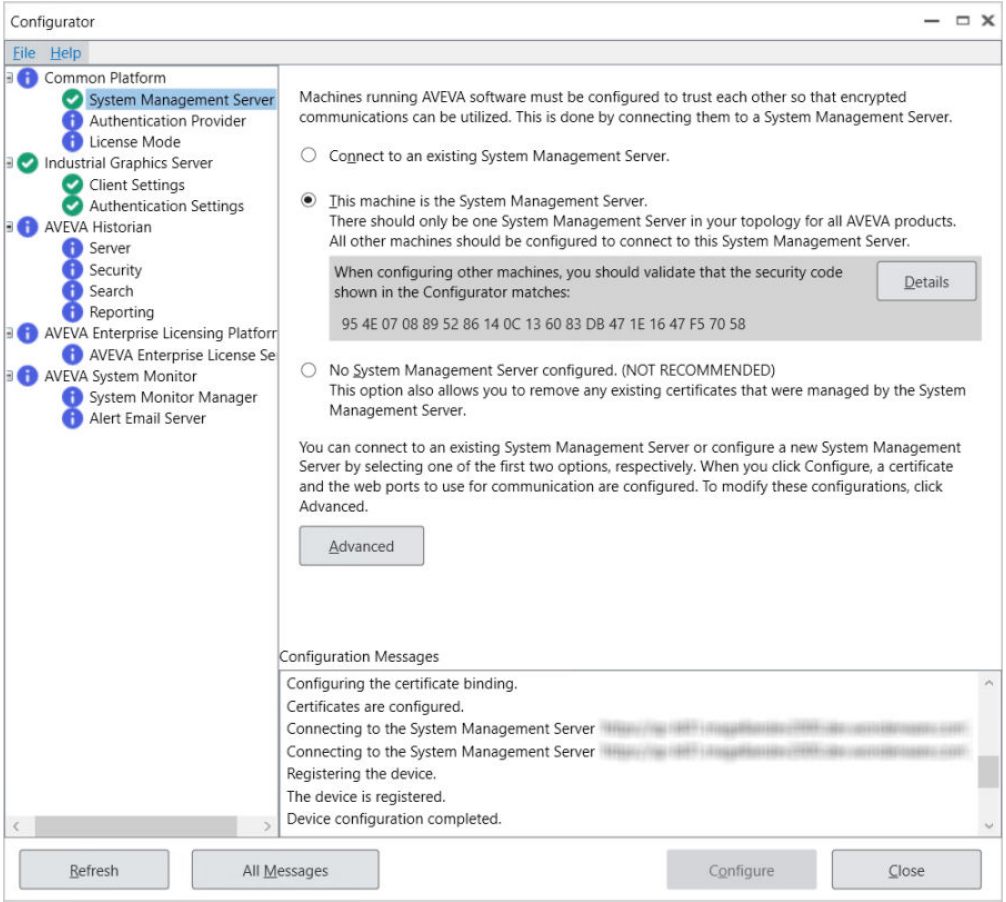
如果某些节点尚未升级到 System Platform 2017 Update 3 或更高版本，那么与那些旧节点的通讯可能需要利用不安全的通讯。但是，只要将运行 System Platform 2017 Update 3 或更高版本的节点配置为与系统管理服务器进行通讯，就会加密这些节点之间的通讯。

配置系统管理服务器

1. 在配置器中，展开通用平台，然后选择系统管理服务器。

   注意：如果系统提示您输入系统管理服务器的用户凭据，请使用以下格式输入用户名：**DomainName\UserName**。如果您具有域管理员权限但并不是本地计算机上的管理员，那么可能会显示输入用户凭据的提示。您必须是 **Administrators** 或 **aaAdministrators** 操作系统组的成员才能配置系统管理服务器。如需详细信息，请参阅 AVEVA System Platform 帮助中的“用于配置系统管理服务器的用户凭据”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

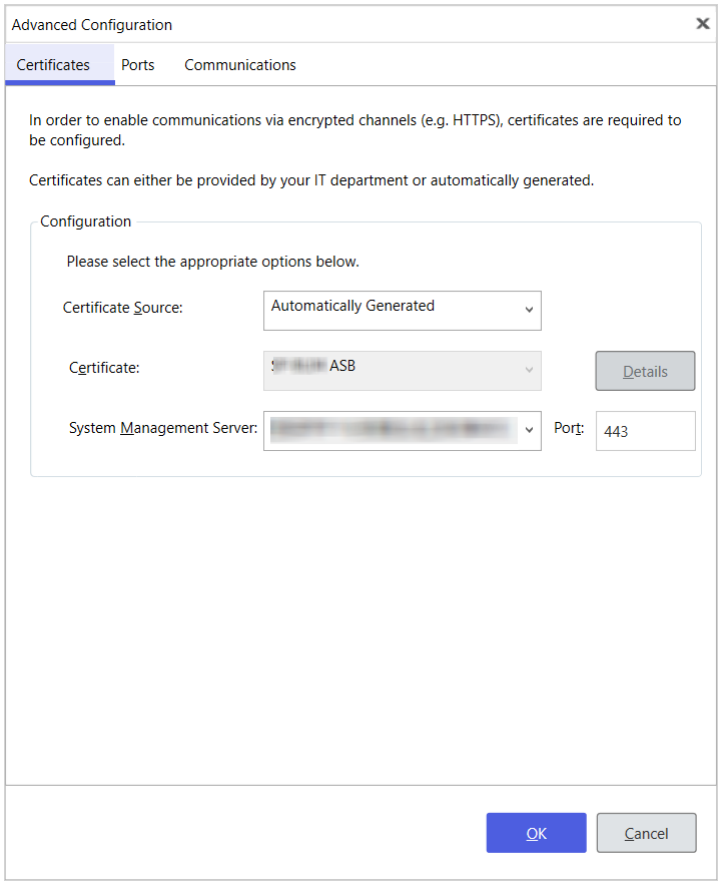


注意：安装完成后会自动调用配置器。您之后也可以随时从任何 System Platform 节点上的 Windows“开始”菜单启动配置器。

2. 将为您提供三个选项：
   - **连接到现有的系统管理服务器：**此为缺省选项。System Platform 发现服务会在其网络上查找任何现有的系统管理服务器。如果找到任何系统管理服务器，就会将它们显示在下拉列表中。选择要使用的服务器，或输入服务器的计算机名称。System Platform 拓扑中的所有计算机都应该连接到同一台服务器。
   - **此机器为系统管理服务器：**如果此计算机将作为系统管理服务器，请选择此选项。System Platform 拓扑中的所有其它计算机应配置为使用**连接到现有的系统管理服务器**选项连接到此服务器。
   - **未配置任何系统管理服务器。（不推荐）：**选择此选项可设置您的计算机，无需进行加密和安全通讯。您仍然可以将拓扑中的其它计算机配置为使用系统管理服务器。
3. 选择高级按钮。

   此时打开高级配置窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



a. 在**证书选项**卡中配置证书参数。

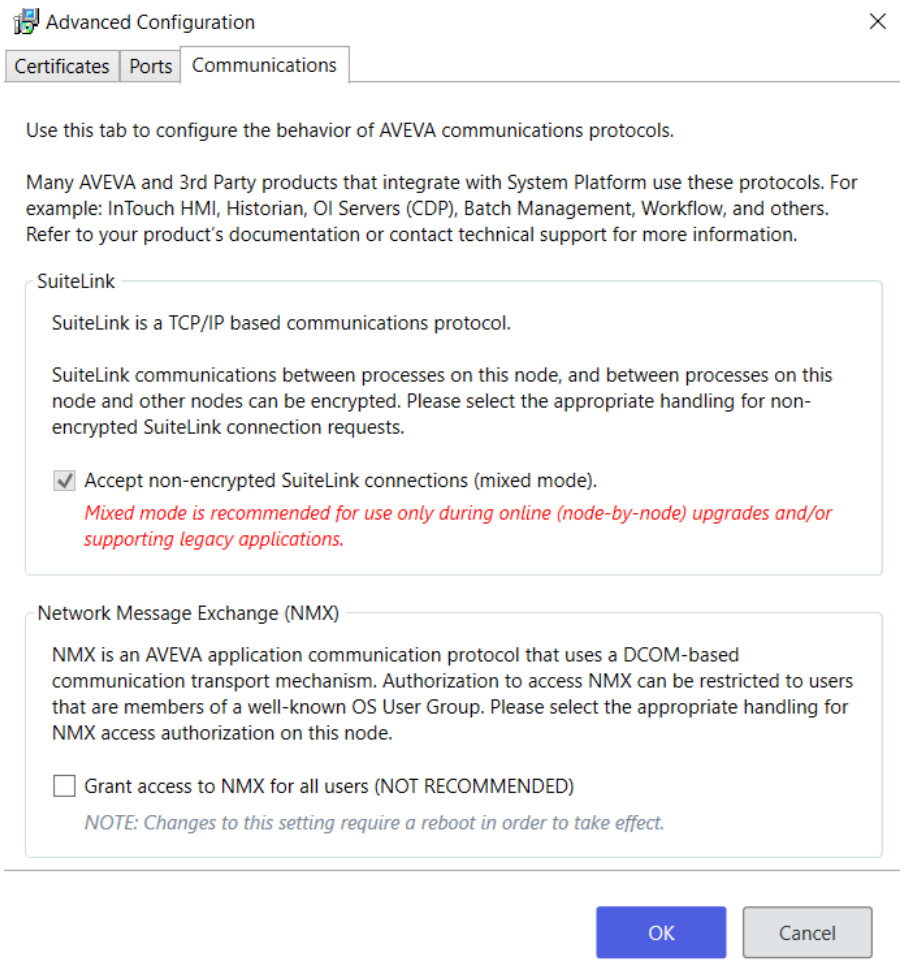
- **证书源：选择自动生成**（缺省值）或**由 IT 提供**。如果您的 IT 部门提供证书，请按**导入**按钮并导航到该证书文件。如需详细信息，请参阅《System Platform 安装》中的“导入证书”。
- **证书**：将显示证书名称。**选择详细信息**以查看导入的证书。无论是在服务器节点还是远程节点上，都会通过自动更新过程定期更新证书。
- **系统管理服务器**：如果您正在连接到现有的系统管理服务器，那么会显示所选服务器的名称和端口号。
- **通用平台端口**：通用平台的端口用于与特定 AVEVA 软件（如 Sentinel System Monitor）进行通讯。通常，您可以使用缺省设置。远程节点必须使用与此处配置的端口号相同的端口号进行配置。
  缺省 HTTP 端口：80
  缺省 HTTPS 端口：443

如需详细信息，请参阅《AVEVA System Platform 安装》的“高级配置选项”一节。

b. 在**通讯选项**卡中配置通讯参数。

- 选中**接受非加密的 SuiteLink 连接（混合模式）复选**框，然后**选择确定**。这样即会启用混合模式。也就是说，同时接受加密连接和非加密连接。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

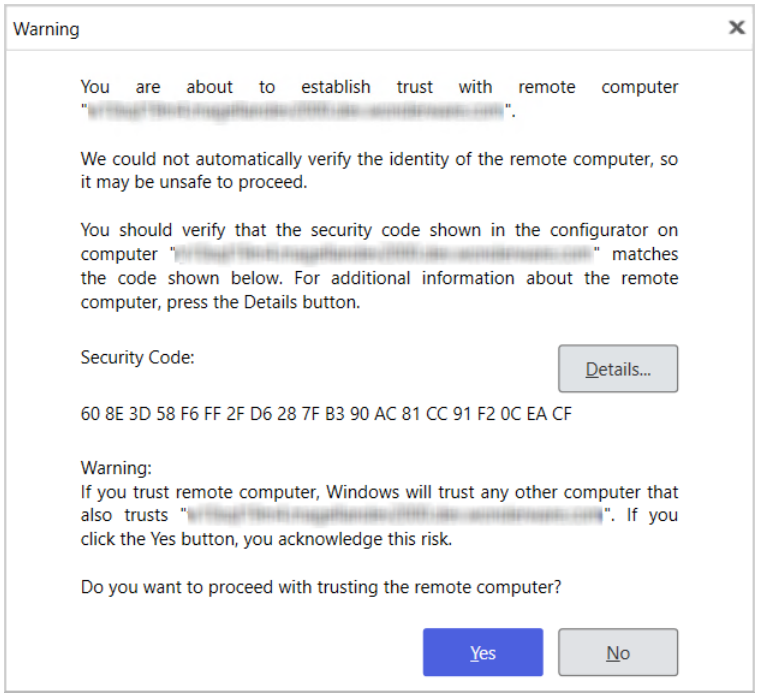


注意：如果您只需要加密连接 (V3)，请取消选中**接受非加密的 SuiteLink 连接（混合模式）**复选框，然后**单击确定**。

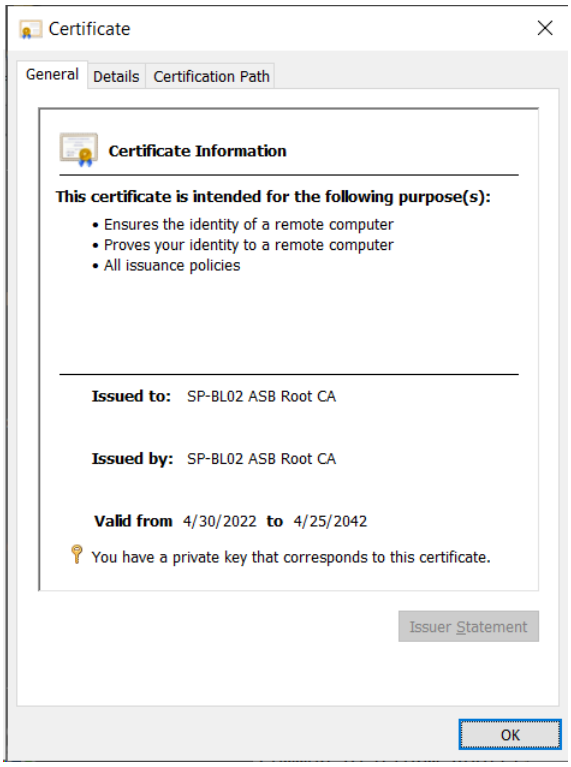
4. **选择确定**以保存更改，然后返回到 SMS 配置窗口。
5. **选择配置**。

   此时将显示“安全警告”窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



通过在计算机之间建立信任，通讯可以畅通无阻。如果您不确定远程计算机的标识，这将成为一个安全问题。如果您对正在连接的计算机有任何疑问，请通过选择“高级配置”对话框中的**详细信息...**按钮打开证书来验证安全代码和证书详细信息。



6. 在左窗格中选择下一个需要配置的项目。配置完所有必需项目后，按**关闭**按钮完成安装。请参阅《Application Server 用户》文档中的“配置后系统重启”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如需有关系统管理服务器配置的详细信息，请参阅《System Platform 安装》的“配置系统管理服务器”一节。

以标准用户身份访问服务器

以标准用户身份访问服务器时，无法建立安全的 SuiteLink 通道。为了实现安全、加密的通讯工作流程，请确保满足以下条件之一：

- **将标准用户添加到用户组：**应将标准用户添加到服务器端的“ArchestrAWebHosting”用户组中。如需有关将用户添加到用户组的详细信息，请参阅特定于 Windows 的文档。
- **将 WindowViewer 作为服务运行：**对于标记服务器架构中基于 InTouch 标记的应用程序，可以通过在标记服务器节点上将 InTouch WindowViewer 作为服务运行来实现安全加密的 Suitelink 通讯。
- **使用管理员权限以交互模式运行 WindowViewer：**这使得 WindowViewer 能够在充当 Suitelink 服务器的非交互式用户下运行，该用户具有访问用来加密 SuiteLink 服务器-客户端通讯的私钥所需的用户权限，这在使用交互式用户时是不可能的。这是必需的，因为运行 WindowViewer 的用户提供了安全上下文，用于检索 SuiteLink 服务器进行安全通讯所需的私钥。如果 WindowViewer 作为交互式应用程序运行，则用户是一个交互式用户，如前所述，该用户无法访问用于安全加密的 Suitelink 通讯的私钥。

将 InTouch 作为标记服务器作为服务运行，可以在客户端节点上与 InTouch 标记服务器客户端许可证结合使用，从而限制 InTouch 只能与标记服务器进行通讯。在这种情况下工作的用户通常以 HyperV 或 VMWare 等 PC 虚拟化技术为例，并在该虚拟机中将 WindowViewer 作为服务运行，以降低其解决方案的总体拥有成本。

以标准用户身份运行 WindowViewer

在以下情况下，WindowViewer 据说会以标准用户身份运行，而无需管理权限：

- 在没有管理权限的情况下以标准用户身份启动 System Platform IDE，在 WindowMaker 中编辑 $InTouchViewApp，然后快速切换到 WindowViewer。
- 从“AVEVA 应用程序管理器”启动 WindowViewer，或者在没有管理权限的情况下，以标准用户身份直接启动它。

SuiteLink 通讯问题疑难排解

如果遇到 SuiteLink 通讯问题：

- 确认 Microsoft TCP/IP 可以在安装 InTouch HMI 的计算机上正常工作。
- 验证计算机节点名不超过 15 个字符。
- 确认 SuiteLink 在安装 InTouch HMI 的计算机上是作为服务在运行。
- SuiteLink 在安装 InTouch 的过程中自动安装。SuiteLink 服务会自动启动。如果 SuiteLink 服务停止，必须重新启动它。

以标准用户身份访问服务器时的 Suitelink 通讯问题

下表显示非管理员用户（标准用户）在不同 System Platform 版本中的 Suitelink 连接行为以及相应的故障排除步骤。

情形	直到 System Platform 2020 R2 SP1 为止	系统平台 2023	疑难排解
未配置 SMS，运行 WindowViewer	无记录器消息	无记录器消息	无

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

未配置 SMS，将客户端连接到 WindowViewer	与 WindowViewer 建立了不安全的连接。记录器中出现警告消息。*	连接失败。 记录器中出现警告消息。*	• 在配置器中，选择允许不安全连接的选项。 建立了不安全的连接，且在记录器中出现警告消息。 • 配置安全的 Suitelink 连接。
已配置 SMS，运行 WindowViewer	连接失败。 记录器中出现错误消息。**	连接失败。 记录器中出现错误消息。**	• 将 WindowViewer 作为服务运行 • 使用管理员权限以交互模式运行 WindowViewer，或者 • 将 WindowViewer 用户添加到 “ArchestrAWebHosting”组。
已配置 SMS，将客户端连接到 WindowViewer	与 WindowViewer 建立了不安全的连接。记录器中出现警告消息。*	连接失败。 记录器中出现警告消息。*	• 在配置器中，选择允许不安全连接的选项。 建立了不安全的连接，且在记录器中出现警告消息。 • 配置安全的 Suitelink 连接
未配置 SMS，未配置安全的 Suitelink 连接 (V3)，并且禁用将不安全的 Suitelink 连接 (V2) 作为回退	• Suitelink 客户端连接到 WindowViewer。 • Web 客户端绑定到 InTouch 标记。 • 记录器中出现警告消息。*	• Suitelink 客户端未连接到 WindowViewer。 • Web 客户端未绑定到 InTouch 标记。 启动 WindowViewer 时记录器中出现错误消息。**	• 在配置器中，配置 SMS。 • 配置安全的 Suitelink 连接。 • 选择允许将不安全的连接作为回退的选项。

警告消息* - 警告 WWSLS - 从 127.0.0.1(VIEW) 建立非加密通讯通道。通过在两个节点上都安装系统管理服务器并升级到客户端和服务器产品的最新版本来保护此通道

错误消息** - 错误 WWSLS [多行消息] - 在此配置中无法实现加密入站 SuiteLink 连接。

OPC

OPC 本身不是通讯协议，但是可用作驱动程序（一组基于 Microsoft OLE/COM 技术的非专有标准接口）。该标准可实现自动化应用程序和控制应用程序之间、现场系统和现场设备之间以及企业应用程序和办公应用程序之间的互操作。

通过定义一种通用的高性能接口，OPC 可以一次完成与现场设备交换数据的任务，并且可让 HMI、SCADA、控件和自定义应用程序轻松地重复利用数据，这样软件/应用程序开发人员就无需再像以前一样通过编写自定义驱动程序来交换数据了。

在网络上，OPC 会使用 DCOM（分布式 COM）进行远程通讯。

OPC UA

OPC 统一架构 (OPC UA) 是用于实现工业互操作性的机器对机器通讯协议。它提供具有增强的安全性的过程控制，高级通讯、安全和信息模型，以及跨平台连接性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

OPC UA 实现为 OI Gateway 中的客户端。

OPC UA 与 OPC 有很大差异。下面提供了经典 OPC 与 OPC UA 之间的主要差异。

经典 OPC	OPC UA
使用 Microsoft 的 COM/DCOM 技术进行通讯。没有可配置的超时。取决于系统中配置的 DCOM 超时。	使用服务架构导出数据，可提高通讯和连接的方便性。
依赖 Windows 操作系统。	平台独立，可以连接到多种设备和平台。
具有有限安全性。	具有内置安全性。
没有内置的处理问题（例如消息丢失）的功能。	具有内置的处理问题（例如消息丢失）的功能。

MQTT

MQTT（以前称为消息队列遥测传输）是在 TCP/IP 上使用的发布/订阅消息协议。MQTT 旨在确保设备可以互相通讯，同时最大程度地降低功率和带宽要求。它是简单的消息协议，非常适合用于依靠慢速或不可靠网络的设备。

MQTT 协议是应用层规范，已作为标准 ISO/IEC PRF 20922 发布。MQTT 使用发布-订阅机制，需要中介代理。发布者向代理发送数据，订阅客户端接收发布给代理的数据。只有订阅了特定主题的客户端可接收关于该主题的消息。该协议支持双向通讯，这样作为发布者的设备也可以接收更新。

设置访问名

您必须将 InTouch I/O 标记或远程标记引用与“访问名”关联起来。“访问名”定义与另一个 I/O 数据源之间的通讯链接。每个“访问名”指定一个由节点名、应用程序名、主题组成的 I/O 地址。

在分布式应用程序中，I/O 引用可以设置为网络“I/O 服务器”的全局地址，也可以是本地“I/O 服务器”的本地地址。

InTouchView 显示专门设计用于 Application Server 环境的 HMI 应用程序的可视化界面。InTouchView 应用程序作为客户端运行，由 Application Server 充当提供大多数 HMI 功能的服务器。

InTouchView 应用程序仅提供全功能 InTouch 应用程序中的某些标准功能。InTouchView 应用程序无法连接到 Application Server Galaxy 之外的其它 I/O 数据源。使用 InTouchView 应用程序时，只能使用缺省的“访问名”Galaxy，且无法创建“访问名”。

创建访问名

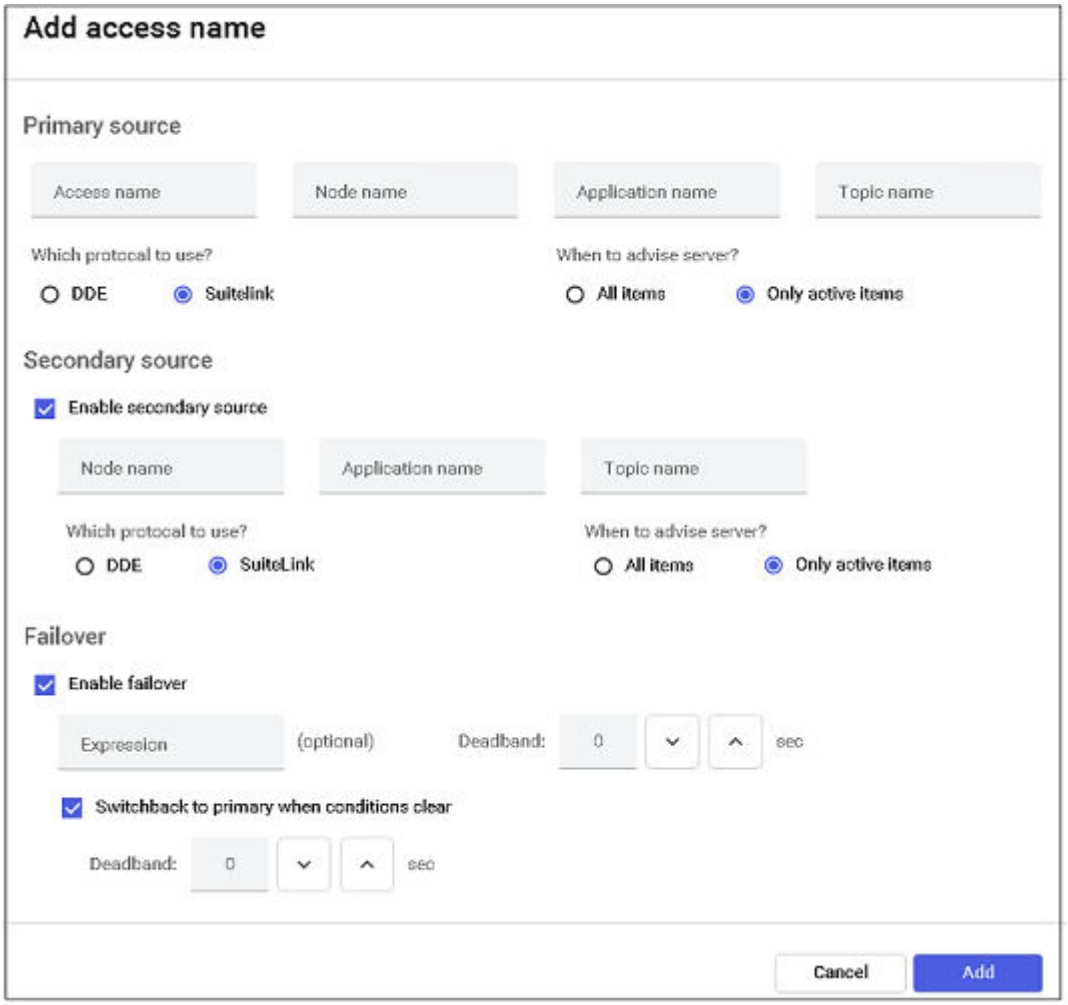
1. 在主页菜单上的标记组中，选择访问名。

   此时出现访问名对话框。

2. 选择添加。

   此时出现添加访问名对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 设置**添加访问名对话**框的属性。**执**行以下操作：
   - 在**访问名**框中，**输**入用于**标识**此“**访问**名”的名称。“**访问**名”的最大字符限制**为** 16 个字符。
   - 如果数据位于网**络**“I/O 服**务**器”上，**请**在**节点名**框中**输**入**该远**程服**务**器的**节**点名。**节**点名的最大字符限制**为** 71 个字符。
   - 在**应用程序名称**框中，**输**入将从中**获**取数据的“I/O 服**务**器”程序的**实际**程序名。

     如果 I/O 数据源是 DAServer，**请输**入 DAServer 程序的名称；不要包含**该**程序的 .exe 文件**扩**展名。**应**用程序名称的最大字符限制**为** 100 个字符。
   - 在**主题名**框中，**输**入希望**访问**的主**题**名。主**题**名的最大字符限制**为** 179 个字符。

     主**题**名是特定于**应**用程序的数据元素子**组**。在数据来自 DAServer 程序的情况中，主**题**名与在 DAServer 程序中**给**主**题**配置的名称相同。与 Microsoft Excel 通**讯时**，主**题**名必**须**是保存工作簿与**电**子表格**时**所指定的名称。例如，[Book1]Sheet1。
   - **选择**与“I/O 服**务**器”**进**行通**讯**的通**讯协议**。
   - **选择**相**应选项**以**轮询**服**务**器上存**储**的信息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Option	定义
提示所有项	轮询所有的数据，不论这些数据是否在可见的窗口中，也不论是否已报警、已记录、已绘制到趋势图上或是用在脚本中。选择此选项会影响性能，因此建议不要使用。
只提示激活项	轮询部分数据，这些数据要么是显示在可见的窗口中，要么是已报警、已记录、已绘制到趋势图上或用在脚本中。 注意：除非出现在可见的窗口中，否则不会轮询到按钮动作脚本。

4. 如果要选择辅助备份服务器，请选择启用辅助数据源，然后输入辅助数据源备份服务器所需的详细信息。
5. 在指定完“访问名”之后，选择添加。

   此时即会将新的访问名添加到列表中。
6. 选择关闭。

如需有关设置辅助服务器以实现故障转移的详细信息，请参阅使用访问名的故障转移功能。

删除访问名

您可以删除不再需要的<1>“</1>访问名<2>”。</2> 在删除访问名之前，确保满足以下条件：

- 没有标记与该<1>“</1>访问名<2>”</2>关联<3>。</3>
- WindowViewer 已停止。

删除访问名

1. 在主页菜单上的标记组中，选择访问名。

   此时出现访问名对话框。当前的各个<1>“</1>访问名<2>”</2>会列出来<3>。</3>
2. 要删除某个“访问名”，请在列表中将其突出显示，然后选择删除。

   此时出现一条消息<1>，</1>要求确认是否应删除该<2>“</2>访问名<3>”。</3>
3. 选择是。
4. 选择关闭；如果要删除定义的其它“访问名”，请重复此操作程序。

使用 I/O 标记访问 I/O 数据

InTouch HMI 可以使用 I/O 标记来发送与接收本地或远程 Windows 应用程序的数据。每个 I/O 型标记引用都是“I/O 服务器”程序中的有效项目。您可以在<1>“</1>标记名词典<2>”</2>中定义不同类型的 I/O 标记<3>。</3>

配置 I/O 标记的属性

在<1>“</1>标记名词典<2>”</2>中可以定义不同类型的 I/O 标记<3>。</3>

指定离散 I/O 标记

I/O 离散标记指出可编程控制器、过程计算机的所有输入与输出以及网络节点数据的二进制状态。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

I/O 离散标记必须指定一个初始值<1>，</1>即<2>“</2>打开<3>”</3>或<4>“</4>关闭<5>”。</5> 您也可以配置离散 I/O 标记，使之切换到二值数据源的相对值。您可以指定与标记关联的过程进入或脱离某种报警状态时，在报警事件窗口中出现的消息。

如需有关从“标记名字典”中创建标记的总体操作程序的详细信息，请参阅创建新标记。

定义 I/O 离散标记

1. <1>打开</1><2>“</2><3>标记名字典</3><4>”</4><5>并为新的标记指定一个名称</5><6>。</6>
2. 从**标记类型**对话框中，将标记指定为 I/O 离散型。此时出现**标记名字典**对话框的详细信息部分。

3. 执行以下操作：
   - 选择打**开**或**关闭**作为与标记关联的初始值。

     InTouch HMI 在应用程序启动时将此值指定给标记，但并不会将这个初始值写入 I/O 设备。
   - 选择直接或取反作为要应用于从远程 I/O 标记接收到的值的输入转换。

输入转换	描述
直接	输入值从“I/O 服务器”程序中直接读取而不加改变。
取反	I/O 离散值在从服务器程序中读取时进行切换。例如，如果 I/O 输入是 0，则值自动设置为 1。

4. 在打**开**消息与**关闭**消息框中输入标记进入或脱离某种报警状态时出现的消息。

   这些消息可用在任何动画链接或脚本中，而不论该标记是否配置了报警。
   - 如果定义在标记值等于 1 (On, True) 时处于活动状态的离散报警，则在打**开**消息框中输入的消息会出现在 ActiveX 报警显示的**值**与**报警限**列中。

     标记的报警状态恢复正常时，在**关闭**消息框中输入的消息会出现在**值**列中，“打开消息”保持在**报警限**列中。
   - 如果定义在标记值等于 0 (Off, False) 时处于活动状态的离散报警，则在**关闭**消息框中输入的消息会出现在 ActiveX 报警显示的**值**与**报警限**列中。

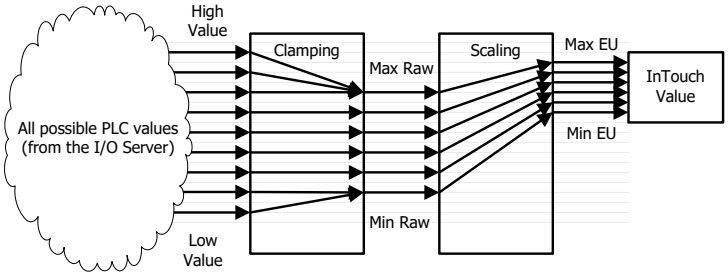
     标记的报警状态恢复正常时，在打**开**消息框中输入的消息会出现在**值**列中，“关闭消息”保持在**报警限**列中。
5. 将更改保存到标记中。

指定整型与实型 I/O 标记

您必须为 I/O 整型与实型标记指定一组属性，这些属性描述在 InTouch 应用程序与外部进程之间发送的数值数据。

InTouch HMI 对来自 PLC 的原始输入数据进行规格化处理。下图显示限定原始 I/O 数据值、再将它们缩放到可从 InTouch 应用程序中显示的工程单位的过程。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



I/O 整型与实型标记包含某些属性，这些属性设置 PLC 发送的原始输入数据的最小与最大限制。InTouch HMI 会对低于或高于原始值范围的 I/O 值进行钳位处理。钳位功能将超出范围的值重新指定为最小或最大原始值。

I/O 整型与实型标记包含某些属性，这些属性将钳位后的原始值缩放到一定的工程单位范围。最小与最大工程单位属性设置缩放后的值的上、下边界。

定义整型与实型 I/O 标记时，可以指定缩放原始值以计算工程单位值时要采用的转换类型。您可以选择<1>“</1>线性<2>”</2>或<3>“</3>平方根<4>”。</4>

对于线性缩放，结果是在最小与最大端点之间使用线性插值进行计算的。输入值的线性缩放算法是：

```
EUValue = (RawValue - MinRaw) * ((MaxEU - MinEU) / (MaxRaw - MinRaw)) + MinEU
```

输出值的线性缩放算法是：

```
RawValue = (EUValue - MinEU) * ((MaxRaw - MinRaw) / (MaxEU - MinEU)) + MinRaw
```

对于平方根缩放，最小与最大原始值用作插值。这对于缩放来自非线性设备（如压力传感器）的输入非常有用。输入值的平方根缩放算法是：

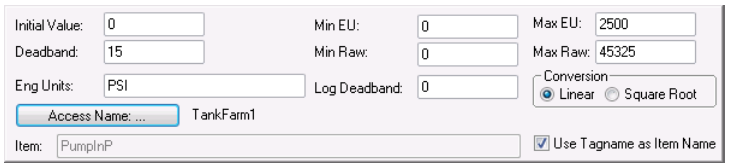
```
EUValue = sqrt(RawValue - MinRaw) * ((MaxEU - MinEU) / sqrt(MaxRaw - MinRaw)) + MinEU
```

输出值的平方根缩放算法是：

```
RawValue = square((EUValue - MinEU) * (sqrt(MaxRaw – MinRaw) / (MaxEU -MinEU))) + MinRaw
```

定义整型与实型 I/O 标记

1. <1>打开</1><2>“</2><3>标记名字典</3><4>”</4><5>并为新的标记指定一个名称</5><6>。</6>
2. 从标记类型对话框中，将“I/O 整型”或“I/O 实型”指定为标记的类型。此时出现标记名字典对话框的详细信息部分。



3. 执行以下操作：
   - 在初始值框中，输入应用程序启动时与标记关联的整数或实数。

     应用程序并不会将这个初始值写入外部进程。
   - 在最小工程单位框中，输入标记的最小工程单位值。
   - 在最大工程单位框中，输入标记的最大工程单位值。
   - 在最小原始数据框中，输入原始 I/O 整型或实型数值的下钳位（最小值）。
   - 在最大原始数据框中，输入原始 I/O 整型或实型数值的上钳位（最大值）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 在工程单位框中，输入要给标记的工程单位使用的标签。
- 选择线性或平方根作为计算工程单位值时缩放原始值所采用的转换类型。

4. 将更改保存到标记中。

指定消息型 I/O 标记

您可以指定一个 I/O 消息标记选项，以指定远程进程的网络地址。它的消息属性与内存消息标记相同。

定义内存与 I/O 消息标记值

1. <1>打开</1><2>“</2><3>标记名字典</3><4>”</4><5>并为新的标记指定一个名称</5><6>。</6>
2. 从标记类型对话框中，选择 I/O 消息作为标记的类型。此时出现标记名字典对话框的详细信息部分。

3. 在最大长度框中，输入标记的消息中允许的最大字符数。

   在消息中最多可以输入 131 个字符。
4. 在初始值框中，输入希望在 WindowViewer 启动应用程序时显示的文本字符串。
5. 将更改保存到标记中。

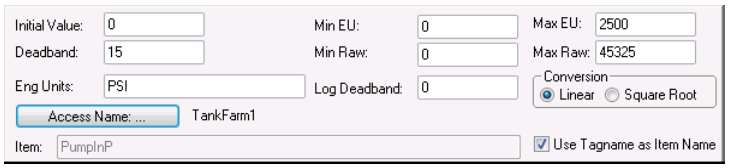
设置 I/O 访问参数

您可以在<1>“</1>标记名字典<2>”</2>中设置标记的 I/O 属性<3>。</3> 这些属性确定与标记关联的外部数据源。

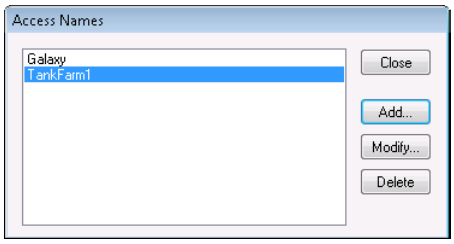
这些步骤仅说明如何从<1>“</1>标记名字典<2>”</2>中指定 I/O 属性<3>。</3> 如需有关配置 Galaxy 与“访问名”的详细信息，请参阅使用 I/O 进行数据访问。

设置标记的 I/O 属性

1. 从标记类型对话框中，将标记指定为某种 I/O 标记类型。此时出现标记名字典对话框的详细信息部分。



2. 选择访问名以定义或选择指定给标记的“访问名”。此时出现访问名对话框，显示 InTouch HMI 识别出的一列当前“访问名”。（Galaxy 是 ArchestrA 连接的缺省“访问名”）。



3. 添加<1>“</1>访问名<2>”</2>或接受缺省值<3>。</3>
4. 选择服务器程序中的数据点供 I/O 标记读取和写入数据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 要在服务器程序中的某个过程数据点上读取和写入数据，请在**项目**框中输入“项目名”。例如，要从 PLC 的寄存器中读取某个值，请输入该寄存器的标识作为“项目名”。例如：

  要将 Allen-Bradley PLC 的寄存器 1 用作“项目名”，请在**项目**框中输入 R1。

  要将 Allen-Bradley PLC 的寄存器 1 的最低有效位用作“项目名”，请在**项目**框中输入 R1:0。
- 要将标记用作项目，请选择将**标记名用作项目名**。“项目名”最多支持 254 个字符。

在运行时检索 I/O 标记的有关信息

<1>您可以使用返回在</1><2>“</2><3>访问名</3><4>”</4><5>定义中指定的节点名</5><6>、</6><7>应用程序名以及主题名的函数编写脚本</7><8>。</8>

IOGetNode() 函数

IOGetNode() 函数将为特定“访问名”定义的节点地址返回给与脚本中的函数关联的标记。

类别

其它

语法

```
IOGetNode("AccessName");
```

参数

AccessName
要返回其节点信息的现有“访问名”。

附注

您可以将“访问名”指定为字符串，或是使用其它 InTouch 标记或函数提供的字符串值。

示例

下例将 ModbusPLC1“访问名”的节点信息返回给 **NodeName** 标记。

```
NodeName = IOGetNode("ModbusPLC1");
```

IOGetApplication() 函数

IOGetApplication() 脚本函数将为特定“访问名”定义的应用程序名返回给某个标记，该标记指定为函数的参数。

类别

其它

语法

```
IOGetApplication("AccessName");
```

参数

AccessName
在其中定义应用程序的现有“访问名”。

附注

您可以将“访问名”指定为字符串，或是使用其它 InTouch 标记或函数提供的字符串值。

示例

本例将为 ModbusPLC1“访问名”指定的应用程序名返回给 **AppName** 标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
AppName = IOGetApplication ("ModbusPLC1");
```

IOGetTopic() 函数

IOGetTopic() 脚本将为特定“访问名”定义的标题名返回给某个标记，该标记与脚本中的函数关联。

类别

其它

语法

```
IOGetTopic("AccessName");
```

参数

AccessName
要返回其主题名的“访问名”。

附注

“访问名”可以指定为字符串，或是使用其它 InTouch 消息标记或函数提供的字符串值。

示例

本例将访问名 ModbusPLC1 的主题信息返回给 **TopicName** 标记。

```
TopicName = IOGetTopic("ModbusPLC1");
```

在运行时动态更改 I/O 标记引用

InTouch HMI 使用动态引用查看只是临时需要它们的值的数据点，如在诊断应用程序中。通过“动态引用寻址”，可以使用单个标记名对多个数据源进行寻址。

您可以通过多种方法使用单个标记来动态引用多个数据源：

- 使用 I/O 标记的 .**Reference** 点域指定不同的“访问名”特征
- 使用 **IOSetItem()** 脚本函数设置 I/O 标记的 .**Reference** 点域
- 使用 **IOSetAccessName()** 脚本函数在运行时更改“访问名”的特征

.Reference 点域

通过将有效的引用指定给 I/O 标记的 .**Reference** 点域，可以实现“动态引用寻址”。通过使用 **.Reference** 点域来修改指定给 I/O 标记的“访问名”的特征，可以动态更改数据源。

.**Reference** 点域的语法是：

tag.Reference="accessname.item"	更改指定给标记的“访问名”与项目。
tag.Reference="[.]item"	更改指定给 I/O 标记的项目。
tag.Reference="accessname."	更改 I/O 标记的“访问名”。
tag.Reference=""	使 I/O 标记不再活动。

每个 I/O 类型标记都有一个 .**ReferenceComplete** 点域。此离散点域的值指出 .**Reference** 点域中请求的项目是否会反映在 .**Value** 点域中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

应用程序在 WindowViewer 中启动时，.**ReferenceComplete** 域设置为假 (0)。.**Value** 点域经确认使用 .**Reference** 点域中指定的数据源进行更新时，.**ReferenceComplete** 值设置为真 (1)。如果 .**Reference** 点域发生更改，则 .**ReferenceComplete** 点域会自动设置为假 (0)，然后在更新新的值时更新为真 (1)。

IOSetItem() 函数

通过在脚本中使用 **IOSetItem()** 函数，可以实现“动态引用寻址”。**IOSetItem()** 包含一些参数，可以在运行时更改指定给 I/O 标记的 .**Reference** 点域的值。

类别

其它

语法

```
IoSetItem ("Tag", "AccessName", "Item");
```

参数

Tag
使用英文双引号括起来的任何 InTouch I/O 标记。

AccessName
指定给 I/O 标记的“访问名”。

Item
指定给 I/O 标记的“项目”。

Tag、AccessName 以及 Item 参数可以指定为字符串，它们也可以是其它 InTouch 标记或函数提供的字符串值。

示例

在下例中，**PumpInP1** 标记的 .**Reference** 点域更改为指向 Excel“访问名”与 R1C1 项目。

```
IOSetItem("PumpInP1", "excel", "R1C1");
```

或者

```
Number = 1;
TagNameString = "PumpInP" + Text(Number,"#");
IOSetItem(TagNameString, "excel", "R1C1");
```

如果将“访问名”与“项目”的值都指定为空字符串 (“”)，则可以使标记不再活动。例如，通过以下方法可以使 **PumpInP2** 不再活动：

```
IOSetItem("PumpInp2", "", "");
```

如果只是给“项目”指定空值，则标记的当前项目值保持不变，它的“访问名”值会进行更新。例如，以下代码将 **PumpInP3** 标记的“访问名”更改为 excel2，而不影响它当前的“项目”值：

```
IOSetItem("PumpInP3", "excel2", "");
```

类似地，如果只是给“访问名”指定空字符串，则标记的当前“访问名”值保持不变，它的“项目”值会进行更新。下例将 **PumpInP4** 标记的“项目”更改为 R1C2，而不影响它当前的“访问名”值：

```
IOSetItem("PumpInP4", "", "R1C2");
```

IOSetAccessName() 函数

通过在脚本中使用 **IOSetAccessName()** 函数，可以实现“动态引用寻址”。**IOSetAccessName()** 在运行时修改 I/O 标记的“访问名”的应用程序名或主题名的特征。

备注：处理 IOSetAccessName() 函数时，在结束当前的对话并启动新的对话期间，会有一个时间上的延迟。在此期间，任何“插入”或写入新主题的尝试都将丢失。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

其它

语法

```
IOSetAccessName("AccessName", "NodeName", "AppName", "TopicName");
```

参数

AccessName
要将新的“应用程序名”与“主题名”值指定给它的现有“访问名”。实际的字符串或消息标记。

NodeName
要指定的新“节点名”。实际的字符串或消息标记。

AppName
要指定的新“应用程序名”。实际的字符串或消息标记。

TopicName
要指定的新“主题名”。实际的字符串或消息标记。

附注

指定给 AccessName、AppName 以及 TopicName 参数的值可以指定为字符串，或是使用其它 InTouch 标记或函数提供的字符串值。

如果使用**添加访问名对话**框为访问名故障转移配置了辅助数据源，则无法在运行时使用 IOSetAccessName() 函数来更改辅助数据源配置。

备注：在 WindowMaker 中创建“访问名”时，如果“访问名”是 SuiteLink 类型，则 InTouch HMI 会阻止“访问名”去访问相同的节点、应用程序以及主题。请勿允许 **IOSetAccessName()** 函数在运行时将“访问名”重定向到重复项。通过在运行时使用 **IOSetAccessName()** 函数，SuiteLink 类型的“访问名”可以重定向到重复的主题。重定向后的“访问名”将不起作用。

示例

通过使用以下脚本函数，可以将 **MyAccess1**“访问名”更改为指向 **Excel** 应用程序与 **[Book1]Sheet1** 主题，而不影响当前的 NodeName：

```
IOSetAccessName("MyAccess1", "", "EXCEL", "[Book1]Sheet1");
```

如果给“主题”指定空字符串，则“访问名”的当前“应用程序”值会进行更新，它的“主题”值会保持不变。

例如，以下脚本将 **MyAccess2**“访问名”的“应用程序名”更改为 EXCEL，而不影响当前的“主题”值：

```
IOSetAccessName("MyAccess2", "", "EXCEL", "");
```

如果仅仅给“应用程序名”指定空字符串，则标记的当前“主题”值会进行更新，但它的“应用程序”值会保持不变。例如，以下脚本将 **MyAccess3** 标记的“主题”更改为 **[Book2]Sheet1**，而不影响它的当前“应用程序名”值：

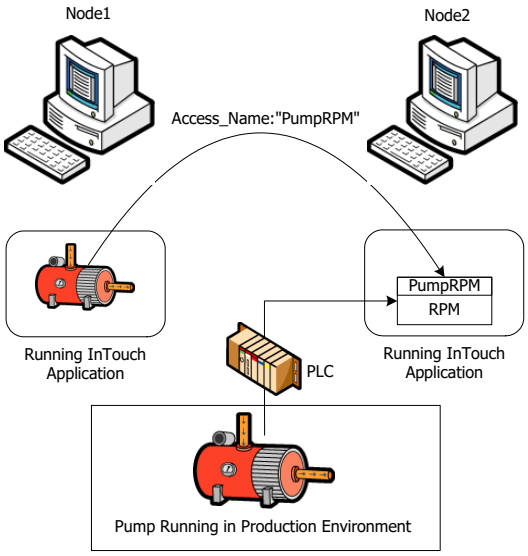
```
IOSetAccessName("MyAccess3", "", "", "[Book2]Sheet1");
```

本例可用于必须采用 PLC 冗余配置的情况。

将标记转换为远程引用

您可以根据客户端-服务器架构创建分布式 InTouch 应用程序。客户端应用程序可以运行在使用其它远程节点上定义的标记的一个网络节点上。下图显示在 Node1 上运行的 InTouch 应用程序，它远程引用 Node2 上的 PumpRPM 标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



在本例中，可以使用两种方法来检索 Node2 上 PumpRPM 标记的值：

- 在 Node1 的“标记名字典”中创建一个 I/O 型标记，在同这个标记关联的“访问名”中 Node2 被用作节点名。
- 使用指向 **PumpRPM** 标记的直接远程引用。例如，**PLC1:"PumpRPM"**。

  在窗口或 QuickScript 中<1>，</1>通过将<2>“</2>访问名<3>”</3>作为前缀附加到远程标记名之前<4>，</4>可以引用远程标记<5>，</5>具体形式如下<6>：</6>

  access_name:"tag_name"

导入窗口或 QuickScript 时，可以将占位符标记转换为远程标记引用。例如，您可以将占位符标记转换为指向从中导入窗口的应用程序。这些标记不需要在本地<1>“</1>标记名字典<2>”</2>中定义<3>。</3>

您可以使用多种方法将本地标记转换为远程标记引用：

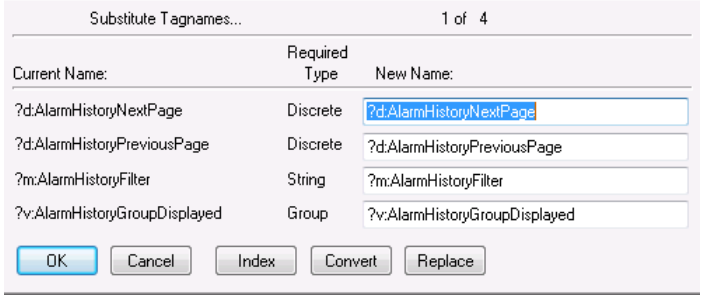
- 附加远程标记引用
- 转换与导入的窗口关联的占位符标记
- 启动“标记浏览器”，然后打开标记源的“标记名字典”以选择远程标记引用。

手动将标记转换为远程标记引用

1. 在 WindowMaker 中打开一个应用程序窗口。
2. 选择同希望更改为远程标记引用的本地标记关联的对象。
3. 在动画选项卡上的替换组中，选择标记。

   此时出现替换标记名对话框，其中列出与该对象关联的标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

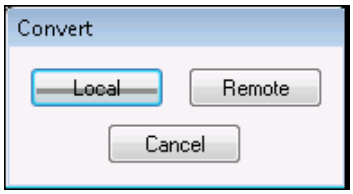


4. 选择**索引**将索引添加到每个**标记**名。
5. 选择**转换**。此时出现**转换**对话框，显示将**标记转换为**本地或**远**程引用**标记**的**选项**。
6. 选择**远**程。此时出现**访问**名对话框。对话框中显示在本地 InTouch 应用程序中定义的所有“**访问**名”。
7. <1>从列表中**选择**一个</1><2>“</2><3>**访问**名</3><4>”。</4>
8. 选择**关闭**。此时替代**标记**对话框中列出的所有**标记**都自**动转换为远**程**标记**引用，“**访问**名”会附加到**标记**名上。
9. 选择确定。

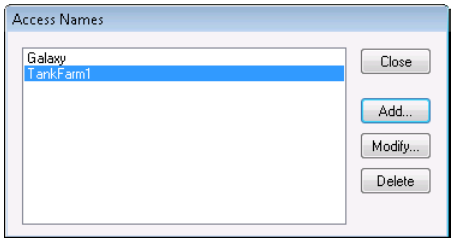
将导入的窗口的标记转换为远程引用

1. 打**开导**入的窗口并**选择**所有的对象。
2. 在**动画选项**卡上的**替换组**中，**选择标记**。

   此时出现**替换标记**名对话框。
3. 选择**转换**。此时出现**转换**对话框。



4. 选择**远**程。此时出现**访问**名对话框。对话框中显示在本地 InTouch 应用程序中定义的所有“**访问**名”。

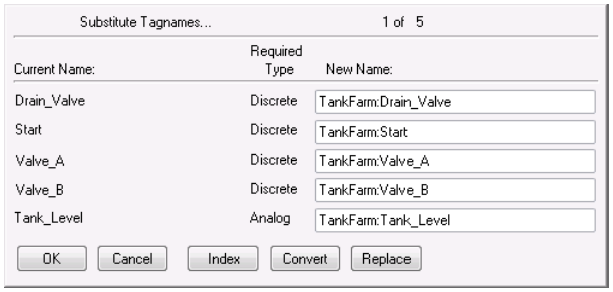


5. <1>从列表中**选择**一个</1><2>“</2><3>**访问**名</3><4>”。</4>

   要**验证**“**访问**名”是否正确配置，**请选择修改**。

   如果当前尚未定义指向**该标记**源的“**访问**名”，**请选择添加**并定义它。<1>“</1>**访问**名<2>”</2>必须包含**应**用程序所在**远**程**节**点的名称<3>。</3>
6. 选择**关闭**。此时替代**标记**对话框中列出的所有**标记**都自**动转换为远**程**标记**引用，“**访问**名”会附加到**标记**名上。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

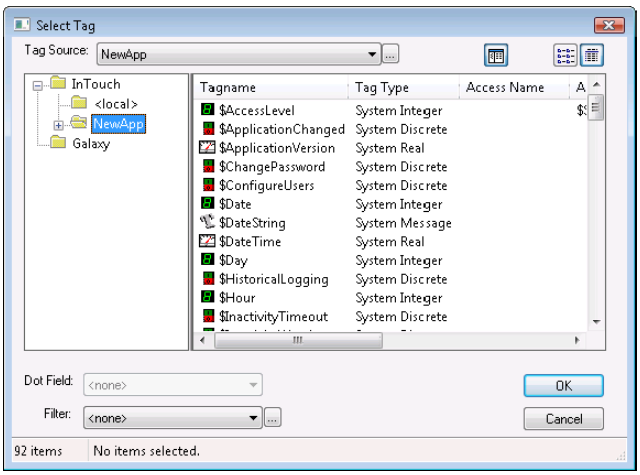


7. 选择确定。

在“标记浏览器”中选择远程标记引用

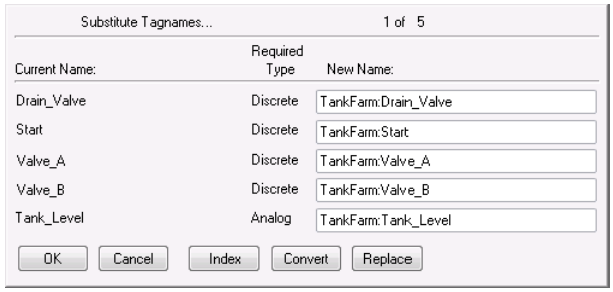
1. 选择同要转换为远程标记引用的本地标记关联的对象。
2. 在动画选项卡上的替换组中，选择标记。

   此时出现显示所选标记的替换标记名对话框。
3. 删除新名框中要替换为远程标记引用的标记名。
4. 双击新名框。此时出现选择标记对话框，其中列出与该应用程序关联的标记。
5. 使用“目录树”视图选择一个远程标记。
   a. 选择目录树图标，以便在左侧窗格中显示所有本地与远程“访问名”的层次结构化列表。



   b. 选择一个远程<1>“</1>访问名<2>”</2>文件夹<3>，</3>以便在右侧窗格中显示已指定给它的标记<4>。</4>
   c. 选择要用作远程引用的远程标记。
   d. 选择确定。此时出现替换标记名对话框，新名框中的内容是所选的远程标记名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

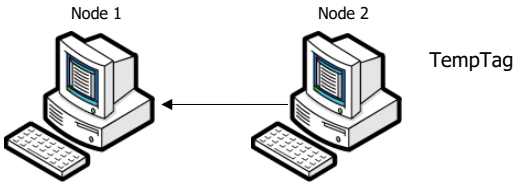


6. 选择确定以关闭对话框并将远程标记关联到所选的对象。
7. 针对要与远程引用关联的每个标记重复这些步骤。

通过远程引用访问 I/O 数据

InTouch HMI 支持工厂自动化应用程序的真正客户端-服务器架构。您可以设计这样的客户端应用程序，它不使用与运行 InTouch 应用程序位于相同节点的本地“标记名字典”中的任何标记。您可以在一个节点上运行应用程序，而此应用程序通过使用远程标记引用来使用远程节点的标记。

下图显示一个简单的示例，其中 **TempTag** 在 Node2 的本地定义：



在本例中，Node1 上运行的 InTouch 应用程序可以通过两种方法来检索 Node2 上 TempTag 的值：

- 在 Node1 的“标记名字典”中创建一个 I/O 型标记，在与该 I/O 标记关联的“访问名”中 Node2 被用作“节点名”。
- 使用一个直接指向 **TempTag** 的远程引用。例如，Node2:"TempTag"。

要在任何其它应用程序中直接引用远程标记，仅需要使用 AccessName:item。您不必在本地“标记名字典”中定义远程标记。远程引用还可以访问任何 I/O 数据源（如 DAServer 或 Microsoft Excel）中的数据。

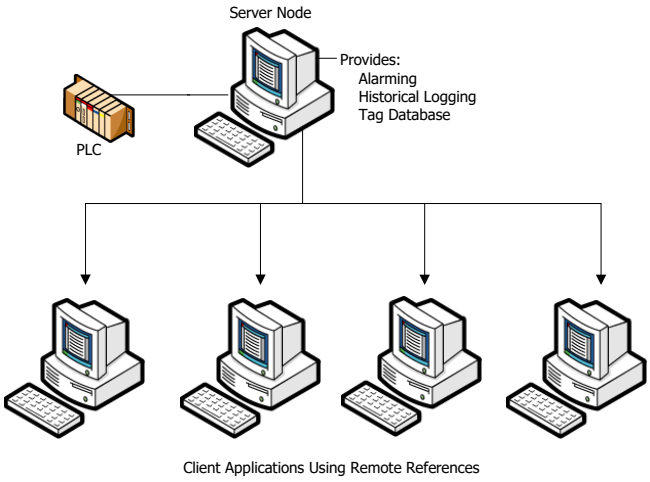
您也可以远程引用 SuperTag。对 SuperTag 进行远程标记引用的有效语法如下：

Access_name:Parent_Instance\ChildMember\SubMember。

如需有关远程引用 SuperTag 的详细信息，请参阅引用 SuperTag 成员。

导入窗口或 QuickScript 时，可以将占位符标记转换为远程标记引用。您不必在本地<1>“</1>标记名字典<2>”</2>中定义标记<3>。</3> 您可以从网络上的任何应用程序中访问远程引用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



在运行时重定向远程引用

在运行时通过使用脚本，可以将 Application Server 对象引用或远程引用重定向到 InTouch 标记。根据满足的特定条件，或是直接通过操作员的动作，可以切换图形符号的对象实例。

IOSetRemoteReferences() 函数

在 InTouch 应用程序运行期间，可以使用 **IOSetRemoteReferences()** 脚本函数将 Application Server 对象引用或远程引用重定向到标记。**IOSetRemoteReferences()** 查找与指定的字符串匹配的所有远程引用，并根据指定的参数值来更改这些引用。您可以创建这样的脚本：它根据满足的条件或用户的动作来触发重定向引用的函数。

类别

其它

语法

```
IOSetRemoteReferences(BaseAccess, NewAccess, MatchString, SubstituteString, Mode)
```

参数

BaseAccess
此字符串参数指定配置的原始“访问名”，此访问名要在引用中进行匹配。

NewAccess
新的“访问名”。新的“访问名”应用于满足以下条件的所有引用：原始“访问名”与 BaseAccess 提供的字符串匹配，并且原始“项目名”与 MatchString 值（如果有指定）匹配。

MatchString
引用中配置的原始“项目名”所要匹配的字符串。如果 MatchString 值是空字符串，则视为同任何“项目名”都匹配。

SubstituteString
要替换原始“项目名”的字符串。此字符串会替换 MatchString 值，以便为引用创建新的活动“项目名”。如果 SubstituteString 是空字符串，则不进行替换。

模式
确定对配置的原始“项目名”与 MatchString 值进行比较时要使用的方法。匹配操作总是从“项目名”的开头进行。Mode 值为 0 时，指定必须匹配整个“项目名”，或是匹配到名称中的英文句号 (.)。Mode 值为 1 时，指定允许部分匹配，即便下一个字符不是英文句号。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

IOSetRemoteReferences() 不会在更改对象引用之前检查新的标记或“访问名”是否有效。

- **IOSetRemoteReferences()** 只更改远程引用。函数重定向满足以下条件的那些引用：配置的原始“访问名”与 BaseAccess 参数指定的值匹配，并且原始“项目名”与 MatchString 值匹配。
- 只需调用一次 **IOSetRemoteReferences()**，便会影响内存中的活动窗口中满足以下条件的所有远程引用：配置的原始名称字符串与指定给 BaseAccess 和 MatchString 参数的值匹配。
- 如果没有给 BaseAccess 参数指定任何值，则 **IOSetRemoteReferences()** 不会重定向任何远程引用。
- 如果 MatchString 参数为空，则 **IOSetRemoteReferences()** 重定向满足以下条件的所有远程引用：原始访问名与指定给 BaseAccess 参数的值匹配。
- Mode 参数设置为 0 时，仅替换完整对象名（或标记）或完整属性名（或点域）中的“项目名”。MatchString 参数的值必须与整个原始“项目名”匹配，或是匹配到英文句号前的一个字符。
- Mode 参数设置为 1 时，如果项目字符串以匹配项目字符串开头，则允许对项目字符串进行部分替换。也就是说，MatchString 必须与原始项目字符串的某个部分匹配，但这个部分必须从项目字符串的开头部分开始。匹配字符串中的最后一个字符后不必是英文句号。
- 给远程引用配置的原始名称保持不变。对 **IOSetRemoteReferences()** 的后续调用不需要识别当前的活动名称。对 **IOSetRemoteReferences()** 的调用可以按任何顺序进行。
- 如果让两个或更多的窗口指向同一个远程引用，则该远程引用的作用类似于 I/O 标记。对它进行重定向时，所有的窗口都会看到相同的内容。请勿使用单个名称来同时指向两个单独的目标。

备注：同时更改许多个引用（例如在“显示时”窗口脚本中）时，可能需要一些时间才能完成对所有引用的解析。

示例

下例将满足以下条件的所有远程引用重定向到 Galaxy“访问名”的 pump001 项目名：原始项目名与 pumpX 完全匹配。

```
IOSetRemoteReferences("Galaxy","","pumpX", "pump001",0);
```

下例将项目名与 pumpX 完全匹配的 Galaxy“访问名”更改为 TagServer1。此外，项目名也更改为 p2。

```
IOSetRemoteReferences("Galaxy","TagServer1", "pumpX","p2",0);
```

下例在项目名是 pumpX 时将 TagServer1“访问名”更改为 TagServer2。此外，项目名也更改为 backpump3。

```
IOSetRemoteReferences("TagServer1","TagServer2", "pumpX","backpump3",0)
```

下例将 TagServer1“访问名”的 Tank 项目名更改为 Plant。

```
IOSetRemoteReferences("TagServer1","","Tank", "Plant",1)
```

因为没有给 *BaseAccess* 参数指定任何值，下例不会重定向任何远程引用。

```
IOSetRemoteReferences("","Galaxy","pumpX", "pump001",0);
```

恢复引用

<1>如果 NewAccess 参数为空</1><2>（</2><3>即没有指定的值</3><4>），</4><5>则 IOSetRemoteReferences() 会将活动的</5><6>“</6><7>访问名</7><8>”</8><9>恢复为原始的基本</9><10>“</10><11>访问名</11><12>”。</12>

如果 MatchString 参数为空（即没有指定的值），则 **IOSetRemoteReferences()** 会将活动的“项目名”恢复为原始“项目名”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

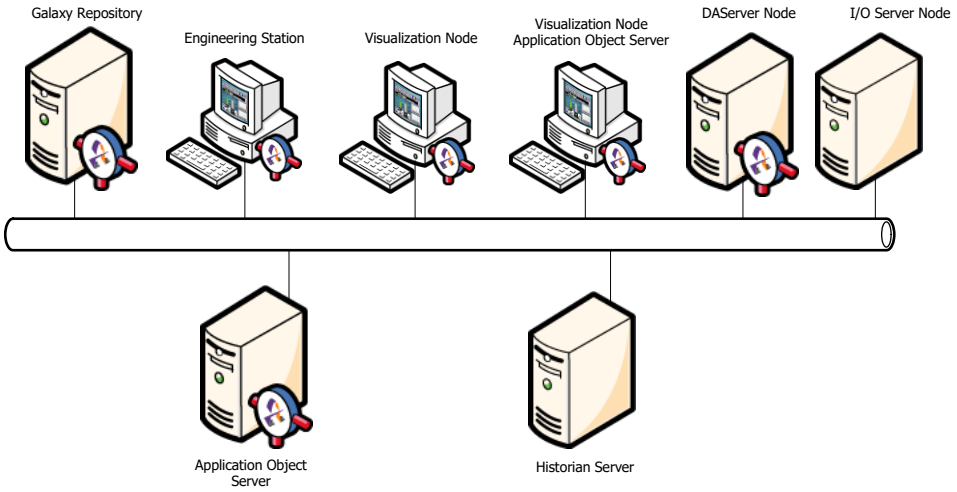
注意：即便 SubstituteString 不为空，但如果 MatchString 为空，则项目名也会恢复为原始项目名。在名称的开头插入文本是不允许的。例如，运行脚本 IOSetRemoteReferences("Access1", "", "","Valve",0); 时，并不会在任何原始“项目名”的开头或结尾附加字符串 Valve。

如果 SubstituteString 为空<1>（</1>即没有指定任何值<2>），</2>IOSetRemoteReferences() 会将活动的<3>“</3>项目名<4>”</4>恢复为原始的基本<5>“</5>项目名<6>”。</6> 通过使用非空的 MatchString 与空的 SubstituteString，可以根据指定的原始基本访问名选择一部分远程引用，并恢复它们的原始“项目名”。

从 InTouch 访问 Application Server 数据

ArchestrA 为整套产品提供一组公共服务和底层架构。通过选择这组产品中的某些产品并使用模块化的 ArchestrA 组件，可以构建工厂自动化与信息系统。

Application Server 提供一组用于构建工厂自动化应用的服务。Application Server 服务分布在系统中的一组节点上。



通常，InTouch HMI 与 Application Server 相互配合，共同为操作员与之交互的应用程序提供可视化界面，从而对工厂的生产过程进行管理。

将 Application Server 对象属性与 InTouch 标记一起使用

您可以将 InTouch 标记用于同 Application Server 对象属性进行交互，以便在 InTouch 应用程序与 Application Server 数据储备库之间传输数据。

InTouch HMI 支持的通讯协议包括“消息交换”。WindowViewer 运行 InTouch 应用程序时，“消息交换”将 WindowViewer 视为匿名引擎。

这种匿名性质表示 InTouch 应用程序没有任何属性可供其它“消息交换”客户端访问。在 Application Server Galaxy 中，WindowViewer 未配置成或当作 AutomationObject 进行管理，也不视为 AutomationObject。InTouch HMI 仅使用“消息交换”来预订 Application Server 中的那些活动项目。

您可以使用 InTouch 的“标记浏览器”将 Galaxy 选作远程标记的标记源，以及浏览 Galaxy 的域名空间。Application Server 对象属性或是属性的特性可以用在远程引用中，或是用作 InTouch I/O 标记的项目。

如需有关将 Application Server 对象用作远程标记源的详细信息，请参阅配置 InTouch HMI 以便将 Galaxy 用作远程标记源。

WindowViewer 中提供一个预先配置好的“访问名”Galaxy 供“消息交换”来访问。Galaxy“访问名”仅与 ArchestrA 环境中的 InTouch HMI 相关。<1>Galaxy</1><2>“</2><3>访问名</3><4>”</4><5>没有用户可以配置的属性</5><6>。</6>

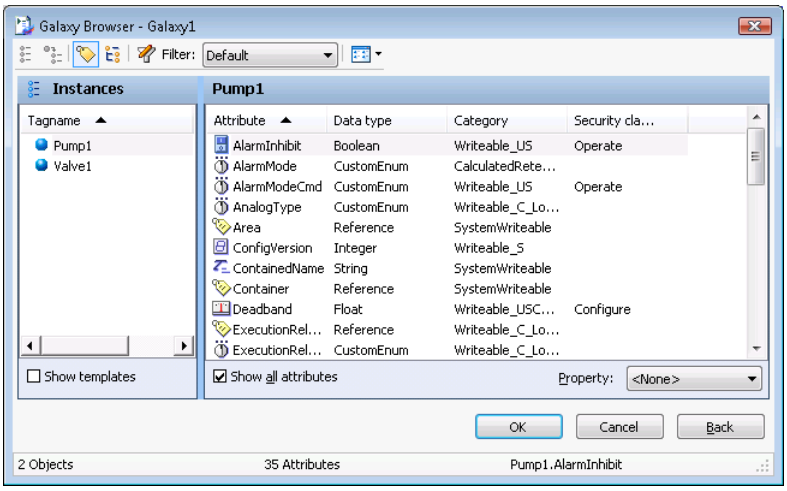
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从 InTouch 浏览应用程序服务器对象属性

要从 InTouch HMI 浏览并选择“应用程序服务器”属性，必须首先在 InTouch HMI 中设置 Galaxy 的标记源。如需有关详细信息，请参阅配置 InTouch HMI 以便将 Galaxy 用作远程标记源。

从 InTouch **选择标记**对话框中选择标记源。

InTouch 的 Galaxy Browser（Galaxy 浏览器）对话框列出目标 Galaxy 中的所有对象。您可以展开一个对象来查看它包含的对象或运行时可以访问属性。Galaxy Browser（Galaxy 浏览器）对话框不显示那些以“_”（标记为隐藏）开头的属性，也不显示任何 QualifiedStruct 类型的属性。



要从“Galaxy 浏览器”对话框返回到普通的 InTouch 选择标记对话框，请选择返回。

Application Server 浏览器限制

使用 InTouch 的 Attribute Browser（属性浏览器）对话框查看 Application Server 对象时，存在以下限制：

- 只能查看到单个 Galaxy 的域名空间中的运行时可见属性。这包括在对象的 TagName 及其 HierarchicalName 之间切换的功能。

  如果对象属性满足以下要求，则可以从 **Attribute Browser**（属性浏览器）对话框中选择它：

  - 在运行时可见
  - 是 AutomationObject 中当前选中的属性
  - 属性名的“.”后面没有跟着“_”的情况
- Attribute Browser（属性浏览器）对话框仅显示 InTouch HMI 所支持的 Application Server 对象属性数据类型。如需有关支持的数据类型的详细信息，请参阅将 Application Server 数据类型映射到 InTouch 数据类型。
- InTouch 的 Attribute Browser（属性浏览器）对话框不显示会导致 InTouch“项目名”超过 95 个字符这一最大限制的任何属性。
- 通过将“标记名.属性名[索引]”用作引用，可以在 InTouch HMI 中显示或检索自动化对象数组元素。使用索引 -1 可以显示或检索所有数组元素的值。
- 您可以使用<1>“</1>标记浏览器<2>”</2>来选择对象属性的特性<3>。</3> 缺省条件下，选择属性时会选择 Value 特性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Application Server 对象中的特殊扩展

WindowMaker 的“标记浏览器”与 WindowViewer 中的“消息交换”客户端可以给每个 Application Server 对象属性添加特殊扩展，并能识别它们。通过这些扩展，InTouch HMI 可以访问无法以其它方式获取的信息。

这些特殊扩展是可选的，并且 InTouch 应用程序不需要使用它们。不过，经常处理状态与质量信息的应用程序需要用到这些扩展。

这些项目扩展了属性的域名空间，使之包含一些 WindowViewer 可以提供给应用程序脚本与窗口的附加属性。例如，通过引用“TIC101.PV.#ReadSts”，可以访问有关 TIC101.PV 预订情况的 MxStatus 信息。此信息对于显示由<1>“</1>消息交换<2>”</2>提供的扩展信息非常有用<3>。</3>

在 Application Server 中，这些属性不像具名元素那样作为对象属性而存在。这些属性是客户端抽象层中提供的客户端扩展，可使得对象属性对于 InTouch HMI 而言是可见的。下表介绍 InTouch HMI 的这些属性扩展：

属性扩展	数据类型	用途
无	强制转换结果	缺省的扩展。表示未提供任何扩展。项目根据 InTouch 的要求使用强制转换的值数据类型进行读取/写入。如果客户端预订下文介绍的 #ReadSts 或 #WriteSts 项目，则可以获取失败的读取/写入信息。示例：“Pump1.PV”。
.#VString 仅限浮点/双精度属性： .#VString1 .#VString2 .#VString3 .#VString4	字符串（读取/写入）	设置对带“.#VString”后缀的引用的预订。这是底层引用。读取和写入都可以正常工作时，将底层引用的当前值作为字符串返回。如果 UserGetAttribute 返回错误的状态，此项目根据 MxStatus 返回简短的状态描述字符串，而不返回值。这些简短的状态描述字符串是： “?Pending”- 待处理 “?Warning”- 警告 “?Comms”- 通讯错误 “?Config”- 配置错误 “?Oper”- 操作错误 “?Security”- 安全性错误 “?Software”- 软件错误 “?Other”- 其它错误

对于 .#VString<1>，</1>如果状态正常但质量不佳<2>，</2>则此项目返回<3>“</3>消息交换<4>”</4>所提供的质量描述字符串<5>，</5>而不返回值<6>。</6>

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

仅当 UserGetAttribute 的质量和状态都正常，或是质量正常而状态不定时，才会将值作为字符串返回。这可能需要进行强制转换<1>, “</1>消息交换<2>”</2>返回的数据类型不是字符串时便是如此<3>。</3> 质量或状态不定时，值会带一个后缀“?”。例如，“3.27?”或“True?”。

将 Application Server 数据类型映射到 InTouch 数据类型

Application Server 包括某些属性与数据类型，它们并未直接映射到 InTouch 标记所支持的四种基本数据类型。

下表显示客户端抽象层如何映射数据类型，以便进行读取和写入操作。它还显示 Galaxy 字典展现给 InTouch 的数据类型。

属性 特性数据类型	InTouch 数据类型	备注
浮点	实型 - 32 位	直接传递。
双精度	实型 - 32 位	如果双精度值是 IEEE NAN，则转换成浮点型 IEEE NAN。如果这会发生上溢，则将 **Quality** 设置为“不佳”，并传递浮点型 IEEE NAN。如果双精度小数值比最小的浮点小数 1.17549E-38 还小，则将它视为浮点数 0.0，并将 **Quality** 设置为“良好”。
布尔型	离散	假 = 0、真 = 1。
整型	整型 - 32 位	直接传递。
字符串型（总是 Unicode）	消息 - MBCS（多字节字符集编码）	如果字符串对于 InTouch 而言太长，则截断它并将 Quality 设置为“不定”。每个 Unicode 字符的两个字节都保持不变。
时间	消息 - MBCS	按照语言环境设置成适当格式的字符串。使用 **MxValue** 来转换字符串。
经过时间	实型	作为浮点数传递，以秒计。**MxValue** 支持强制转换为这种类型。
MxDataType	消息 - MBCS	传递字符串。
MxSecurityClassification	消息 - MBCS	传递字符串。
MxQuality	消息 - MBCS	传递字符串。
MxReference	消息 - MBCS	将引用字符串仅当作 Unicode 传递。
MxCategorizedStatus	消息 - MBCS	传递字符串。
MxQualifiedStruct	不支持	不支持。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性 特性数据类型	InTouch 数据类型	备注
MxQualifiedEnum	消息 - MBCS	传递<1>“</1>枚举<2>”</2>型字符串<3>。</3> 应用程序可以通过引用 #EnumOrdinal 来访问整型序数。例如，“Pump1.PV.#EnumOrdinal”。
字符串数组	消息 - MBCS（只读）	将数组的每个元素放入一个由逗号分隔的字符串，如： “String1,String2,String3” 最长为 InTouch 字符串值的最大限制。如果这会发生截断情况，则发送给 InTouch HMI 的关联质量是“不定”。您无法写入整个字符串数组，但可以写入数组的单独元素。
所有数组	整型、实型、消息、离散	仅支持预订数组的单个元素。在这种情况下，将采用上文所述的转换。否则返回空字符串<1>，</1>并且质量为<2>“</2>不佳<3>”。</3>
MxInternationalizedText	消息	这在运行时作为字符串类型来访问。

Application Server 属性的读取/写入行为

系统写入自动化对象属性时，属性的写入状态最初设置为“?Pending”。

完成写入时，#WriteSts 字符串更新为写入操作的结果。如果成功完成写入，则 #WriteSts 值设置为空字符串。如果写入操作返回错误并处于待处理状态，即便读取操作上仍然在继续更新预订，#WriteSts 项目也会继续显示最新的写入状态。

您也可以使用 #VString1 到 #VString4 这些项目将浮点值或双精度值转换为字符串格式，数字 N 表示要返回的小数位数。例如，“3.1234”是 #VString4 的字符串。通过使用不带数字的 #VString 项目，可以将浮点值或双精度值舍入成整数值，然后将它作为字符串值返回。

属性扩展	数据类型	用途
.#EnumOrdinal	整型（读写）	包含<1>“</1>限定枚举<2>”</2>类型属性的当前读取序数值<3>。</3> 这是一种为枚举表达式返回整数（而不是返回字符串）的方法。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性扩展	数据类型	用途
.#ReadSts	字符串（只读）	包含项目预订情况的当前读取状态，字符串将与之串联起来。它显示为“TIC101.PV.#ReadSts”。它由<1>“</1>消息交换<2>”</2>作为字符串提供<3>。</3> 它的值可以是如下内容之一： “?Config”- 配置错误 “?Comms”- 通讯错误 “?Oper”- 操作错误 “?Pending”- 待处理 “?Warning”- 警告 “?Security”- 安全性错误 “?Software”- 软件错误 “?Other”- 其它错误 **注意：**如果未预订关联的项目（例如，TIC101.PV），则返回的字符串为空。
.#WriteSts	字符串（只读）	包含此字符串要串联的项目的最新写入状态，例如 Pump1.Cmd.#WriteSts。它由<1>“</1>消息交换<2>”</2>作为字符串提供<3>。</3> 如果字符串为空，则说明上次对项目的写入操作成功。否则，#WriteSts 可以是以下各个值之一： “?Config”- 配置错误 “?Comms”- 通讯错误 “?Oper”- 操作错误 “?Pending”- 待处理 “?Warning”- 警告 “?Security”- 安全性错误 “?Software”- 软件错误 “?Other”- 其它错误 **注意：**如果未预订关联的项目（例如，TIC101.PV），则返回的字符串为空。

配置 InTouch HMI 以便将 Galaxy 用作远程标记源

您可以使用 InTouch 的“标记浏览器”选择 Application Server 对象作为标记源，也可以使用它来浏览 Galaxy 数据库。Application Server 对象的属性或属性的特性可以用在远程引用中，或是用作 InTouch I/O 标记的项目。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 应用程序作为客户端运行，为 Application Server 应用程序提供可视化界面时，必须在 InTouch 应用程序所在的相同节点上安装 Application Server Bootstrap 与 WinPlatform 对象。

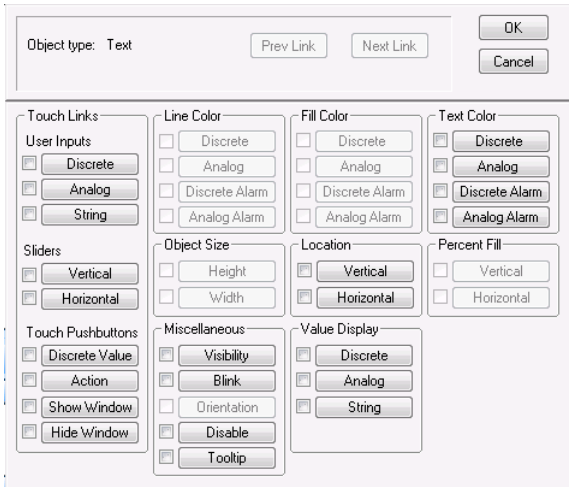
要从 InTouch HMI 中浏览 Galaxy 域名空间，还需要安装“集成开发环境”(IDE)。

InTouch HMI 使用平台的“消息交换”功能来查看 Galaxy 域名空间，并提供更好的数据通讯。

要配置 InTouch 将 Galaxy 用作远程标记源

1. 在 WindowMaker 中打开一个应用程序窗口。
2. 双击一个文本对象。

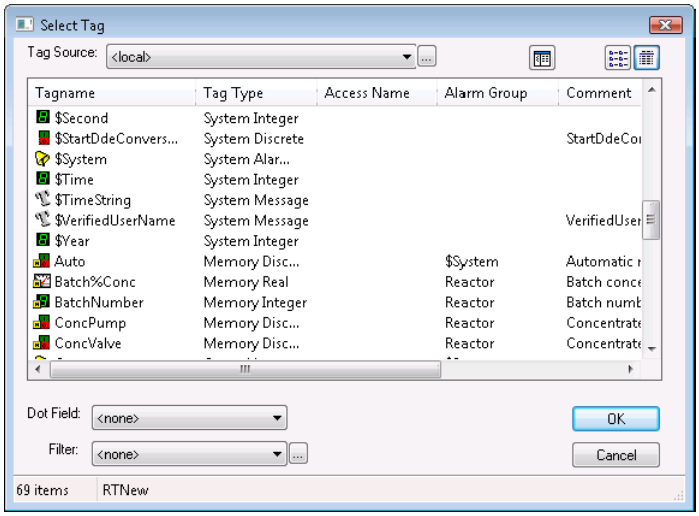
   此时出现**动画链接**对话框。



3. 在**值显示**区域中，选择**模拟**。

   此时出现一个用于插入表达式的**对话**框。
4. **删**除**表达式**框中的任何表达式。
5. 在**表达式**框内双击。

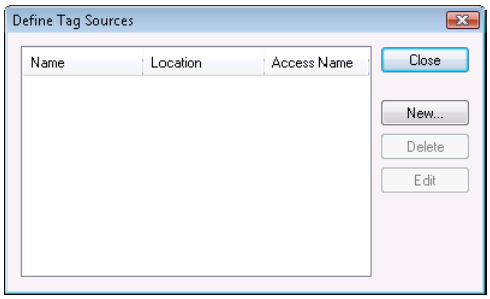
   此时出现**选择标记**对话框。



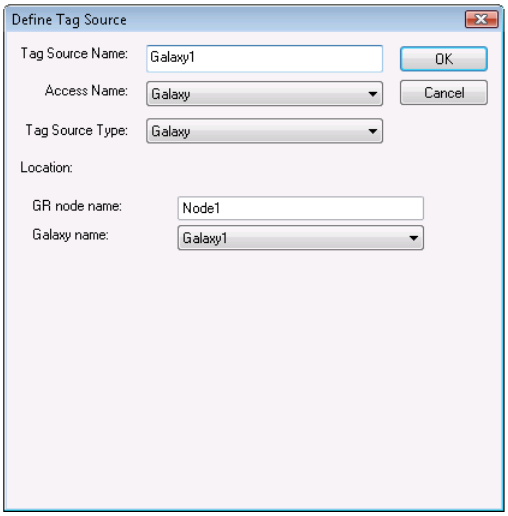
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 选择**标记源**框右侧的按钮。

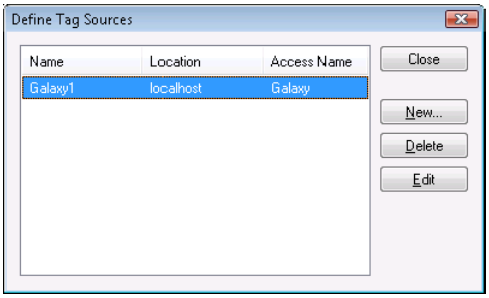
   此时出现定**义标记源**对话框。



7. 选择新建以显示定**义标记源**对话框。

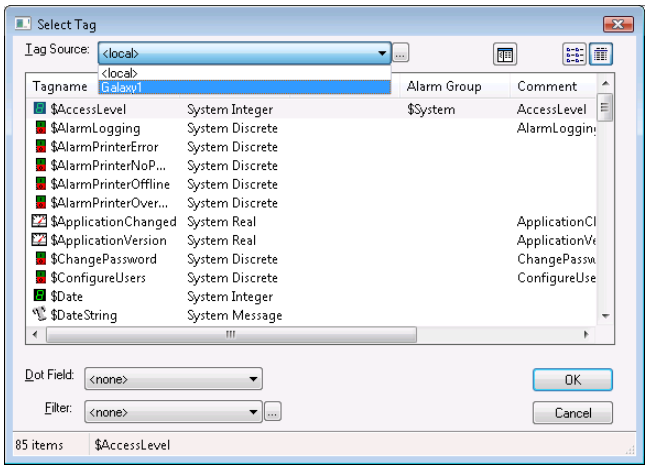


8. 为定**义标记源**对话框中的各个框**输入值**。**执**行以下操作：
   a. 在**标记资源名**框中，**输入远**程 Galaxy **标记**源的名称。
   b. 在**访问名**框中，从列表中**选择** Galaxy。
   c. 在**标记源类型**框中，从列表中**选择** Galaxy。
   d. 在**位置**区域中，**输入**“Galaxy **储备库节**点”的名称，然后从列表中**选择** Galaxy。
   e. **选择确定**。**定义标记源**对话框**显**示您在它的列表中定**义**的**远**程**标记**源。

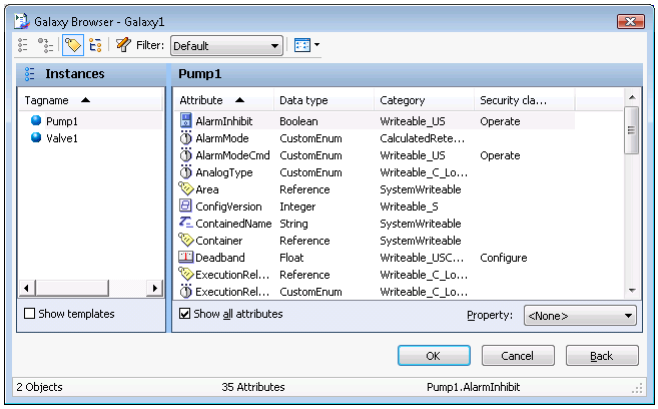


9. 选择**关闭**以关闭“定义标记源”对话框。此时**选择标记**对话框在**标记源**框的列表中**显**示新的**标记**源。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

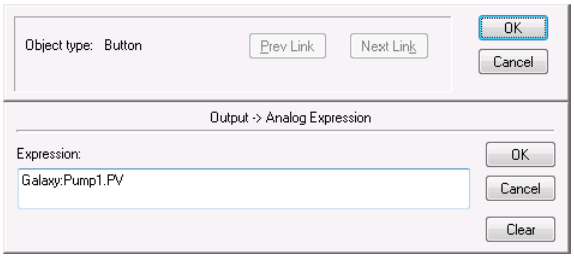


10. 从**标记源**框中**选择**所**创**建的新**标记**源。此**时 Galaxy 浏览**器对话框打**开**，并在其左**侧**窗格中**显**示**标记**列表。



11. 从 **Galaxy 浏览**器对话框的左**侧**窗格中**选择**一个**标记**。此**时 Galaxy 浏览**器对话**框的右**侧**窗格会**进**行刷新，以**显**示所**选标记**的属性。

12. 突出**显**示要使用的属性，然后**选择确定**。此**时**出**现输出 -> 模拟表达式对话**框，并且其**表达式**框中会出**现**一个表达式。



13. 确**认该**字符串表达式是否正确。

    表达式采用以下形式：

    Galaxy:tag_name.attribute_name

    示例：

```
Galaxy:Pump1.PV
```

14. **选择确定**以**关闭输出 > 模拟**表达式对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

15. 根据需要配置其余的对象链接。
16. 在动画链接对话框中选择确定。
17. 选择运行时。此时文本对象显示所配置的标记属性的值。

查看 I/O 标记的时间标签与质量信息

InTouch HMI 在传递给支持 VTQ 的客户端的所有数据值上都会插入数值、时间及质量 (VTQ) 指示符。InTouch 将一组点域用作数据质量的指示符，它们对于疑难排解非常有用。

- .Value 点域包含指定的标记的值。这也是每个 InTouch 标记的缺省点域。如果没有指定其它点域，则假定使用 .Value 点域。
- **Time** 点域是时间标签，表示标记上次更新的时间。
- **Quality** 点域显示指定给 I/O 标记的数据值的可靠性。

查看 I/O 标记的时间标签信息

Time 点域按以下格式指定给标记：

Tag_name.Time_Dotfield

.TimeDate 点域

.**TimeDate** 点域显示自 1970 年 1 月 1 日到上次从“I/O 服务器”更新标记值之间已经过的天数的整数部分。

类别

标记

用法

```
Tag_name.TimeDate
```

参数

Tag_name
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）

另请参阅

.TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeDateString 点域

.TimeDateString 点域显示从“I/O 服务器”中更新标记值时的日期与时间。

类别

标记

用法

```
Tag.TimeDateString
```

参数

Tag

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

消息（只读）。

另请参阅

.TimeDate, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeDateTime 点域

.**TimeDateTime** 点域显示自 1970 年 1 月 1 日到上次从“I/O 服务器”更新标记值之间已经过的天数的小数部分。

类别

标记

用法

```
Tag.TimeDateTime
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

实型（只读）。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeDay 点域

.**TimeDay** 点域显示当月自上次从“I/O 服务器”更新标记值之后已经过去的天数。

类别

标记

用法

```
Tag.TimeDay
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 1 到 31。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

.TimeDate, .TimeDateString, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeHour 点域

.**TimeHour** 点域显示当天自上次从“I/O 服务器”更新标记值之后已经过去的小时数。

类别

标记

用法

```
Tag.TimeHour
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 23。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeMinute 点域

.**TimeMinute** 点域显示上次从“I/O 服务器”更新标记值时的分钟数部分。

类别

标记

用法

```
Tag.TimeMinute
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 59。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.TimeMonth 点域

.**TimeMonth** 点域显示上次从“I/O 服务器”更新标记值时所属的月份 (1-12)。

类别

标记

用法

```
Tag.TimeMonth
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 1 到 12。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeMsec 点域

.**TimeMsec** 点域显示上次从“I/O 服务器”更新标记值时的毫秒部分。

类别

标记

用法

```
Tag.TimeMsec
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 999。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString, .TimeYear

.TimeSecond 点域

.**TimeSecond** 点域显示上次从“I/O 服务器”更新标记值时的秒数部分。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

标记

用法

```
Tag.TimeSecond
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 59。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeTime, .TimeTimeString, .TimeYear

.TimeTime 点域

.TimeTime 点域显示从“I/O 服务器”更新标记值时的时间标签，使用从午夜之后已经过去的毫秒数来表示。

类别

标记

用法

```
Tag.TimeTime
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）。

有效值

值的范围是 0 到 86399999。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTimeString, .TimeYear

.TimeTimeString 点域

.TimeTimeString 点域显示从“I/O 服务器”中更新标记值时的时间。

类别

标记

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用法

```
Tag.TimeTimeString
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

消息（只读）。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeYear

.TimeYear 点域

.**TimeYear** 点域显示从“I/O 服务器”更新标记值时的四位数的年份。

类别

标记

用法

```
Tag.TimeYear
```

参数

Tag
任何离散、整型、实型、消息、间接模拟、间接离散或间接消息标记。

数据类型

整型（只读）。

有效值

以四位数形式表示的任何年份。

另请参阅

.TimeDate, .TimeDateString, .TimeDay, .TimeDateTime, .TimeHour, .TimeMinute, .TimeMsec, .TimeMonth, .TimeSecond, .TimeTime, .TimeTimeString.

查看 I/O 标记的质量信息

您可以使用一组质量点域来确保在“I/O 服务器”与 InTouch 应用程序之间发送的数据的完整性。质量点域代表项目的数据值的质量状态。通过此质量属性，可以非常轻松地监视在网络节点之间发送的 InTouch 数据的完整性。

数据质量标准基于建议的标准“过程控制 OLE”（OLE for Process Control，简称 OPC)，后者又基于各种“现场总线数据质量规格”。

可以根据数据类型和数据质量配置数字值在运行时所使用的格式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

质量数据格式

通过将值指定给一组 InTouch .Quality 点域，“I/O 服务器”可以就发送给客户端的数据质量报告六个互异的状态。Quality 点域的低八位（最低有效字节）目前以三个位域的形式定义：“质量”(Q)、“子状态”(S) 以及“限制”(L)，它们的格式如下：QQSSSSLL。客户端应用程序无法与服务器通讯时，**Quality** 点域的值为 0。

下表显示<1>“</1>I/O 服务器<2>”</2>使用 .Quality 点域报告的数据质量状态<3>：</3>

质量状态	十进制值	十六进制值	最高有效字节 XXXXXXXX	最低有效字节 QQSSSSLL	质量	质量子状态	限制
较好	192	0x00C0	00000000	11000000	Q=3	S=0	L=0
上钳位（超出范围）	86	0x0056	00000000	01010110	Q=1	S=5	L=2
下钳位（超出范围）	85	0x0055	00000000	01010101	Q=1	S=5	L=1
不能转换	64	0x0040	00000000	01000000	Q=1	S=0	L=0
通讯失败	24	0x0018	00000000	00011000	Q=0	S=6	L=0
不能访问点	4	0x0004	00000000	00000100	Q=0	S=1	L=0

关于数据质量点域

.Quality 点域表示上次收到数据时数据值的质量。仅当数据改变由“I/O 服务器”提供时，SuiteLink 与 DDE 协议才会向客户端（例如，WindowViewer）发送更新的质量。因此仅当收到新的数据值时，才能观察到质量改变。某些“I/O 服务器”可以在与数据关联的质量发生改变时，随当前的数据值一起发送更新的质量。

使用 SuiteLink 与 DDE 协议时，可能无法直接引用服务器项目值的质量。<1>要这样做</1><2>, “</2><3>I/O 服务器</3><4>”</4><5>必须直接支持 Item.Quality</5><6>。</6> 没有这种支持，项目便无法继续给出提示，因此 .Quality 点域的值永远保持为 0。

<1>TestProt</1><2>“</2><3>I/O 服务器</3><4>”</4><5>模拟器不直接支持 Item.Quality</5><6>。</6> 在使用 Quality（质量）菜单命令修改质量时，该模拟器不发送新的数据值。

如果要观察 I/O 项目的数据质量<1>，</1>但<2>“</2>I/O 服务器<3>”</3>又不直接支持 Item.Quality 的寻址<4>，</4>请定义一个查看服务器项目的 InTouch I/O 标记<5>，</5>然后监视该 InTouch 标记的 .Quality<6>。</6> 如果超出了标记限制，请考虑在脚本中使用 IOSetRemoteReferences() 函数来动态调整 I/O 点。

SuiteLink 与 DDE 协议不会将连接状态或“I/O 服务器”状态的其它改变诠释为发送给客户端的质量项目。因此，项目的质量不一定要指出当前的数据服务器状态，也不一定要指出客户端-服务器连接的当前状态。<1>“</1>I/O 服务器<2>”</2>进程可以停止<3>，</3>而 .Quality 域并不改变<4>。</4> 如果通讯链接丢失，.Quality 域的值不一定会改变。

使用 DDE 或 SuiteLink 内部状态项目来监视“I/O 服务器”连接。

.Quality 点域

.**Quality** 点域以数值形式评估“I/O 服务器”所提供的数据的质量。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

标记

用法

```
Tag.Quality
```

参数

Tag
任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

整型（只读）。

有效值

值的范围是 0 到 255。

另请参阅

.QualityLimit, .QualityStatus, .QualitySubstatus

示例

```
IF I0Tag.Quality <> 192 THEN
   LogMessage("This data is not Good, assuming high-byte of 2-byte quality is zero");
   LogMessage("Better to check .QualityStatus to avoid this assumption");
ENDIF;
```

.QualityLimit 点域

.**QualityLimit** 点域显示所连接的“I/O 服务器”提供的数据值的质量限制。

类别

标记

用法

```
Tag.QualityLimit
```

参数

Tag
任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

整型（只读）。

有效值

0 = 无限制

1 = 低限制

2 = 高限制

3 = 常数

另请参阅

.Quality

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.QualityLimitString 点域

.**QualityLimitString** 点域显示所连接的“I/O 服务器”提供的数据值的质量限制字符串。

类别

标记

用法

```
Tag.QualityLimitString
```

参数

Tag
任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

消息（只读）。

另请参阅

.Quality, .QualityLimit

.QualityStatus 点域

.**QualityStatus** 点域显示“I/O 服务器”所提供的数据值的整数质量状态。

类别

标记

用法

```
Tag.QualityStatus
```

参数

Tag
任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

整型（只读）。

有效值 (SSSS)

0 = 不佳

1 = 不定

3 = 良好

示例

```
IF I0Tag.QualityStatus <> 3 THEN
   LogMessage("This data is not Good!");
ENDIF;
```

另请参阅

.Quality, .QualitySubStatus

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.QualityStatusString 点域

.**QualityStatusString** 点域显示“I/O 服务器”所提供的数据值的质量状态字符串。

类别

标记

用法

```
Tag.QualityStatusString
```

参数

Tag
任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

消息（只读）。

另请参阅

.QualityStatus, .QualitySubStatus, .Quality

.QualitySubstatus 点域

.**QualitySubstatus** 点域显示“I/O 服务器”所提供的数据值的质量子状态。

类别

标记

用法

```
Tag.QualitySubstatus
```

参数

Tag
任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

整型（只读）。

有效值 (SSSS) 与 (QQ)

“不佳”质量 (QQ=0) 的子状态 (SSSS)。

0 = 非具体

1 = 配置错误

2 = 未连接

3 = 设备故障

4 = 传感器故障

5 = 最后已知值

6 = 通讯故障

7 = 停止服务

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

“不定”质量 (QQ=1) 的子状态 (SSSS)。

0 = 非具体

1 = 最后可用值

4 = 传感器不精确

5 = 超出工程单位

6 = 次正常

“良好”质量 (QQ=3) 的子状态 (SSSS)。

0 = 非具体

6 = 本地重写

另请参阅

.QualityStatus, .QualitySubStatus, .Quality

.QualitySubstatusString 点域

.**QualitySubstatusString** 点域显示“I/O 服务器”所提供的数据值的质量子状态字符串。

类别

标记

用法

```
Tag.QualitySubstatusString
```

参数

Tag
任何离散、整型、实型、间接模拟或消息标记类型。

数据类型

消息（只读）。

另请参阅

.QualityStatus, .QualitySubstatus, .Quality

在运行时初始化与重置 I/O 连接

WindowViewer 在启动 InTouch 应用程序时会启动所有的 I/O 对话。您也可以在 InTouch 应用程序运行期间手动重新启动 I/O 对话。通过使用 WindowViewer 命令或脚本，可以在运行时初始化与重置 I/O 连接。

您也可以指定 I/O 连接应该根据它们的缺省值来重新初始化。如果选择此选项<1>，</1>则重新初始化<2>“</2>访问名<3>”</3>时使用缺省设置<4>，</4>而忽略当前的设置<5>。</5> 要根据“访问名”重新初始化 I/O 对话，InTouch 应用程序必须已经定义“访问名”。

使用命令重新初始化 I/O 连接

WindowViewer 中的特别菜单包含一组命令，可用于重新初始化所有的 I/O 对话或是选择特定的 I/O 对话。

您可以使用 InTouch 的缺省设置来重新初始化“访问名”。使用缺省的重新初始化设置时<1>，“</1>访问名<2>”</2>会忽略指定给节点名<3>、</3>应用程序名以及主题的当前值<4>。</4> 此时使用原始的“访问名”设置来重新初始化“访问名”。

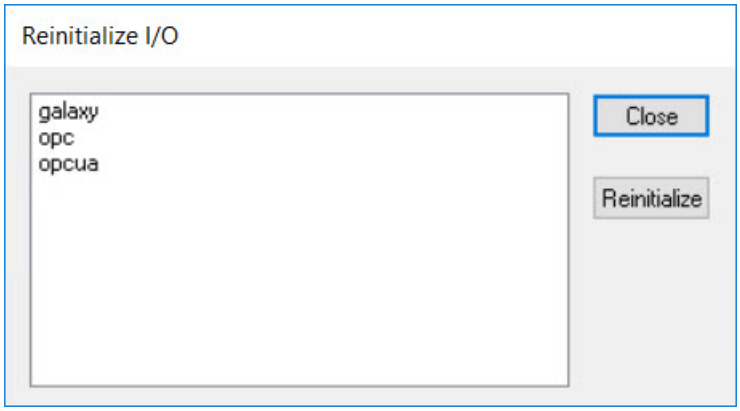
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在运行时重新初始化所有的“访问名”

1. 在特别菜单上，选择重新初始化 I/O。
2. 选择全部重新初始化。此时重新初始化所有的<1>“</1>访问名<2>”。</2>

在运行时重新初始化所选的“访问名”

1. 在特别菜单上，选择重新初始化 I/O，然后单击选择。此时出现重新初始化 I/O 对话框，它显示一个“访问名”列表。



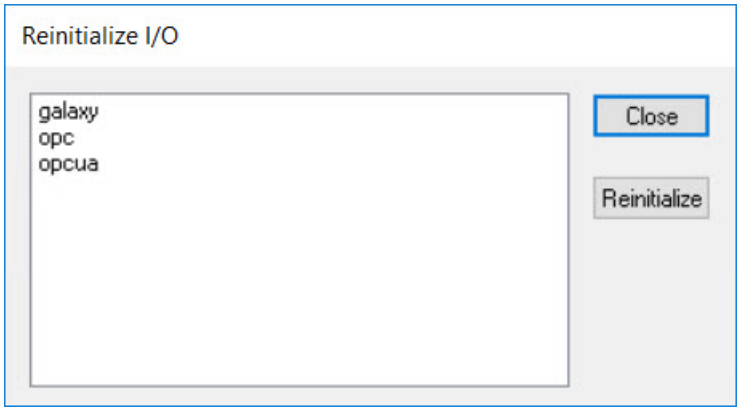
2. 突出显示要重新初始化的一个或多个“访问名”，然后选择重新初始化。此时重新初始化所选的“访问名”。

使用缺省设置重新初始化“访问名”

1. 在 WindowMaker 中打开一个应用程序。
1. 在文件菜单上，指向配置，然后选择 WindowViewer。此时出现 WindowViewer 配置屏幕。
2. 在首选项选项卡上，选中 I/O 区域中的重新初始化缺省值复选框。

3. 选择确定。
4. 在 WindowViewer 中打开应用程序。
5. 在特别菜单上，突出显示重新初始化 I/O，然后单击选择。此时出现重新初始化 I/O 对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



6. 突出显示要重新初始化的一个或多个“访问名”，然后选择重新初始化。此时忽略节点名、应用程序名称以及主题的当前设置。同时使用原始的“访问名”设置来重新初始化“访问名”。

使用脚本重新初始化 I/O 连接

通过创建包含以下函数的脚本，可以重新初始化与一个或多个“访问名”的 I/O 连接：

- **IOReinitAccessName()**
- **IOReinitialize()**
- **IOStartUninitConversations()**

IOReinitAccessName() 函数

IOReinitAccessName() 函数初始化与特定“访问名”的 I/O 连接。

类别

I/O 通讯

语法

```
IOReinitAccessName("AccessName", Default);
```

参数

AccessName
要重新初始化的“访问名”。

Default
缺省值 = 1。I/O 重新初始化过程使用从 WindowMaker 中指定的原始“访问名”缺省值。
缺省值 = 0。I/O 重新初始化过程使用指定给“访问名”的当前节点、应用程序以及主题值。

附注

缺省设置由“访问名”配置面板中的各个设置以及 WindowViewer 配置中的各个设置（重试间隔、启动本地服务器、重新初始化缺省值）确定。

示例

本例使用指定给节点、应用程序以及主题的缺省值重新初始化与 AccessName1 的 I/O 连接。

```
IOReinitAccessName("AccessName1", 1);
```

本例使用指定给节点、应用程序以及主题的当前值重新初始化与 AccessName2 的 I/O 连接。

```
IOReinitAccessName("AccessName2", 0);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

IOReinitialize() 函数

IOReinitialize() 函数首先关闭为 InTouch 应用程序定义的所有活动的 I/O 连接，然后重新启动它们。

类别

其它

语法

```
IOReinitialize();
```

参数

无。

附注

IOReinitialize() 函数与 WindowViewer 特别菜单上的重新初始化 **I/O** 命令执行相同的操作。

如果 WindowViewer 作为服务运行，**IOReinitialize()** 函数与 WindowViewer 特别菜单上的全部重新初始化命令不会重新初始化所有“访问名”。如果选择了在重新初始化 **I/O** 对话框中列出的“访问名”并选择重新初始化，将重新初始化所选的“访问名”。

如需有关导航至重新初始化 **I/O** 对话框的详细信息，请参阅使用命令重新初始化 I/O 连接。

示例

本例关闭任何活动的 I/O 连接，然后重新启动为 InTouch 应用程序定义的所有 I/O 连接。

```
IOReinitialize();
```

IOStartUninitConversations() 函数

WindowViewer 开始运行 InTouch 应用程序时，会自动处理启动请求以启动所有的 I/O 对话。如果“I/O 服务器”程序没有响应 WindowViewer 的启动请求，则可以使用 **IOStartUninitConversations()** 脚本函数来强制 WindowViewer 再次试图启动 I/O 对话。

类别

其它

语法

```
IOStartUninitConversations();
```

参数

无。

附注

IOStartUninitConversations() 函数与 WindowViewer 特别菜单上的启动未初始化的对话命令执行相同的操作。

示例

本例强制 WindowViewer 再提交一个初始化请求，以启动为 InTouch 应用程序定义的所有 I/O 连接。

```
IOStartUninitConversations();
```

使用访问名的故障转移功能

您可以指定让 InTouch HMI 在主“I/O 服务器”遇到通讯问题时自动切换到辅助“I/O 服务器”。这就是所谓的 I/O 故障转移。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置故障转移

您可以指定让 InTouch 应用程序在无法继续与主“I/O 服务器”通讯时切换到用于故障转移的辅助“I/O 服务器”。

设置故障转移时，需要指定故障转移死区。故障转移死区是从主<1>“</1>访问名<2>”</2>切换到辅助<3>“</3>访问名<4>”</4>之前延迟的秒数<5>。</5> 表达式为真所持续的时间或是 I/O 通讯故障所持续的时间超过死区时段长度时，InTouch HMI 便会触发故障转移。故障转移死区设置为 0 或是为空时，检测到 I/O 通讯故障后会立即触发故障转移。

配置“访问名”的故障转移

1. 如果需要，停止 WindowViewer。
2. 在主页菜单上的**标记**组中，选择**访问名**。

   此时出现**访问名对话**框，其中列出所有定义的“访问名”。
3. 从列表中选择要添加故障转移服务器的<1>“</1>访问名<2>”。</2>
4. 选择**编辑**。

   此时出现添加**访问名对话**框。
5. 在**辅助数据源**部分中，选中**启用辅助数据源**复选框。
6. 执行以下操作：
   - 在**节点名**框中，输入辅助“I/O 服务器”的节点名。
   - 在**应用程序名称**框中，输入将从中获取数据的辅助“I/O 服务器”程序的程序名。
   - 在**主题名**框中，输入要从辅助 I/O 数据源中访问的主题名。
   - 在**要使用的协议**区域，选择 **DDE** 或 SuiteLink 作为辅助“I/O 服务器”通讯协议。
   - 在**要对服务器提示时**区域中，为辅助 I/O 数据源选择**提示所有项**或**只提示激活项**。
7. 在**故障转移**部分中，选中**启用故障转移**复选框。
8. 执行以下操作：
   - 在**故障转移表达式**框中输入可选的故障转移表达式，或双击该框以选择一个标记。如需有关故障转移表达式的详细信息，请参阅强制故障转移到备份访问名。
   - 在**故障转移死区**框中，输入以秒为单位的故障转移死区长度。
   - 如果希望启用在故障转移条件消失之后从辅助“访问名”切换回主“访问名”，选择**故障转移条件消失时切换回主数据源**。

     缺省值是不切换回主<1>“</1>访问名<2>”。</2> 如果选择**故障转移条件消失时切换回主数据源**，则**故障转移配置对话**框中的**故障恢复死区选项**变为可以选择的。
   - 从**故障恢复死区**中，输入以秒为单位的故障恢复死区长度。

     在表达式与任何关联的 I/O 通讯故障消失后所持续的时间超过死区时段之后，InTouch HMI 便会触发故障恢复，以恢复到主“访问名”。表达式留为空白或是为 0 时，I/O 通讯故障条件消失后便会立即发生故障恢复。
9. 选择**更新**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

编辑故障转移对的“访问名”参数

要编辑作为故障转移对的一部分的“访问名”参数，必须已经使用辅助 I/O 数据源给“访问名”配置故障转移。

编辑故障转移对的“访问名”参数

1. 如果需要，停止 WindowViewer。
2. 在主**页**菜单上的**标记**组中，选择**访问名**。

   此时出现**访问名**对话框。
3. 突出显示“访问名”对，然后选择**编辑**。

   此时出现**添加访问名**对话框，显示主“访问名”与辅助“访问名”的参数。
4. 更改主“访问名”与辅助“访问名”的“访问名”参数。
5. 选择**更新**。

删除访问名的故障转移

要删除<1>“</1>访问名<2>”</2>的故障转移<3>，</3>必须已经使用辅助 I/O 数据源给<4>“</4>访问名<5>”</5>配置了故障转移<6>。</6>

删除“访问名”对的故障转移

1. 在主**页**菜单上的**标记**组中，选择**访问名**。
2. 突出显示“访问名”对，然后选择**编辑**。

   此时出现**添加访问名**对话框。
3. 清除**启用辅助数据源**复选框。
4. 选择**确定**。

   此时“访问名”对的故障转移功能便已禁用。

强制故障转移到备份访问名

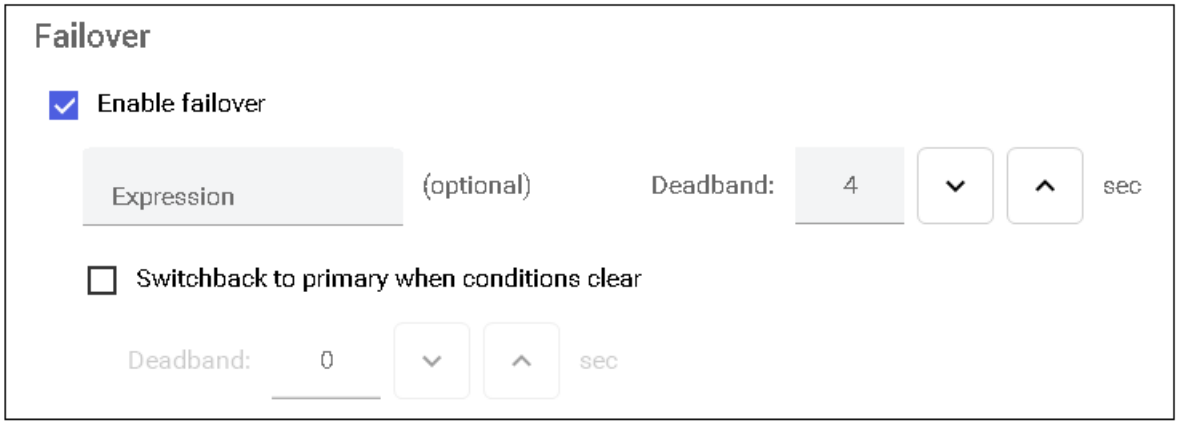
您可以在没有遇到故障转移的情况下，手动在“访问名”的主数据源与辅助数据源之间切换。这就是所谓的强制故障转移。要强制进行故障转移<1>，</1>必须已经使用辅助 I/O 数据源给<2>“</2>访问名<3>”</3>配置了故障转移<4>。</4>

您可以使用故障转移表达式或 **IOForceFailover()** 脚本函数来强制进行故障转移。

故障转移表达式

故障转移配置部分显示一个**故障转移表达式选项**，用以包含触发故障转移的**标记**或表达式。下图显示作为**故障转移表达式**的**值**输入的故障转移内存离散**标记**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



将 failover 表达式设置为真<1>（</1>例如<2>，</2>通过将 failover 标记设置为真<3>），</3>可以将<4>“</4>访问名<5>”</5>从主<6>（</6>假<7>）</7>I/O 数据源切换到辅助<8>（</8>真<9>）</9>I/O 数据源<10>。</10>

IOForceFailover() 函数

IOForceFailover() 脚本函数在“访问名”的主数据源与辅助数据源之间进行切换。每次激活脚本函数时，活动的 I/O 节点在主节点与辅助节点之间切换。

通常，**IOForceFailover()** 函数是与按钮或另一个窗口对象关联的脚本中的一部分。操作员从应用程序窗口中选择该对象，以便强制进行故障转移。操作员再次选择该对象之后，**IOForceFailover** 函数强制将 I/O 连接切换回之前活动的 I/O 节点。

类别

I/O 通讯

语法

```
IOForceFailover("AccessName");
```

参数

AccessName
<1>配置了故障转移的</1><2>“</2><3>访问名</3><4>”。</4>

示例

Acc1<1>“</1>访问名<2>”</2>有<3>“</3>主<4>”</4>数据源与<5>“</5>辅助<6>”</6>数据源<7>，</7>其中<8>“</8>主<9>”</9>数据源处于活动状态<10>。</10> 脚本运行时<1>，</1>Acc1 发生故障转移并切换到<2>“</2>辅助<3>”</3>数据源<4>。</4>

```
IOForceFailOver("Acc1");
```

临时禁用故障转移功能

您可以手动禁用在“访问名”的主 I/O 节点与辅助 I/O 节点之间进行切换的故障转移功能。InTouch 系统的各个组件已经启动但尚未准备就绪的很短时间，便是需要临时禁用故障转移的典型情况。在各个组件都稳定之后，便可以恢复故障转移切换功能。

要禁用<1>“</1>访问名<2>”</2>的故障转移<3>，</3>必须已经使用辅助 I/O 数据源给<4>“</4>访问名<5>”</5>配置了故障转移<6>。</6>

您可以通过两种方法来手动禁用故障转移切换功能：

- 从故障转移部分中选择禁用故障转移选项。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 运行包含 **IODisableFailover()** 函数的脚本。

禁用故障转移配置选项

在故障转移配置部分中，清除启用故障转移选项的选择，以防止“访问名”在主 I/O 节点与辅助 I/O 节点之间进行切换。

您必须编辑“访问名”定义才能将禁用故障转移选项设置为活动状态。只要“访问名”的该选项处于活动状态，便会禁用故障转移。

IODisableFailover() 脚本函数

您可以在脚本中使用 **IODisableFailover()** 函数给指定的“访问名”禁用故障转移功能。**IODisableFailover()** 禁用除 **IOForceFailover()** 脚本函数方法之外的所有故障转移方法的切换功能。

类别

I/O 通讯

语法

```
IODisableFailover("AccessName",Option);
```

参数

AccessName
配置了故障转移的“访问名”。

Option
1 = 禁用故障转移
0 = 启用故障转移

附注

“访问名”可以指定为字符串，或是使用其它 InTouch 标记或函数提供的字符串值。

示例

本例禁用 ModbusPLC1“访问名”的故障转移功能。

```
IODisableFailover ("ModbusPLC1",1)
```

本例启用 ModbusPLC1“访问名”的故障转移功能。

```
IODisableFailover ("ModbusPLC1",0)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用脚本检索故障转移对的有关信息

您可以编写包含一些函数的脚本，这些函数返回“访问名”的主 I/O 数据源、辅助 I/O 数据源以及活动的 I/O 数据源的状态。通常，操作员在强制进行故障转移之前，会运行脚本以确定“访问名”的辅助 I/O 数据源的状态。

要创建脚本来返回“访问名”信息，必须已经使用辅助 I/O 数据源给“访问名”配置了故障转移。

IOGetAccessNameStatus() 函数

IOGetAccessNameStatus() 脚本函数返回一个整数，指出“访问名”的主 I/O 数据源、辅助 I/O 数据源或活动的 I/O 数据源的连接状态。

通常，**IOGetAccessNameStatus()** 返回值与整型标记关联。标记的值可以驱动离散值显示动画链接，向操作员显示“访问名”活动的 I/O 数据源、主 I/O 数据源以及辅助 I/O 数据源的状态。

类别

其它

语法

```
Result=IOGetAccessNameStatus("AccessName", Mode);
```

参数

AccessName
要返回其状态的现有“访问名”。

Mode
指定给此参数的值确定查询故障转移对哪个“访问名”的当前状态。
0 -“访问名”的活动 I/O 数据源的状态
1 -“访问名”的主 I/O 数据源的状态
2 -“访问名”的辅助 I/O 数据源的状态

结果

返回值	描述
-1	“访问名”中存在配置错误。“访问名”不存在，或未定义“访问名”的辅助 I/O 数据源。
0	未成功连接到所请求的 I/O 数据源。
1	成功连接到所请求的 I/O 数据源。

附注

IOGetAccessNameStatus() 函数通常在脚本中用于确定辅助 IO 数据源（当前在不活动状态）的状态。在强制进行故障转移之前，操作员运行此脚本以验证辅助连接的状态。

示例

本例返回 ModbusPLC1“访问名”的辅助 I/O 数据源的状态。返回值与 **ANStatus** 标记关联。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
ANStatus = IOGetAccessNameStatus ("ModbusPLC1",2)
```

IOGetActiveSourceName() 函数

IOGetActiveSourceName() 脚本函数返回**访问**名是否当前使用其主数据源或**辅**助数据源。

通常，**IOGetActiveSourceName()** 函数包含在与按**钮**或另一个窗口**对**象相**关联**的脚本中。随后，操作**员**从**应**用程序窗口中**选择该对**象，以**请**求**应**用程序的“I/O 服**务**器”的状**态**。

类别

其它

语法

```
Result=IOGetActiveSourceName("AccessName");
```

参数

AccessName
要返回其数据源名的**现**有“**访问**名”。

附注

IOGetActiveSourceName() 返回一个字符串，指出正在**轮询**“**访问**名”的主**节**点**还**是**辅**助**节**点。
IOGetActiveSourceName() 函数可能的返回**值**如下：

主要	正在**轮询**“**访问**名”的主**节**点。
辅助	正在**轮询**“**访问**名”的**辅**助或故障**转**移**节**点。
Null	“**访问**名”的主**节**点与**辅**助**节**点都**处**于不活**动**状**态**。

示例

在本例中，返回**值**（Primary、Secondary 或 Null）指定**给 ActiveServer** 消息**标记**，用于确定 ModbusPLC1“**访问**名”的当前活**动节**点。

```
ActiveServer = IOGetActiveSourceName ("ModbusPLC1");
```

监视 I/O 连接的状态

WindowViewer 包含一个内置的主**题** IOStatus，用于**监视** InTouch **应**用程序与“I/O 服**务**器”（同 PLC 通**讯**）之**间**的特定 I/O **对话**的状**态**。

注意：在 7.0 之前的 InTouch 版本中，此主**题**名是 DDEStatus。

您可以**设**置 **IOStatus** 主**题**以**监视** I/O **对话**。

使用 IOStatus 主题名

您可以**预备 IOStatus** 主**题**，以便**监视** WindowViewer 与“I/O 服**务**器”之**间**的 I/O 通**讯**情况。在本例中，WindowViewer 与“模**拟** I/O 服**务**器”通**讯**，后者又与“I/O 服**务**器”中定**义**的主**题**名**为** PLC1 的 PLC **进**行通**讯**。

注意：Simulate Server 是一个用作培**训**工具的通用型 DAServer。Simulate Server 位于 c:\program files\common files\ArchestrA 文件**夹**。

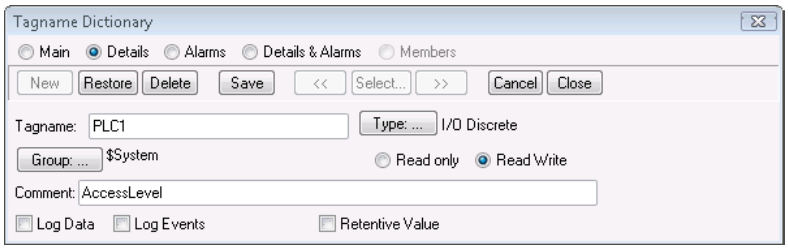
监视 I/O 通讯的状态

1. 在 WindowMaker 中打**开**一个**应**用程序。

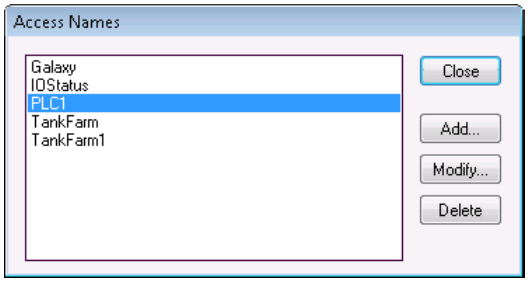
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 打开“标记名字典”。
3. 创建一个<1>“</1>I/O 离散<2>”</2>标记<3>。</3>

   使用 IOStatus 监视 I/O 对话时，必须为要监视的“访问名”定义至少一个 I/O 型标记。

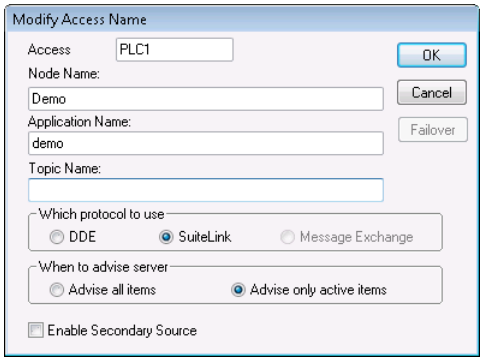


4. 选择访问名以便将标记指定给“访问名”的定义，此定义将 IOStatus 定义为主题名。



   请注意，**PLC1** 的“访问名”定义当前已经存在。

5. 突出显示 PLC1 并选择修改。

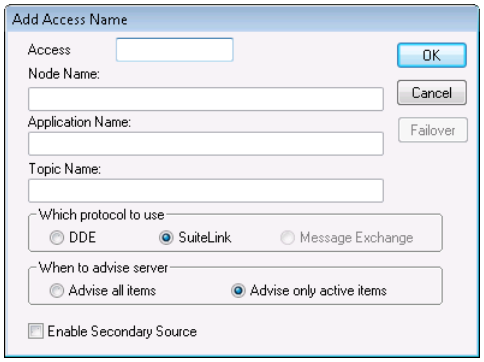


   在本例中，由于标记与主题名相同，因此很容易找到包含正确主题名的“访问名”。

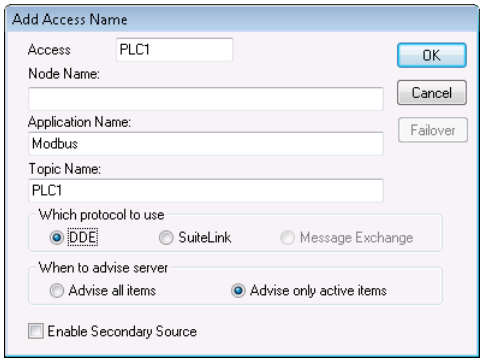
6. 选择取消以关闭对话框，并返回到最初的**访问名**对话框。
7. 选择添加。

   此时出现添加**访问**名对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



8. 执行以下操作：
   a. 在**访问名**框中，输入 IOStatus。
   b. 由于打算要从 WindowViewer 中监视状态，因此请在**应用程序名称**框中输入 **View**。
   c. 在**主题名**框中，输入 **IOStatus** 作为 InTouch 内部主题。
   d. 选择**只提示激活项**。
9. 选择**确定**以关闭对话框。此时再次出现最初的**访问名**对话框，并且列表中显示新的“访问名”IOStatus：



10. 选择**关闭**以关闭对话框，并将新的访问名关联到 I/O 离散标记。

    在项目框中，为要监视的实际主题名输入“访问名”。

11. 由于标记与主题名相同，因此可以选择将“标记名”用作“项目名”，并自动将它作为项目进行输入。

注意：使用内置的主题 IOStatus（在 InTouch 7.0 版之前是 DDEStatus）监视 I/O 对话时，在“访问名”框中输入的名称总是同时用于项目。

“项目名”可以是最多包含 254 个字符的字符串。项目名长度为主题名、项目名和 1 个分隔符之和。

在 Excel 中使用主题名 IOStatus

通过输入相同的信息作为电子表格单元格的公式，可以使用 Excel 来监视 I/O 状态。例如，要监视与前面的操作程序所介绍的相同主题，请在单元格中输入以下公式：

```
=view|IOStatus!'PLC1'
```

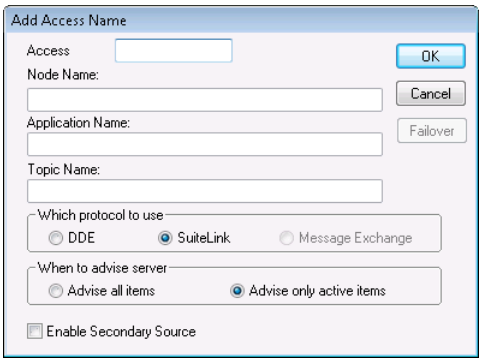
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

监视 I/O 服务器通讯状态

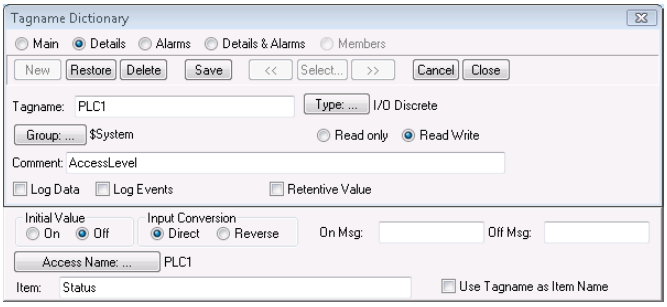
对于使用的每个主题名，都可以使用内置的离散项 **Status** 来监视与“I/O 服务器”程序之间的通讯状态。发生通讯故障时，**Status** 项目设置为 0。与“I/O 服务器”程序之间正常通讯时，**Status** 项目设置为 1。

注意：使用 IOStatus 项目来监视主题的状态时，所监视的主题至少必须有一个活动的 I/O 点。

从 InTouch HMI 中，通过定义标记并将它关联到给设备配置的主题（通过将 **Status** 用作“项目名”），可以读取服务器通讯的状态。例如，如果 WindowViewer 正在使用 Modbus DAServer 与 PLC 进行通讯，则“访问名”定义类似于下例：



要监视与 PLC1 主题之间的所有通讯的状态，请创建以下标记定义：



在 Excel 中，通过在单元格中输入以下公式，可以读取 PLC 通讯的状态：

```
=SIMULATE|PLC1!'STATUS'
```

从其它应用程序中访问 InTouch 标记数据

其它应用程序向 InTouch HMI 请求数据值时，它必须知道三个 I/O 地址项。请遵循这些 InTouch I/O 地址惯例。

VIEW（应用程序名）确定包含数据元素的 InTouch 运行时程序。

TAGNAME（主题名）是读取/写入标记时总是要使用的字。

ActualTagname（项目名）是在 InTouch 的“标记名字典”中给项目定义的实际标记。

例如，要从 Excel 访问 InTouch HMI 中的数据值，需要为将写入数据的单元格指定一个“DDE 远程调用”公式：

=VIEW|TAGNAME!'ActualTag_name'

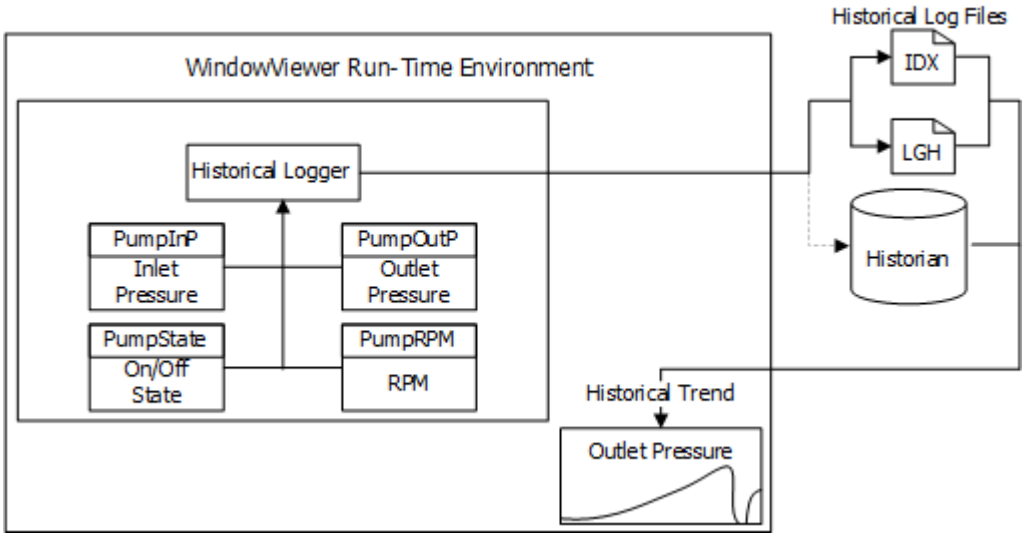
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

记录标记值

InTouch 应用程序运行期间，其标记值可以记录并永久保存。您可以使用此日志数据来创建历史趋势图，以显示一段时间内工厂过程的某些方面。

注意：对于大型应用程序，或是要创建详细报告时，应使用 Historian 服务器保存 InTouch 历史数据。如需有关配置历史记录功能的详细信息，请参阅 Historian 服务器文档。

下图显示一个泵的标记数据，这些数据保存在历史日志文件和/或 Historian 服务器数据库中。每次标记值的变化超出指定的记录死区范围时，InTouch Historical Logger 便写入一个日志项。



将数据存储到历史日志文件时，InTouch HMI 创建两个日志文件。一个文件包含按专有格式存储的记录数据。另一个文件是这些数据的索引。

这两个日志文件的名称按以下格式指定：

YYMMDD00.LGH 和 *YYMMDD00.IDX*

其中：

YY	创建日志文件的年份的后两位数字
MM	创建文件的月份的两位数字 (01-12)
DD	创建文件的日期的两位数字 (01-31)
00	日志文件名中的常数值 00

每天的记录周期从午夜开始并在午夜结束。Historical Logger 在午夜将最后一些项目写入活动的日志文件，并将它们进行归档。此时创建要在第二天使用的两个新文件，并在其中记录数据。

日志文件会保存指定的天数。超过保留期限的日志文件将会删除。如需有关设置日志文件保留天数的详细信息，请参阅配置历史日志文件的常规记录属性。

配置历史记录

为 InTouch 应用程序配置历史记录功能时，需要执行三个主要任务：

- 配置要记录历史的标记

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 配置 InTouch 应用程序的常规记录属性
  - 配置在 InTouch 历史日志文件中存储标记值的属性和/或
  - 配置在 Historian 服务器中存储标记值的属性
- 作为可选项，设置历史记录频率

注意：可以配置为记录到 LGH 文件和/或 Historian 服务器。

配置要记录历史的标记

您可以从“标记名字典”中选择要记录历史的标记。所选标记的值发生改变时，Historical Logger 根据每个标记的记录死区与当前值来确定是否应该将一个项目写入日志。

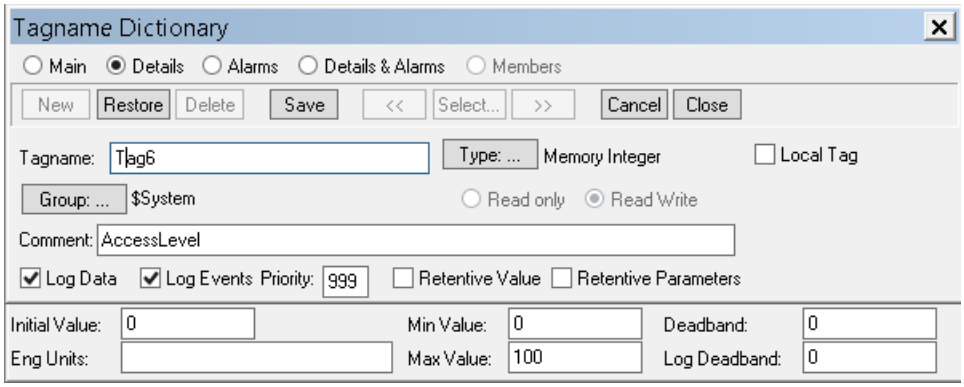
如果将某个标记从要记录更改为不记录，则与该标记关联的数据将不再保存到日志文件。重新启用记录时会恢复记录。不过历史趋势会在禁用记录功能的期间显示一段空隙。

在 WindowViewer 运行应用程序期间，对标记记录功能所作的更改会被忽略。只有停止并重新启动正在运行的应用程序之后，对标记记录所作的更改才会生效。

您可以从<1>“</1>标记名字典<2>”</2>中为每个单独的标记配置记录功能<3>。</3>

为标记启用历史记录功能

1. 如果需要，停止 WindowViewer 中正在运行的应用程序。
2. 使用 WindowMaker 打开应用程序。
3. 打开“标记名字典”。
4. 从<1>“</1>标记名字典<2>”</2>的列表中选择要记录其数据的标记<3>。</3>
5. 选中记录数据复选框。



标记名字典对话框包含与记录功能紧密关联的其它标记属性。

- 记录死区设置一个工程单位阈值，标记的值必须超过这个值才会写入日志文件。只有超出死区的新值才会写入日志文件。死区范围内的微小值变化将被忽略。
- 最小工程单位与最大工程单位属性将原始值缩放到一定的工程单位范围内。最小及最大工程单位属性设置缩放值的上下边界。

  最小/最大工程单位确定趋势中显示的日志值的范围边界。缺省条件下，InTouch 历史趋势显示 0-100% 工程单位范围内的日志数据。

6. 选择保存。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

7. 重复这些步骤，为要记录数据的每个标记启用记录功能。
8. 完成时选择**关闭**，以关闭“标记名字典”对话框。

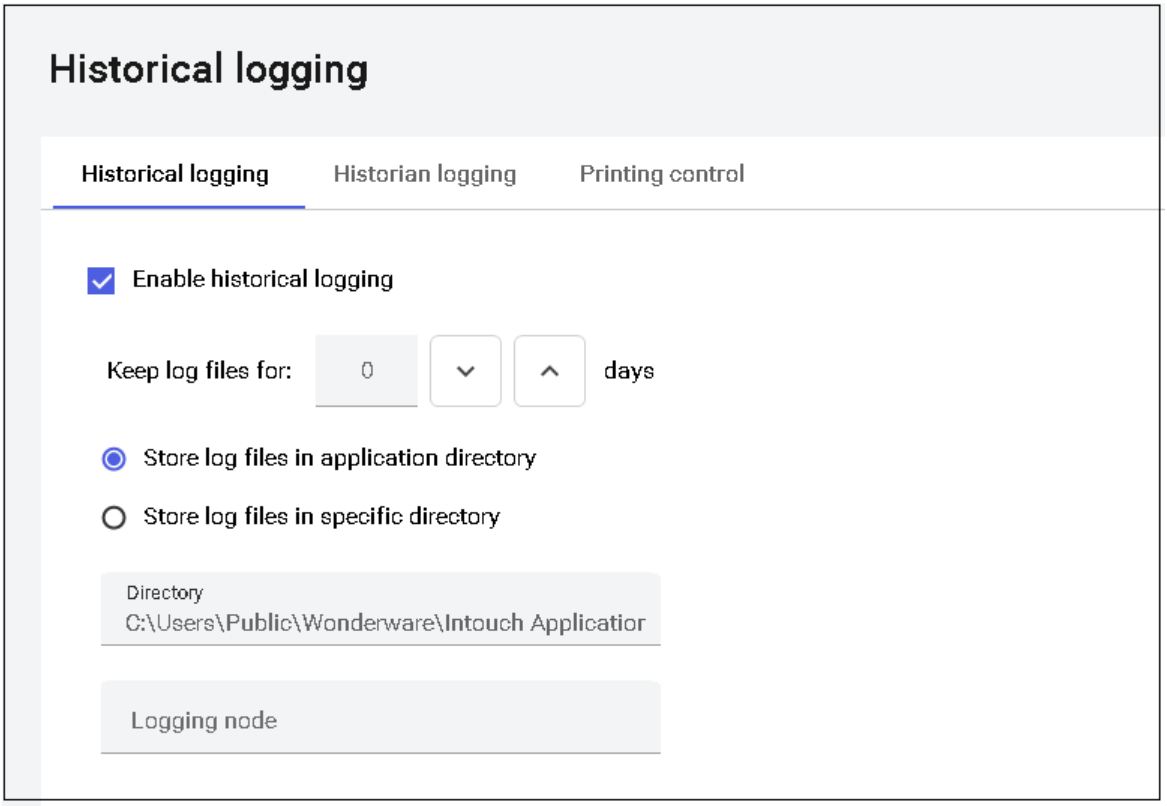
配置历史日志文件的常规记录属性

您可以设置要应用于所选应用程序的常规记录属性。

配置历史记录

1. 如果需要，请关闭 WindowViewer 中正在运行的 InTouch 应用程序。
2. 打开 WindowMaker。
3. 在文件菜单上，选择配置，然后选择**历史记录**。

   此时出现**历史记录**配置屏幕。



4. 选中**启用历史记录**复选框。
5. 在**保持日志文件时间**框中，输入当前日期之前要保留日志文件的天数。

   当天以及指定的保留期限内的日志文件都会保留下来。超过保留期限的日志文件将会删除。将此值设置为 0 时，则会无限期保留所有日志文件。

   示例：

   将保留期限设置为五天，并从当月的第一天开始记录。在当月的第七天，保留的是前五天以及当天 (02-07) 的日志文件。当月第一天创建的日志文件会被删除。

   设置保存记录数据的天数时，要考虑磁盘空间使用情况。如果硬盘可用空间不足，历史记录功能会停止。您必须释放磁盘空间才能恢复记录功能。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 选择要在其中保存日志文件的文件夹位置。

   注意：存储日志数据的文件夹路径与文件名最多可以包含 55 个字符。

   选择在应用程序目录中存储日志文件，以便将日志文件保存在创建记录数据的 InTouch 应用程序的相同文件夹中。

   选择在指定目录中存储日志文件，以指定要用于存储日志文件的另一个文件夹。您可以按以下格式指定要用于存储日志文件的文件夹：

   - Windows 文件夹路径，如 C:\History Log Files
   - “通用命名惯例”（Universal Naming Convention，简称 UNC）路径，如 \\node\share\directory。

   如果要将日志文件保存到分布式节点上，则必须按 UNC 路径格式来指定目录。

   如果配置为将历史数据写入 NAD 储备库应用程序节点的“应用程序目录”，则所有 NAD 节点都会尝试将历史数据写入 NAD 储备库应用程序。为避免这种情况，请将每个 NAD 节点上的历史数据配置为写入本地目录，而不是 NAD 储备库应用程序节点。

7. 在记录节点框中，输入创建日志数据的 InTouch 应用程序运行所在的计算机的节点名。
8. 选择保存。

   对记录功能的配置所作的更改将立即生效。记录功能将在下次运行应用程序时开始记录数据。

如需有关设置趋势打印的详细信息，请参阅在运行时打印趋势。

配置常规记录属性存储到 Historian

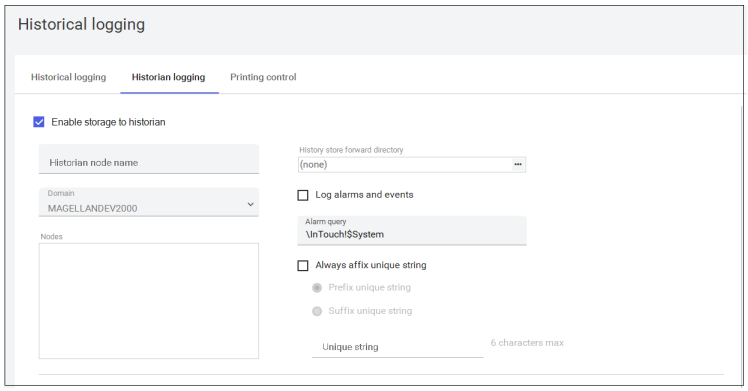
您可以设置要应用于所选应用程序的常规记录属性。

配置存储到 Historian

1. 如果需要，请关闭 WindowViewer 中正在运行的 InTouch 应用程序。
2. 打开 WindowMaker。
3. 在文件菜单上，选择配置，然后选择历史记录。

   此时出现历史记录配置屏幕。

4. 选择 Historian 记录选项卡。



5. 选中启用存储到 Historian 复选框。

6. 在 **Historian 节点名**框中，输入运行 Historian 服务器的计算机的节点名。您也可以指定 IP 地址，如果服务器安装在同一台计算机上，则可以使用“localhost”。只允许使用以下特殊字符：句点 (.)、下划线 (_) 和连字符 (-)。
7. 在**历史存储转发目录**框中，输入本地文件夹位置的路径，该位置用于存储与存储转发相关的文件。如果失去与 Historian 服务器的连接，这些文件将允许临时存储历史数据。建立连接后，Historian 服务器将与该存储转发目录中的文件同步并保留所有信息。
8. 选择**记录报警和事件**以启用记录报警和事件。在**报警查询**文本框中指定报警查询。

   注意：如果不同的 InTouch 应用程序将标记数据存储到同一 Historian 服务器并使用相同的标记名，则 InTouch 应用程序不会检测到这种情况，并可能导致 Historian 数据重叠。使用唯一的前缀或后缀来区别应用程序。如需详细信息，请参阅配置附加字符串。

9. 选择保存。

   对记录功能的配置所作的更改将立即生效。记录功能将在下次运行应用程序时开始记录数据。

如需有关设置趋势打印的详细信息，请参阅在运行时打印趋势。

注意：当通过 OCMC 使用 tagname.x 文件将 InTouch 标签手动导入 Historian 时，WindowViewer 必须以管理员模式运行才能将数据归档。在非管理员模式下，此方法不支持数据归档。有关如何使用 tagname.x 文件导入 InTouch 标签的信息，请参阅 Historian 帮助中的导入或重新导入字典主题。

配置附加字符串

在多个节点上使用相同的应用程序，并连接到相同的 Historian 服务器时，可以使用唯一后缀或前缀来区分不同节点的标记。您可以通过 WindowMaker 或者通过手动编辑每个节点上的“dhistcfg.ini”文件来配置字符串。

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择历史记录。在“历史记录”配置屏幕中，选择 **Historian 记录**选项卡。
3. 选中启用存储到 **Historian** 复选框。
4. 选中始终附加唯一字符串复选框。
5. 在前缀唯一字符串和后缀唯一字符串中选择一个。
6. 在唯一字符串字段中输入字符串。字符串最多可以包含 6 个字符。
7. 选择确定。

在 .ini 文件中手动编辑附加字符串。

您可以通过编辑每个节点上的应用程序文件夹中提供的 dhistcfg.ini 文件来更新附加字符串。

1. 导航到应用程序文件夹。
2. 编辑 dhistcfg.ini 文件。

   示例 1：指定前缀唯一字符串“aa”

```
szHistorianNode=<MachineName>
bStorageLoggingEnabled=1
bAffixEnabled=1
bPrefixEnabled=1
bSuffixEnabled=0
szHistUniqueString=aa
```

   示例 2：指定后缀唯一字符串“xx”

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

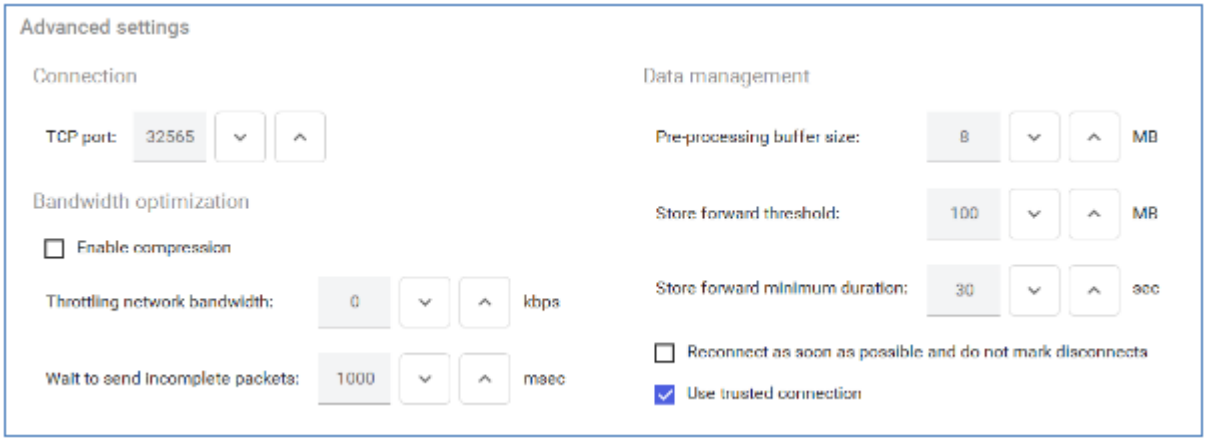
```
szHistorianNode=<MachineName>
bStorageLoggingEnabled=1
bAffixEnabled=1
bPrefixEnabled=0
bSuffixEnabled=1
szHistUniqueString=xx
```

3. 保存并关闭 .ini 文件。

重新启动 WindowViewer 之后，所有更改都将生效。

配置高级设置

更新高级设置部分，以指定与 Historian 连接相关的设置。



1. 在连接下，可以指定 **TCP 端口**。历史数据将发送到的 Historian 服务器节点上的 TCP 端口。TCP 端口在安装 Historian 服务器时配置。缺省端口为 32565。缺省 Historian TCP 端口是可配置的，用于数据复制以及与远程 IDAS 2023 R2 版及更高版本的通讯（也就是 gRPC 通讯）。

   注意：支持 WCF 通讯的经典 Historian TCP 端口 32568 用于数据复制以及与 Historian 2023 版及更早版本进行远程 IDAS 通讯，并且保留与以前版本的兼容性。在 InTouch HMI 2023 及更早版本中创建的应用程序的缺省端口号是 32568。在升级过程中，迁移后的应用程序的 TCP 端口将自动更新为 32565。但是，如果您已手动修改端口号，那么修改的端口号将保留在升级的应用程序中。

2. 在**带宽优化**部分下，可以选中**启用压缩**复选框。选中后，可以指定：
   - **调节网络带宽**：指定当与 Historian 通讯时 HCAL 使用的网络通讯的带宽使用限制（以 kbps 为单位）。值为 0 时禁用此功能（缺省值）。如需有关估计带宽需求的详细信息，请参阅《System Platform 安装》中针对 Historian 服务器的性能和规模调整建议。允许的值为 0 kbps 到 65535 kbps。
   - **等待发送不完整的包**：指定在将部分填充的 Historian 客户端访问层 (HCAL) 缓冲区发送到 Historian 服务器之前将其保留的最长时间（以毫秒为单位）。如果该缓冲区已满，则会立即发送，不管此字段中配置的值如何。如果数据快速变化，则此设置不相关。如果数据变化较慢且网络带宽有限，则最好增加此值，以减少网络的“抖动”。允许的值为 1000 毫秒到 30000 毫秒。

3. 在**数据管理**部分下，可以指定：
   - **预处理缓冲区大小**：Historian 客户端访问层 (HCAL) 使用的所有缓冲区的总大小（以 MB 为单位）。缺省值（同时也是最小值）为 8。如果有非常高的数据突发，则应该增加此值。如果此值过低，会发生数据丢失，并在 Logger 中出现与缓冲区溢出有关的错误消息。以 10 MB 为增量增加此值。允许的值为 8 MB 到 65535 MB

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- **存储转发阈值**：HCAL 存储和转发磁盘上保留的可用空间的大小（以 MB 为单位）。在存储和转发期间不会使用指定的空间。此值不能是负数。允许的值为 0 MB 到 65535 MB
- **存储转发最小期间**：HCAL 在存储和转发模式下运行的最小期间（以秒为单位）。即使导致 HCAL 在存储和转发模式下运行的条件不再存在，HCAL 也将在此时长范围内在存储和转发模式下运行。允许的值为 30 秒到 3600 秒。
- **选中尽早重新连接，不要标记断连复选**框：指定在 Application Server 与 Historian 之间的通讯断开期间如何显示趋势。通讯恢复后，历史数据趋势的显示方式不受影响。如果为 TRUE，客户端趋势在断开间隔不会出现空隙。断开后，将使用断开前最新收到的值填充间隔。如果为 FALSE，客户端趋势在断开间隔将出现空隙。断开时将注入 NULL 值以产生空隙。在这两种情况下，重新连接后，都将使用存储转发数据填充间隔。
- **选中使用受信任的连接复选**框：确保仅与 Historian 服务器进行安全通讯。
  缺省条件下，所有新应用程序和迁移的应用程序都会启用**使用受信任的连接复选**框。
  - 当选中**使用受信任的连接复选**框时，只有信任的连接才能与 Historian 服务器进行通讯。如果客户端与 Historian 节点之间没有共同的证书，连接将会失败。
  - 当未选中**使用受信任的连接复选**框时，即使记录器中存在 SSL 错误，也会建立与 Historian 的连接。

注意：我们建议您在配置器中配置 SMS 并选中此复选框以确保安全连接。

4. 选择保存。

支持 WCF 和 gRPC 通讯的服务

当 Historian 版本为 2023 R2 或更高版本时，确保以下服务正在运行以支持向后兼容性。

- aahClientAccessPoint（WCF 通讯）- 支持 2023 版及更早版本的客户端。
- aahClientAccessPointNG（gRPC 通讯）- 支持 2023 R2 版及更高版本的客户端。

控制历史记录频率

作为可选项，您可以基于两个条件来指定要记录的项：

- 只要标记值的变化超出记录死区值一个工程单位值，则立即写入一个日志项。
- 按固定的间隔写入记录的所有标记的当前值。记录的所有标记对应的日志项都会写入日志，而不论它们的当前值如何。缺省的固定间隔为 60 分钟。

您可以接受缺省的固定记录间隔，也可以在 intouch.ini 文件中添加两个参数来更改这个间隔。

- *ForceLogging*

  ForceLogging 以分钟为单位指定固定记录间隔的长度。*ForceLogging* 可以设置为 5 到 120 分钟之间的某个值。缺省值是 *ForceLogging=60*。

- *ForceLogCurrentValue*

  ForceLogCurrentValue 指定按固定的记录间隔记录当前标记值，即使它仍然在上一个记录值的记录死区范围内。设置为 0 时，再次记录上一个记录值。缺省值是 *ForceLogCurrentValue=0*。

下例显示包含这两个记录参数的示例 intouch.ini 文件。

```
WinFullScreen=1
WinWidth=808
AlarmBufferSize=5000
ForceLogging=5
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
ForceLogCurrentValue=1
```

在本例中<1>，</1>标记值按照五分钟的间隔写入<2>“</2>历史日志<3>”</3>文件<4>。</4>

修改历史记录频率

1. 关闭 WindowMaker 和 WindowViewer。
2. 在与 InTouch 应用程序相同的文件夹中找到 intouch.ini 文件。
3. 编辑 intouch.ini 文件。
4. 在 *ForceLogging* 语句中插入 5 与 120 之间的一个值。
5. 插入 *ForceLogCurrentValue=1* 语句。
6. 保存更改并关闭 intouch.ini 文件。
7. 重新启动 WindowViewer。

在运行时启动与停止历史记录

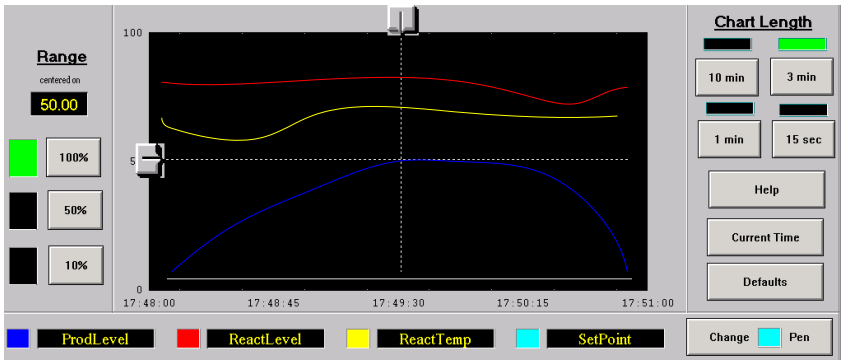
应用程序正在运行时，可以使用 WindowViewer 特别菜单中的命令来手动停止然后再重新启动历史记录功能。

- **停止历史记录**命令可以在当前应用程序会话期间停止记录功能。记录功能在当前会话期间一直处于停止状态，直到再次手动启动它。
- 使用**停止历史记录选项**手动停止记录功能之后，**重新启动历史记录**命令可以重新启动记录功能。

您也可以在应用程序中添加一个按钮，然后编写一个包含 **$HistoricalLogging** 系统标记的 QuickScript 来启动与停止历史记录功能。当为 **$HistoricalLogging** 指定值 1 时，开始记录。当为 **$HistoricalLogging** 指定值 0 时，停止记录。有关 **$HistoricalLogging** 系统标记的详细信息，请参阅系统标记。

绘制标记数据的趋势

您可以创建一些趋势图，以图形化的方式显示从 InTouch 应用程序采集的数据。WindowViewer 包含一组可用于创建历史与实时趋势的实用程序与向导。下例显示一个 InTouch“平均/散点”趋势的示例。



您也可以使用一组趋势控件。通过使用这些控件，可以选择要在趋势中显示的数据及其在趋势中的显示方式。

您可以配置实时与历史趋势。这两种类型的趋势都包含一些配置选项，可用于设置趋势的数据采集间隔和视觉外观。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 趋势的类型

历史趋势显示过去采集并存储在 InTouch 数据储备库中的记录数据。

通过使用分布式历史系统，从可访问的网络节点上的任何 InTouch 历史记录文件中，都可以检索到历史数据。分布式历史系统将远程记录数据库涵括在内，从而扩展了历史趋势的检索能力。

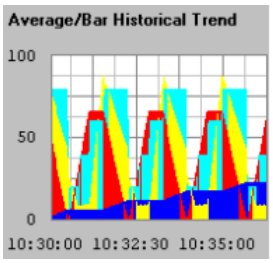
实时趋势会持续进行更新，可以在数据产生之后相对较短的时间段内显示它们。您可以使用 WindowMaker 的“实时趋势”工具在窗口中创建趋势对象。如果安装了可选的“16 笔趋势”工具，则也可以创建实时趋势来最多显示 16 个标记的数据。

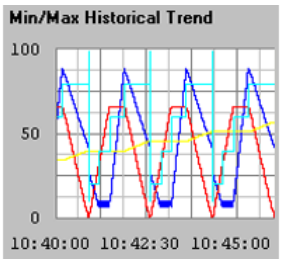
理解历史趋势

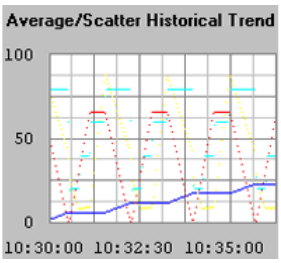
历史趋势显示过去一个连续时间段内的数据。与实时趋势不同，历史趋势只能通过脚本或操作员动作进行更新。

历史趋势使用图形化的表示法来最多显示八个标记的数据。通过将标记指定给趋势笔，可以指定要在历史趋势中显示的数据。

下图显示三种类型的 InTouch 历史趋势。







- “均值/棒图”历史趋势以条形图的形式显示一定时间间隔内数据点的平均值。
- <1>“</1>最小/最大<2>”</2>历史趋势以工程单位的百分比为纵坐标<3>，</3>以时间宽度为横坐标来显示所发生的变化<4>。</4> 它强调的是时间流与变化率，而不是变化量。
- <1>“</1>平均/散点<2>”</2>历史趋势显示每个趋势时间间隔内数据点的平均值<3>。</3>

您可以创建一些被称作指示器的图形游标，以便根据指示器在趋势中的当前位置来访问趋势数据的详细信息。例如，操作员将指示器放在趋势上有可见数据的区域时，要在该位置上绘制趋势的所有数据库值的时间和值都会显示出来。

您也可以创建一些按钮或游标，以放大或缩小指示器之间的区域；也可以缩放到数据范围，如最大值到最小值。此时可以显示完整图表或指示器之间区域的平均值和标准偏差。

历史趋势也可以按任何时间量进行滚动。您可以创建自定义刻度，并将其链接到 **.MinEU** 和 **.MaxEU** 点域，以创建一个趋势，按工程单位显示整个数据集范围。

理解实时趋势

实时趋势显示当前正在运行的 InTouch 应用程序中的数据。实时趋势持续进行更新。实时趋势图绘制最多与四个本地标记或表达式关联的当前数据值。

您可以：

- 创建实时趋势
- 给趋势选择标记
- 指定趋势的时间跨度与更新间隔

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 配置趋势的显示选项

在历史趋势中显示保存的标记值

通过使用以下任何一种 WindowMaker 实用程序，都可以创建历史趋势：

- 历史趋势工具
- 历史趋势向导
- 16-笔趋势向导（可选）

此外，您可以采用 ActiveFactory 趋势，以显示保存到 Historian 数据库中的 InTouch 历史趋势数据。

使用历史趋势对象

您可以使用 WindowMaker 的“历史趋势”工具来创建和配置趋势。您可以：

- 创建历史趋势
- 给趋势选择标记
- 指定趋势的时间跨度与更新间隔
- 配置趋势的显示选项

创建历史趋势

您可以使用<1>“</1>历史趋势<2>”</2>工具在窗口中创建一个趋势对象<3>。</3> 第一次创建历史趋势对象时，使用 InTouch 缺省配置设置。

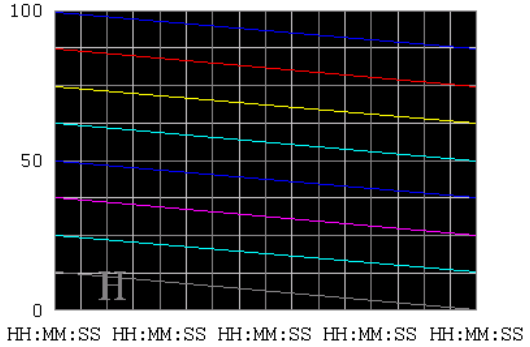
配置历史趋势之后，WindowMaker 使用最新的配置值作为新趋势对象的初始设置。

您可以在窗口边框内绘制任意大小的趋势图。

创建历史趋势

1. 在 WindowMaker 中，打开要在其中放置一个历史趋势的窗口。
2. 在**绘图**菜单上的**趋势**组中，选择**历史**。
3. 将鼠标移到要放置历史趋势的窗口区域上。沿着对角线拖动鼠标，根据所需的趋势大小来创建一个长方形。

   此时“历史趋势”对象出现在窗口中。



4. 如果需要，使用对象手柄来调整趋势的高度与宽度。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置要在历史趋势上显示的标记

历史趋势笔创建指定时间段内所记录数据的图形化表示。您将趋势笔指定给采集历史数据的标记。

历史趋势最多支持八支笔。

注意：WindowViewer 必须关闭。否则无法选择“笔”框。

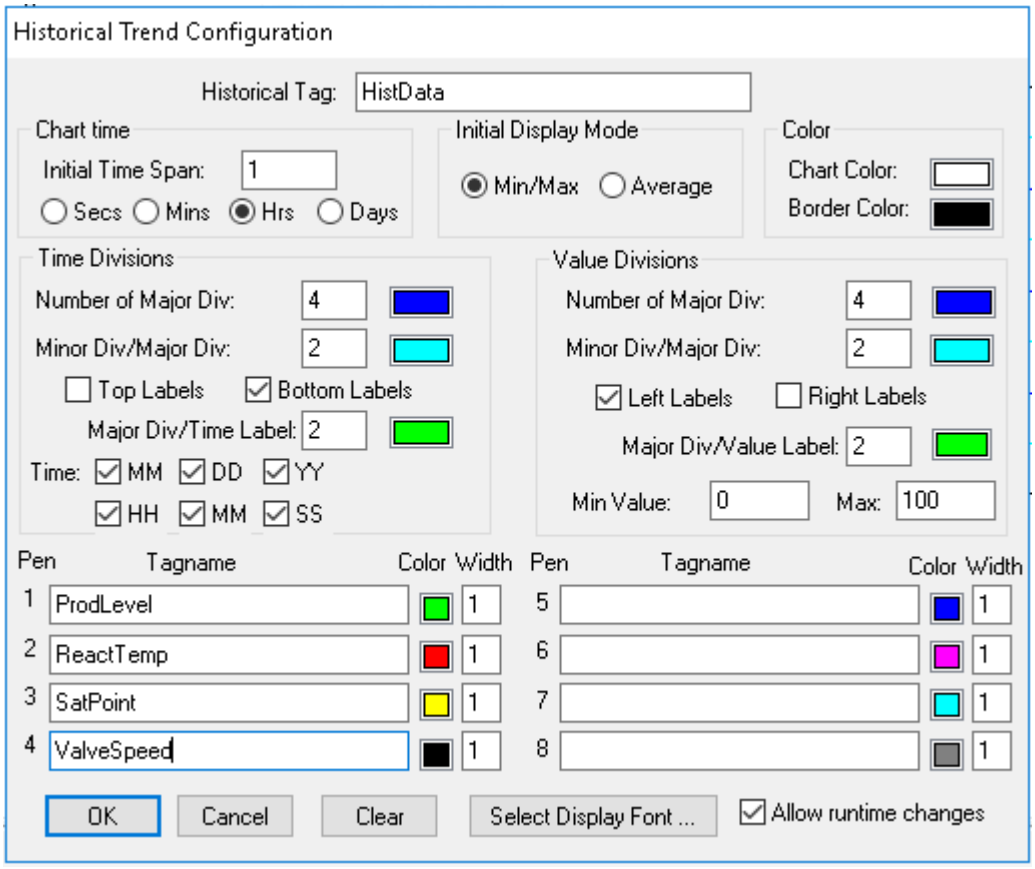
如果配置了远程历史供应器，则可以从这些供应器中选择标记。如需有关设置远程历史供应器的信息，请参阅《*AVEVA™ InTouch HMI 应用程序部署*》中的分发应用程序。

注意：您也可以配置一个 IndustrialSQL Server 历史供应器，以便使用图形化的方式来显示历史趋势数据。为获得更多的功能性以及其它的绘图选项，请使用 ActiveFactory 趋势工具基于 IndustrialSQL Server 数据库中保存的 InTouch 历史数据来创建趋势。

配置希望在历史趋势上显示的标记

1. 双击窗口中的趋势对象。

   此时出现历史趋势配置对话框。



2. 在历史标记框中，输入要给趋势使用的标记。

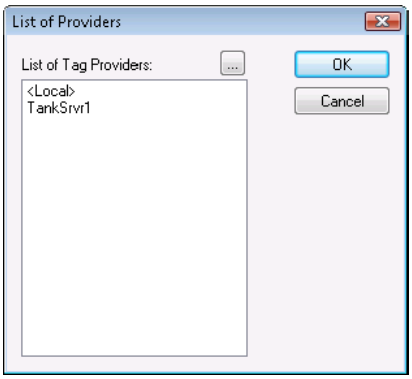
   该标记必须定义为“历史趋势”类型。您必须给 InTouch 应用程序中的每个历史趋势指定一个不同的“历史趋势”标记。

   如果输入的标记当前尚未在“标记名字典”中定义，则出现一个对话框，询问是否要创建标记。如果选择确定以定义标记，则出现标记名字典对话框。此标记不支持 128 个字符。

3. 在标记名区域的一个或多个笔框中，指定现有的本地或远程标记的名称。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 要直接指定现有的本地或远程标记，请选择笔框并输入标记的名称。
5. 要浏览到所要指定的标记：
   a. 在笔框中双击。此时出现供应器列表对话框。



   b. 选择要给笔使用的标记供应器。
   c. 选择确定以显示一个列出所选供应器的标记的对话框。
   d. 从列表中双击某个标记来选择它。
6. 对于指定了标记的每支笔，双击旁边的颜色框以显示调色板。为笔选择一种颜色。
7. 在宽度框中，为趋势中显示的每支笔输入以像素为单位的线条宽度。
8. 对于要指定给历史趋势笔的每个标记，重复步骤 3 到 7。
9. 选中允许运行时更改复选框，以允许操作员在应用程序运行期间配置历史趋势。

   如需有关在运行时更新历史趋势的详细信息，请参阅在运行时更改趋势的配置。

配置历史趋势的时间跨度

您可以配置历史趋势的时间跨度。

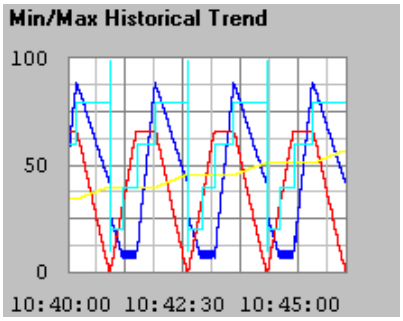
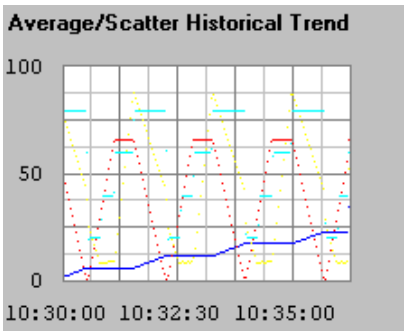
配置历史趋势的时间跨度

1. 双击趋势对象以显示历史趋势配置对话框。
2. 在图表时间区域的初始时间跨度中，输入希望在趋势的 x 横坐标轴上出现的时间长度。
3. 选择时间单位：“秒”、“分”、“小时”或“日”。

   例如，如果在初始时间跨度中输入 8，然后选择小时，则趋势上显示的时间跨度为 8 小时。
4. 在初始显示模式区域中，选择 WindowViewer 最初显示包含趋势的窗口时出现的历史趋势类型。

初始显示模式	描述

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

最小/最大	图表以工程单位的百分比为纵坐标，以时间宽度为横坐标来显示所发生的变化。 
均值	趋势时间段内的每个像素都显示该时间段内标记的平均值。 

5. 转到配置历史趋势显示选项以配置历史趋势显示选项。

配置历史趋势显示选项

您可以配置历史趋势的视觉外观。

配置历史趋势显示选项

1. 双击趋势对象。此时出现历史趋势配置对话框。
2. 设置颜色选项。执行以下操作：
   - 在颜色部分中，选择图表颜色框以打开调色板。
   - 选择调色板中的某种颜色，以便将它用作趋势的背景色。

     白色是缺省的背景色。任何其它背景色都会增加打印趋势所需的时间。
   - 选择边框颜色以打开调色板。
   - 选择调色板中的某种颜色，以便将它用作趋势的边框颜色。
3. 设置时间刻度选项。执行以下操作：
   - 在时间刻度区域的主刻度数中，输入趋势的主时间刻度数。

     此时主时间刻度出现在趋势的时间横坐标轴上。主时间刻度之间的最大时间间隔是 65536 秒，即 18 小时 12 分 16 秒。
   - 选择主刻度线的颜色。
   - 在副/主刻度中，输入每个主时间刻度内的副时间刻度数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

副时间刻度数应该是主刻度的偶数倍。例如，如果主刻度设置为 60 秒，则在副/主刻度中输入值 2 时，会将副时间刻度设置为 30 秒。

- 选择副刻度线的颜色。
- 选择顶部标签或底部标签，以便指定时间标签在趋势上的位置。
- 如果使用时间标签，请在主刻度/时间标签框中输入每个时间标签中的主时间刻度线数。
- 选择时间刻度标签的颜色。
- 选择显示为主时间刻度标签的时间单位。

月 (MM)	时 (HH)
天 (DD)	分 (MM)
年 (YY)	秒 (SS)

4. 在值刻度区域中，配置趋势纵坐标轴的外观。

   值刻度选项的配置方法与时间刻度选项的方法相同。纵坐标轴根据所有标记的工程单位来指定趋势中出现的数据值范围。

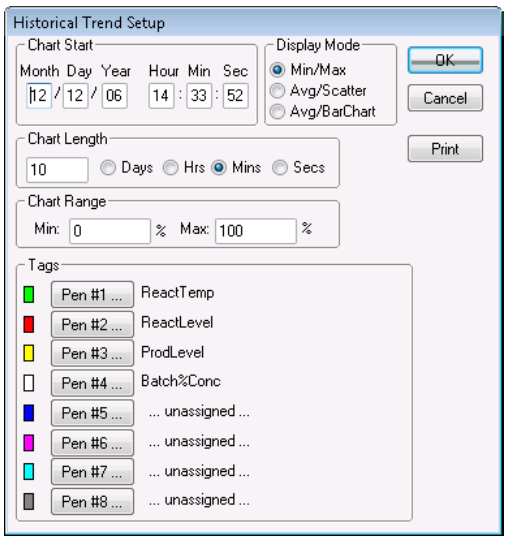
5. 选择确定以保存对配置所作的更改，然后关闭历史趋势配置对话框。

在运行时更改趋势的配置

如果在配置历史趋势时选择允许运行时更改选项，则操作员可以在运行期间配置历史趋势。操作员使用一个对话框来配置趋势，该对话框在从显示的窗口中选择趋势之后出现。

在运行时配置历史趋势

1. 在运行期间选择历史趋势。此时出现历史趋势设置对话框。



2. 在图表开始区域中，输入历史趋势数据采集间隔的开始日期与时间。
3. 在显示模式区域中，选择历史趋势图表的类型。

   趋势显示模式对性能有一定的影响<1>。</1> 决定趋势性能的主要因素是趋势中显示的线条的长度。线条越长，生成趋势所需的时间也越长。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

线条宽度也会对性能产生影响。绘制宽线条花费的时间要明显长许多。创建<1>“</1>最小/最大<2>”</2>或<3>“</3>平均/散点<4>”</4>趋势比创建<5>“</5>均值/棒图<6>”</6>要更为快速<7>。</7>

4. 在**图表长度**区域中，输入要在趋势上显示的时段，然后选择度量单位。

   例如，如果输入 2 并选择小时，则趋势持续时间为 2 小时。

5. 在**图表范围**区域中，输入历史趋势纵坐标轴上显示的工程单位范围的百分比。

   趋势的缩放范围是使用某个百分比范围定义的要绘制趋势的标记的一段工程单位范围。值的范围是从 0 到 100。例如，对于所选的标记，如果要绘制它在 40% 到 60% 工程单位范围内的变化趋势，请分别在最小与最大范围框中输入 40 和 60。

6. 在**标记**区域中，选择笔号以指定标记。

   此时出现**选择标记**对话框，显示可以指定给历史趋势笔的一列标记。

7. 在 QuickScript 或按钮中使用以下语句，以便操作员可以更新图表：

```
Hist_TrendTag.UpdateTrend = 1;
```

8. 在 QuickScript 中或按钮上使用以下任何一个函数：

   HTUpdateToCurrentTime(Hist_Tag);

   HTScrollLeft(Hist_Tag,Percent);

   HTScrollRight(Hist_Tag,Percent);

   HTZoomIn(Hist_Tag,LockString);

   HTZoomOut(Hist_Tag,LockString);

   HTSetPenName(Hist_Tag,PenNum,Tagname);

   如需有关使用包含趋势函数的脚本的详细信息，请参阅使用脚本控制历史趋势向导。

9. 更改以下任何一个趋势标记点域：

   .ChartStart

   .ChartLength

   .MaxRange

   .MinRange

   .Pen1-.Pen8

   如需有关给历史趋势使用点域的详细信息，请参阅使用点域控制历史趋势。

使用点域控制历史趋势

在运行时，可以使用点域来管理历史趋势。

.DisplayMode 点域

.**DisplayMode** 点域指定用于显示标记值的趋势格式。

类别

历史

用法

```
tag_name.DisplayMode
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

tag_name
任何“历史趋势”标记。

数据类型

模拟（可读写）。

有效值

1 = 显示每个采样周期中发生的最小/最大值（缺省）。

2 = 在散点历史趋势中显示每个采样周期的平均值。

3 = 在棒图历史趋势中显示每个采样周期的平均值。

示例

此语句指定历史趋势中的值由 "HistTrend_Tag" 表示，且设置为条形图历史趋势的格式。

```
HistTrend_Tag.DisplayMode=3;
```

另请参阅

.ChartLength, .ChartStart

.MinRange 点域

.MinRange 点域指定给“历史”趋势中的每个标记显示的标记工程单位范围的最小百分比。

类别

历史。

用法

```
tag_name.MinRange
```

参数

tag_name
任何“历史趋势”标记。

附注

历史趋势可以同时显示多个类型的标记。由于不同类型的标记有不同的工程范围，要按工程单位来指定值范围的最小与最大边界非常困难。因此，最小与最大范围值使用每个标记工程范围的百分比来表示。这样，无论标记的真正工程范围如何，历史趋势都显示该标记的指定百分比的特定工程范围。

数据类型

实型（可读写）。

有效值

.MaxRange 与 .MinRange 点域的极限介于 0 到 100 之间。.MinRange 总是小于 .MaxRange。如果将小于 0 或大于 100 的值指定给这两个点域中的任何一个，则该值将调整为 0 或 100。如果 .MinRange 大于或等于 .MaxRange，则趋势不显示任何数据。

示例

此示例点域语句将历史趋势的最小百分比范围设置为 Hist Trend 标记可能的工程单位范围的 25%。

```
HistTrend.MinRange=25
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

.ChartStart, .ChartLength, .DisplayMode, .EngUnits, .MinEU, .MaxEU, .MaxRange, .MinRaw, .MaxRaw, .RawValue

.MaxRange 点域

.**MaxRange** 点域指定给“历史”趋势中的每个标记显示的工程单位范围的最大百分比。

类别

历史。

用法

```
tag_name.MaxRange
```

参数

tag_name
任何“历史趋势”标记。

附注

历史趋势可以同时显示许多种类型的标记。由于标记可以有不同的工程范围，要按工程单位来指定值范围的最小与最大边界非常困难。因此，最小与最大范围值使用每个标记工程范围的百分比来表示。这样，无论标记的真正工程范围如何，历史趋势都显示该标记的指定百分比的工程范围。

数据类型

实型（可读写）。

有效值

.MaxRange 与 .MinRange 点域的极限介于 0 到 100 之间。.MinRange 总是小于 .MaxRange。如果将小于 0 或大于 100 的值指定给这两个点域中的任何一个，则该值将调整为 0 或 100。如果 .MinRange 大于或等于 .MaxRange，则趋势不显示任何数据。

示例

此示例点域语句将历史趋势的最大范围设置为 Hist Trend 标记可能的工程单位范围的 75%。

```
HistTrend.MaxRange=75
```

另请参阅

.ChartStart, .ChartLength, .DisplayMode, .EngUnits, .MinEU, .MaxEU, .MinRange, .MinRaw, .MaxRaw, .RawValue

.UpdateCount 点域

每次历史趋势更新时，.**UpdateCount** 点域将计数值递增一次。.**UpdateCount** 点域可以用作其它函数的触发器。

类别

历史。

用法

```
HistTrendTag.UpdateCount
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

整型（只读）。

有效值

任何正整数。

示例

本例使用 **HTGetValueAtScooter()** 函数来检索右指示器当前位置上 Pen1 的值。对任何函数参数所作的更改都会导致对该函数进行重新求值。完成更新时，**.UpdateCount** 的值会递增，此语句将重新求值。

```
MyRealTag=HTGetValueAtScooter( MyHistTrendTag,MyHistTrendTag.UpdateCount, 2,
MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

另请参阅

.UpdateInProgress, .UpdateTrend

.UpdateInProgress 点域

.UpdateInProgress 点域指出历史趋势更新操作的当前状态。如果正在检索历史，则此点域的值设置为 1；否则此点域设置为 0。

类别

历史。

用法

```
HistTrendTag.UpdateInProgress
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

附注

只要历史趋势要求提供新数据，此点域的值便设置为 1。在过程结束之后，**.UpdateInProgress** 重置为 0。**.UpdateInProgress** 可以用于同历史趋势有关的函数。

如果操作员将趋势滚动到当前显示的时段以外的区域，则检索历史数据需要花费一些时间。**.UpdateInProgress 点域**提供了一种方法来提示操作员目前正在检索要求提供的数据。如果没有这个反馈信息，操作员可能不会了解到目前正在更新趋势。

数据类型

离散（只读）。

有效值

0 = 未在更新

1 = 正在更新

示例

.UpdateInProgress 点域通常用作“历史趋势”滚动按钮上的或旁边的文本对象上的可视化链接中的表达式。通过以下消息值显示动画链接，可以使用 .UpdateInProgress 点域在检索数据时将 "Busy" 显示到窗口上。

```
DText(HistTrend1.UpdateInProgress,"Busy","Ready")
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

.UpdateCount, .UpdateTrend

.UpdateTrend 点域

.UpdateTrend 点域触发对历史趋势的更新。在按钮动作脚本中使用 .**UpdateTrend** 点域时，操作员可以在运行时手动更新趋势。

类别

历史。

用法

```
HistTrendTag.UpdateTrend
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

附注

历史趋势不自动更新。要更新图表并显示指定的标记的当前值，必须更改 .**ChartStart** 或 .**ChartLength** 点域之一。通过在按钮动作脚本中使用此点域，操作员可以在运行时更新图表。如果要更改与历史趋势关联的其它点域，则也可以在 QuickScript 中使用此点域。

您只能将 .UpdateTrend 点域设置为 1 这个值。

数据类型

离散（只写）。

有效值

1

示例

本例触发与 **MyHistTrendTag** 标记关联的历史趋势使用所有参数的当前值进行更新。

```
MyHistTrendTag.UpdateTrend=1;
```

.ChartLength 点域

.**ChartLength** 点域指定“历史”趋势中显示的时间长度。

类别

历史。

用法

```
HistTrendTag.ChartLength
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

附注

指定给 .ChartLength 的值以秒为单位指定图表的长度。长度定义为“历史趋势图表”上当前显示的时间量。更具体一些说，从“历史趋势图表”检索“图表长度”的计算公式是：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
ChartLength=(图表右侧的日期/时间标签) - (图表左侧的日期/时间标签);
```

由于“日期/时间标签”使用自 1970 年 1 月 1 日午夜以来的秒数来表示，因此计算结果为图表左右两侧之间显示的时间秒数之差。

对 .**ChartLength** 进行加减运算时，都使用秒来表示时间。因此，要从当前的 .**ChartLength** 中减去两小时，则必须在执行计算之前先将小时转换为秒。例如：

```
(2 小时) * (60 分钟/小时) * (60 秒/分钟) = 7200 秒。
```

数据类型

整型（可读写）。

有效值

任何正整数。

示例

本例将历史趋势的长度强制设置为一小时。

```
HtTag.ChartLength=3600 {60 分钟 * 60 秒/分钟};
```

本例将趋势向左滚动 50%。

```
HtTag.ChartStart=HtTag.ChartStart - HtTag.ChartLength / 2;
```

本例将图表向左滚动 10%。

```
HtTag.ChartStart=HtTag.ChartStart - (.10 * HtTag.ChartLength);
```

另请参阅

.ChartStart

.ChartStart 点域

.**ChartStart** 点域可以用于设置或验证历史趋势的开始（左侧）日期/时间标签的值。

类别

历史。

用法

```
HistTrendTag.ChartStart
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

附注

这个可读写点域用于设置或验证历史趋势的开始日期/时间标签的值。.**ChartStart** 点域使用自 1970 年 1 月 1 日午夜以来所经过的秒数来表示。起始点定义为历史趋势上的第一个日期/时间标签。

数据类型

整型（可读写）。

有效值

任何正整数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

以下语句使图表向右滚动一分钟。

```
HtTagname.ChartStart=HtTagname.ChartStart + 60;
```

另请参阅

.ChartLength

.Pen1-8 点域

.**Pen1-8** 点域将记录的标记指定给某个历史趋势笔。

类别

历史

用法

```
HistTrendTag{.Pen1 | .Pen2 | .Pen3 | .Pen4 | .Pen5 | .Pen6 | .Pen7 | .Pen8};
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

附注

使用 .Pen1-8 点域采用以下格式将标记指定给趋势笔：

HistTrend.Pen*X* = Tag_Name.TagID

其中 *X* 是 1 到 8 之间的整数。

建议尽量使用 HTSetPenName() 与 HTGetPenName() 函数。

备注：只有本地标记可以指定给 .PenX 点域。无法使用 provider.tag 表示法。provider.tag 仅可以用于 HTSetPenName() 函数。
在了解这些点域的工作原理时，一个很好的参考就是放置到屏幕上并进行了分解的历史趋势向导。

数据类型

TagID（可读写）。

有效值

点域数据类型是 **TagID** 类型。这就意味着，只有标记的句柄可以指定给 **.Pen1-8** 点域。您无法将标记的名称直接指定给 **.Pen1-8** 点域。您必须使用以下语法将与标记关联的 .**TagID** 点域关联到 **Pen1-8** 点域：

```
HistTrendTag.Pen1=LoggedTag.TagID;
```

一般而言，TagID 类型标记仅可赋给另一个 TagID 标记。它无法与任何其它标记类型配合使用，除非其它标记中添加了 .**TagID** 点域扩展名。

尽管 .**Pen1-8** 点域被视为可读写，但无法直接在屏幕上显示它们的值。

示例

下例将新标记指定给与“历史趋势”标记关联的历史趋势的 .**Pen5** 点域。要将所记录的标记指定给 .**Pen5** 点域，必须将 .**TagID** 点域附加到所记录的标记的名称中。

```
HistTrendTag.Pen5=PumpPress.TagID;
```

从上例开始，可以显示指定给 HistTrendTag.Pen5 的标记的名称。对于操作员来说，创建一个图例来显示指定给每个趋势笔的标记，这是非常有用的信息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您无法在“消息显示”链接中显示指定给 HistTrendTag.Pen5 的标记。.Pen5 点域的实际值是一个整数，表示 WindowViewer 中的内存位置，这对于显示目的而言毫无意义。您需要创建一个名称为 **Pen05** 的 TagID 型标记。将以下语句放到上例中的语句之下：

```
Pen05=HistTrendTag.Pen5;
```

在第一个示例中，PumpPress 标记指定给 HistTrendTag 的 5 号笔。在本例中，Pen05 被指定 HistTrendTag 的 Pen5 的值，而 HistTrendTag 则是 PumpPress 标记的 TagID。

.Pen1-8 这些点域是一些指针，它们指向与选择要在趋势中显示的笔关联的标记。.Pen1-8 这些点域是一种特殊的数据类型，即 .TagID。执行指定之后，您可以使用 TagID 标记的 .Name 点域显示标记的名称。

.TagID 点域

.**TagID** 点域可以同 .**Pen1** - .**Pen8** 点域配合使用，以便将标记指定给历史趋势笔。

类别

历史标记。

用法

```
tag_name.TagID
```

参数

tag_name
任何离散、整型、实型、间接离散或间接模拟标记。

附注

.**TagID** 点域提供标记句柄并主要用于将标记指定给历史趋势中的笔。

数据类型

TagID（只读）。

示例

本例使用 .**TagID** 点域将 **PumpRPM** 标记指定给历史趋势的 6 号笔。

```
HistTrendTag.Pen6=PumpRPM.TagID;
```

另请参阅

.Pen1-.Pen8

.ScooterLockLeft 点域

.**ScooterLockLeft** 点域指定操作员是否可以将右指示器进一步移到历史趋势上左指示器当前位置的左侧。

类别

历史。

用法

```
HistTrendTag.ScooterLockLeft
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

一般而言，应该阻止操作员将右指示器进一步移到左指示器当前位置的左侧。左指示器锁定时，每次将右指示器移到它的左侧时，它便迫使右指示器处在与它相同的位置。

数据类型

离散（可读写）。

有效值

0 = False。右指示器可以进一步移到历史趋势上左指示器当前位置的左侧

1 = True。右指示器无法移到历史趋势上左指示器当前位置的左侧。

示例

下例阻止将右指示器移到历史趋势上左指示器当前位置的左侧。

```
HistTrendTag.ScooterLockLeft=1;
```

另请参阅

.ScooterPosRight、.ScooterPosLeft、.ScooterLockRight

.ScooterLockRight 点域

.**ScooterLockRight** 点域指定操作员是否可以将左指示器进一步移到历史趋势上右指示器当前位置的右侧。

类别

历史。

用法

```
HistTrendTag.ScooterLockRight
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

附注

一般而言，应该阻止操作员将左指示器进一步移到右指示器当前位置的右侧。右指示器锁定时，每次左指示器移到它的右侧时，它便迫使左指示器处在与它相同的位置。

数据类型

离散（可读写）。

有效值

0 = False。左指示器可以移到历史趋势上右指示器当前位置的右侧。

1 = True。左指示器无法移到历史趋势上右指示器当前位置的右侧。

示例

下例阻止将左指示器移到历史趋势上右指示器当前位置的右侧。

```
HistTrendTag.ScooterLockRight=1;
```

另请参阅

.ScooterPosRight、.ScooterPosLeft、.ScooterLockLeft

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.ScooterPosLeft 点域

.ScooterPosLeft 点域动态控制历史趋势上左指示器的位置。

类别

历史

用法

```
HistTrendTag.ScooterPosLeft
```

参数

HistTrendTag
指定了趋势的名称的“历史趋势”标记。

附注

这个可读写点域动态控制左指示器的位置。您可以在 QuickScript 函数中使用此点域来检索左指示器的当前位置；或者将一个值指定给此点域，以便将左指示器调整到趋势上的另一个位置。

此点域最常与一组 **HTGetValue()** 函数配合使用。这些函数必须指定要查询哪个历史趋势以及趋势指示器的当前位置。

数据类型

实型（可读写）。

有效值

0.0 到 1.0；其中 0.0 是历史趋势图表的最左侧，1.0 是历史趋势图表的最右侧。

示例

下例重新调整左指示器的位置。左指示器移到距离历史趋势图表（当前与 MyHistTrendTag 标记关联的）左侧 34% 图表总长的位置。

```
MyHistTrendTag.ScooterPosLeft=.34;
```

在以下语句中，QuickScript 函数 **HTGetValueAtScooter()** 检索左指示器当前位置上 1 号笔的值。对函数参数列表中的任何值所作的更改都会导致对函数重新进行求值。每次左指示器的位置发生改变时，此语句都会重新求值。

```
MyRealTag=HTGetValueAtScooter (MyHistTrendTag,MyHistTrendTag.UpdateCount, 1,
MyHistTrendTag.ScooterPosLeft, 1, "PenValue");
```

另请参阅

.ScooterPosRight、.ScooterLockLeft、.ScooterLockRight

.ScooterPosRight 点域

可读写的 **.ScooterPosRight** 点域动态控制右指示器的位置。

类别

历史。

用法

```
HistTrendTag.ScooterPosRight
```

参数

HistTrendTag

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

指定了趋势的名称的“历史趋势”标记。

附注

这个可读写点域动态控制右指示器的位置。您可以在 QuickScript 函数中使用此点域来检索右指示器的当前位置。您也可以将某个值指定给此点域，以便将右指示器移到趋势上的另一个位置。

此点域最常与 **HTGetValue()** 函数配合使用。这些函数必须指定要查询哪个历史趋势以及趋势指示器的当前位置。

数据类型

实型（可读写）

有效值

0.0 到 1.0；其中 0.0 是历史趋势图表的最左侧，1.0 是历史趋势图表的最右侧。

示例

以下语句给右指示器指定一个新的位置。右指示器移到距离历史趋势图表（与 MyHistTrendTag 标记关联的）左侧 34% 图表总长的位置。

```
MyHistTrendTag.ScooterPosRight=.34;
```

以下语句使用 QuickScript 函数 **HTGetValueAtScooter()** 检索右指示器新位置上 1 号笔的值。对函数参数列表中的任何变量所作的更改都会导致对函数重新进行求值。每次右指示器的位置发生改变时，此语句都会重新求值。

```
MyRealTag=HTGetValueAtScooter(MyHistTrendTag, MyHistTrendTag.UpdateCount, 2,
MyHistTrendTag.ScooterPosRight, 1, "PenValue");
```

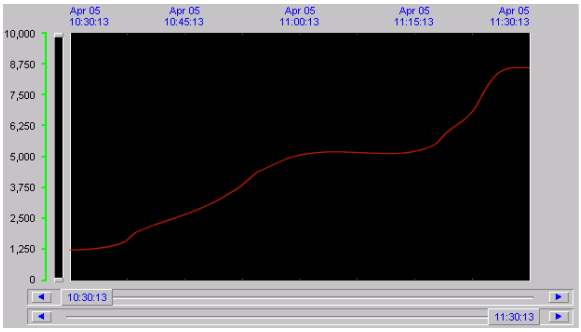
另请参阅

.ScooterPosLeft、ScooterLockLeft、.ScooterLockRight

使用历史趋势向导

<1>“</1>历史趋势向导<2>”</2>自动创建一个历史趋势<3>。</3> 除了手动将标记指定给历史趋势笔，此向导还使用一些标准的值来自动配置历史趋势。

下图显示使用<1>“</1>历史趋势向导<2>”</2>创建的标准趋势<3>。</3> 该趋势图包含被称作指示器的游标，以显示趋势图中特定位置上的数据，或放大到所选的趋势数据范围。



要将缩放与移动功能或一些笔控件添加到历史趋势中<1>，</1>请使用趋势<2>“</2>缩放/平移<3>”</3>面板以及<4>“</4>趋势笔图例<5>”</5>向导<6>。</6>

您可以创建与配置历史趋势。您可以：

- 使用一些向导来创建历史趋势

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 给趋势选择标记
- 配置历史趋势时间跨度
- 使用 QuickScript 控制趋势

使用历史趋势向导创建趋势

您可以使用<1>“</1>历史趋势向导<2>”</2>的自动化功能来创建标准的历史趋势<3>。</3> 其它部分将介绍如何使用向导选项来手动配置历史趋势。

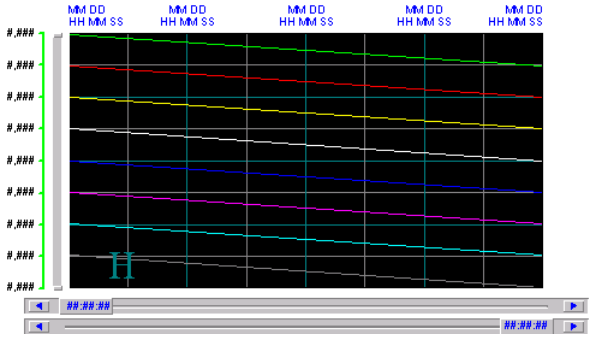
使用一些向导来创建历史趋势

1. 从 WindowMaker 打开一个要放置历史趋势的窗口。
2. 在**绘图**菜单上的**插入组**中，选择**向导**。

   此时出现**向导选择**对话框。
3. 从向导列表中选择**趋势**。

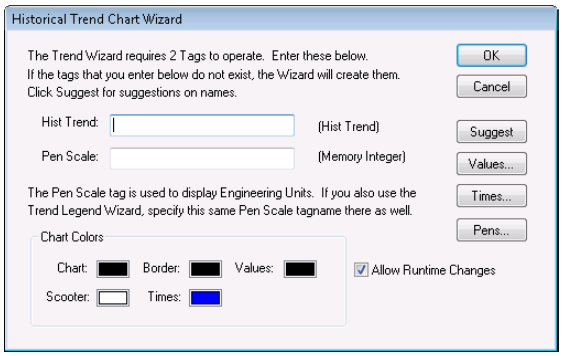
   此时**向导选择**对话框的右侧面板显示一组趋势向导图标。
4. 选择含指示器和刻度的**历史趋势**向导，然后选择**确定**。

   此时**向导选择**对话框关闭，您的窗口再次出现。
5. 将光标移到窗口上希望放置历史趋势左上角的位置。选择将趋势放置到窗口中。



6. 双击该趋势。

   此时出现**历史趋势图表向导**对话框。



7. 选择**建议**。 此时历史趋势图表向导自动将一些缺省配置值指定给该趋势。

   剩余的唯一配置任务就是将标记指定给趋势笔。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置要在趋势图上显示的标记

在“历史趋势图表向导”中将标记指定给趋势笔同“历史”与“实时”工具中的过程类似。

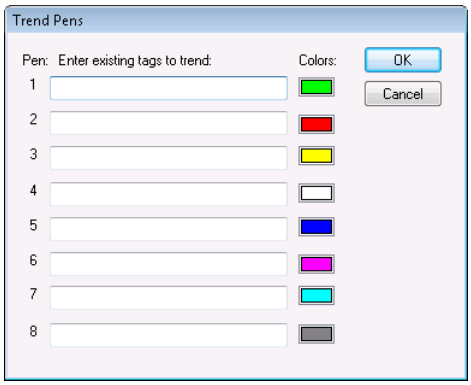
从“历史趋势图表向导”中指定标记

1. 双击历史趋势。

   此时出现历史趋势图表向导对话框。

2. 选择笔。

   此时出现趋势笔对话框。



3. 在笔框中输入现有的本地标记的名称。最多可以输入 49 个字符。

   注意：WindowViewer 必须关闭。否则无法选择“笔”框。

   如果双击笔框，则出现选择标记对话框，其中列出指定了记录数据选项的应用程序标记。通过从选择标记对话框中选择标记，可以将它指定给笔。

4. 如果希望更改笔的缺省颜色，请选择每支笔旁边的颜色框，然后选择另一种颜色。否则跳过此步骤并接受缺省颜色。

5. 选择确定以关闭趋势笔对话框。

6. 选择确定以关闭历史趋势图表向导对话框。

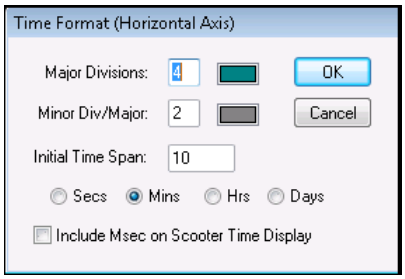
配置历史趋势时间跨度

历史趋势图表向导对话框包含一个选项，可用于手动配置使用“历史趋势向导”创建的趋势中所显示的时间跨度。您可以手动配置趋势的时间，而不是接受“历史趋势向导”的缺省配置。

配置历史趋势的时间跨度

1. 双击历史趋势。此时出现历史趋势图表向导对话框。

2. 选择时间。此时出现时间格式对话框。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 配置时间格式。执行以下操作：
   a. 在主刻度框中，输入趋势的时间横坐标轴上显示的主时间刻度数。
   b. 在副/主刻度框中，输入每个主刻度内的副时间刻度数。
   c. 在初始时间跨度框中，输入在趋势的横坐标轴上显示的时间段的长度。应用程序在 WindowViewer 中运行时，使用“历史趋势向导”创建的趋势可以进行更新。操作员可以更改趋势时间段的长度。但历史趋势总是从时间格式对话框中设置的时间段开始。
   d. 选择趋势时间段的度量单位：秒、分钟、小时、日。
   e. 作为可选项，可以在显示的指示器时间中包含毫秒。下例显示的指示器游标中包含附加到当前时间的毫秒。

4. 选择确定以关闭时间格式对话框。
5. 选择确定以关闭历史趋势图表向导对话框。

配置显示选项

历史趋势图表向导对话框包含一个可配置水平趋势垂直单位的选项。您可以手动配置趋势纵坐标轴上显示的主、副数值刻度。

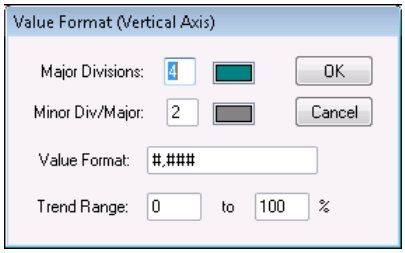
使用“历史趋势图表向导”配置显示选项

1. 双击历史趋势。

   此时出现历史趋势图表向导对话框。

2. 选择值。

   此时出现值格式对话框，其中包含一些可配置趋势数值纵坐标轴的选项。



3. 配置值格式。执行以下操作：
   a. 在主刻度框中，输入趋势的纵坐标轴上显示的主数值刻度数。选择颜色框以访问调色板，然后选择要指定给数值坐标轴主刻度线的颜色。
   b. 在副/主刻度框中，输入每个数值坐标轴主刻度内可见的副刻度数。选择颜色框以访问调色板，然后选择要指定给数值坐标轴副刻度线的颜色。
   c. 在值格式框中，输入趋势的数值纵坐标轴中出现的数值格式。缺省数字格式是 #,###。
   d. 在趋势范围框中，输入趋势中出现的标记工程单位的上、下百分比边界。
4. 选择确定以关闭值格式对话框。
5. 选择确定以关闭历史趋势图表向导对话框。

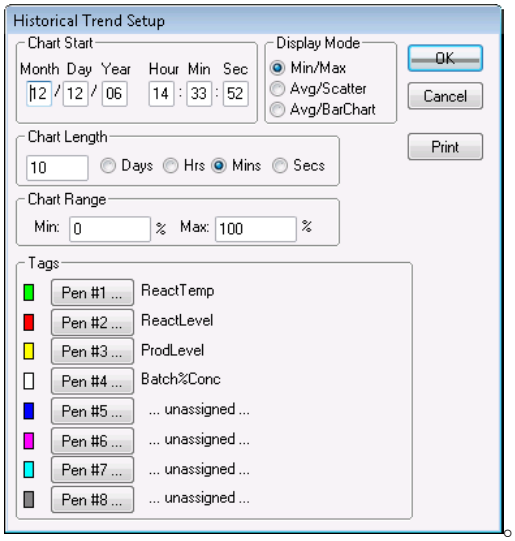
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在运行时更改配置

在配置历史趋势时，如果选择了允许运行时更改选项，则操作员可以在运行时对历史趋势的某些方面进行更改。

在运行时配置历史趋势

1. 在 WindowViewer 中选择趋势。此时出现历史趋势设置对话框



2. 在图表开始区域中，输入图表的开始日期与时间。
3. 在显示模式区域中，选择历史趋势的类型。
4. 在图表长度区域中，输入趋势上显示的时间长度，然后选择该长度的时间增量。
5. 在图表范围区域中，输入作为趋势垂直范围的工程单位范围百分比。
6. 在标记区域中，单击每个笔 # 以便将标记指定给趋势笔。此时出现选择标记对话框，并显示那些启用了记录功能的标记。
7. 双击标记的名称将它指定给趋势笔。
8. 选择确定以保存在运行时对趋势所作的更改。

使用脚本控制历史趋势向导

您可以给趋势对象或动画链接表达式使用 QuickScript 函数，以便在运行时控制历史趋势。例如，您可以使用 QuickScript 将趋势更新到当前时间、将标记指定给趋势笔、将笔连接到图表、重新绘制网格，以及删除或重新绘制指示器。

将趋势更新到当前时间

您可以创建脚本来更新历史趋势以显示最新标记数据。

HTUpdateToCurrentTime() 函数

HT**UpdateToCurrentTime()** 函数检索并显示结束时间等于当前时间的数据。开始时间等于结束时间减去图表的宽度。

类别

历史

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
HTUpdateToCurrentTime(Hist_Tag);
```

参数

Hist_Tag
指定了历史趋势的名称的历史趋势标记

示例

以下语句检索并显示当前时间 **Trend1** 历史标记的数据：

```
HTUpdateToCurrentTime("Trend1");
```

如果当前时间是下午 3:04，趋势的宽度是 60 秒，则新的结束时间是下午 3:04。新的趋势开始时间是下午 3:03。

更改趋势配置

您可以使用以下这些脚本函数来更改指定给历史趋势笔的标记：

- HTSelectTag() 函数
- HTSetPenName() 函数

HTSelectTag() 函数

HTSelectTag() 函数打开选择标记对话框，以便操作员将不同的标记指定给趋势笔。

备注：选择标记对话框仅列出定义了历史记录功能（在“标记名字典”中选择了**记录数据选项**）的那些标记。

类别

历史

语法

```
HTSelectTag();
```

附注

HTSelectTag() 函数仅显示从“标记名字典”中选择了**记录数据选项**的那些标记。不过也可以使用“标记浏览器”的过滤器来显示较少的一组标记。例如，所有以“A”开头的标记。此函数返回所选的标记，并可以用作给笔指定标记的函数参数。

示例

以下 QuickScript 导致在 InTouch 中出现**选择标记**对话框。然后用户可以从列表中选择标记。此标记由名称为 HistTrend 的“历史对象”指定给 1 号笔。

```
HTSetPenName("HistTrend",1,HTSelectTag());
```

另请参阅

HTSetPenName()

HTSetPenName() 函数

HTSetPenName() 函数将一个不同的标记指定给趋势的某支笔。

类别

历史

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
HTSetPenName(Hist_Tag,PenNum,Tagname);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

PenNum
代表趋势的笔号（从 1 到 8）的整型标记或值。

Tagname
指定给笔的新标记的名称。

附注

此 QuickScript 函数是在运行时从分布式历史供应器添加标记的唯一方法。

最多可以为笔名称中的引用输入 49 个字符。

试图取消指定趋势笔时，可能会看到以下错误消息：

```
VIEW /UpdateData:Invalid DBS.TAGNAME handle - 0
```

如果试图取消指定先前按照 *histprovider.tag_name* 的形式指定给某个远程标记的笔，则会发生此错误。要解决此错误，请创建一个选择了记录数据选项的本地标记。然后使用以下脚本来取消指定这支笔：

```
HTSetPenName( "HistTrend", 1, "localtag" );
{assigns the pen to a locally logged tag---localtag}
HistTrend.Pen1=None;
{unassigns the pen}
```

其中，None 是 TagID 类型标记。

示例

以下语句将 OutletPressure 标记指定给 Trend1 的 3 号笔。

```
HTSetPenName("Trend1",3,"OutletPressure");
```

以下语句将 **HistPrv1.Tag1** 标记指定给 Trend1 的 TrendPen4。

```
HTSetPenName("Trend1",TrendPen4,"HistPrv1.Tag1");
```

另请参阅

HTSelectTag()

检索趋势与历史数据的有关信息

您可以创建一些脚本在运行从历史趋势中检索信息。使用以下函数：

- HTGetPenName() 函数
- HTGetTimeAtScooter() 函数
- HTGetTimeStringAtScooter() 函数
- HTGetValue() 函数
- HTGetValueAtScooter() 函数
- HTGetValueAtZone() 函数
- HTScrollLeft() 函数

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- HTScrollRight() 函数
- HTZoomIn() 函数
- HTZoomOut() 函数

HTGetPenName() 函数

HTGetPenName() 函数返回当前指定给历史趋势的指定笔号的标记的名称。

类别

历史

语法

```
MessageResult=HTGetPenName(Hist_Tag,UpdateCount, PenNum);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

UpdateCount
代表趋势的 **.UpdateCount** 点域的整数。此参数值起着数据改变触发器的作用，可以重新对函数求值。

PenNum
代表趋势的笔号（从 1 到 8）的整型标记或值。

示例

以下语句检索指定给 Trend1 趋势的 2 号笔的标记的名称，然后将该名称放入 **TrendPen** 消息标记中：

```
TrendPen=HTGetPenName("Trend1", Trend1.UpdateCount,2);
```

HTGetTimeAtScooter() 函数

HTGetTimeAtScooter() 返回某个样本时间（使用自 GMT 时间 1970 年 1 月 1 日 00:00:00 以来经过的秒数来表示），该样本在 *ScootNum* 与 *ScootLoc* 参数所指定的指示器位置上。

类别

历史

语法

```
IntegerResult=HTGetTimeAtScooter(Hist_Tag, UpdateCount,ScootNum,ScootLoc);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

UpdateCount
代表趋势的 **.UpdateCount** 点域的整数。

ScootNum
代表左指示器或右指示器的整数：
1 = 左指示器
2 = 右指示器

ScootLoc
代表趋势中 .ScooterPosRight 或 .ScooterPosLeft 位置上的值的实数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

指定给 UpdateCount、ScootNum、及 ScootLoc 参数的值所发生的任何变化都会导致对该表达式重新求值。这确保了在检索新的数据或移动指示器之后会对表达式重新进行求值。

示例

以下语句检索 Trend1 趋势的当前左指示器位置上的值的时间（以秒为单位）：

```
HTGetTimeAtScooter("Trend1",Trend1.UpdateCount,1, Trend1.ScooterPosLeft);
```

HTGetTimeStringAtScooter() 函数

HTGetTimeStringAtScooter() 函数为指定的指示器位置上的样本返回一个包含日期/时间的字符串。

类别

历史

语法

```
MessageResult=HTGetTimeStringAtScooter(Hist_Tag, UpdateCount, ScootNum, ScootLoc,
Format_Text);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

UpdateCount
代表趋势的 **.UpdateCount** 点域的整数。

ScootNum
代表左指示器或右指示器的整数：
1 = 左指示器
2 = 右指示器

ScootLoc
代表趋势中 .ScooterPosRight 或 .ScooterPosLeft 位置上的值的实数。

Format_Text
指定要使用的日期/时间格式的字符串。以下是可以接受的 Format_Text 字符串：
"Date"、"Time"、"DateTime"、"DOWShort"（例如 Wed）以及 "DOWLong"（例如 Wednesday）。

附注

指定给 UpdateCount、ScootNum、及 ScootLoc 参数的值所发生的任何变化都会导致对该表达式重新求值。这确保了在检索新的数据或移动指示器之后会对表达式重新进行求值。字符串的格式确定了返回值的内容。

示例

以下语句检索 Trend1 趋势右指示器当前位置上的值的日期与时间。该值按照“Time”格式存储在 NewRightTimeString 消息标记中。

```
NewRightTimeString=HTGetTimeStringAtScooter ("Trend1",Trend1.UpdateCount,2,
Trend1.ScooterPosRight,"Time");
```

HTGetValue() 函数

HTGetValue() 函数为指定的趋势笔返回所请求的类型的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

历史

语法

```
RealResult=HTGetValue(Hist_Tag,UpdateCount, PenNum,ValType_Text);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

UpdateCount
代表趋势的 **.UpdateCount** 点域的整数。

PenNum
代表趋势的笔号（从 1 到 8）的整型标记或值。

ValType_Text
代表要返回的值的类型的字符串：

PenAverageValue = 整个趋势的平均值。

PenMaxValue = 整个趋势的笔的最大值。

PenMinValue = 整个趋势的笔的最小值。

PenMaxEU = 整个趋势的最大工程单位值。

PenMinEU = 整个趋势的最小工程单位值。

PenStdDev = 整个趋势的标准偏差。

附注

函数按实数形式返回所请求的值。

示例

以下语句获取从 PumpPress 趋势中检索的 2 号笔数据的标准偏差。该值存储在 LeftHemisphereSD 内存实型标记中：

```
LeftHemisphereSD=HTGetValue("PumpPress", PumpPress.UpdateCount,2,"PenStdDev");
```

HTGetValueAtScooter() 函数

HTGetValueAtScooter() 函数返回指定的指示器位置、趋势以及笔号处的所请求的类型的样本值。UpdateCount 参数在函数处理完毕之后导致对表达式进行求值。

类别

历史

语法

```
RealResult=HTGetValueAtScooter(Hist_Tag, UpdateCount,ScootNum,ScootLoc,PenNum,
ValType_Text);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

UpdateCount

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

代表趋势的 **.UpdateCount** 点域的整数。

ScootNum
代表左指示器或右指示器的整数：
1 = 左指示器
2 = 右指示器

ScootLoc
代表趋势的 .ScooterPosRight 或 .ScooterPosLeft 点域的实数。

PenNum
代表笔号（从 1 到 8 ）的整型标记或值。

ValType_Text
代表要返回的值的类型的字符串：
PenValue = 指示器位置处的值。
PenValid = 如果该值无效，则返回 0；否则返回 1。

在 HTGetValueAtScooter() 函数中使用 ValType_Text 参数时，使用列出的有效类型之一。

示例

如果值是一个实际的样本，则以下函数返回 1；如果值对于右指示器当前位置上 Trend1 趋势的 3 号笔而言无效，则以下函数返回 0：

```
HTGetValueAtScooter("Trend1",Trend1.UpdateCount, 2,Trend1.ScooterPosRight,3, "PenValid");
```

HTGetValueAtZone() 函数

HTGetValueAtZone() 函数为指定的趋势笔返回左、右指示器位置之间某个数据的所请求的类型的值。

类别

历史

语法

```
RealResult=HTGetValueAtZone(Hist_Tag,UpdateCount,
Scoot1Loc,Scoot2Loc,PenNum,ValType_Text);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

UpdateCount
代表趋势的 **.UpdateCount** 点域的整数。它只用作对函数进行求值的触发器。

Scoot1Loc
代表趋势的 .ScooterPosLeft 点域的实数。它只用作对函数进行求值的触发器。

Scoot2Loc
代表趋势的 .ScooterPosRight 点域的实数。它只用作对函数进行求值的触发器。

PenNum
代表趋势的笔号（从 1 到 8）的整型标记或值。

ValType_Text
代表要返回的值的类型的字符串。
PenAverageValue = 指示器之间的区域的平均值。
PenMaxValue = 指示器之间的区域的最大值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

PenMinValue = 指示器之间的区域的最小值。
PenMaxEU = 指示器之间的区域的最大工程单位值。
PenMinValue = 指示器之间的区域的最小工程单位值。
PenStdDev = 指示器之间的区域的标准偏差。

附注

返回代表给定类型计算值的实数。给 Scoot1Loc 和 Scoot2Loc 指定常数值没有任何效果，它们仅用于对函数进行求值的触发器。此函数直接使用趋势标记的 .ScooterPosLeft 和 .ScooterPosRight 点域值，而不论给 Scoot1Loc 与 Scoot2Loc 参数指定的值如何。

示例

下面的语句计算 Trend1 趋势 1 号笔的左、右指示器之间的数据的平均值。值存储在 AvgValue 内存实型标记中：

```
AvgValue=HTGetValueAtZone("Trend1", Trend1.UpdateCount,Trend1.ScooterPosLeft,
Trend1.ScooterPosRight,1,"PenAverageValue");
```

平移与缩放趋势

您可以创建一些 QuickScript，在其中包含在运行时从历史趋势选择特定数据的函数。

HTScrollLeft() 函数

HTScrollLeft() 函数将趋势的起始时间值设置为比当前起始时间早趋势总时间跨度的一定百分比。其效果是按照指定的趋势总时间跨度百分比向左滚动图表，以回到较早前的时间。

类别

历史

语法

```
HTScrollLeft(Hist_Tag,Percent);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

百分比
代表要向左滚动的图表时间跨度百分比的实数（0.0 到 100.0）。

示例

以下语句将日期/时间向左滚动 PumpPress 趋势总宽度的 10%：

```
HTScrollLeft("PumpPress",10.0);
```

如果当前显示从中午 12:00:00 点开始、显示宽度为 60 秒，则新趋势将从上午 11:59:54 开始（在处理函数之后）。

HTScrollRight() 函数

HTScrollRight() 函数将趋势的起始时间值设置为比当前起始时间晚趋势宽度的一定百分比。其效果是按照指定的趋势宽度百分比向右滚动图表的日期/时间。

类别

历史

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
HTScrollRight(Hist_Tag,Percent);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

Percent
代表要向右滚动的图表百分比的实数（0.0 到 100.0）。

示例

以下语句将 PumpPress 趋势向右滚动 20%：

```
HTScrollRight("PumpPress",20.0);
```

如果当前显示从中午 12:00:00 点开始、显示宽度为 60 秒，则新趋势将从 12:00:12 开始（在处理函数之后）。

HTZoomIn() 函数

HTZoomIn() 函数计算新的图表宽度和开始时间。如果趋势的指示器在趋势的左侧和右侧，则新图表的宽度将等于旧图表的宽度除以二。新的开始时间基于 *LockString* 参数的值进行计算。

如果指示器不在趋势的左侧和右侧，HTZoomIn() 函数将按指示器定义的区域来缩放趋势，而忽略 *LockString* 参数。

类别

历史

语法

```
HTZoomIn(Hist_Tag,LockString);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

LockString
代表缩放类型的字符串：

StartTime	将开始时间设置为与缩放前相等
Center	将中心时间设置为与缩放前相等
EndTime	将结束时间设置为与缩放前相等

附注

如果指示器位置不在趋势的左侧和右侧，则新图表的宽度将是 .ScooterPosLeft 和 .ScooterPosRight 位置之间的时间。在这种情况下不会使用 LockString 的值。最小图表宽度是一秒。在缩放之后，指示器位置设置为 .ScooterPosLeft=0.0、.ScooterPosRight=1.0。

示例

以下语句将显示画面缩小为原来的二分之一，并将 Trend1 趋势的开始时间保持不变。Trend1.ScooterPosRight 等于 1.0，Trend1.ScooterPosLeft 等于 0.0。如果在缩放前开始时间是下午 1:25:00，并且图表宽度是 30 秒，则新的开始时间仍然保持在 1:25:00。新的图表宽度是 15 秒。

```
HTZoomIn("Trend1","StartTime");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

HTZoomOut() 函数

HTZoomOut() 函数计算新的图表宽度和开始时间。新图表的宽度等于旧图表宽度乘以二。新的开始时间基于 *LockString* 参数的值进行计算。

类别

历史

语法

```
HTZoomOut(Hist_Tag,LockString);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

LockString
代表缩放类型的字符串：
StartTime = 将开始时间设置为与缩放前相等
Center = 将中心时间设置为与缩放前相等
EndTime = 将结束时间设置为与缩放前相等

附注

当前的指示器位置对 HTZoomOut() 没有影响。在函数缩放完成之后，新的指示器位置设置为 .ScooterPosLeft=0.0、.ScooterPosRight=1.0。

示例

以下语句将显示画面放大为原来的两倍，并将 Volume 趋势的中心时间保持不变。如果在缩放前开始时间是下午 2:15:00，且图表宽度是 30 秒，则缩放后现在的开始时间是 2:14:45。图表宽度变为 60 秒，趋势的中心仍是 2:15:15。

```
HTZoomOut("Volume","Center");
```

打印趋势

您可以在脚本中使用 PrintHT() 函数打印 WindowViewer 屏幕上当前可见的历史趋势。

PrintHT() 函数

PrintHT() 函数打印屏幕上当前可见的历史趋势。通常，**PrintHT()** 函数与历史趋势窗口上包含的某个屏幕按钮关联。操作员选择该按钮可以打印包含当前值的可见历史趋势。

类别。

历史

语法

```
PrintHT(Trend_Tag);
```

参数

Trend_Tag
历史趋势标记。

示例

本例打印屏幕上当前可见的 PumpPress 历史趋势。

```
PrintHT(PumpPress);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

排解趋势的错误

您可以创建 QuickScript 来验证成功检索的数据是否出现在历史趋势中。使用 HTGetLastError() 函数来排解趋势的错误。

HTGetLastError() 函数

HTGetLastError() 函数可以在脚本中用来确定上一次为指定的历史趋势笔进行数据检索时是否发生了错误。

类别

历史

语法

```
[Result=]HTGetLastError(Hist_Tag,UpdateCount, PenNum);
```

参数

Hist_Tag
指定了趋势的名称的“历史趋势”标记。

UpdateCount
代表趋势的 **.UpdateCount** 点域的整数。

PenNum
代表趋势的笔号（从 1 到 8）的整型标记或值。

结果
指定给某个标记的整数，代表对指定的笔进行的上一个脚本函数调用的状态。

0 = 没有错误

1 = 一般服务器错误

2 = 旧的请求

3 = 文件错误

4 = 未加载服务器

5 = 函数内传递的趋势/笔不存在。

6 = 数据库中不存在此趋势标记

7 = 传递给函数的笔号无效（不在 1 到 8 之间）。

8 = 未给该笔号指定标记，或是未指定启用了记录功能的标记。

示例

以下语句检索上次检索 Trend1 趋势 3 号笔数据的状态，并将结果指定给 ResultCode 整型标记。

```
[ResultCode=]HTGetLastError("Trend1", Trend1.UpdateCount,3);
```

在动画“模拟值显示”QuickScript 中，会使用以下语句：

```
HTGetLastError("Trend1",Trend1.UpdateCount,3);
```

在趋势中显示实时值

您可以通过两种方法来创建实时趋势。“实时趋势”对象提供一套标准控件来选择数据、设置时间范围，以及指定图形的实际外观。InTouch 还包含 16 笔趋势向导，这是可用于创建实时和历史趋势的可选控件。如需有关使用 16 笔趋势向导创建实时趋势的详细信息，请参阅《*AVEVA™ InTouch HMI 管理*》中的创建 16 笔趋势。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用实时趋势对象

您可以创建实时趋势在应用程序中显示当前值。

创建实时趋势

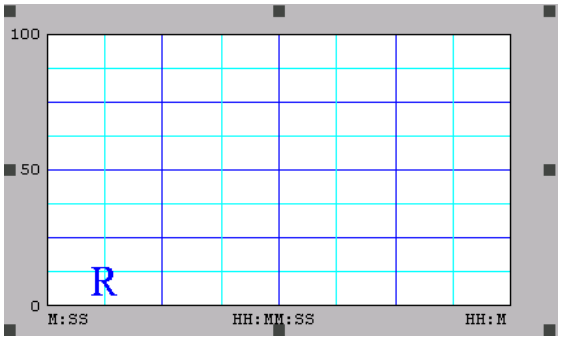
您可以使用“实时趋势”工具在窗口中创建一个趋势对象。第一次粘贴实时趋势对象时，WindowMaker 使用缺省设置。在配置实时趋势之后，WindowMaker 将上次配置的值用作任何新创建的实时趋势对象的初始设置。

您可以在窗口边框内绘制任意大小的趋势图。

创建实时趋势

1. 在**绘图**菜单上的**趋势**组中，选择**实时**。
2. 将鼠标移到要放置实时趋势的窗口区域上。沿着对角线拖动鼠标，根据所需的趋势大小来创建一个长方形。

   此时窗口中出现“实时趋势”对象。



3. 如果需要，使用对象手柄来调整趋势的高度与宽度。

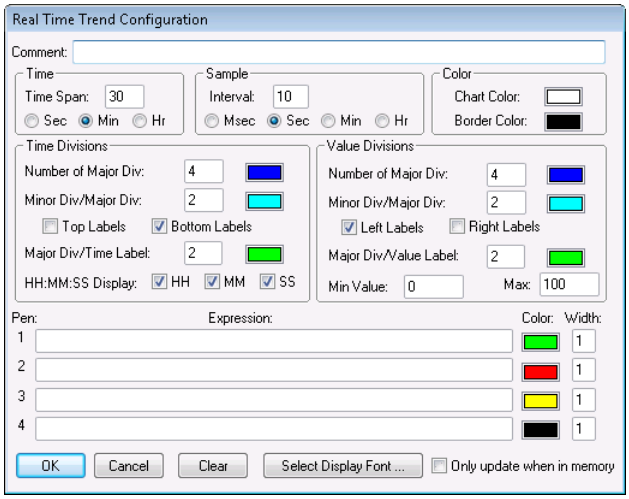
配置要在实时趋势上显示的标记

实时趋势笔创建当前数据的图形化表示，这些数据可来自任何本地标记或包含一个或多个本地标记的表达式。您可以配置这些在实时趋势中显示标记数据的笔。

配置实时趋势标记

1. 双击窗口中的趋势对象。此时出现**实时趋势配置**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**表达式**区域中，**输**入本地**标记**的名称或包含一个或多个本地**标记**的表达式。

   如果在**笔**框中双**击**，**则**出**现选择标记**对话框，**显**示**为应**用程序定**义**的一列**标记**。通**过**从**选择标记**对话框中**选择标记**，可以将它指定**给**笔。

3. **选择**指定了**标记**的**每**支笔旁**边**的**颜**色框以**显**示**调**色板。
4. **选择**要指定**给**笔的**颜**色。
5. 在**宽度**框中，**为趋势**中**显**示的**每**支笔**输**入以像素**为单**位的**线**条**宽**度。

   选择大于 1 的**线**条**宽**度会增加更新或打印**趋势**所需的**时间**。

6. 如果要使**趋势仅**当它**显**示在活**动**窗口中**时**才**进**行更新，**请选择只在在内存中时才更新复选**框。

   如果不**选择**此**选项**，**则**即便窗口**关闭**，**趋势**也在不断更新。持**续**更新**趋势时**会**导**致系**统**性能降低。

7. 保持**实时趋势**配**置**对**话**框打**开**，**转**到配置实时趋势的时间跨度与更新速率中介**绍**的下一个操作程序。

配置实时趋势的时间跨度与更新速率

您可以配置**实时趋势**的**时间**跨度与更新速率。

设置实时趋势的时间跨度与更新速率

1. 双**击趋势**对象。此**时**出**现实时趋势**配**置**对**话**框。
2. 在 **时间**区域的**时间跨度**框中，**输**入要在**趋势**的 x 横坐**标轴**上**显**示的**时间长**度。
3. **选择趋势时间**的度量**单**位。
   - 秒
   - 分
   - 小**时**

   例如，如果在**时间跨度**框中**输**入 30，然后**选择分**，**则图**表上**显**示的**时间**跨度**为** 30 分**钟**。

4. 在**样本**区域的**间隔**框中，**输**入**对趋势**表达式**进**行求**值**并更新**图**表的**间**隔**时间**量。
5. **选择间**隔的度量**单**位。
   - 毫秒
   - 秒

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 分
- 小时

例如，如果在间隔框中输入 10，然后选择秒，则实时趋势每 10 秒更新一次。

6. 保持实时趋势配置对话框打开，转到配置实时趋势的显示选项中介绍的下一个操作程序。

配置实时趋势的显示选项

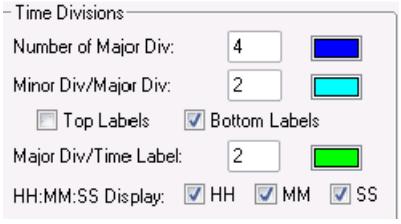
您可以配置实时趋势的视觉外观。

配置实时趋势的显示选项

1. 双击趋势对象。此时出现实时趋势配置对话框。
2. 在颜色区域中配置颜色。执行以下任意操作：
   - 选择图表颜色框可打开调色板。为趋势选择背景颜色。

     白色是缺省的背景色。任何其它背景颜色都会显著增加打印趋势需要的时间。
   - 选择边框颜色框可打开调色板。为趋势选择边框颜色。
3. 在时间刻度区域中，配置时间刻度。执行以下操作：
   - 在主刻度数框中，输入趋势主时间刻度数。主时间刻度出现在趋势的时间水平坐标轴上。

     主时间刻度数必须是主刻度/时间标签 值的偶数倍。例如，刻度数 20 是主刻度/时间标签值 4 的偶数倍。

   - 选择主刻度线的颜色。
   - 在副/主刻度框中，输入每个主时间刻度内可见的副时间刻度数。

     副时间刻度数应该是主刻度时段的偶数倍。例如，如果主刻度时段设置为 60 秒，在副/主刻度中输入值 2，则创建两个为期 30 秒的副时间刻度。
   - 选择副刻度线的颜色。
   - 选择顶部标签或底部标签复选框，以便指定时间标签在趋势上的位置。

     您可以同时选择这两个选项，在趋势的顶部与底部都放上时间标签。将两个选项留为空白会从趋势的横坐标轴中删除时间标签。
   - 如果使用时间标签，请在主刻度/时间标签中输入每个时间标签中的主时间刻度线数。主刻度数必须是主刻度/时间标签值的偶数倍。

     选择时间刻度标签的颜色。
   - 选择作为时间主刻度标签一部分显示的时间单位。

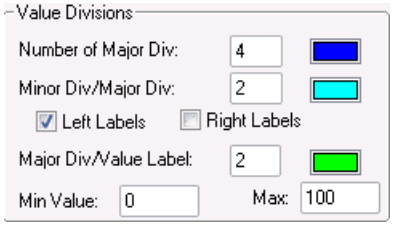
     时 (HH)

     分 (MM)

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

秒 (SS)

4. 在**值刻度**区域中，配置**趋势纵**坐**标轴**的外观。



值刻度选项的配置方法与**时间刻度选项**的**类**似。y 坐**标轴**上的主、副刻度**显**示数据**值**（而非**时间**）的幅度。**纵**坐**标轴**根据所有**标记**的工程**单**位来指定**趋势**中出**现**的数据**值**范**围**。

要**给**数**值**主、副刻度**显**示小数点，**请**在**最小值**与**最大值选项**中**输**入**实**数。例如，0.00 到 100.00。

5. **单击选择显示字体**。此**时**出**现 字体对话**框，它包含一些**选项**，可以**设**置**趋势**中出**现**的文本的字体、字形及大小。
6. **选择确定**。

在运行时打印趋势

有多个因素决定着**趋势**打印的速度。最主要的因素是打印**页**上**趋势**的大小。**趋势**的**显**示模式也会影响打印性能。<1>“</1>最小/最大<2>”</2>或<3>“</3>平均/散点<4>”</4>**趋势**的打印速度可以比<5>“</5>平均/棒**图**<6>”</6>**趋势**的快**许**多<7>。</7> 而且，**趋势**上的**线**条越**长**、越**宽**，所需的打印**时间**也越**长**。

影响打印性能的另一个因素是**趋势**的背景**颜**色。在大多数情况下，白色背景在打印**时**要比彩色背景快**许**多。

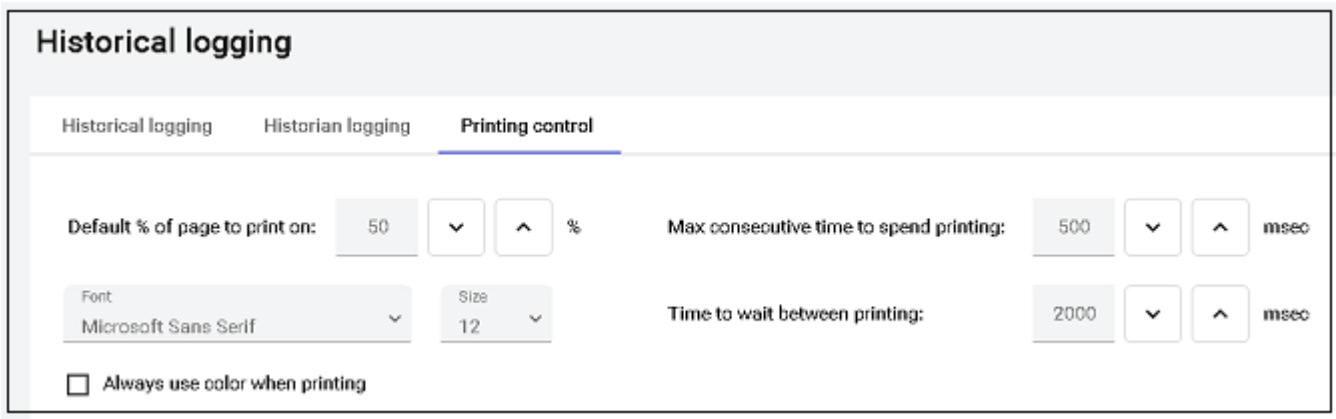
配置趋势打印选项

您可以配置确定如何打印**趋势**的一些**选项**。

配置历史趋势打印

1. 打**开** WindowMaker。
2. 在**文件**菜**单**上，**选择配置**，然后**选择历史记录**。

   此**时**出**现历史记录**配置屏幕。
3. **选择打印控制选项**卡。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 在缺省打印页百分比框中指定用于打印趋势的页面百分比。

   如果输入 50，趋势将打印到垂直和水平方向的一半页面上。按 50% 打印的趋势所花费的时间要比整页打印趋势少许多。

   作为另一个打印选项，您可以使用 **PrintWindow()** QuickScript 函数。

5. 在最长连续打印时间框中，输入以毫秒为单位的处理时间片断。

   时间片断代表分配给计算机处理器以便在前台运行打印模块进程并打印趋势的时段。时间片断越长，趋势的打印速度越快，但这是以牺牲计算机上运行的其它进程为代价的。

6. 在打印等候时间框中，以毫秒为单位输入打印模块在处理器时间片断之间等待的时间。

   处理器时间片断之间的等待时段越短，趋势的打印速度越快。

7. 从字体下拉列表中选择出现在趋势中的打印机字体特征。
8. 对于彩色打印，选中打印时总是使用彩色复选框。
9. 选择保存。

显示来自其它 InTouch 节点或 Historian 服务器的历史标记值

如果要使用远程存储的数据来创建历史趋势，则必须在 InTouch 历史供应器列表中注册远程供应器。此列表指定每个历史供应器的名称与网络位置。只要历史趋势笔指向远程历史供应器上的标记，便会引用这些名称。

您可以定义一个远程历史供应器，将历史趋势笔指定给远程位置上存储的标记数据。您可以：

- 配置远程历史供应器。
- 对笔进行配置以显示远程历史供应器的数据。
- 通过使用<1>“</1>标记浏览器<2>”，</2>将笔指定给远程历史供应器上存储的标记数据<3>。</3>
- 使用 QuickScript 将笔指定给远程标记。

如需有关使用远程历史供应器的数据的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序部署*》中的分发应用程序。

将 InTouch HMI 与 Historian 服务器一起使用

Historian 服务器是一种实时关系型数据库，专门针对工业应用而设计。您可以选择将 Historical Logger 配置为将 InTouch 历史数据存储到 AVEVA 历史服务器数据库中的 Historian。

注意：如需有关为 InTouch HMI 设置远程历史供应器的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序部署*》中的分发应用程序。

如果使用 Historian 存储历史数据，则必须使用 WindowMaker 中的“分布式名称管理器”来指定到数据库的连接。

配置与 Historian 服务器数据库的连接

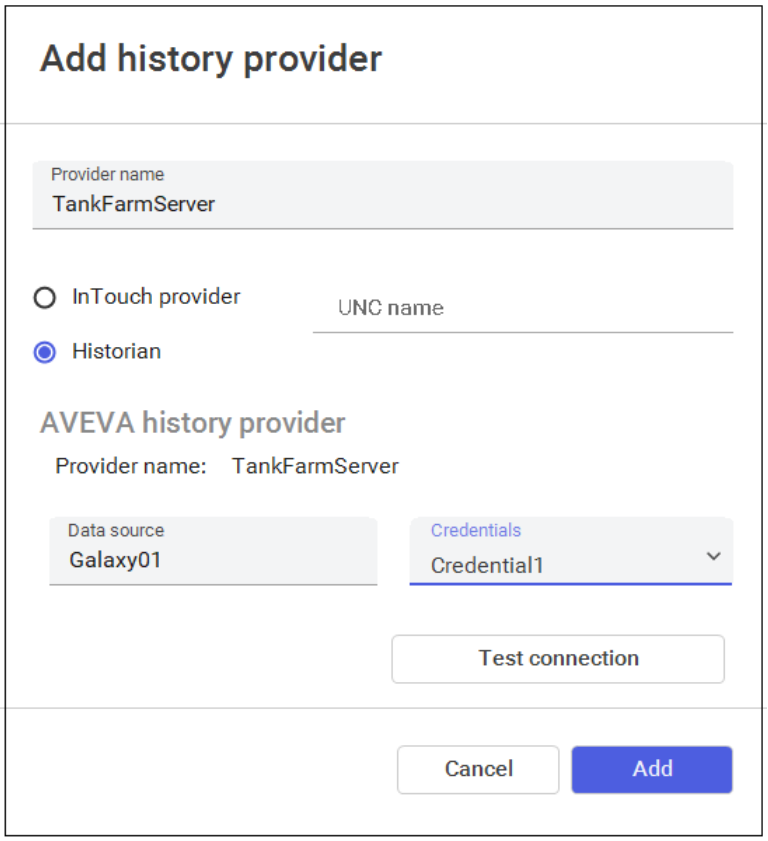
1. 打开 WindowMaker。
2. 在“文件”菜单上，突出显示配置，然后选择分布式名称管理器。

   此时出现“分布式名称管理器”配置屏幕，其中包含用于分布式报警和分布式历史的单独选项卡。

3. 在分布式历史选项卡上，选择 + 图标以添加历史供应器，或按 Alt+A。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现添加历史供应器对话框。



4. 在供应器名框中，为新的历史供应器输入希望使用的名称。

   供应器名的长度不得超过 16 个字母数字字符。

5. 选择 **Historian**。
   a. 在数据源框中，输入安装了 Historian 服务器的服务器的节点名。
   b. 从凭据下拉列表中选择用于身份验证的凭据。

   注意：对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发》中的使用凭据管理器。

   a. 选择**测试连接**，以验证与 Historian 服务器数据库的连接。此时出现一条消息，指出与数据库的连接是否成功。

6. 选择添加以关闭对话框。Historian 服务器节点会出现在历史供应器列表中。
7. 选择保存。

对笔进行配置以显示远程趋势数据

历史趋势同时可以显示本地和远程历史供应器的标记数据。您可以指定趋势笔来显示远程历史供应器的数据。

显示远程历史供应器中的标记

1. 双击历史趋势以显示**历史趋势配置**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 在每支笔的**标记名**框中，输入对远程历史供应器的引用。远程历史供应器的引用格式是：

   history_provider_name.tag_name

   示例：
   ```
   TankFarm1.Pump1RPM
   ```

   历史趋势的每支笔都可以引用不同的远程历史供应器。

3. 选择**确定**保存配置更改。

注意：.TagID 点域无法用在远程历史供应器的标记引用中。

使用标记浏览器将笔指定给远程历史供应器

以下操作程序介绍如何使用<1>“</1>标记浏览器<2>”</2>将趋势笔指定给来自远程历史供应器的标记数据<3>。</3> 通过使用“标记浏览器”来选择标记，消除了手动输入每个标记名的必要，从而减少了出错的可能性。

在<1>“</1>访问名<2>”</2>中指定的远程节点名不必是标记所在节点的实际名称<3>。</3> 但必须创建<1>“</1>访问名<2>”</2>以便将远程历史供应器定义为标记源<3>。</3> 如需有关创建“访问名”的详细信息，请参阅设置访问名

将远程历史供应器定义为标记源

1. 创建一个指定历史供应器所在节点名的<1>“</1>访问名<2>”。</2>
2. 双击历史趋势以打开**历史趋势配置**对话框。
3. 双击某支笔的**标记名**输入框，以显示**选择标记** 对话框。
4. 选择**定义标记源**以便将远程历史供应器定义为标记源。
5. 选择**标记源**箭头，从列表中选择新的远程历史供应器标记源；或选择目**录树视图**按钮，并在目录树视图窗格中选择标记源。此时**选择标记** 对话框刷新，显示来自所选的远程历史供应器的标记。
6. 选择要指定给历史笔的标记并选择**确定**。此时再次出现**历史趋势配置**对话框，所选的标记按照此格式列在该笔的**标记名**框中：*AccessName*:*Item*。
7. 使用您在**分布式名称管理器**中定义的历史供应器名替代 AccessName: 部分。

   例如，HistPrv1.*tag_name*

   这个过程看似麻烦，但在“标记浏览器”中将历史供应器定义为标记源之后，每次双击另一个标记名输入框时，只要在**标记浏览器**中双击标记名，然后将 AccessName: 部分替换成历史供应器名即可。

在 WindowViewer 中，如果允许在运行时更改历史趋势，则用户选择笔按钮更改标记名时，会出现“标记浏览器”，但只能访问应用程序的本地标记。

使用 QuickScript 将笔指定给远程历史供应器

在 InTouch 应用程序正在运行期间，您可以配置趋势笔以便显示来自远程历史供应器的标记数据。创建一个 QuickScript，在 HTSetPenName() 函数中指定远程历史供应器标记引用。例如：

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

在本例中，数字 1 指定历史趋势中的笔号，该笔绘制来自 HistPrv1 远程历史供应器的 **Boiler1** 远程标记的值。

对于远程历史供应器，不支持运行时的**历史趋势设置**对话框与笔点域。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用户定义数据类型

用户定义数据类型 (UDT) 是一种分层数据结构，由各个用户定义的成员所组成。每个成员都可以有自己的数据类型，如整数、布尔值、字符串、其它常见数据类型或其它 UDT。

UDT 可以通过提供创建多层嵌套结构的选项来减少工程工作量。例如，您可以在 InTouch 中创建反应炉 UDT，而不是创建代表该反应炉的一组单个标记。

关于 UDT

InTouch HMI 提供对 UDT 的支持。您可以在 WindowMaker 的左侧看到**用户定义类型**窗格。它包括以下内容：

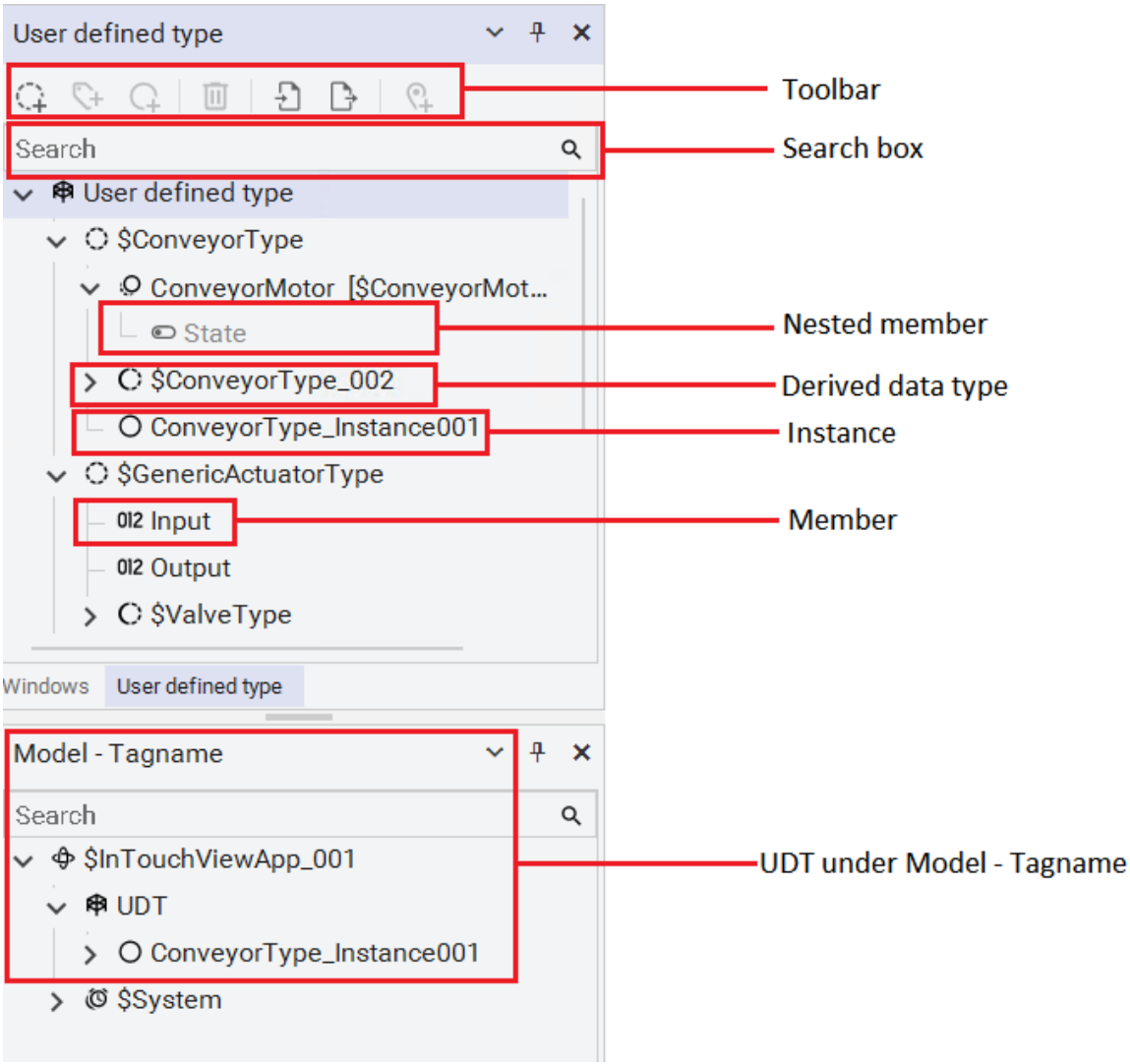
- **搜索框** - 搜索所需的数据类型、衍生数据类型、实例或成员。
- **工具栏** - 工具栏包含以下选项：
  - 添加新数据类型
  - 添加新成员
  - 添加新实例
  - 删除
  - 导入数据类型
  - 导出数据类型
  - 添加位置成员

用户定义类型层次结构可以按以下顺序包含这些项目：

1. **直接成员** - 在当前用户定义类型中定义的成员。
2. **继承成员** - 从其它数据类型继承而来的成员。这些显示为灰色。
3. **衍生数据类型** - 从基本数据类型衍生的数据类型（基本数据类型是只读的，并且是在 InTouch 中预定义的）。它包括衍生它的基本数据类型下的所有成员。除了现有项目之外，还可以向衍生数据类型添加其它成员。缺省条件下，基本数据类型和衍生数据类型使用 $ 符号作为前缀。如果重命名衍生类型，它仍然必须以 $ 作为其名称的第一个字符。
4. **实例** - 实例表示用户定义类型的物理或虚拟对象。您必须创建实例才能在图形动画或脚本中使用用户定义类型。实例标记成员是 InTouch 基本标记。

注意：您可以在属性中编辑 UDT。您无法在标记名字典中查看或编辑 UDT。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



UDT 规格

用户定义数据类型 (UDT) 必须符合以下规格：

名称长度

- 数据类型、衍生数据类型和实例名称的最大名称长度为 32 个字符。
- 成员数据类型和成员的最大名称长度为 32 个字符。

有效名称

- 实例名称可以包含字母数字字符以及特殊字符 $（美元符号）、#（英镑符号）和 _（下划线）。
- $ 字符不能用作第一个字符。

最大嵌套级别

- 最多允许六个嵌套 UDT 级别。

  例如，“Datatype1.Datatype2.Datatype3.Datatype4.Datatype5.Datatype6.Member1”

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

其中：

- “Datatype1.Datatype2.Datatype3.Datatype4.Datatype5.Datatype6”是六个 UDT 嵌套级别。
- “Member1”是一个基本 InTouch 标记类型，如内存整型。

注意：不允许循环引用。

手动创建用户定义数据类型

您可以使用**用户定义类型**窗格在 WindowMaker 中创建和管理用户定义数据类型。您可以创建单独的 UDT 并使用一组离散操作构建 UDT 的层次结构。以下主题中描述了这些操作：

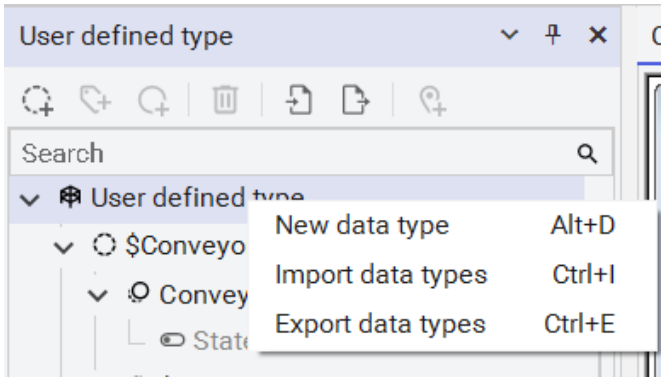
- 创建新数据类型
- 创建新成员
- 创建新衍生数据类型
- 创建新实例
- 将成员嵌套在另一种数据类型下
- 导入数据类型
- 导出数据类型
- 管理用户定义数据类型
- 批量编辑用户数据定义类型

注意：本节中显示的 UDT 名称、成员名称和属性名称仅供说明之用。您应该根据自己的应用程序适当地命名类型和成员。

创建新数据类型

创建新数据类型

1. 在**用户定义类型**窗格中，使用鼠标右键单击**用户定义类型**，然后从上下文菜单中选择**新建数据类型**，或选择工具栏上的**新建数据类型**图标。



或

在工具栏选择“新数据类型” 。

或

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用快捷键 **Alt+D**。

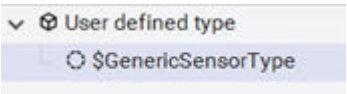
此时会创建数据类型。

缺省数据类型名称是 $Data type_00*n*，其中 *n* 的初始值 = 1。数据类型名称使用 $ 符号作为前缀。您无法删除 $ 符号。数据类型名称可以使用字母数字字符加上 $、# 和 _（下划线）字符。名称的最大长度是 32 个字符。

2. 要重命名数据类型，使用鼠标右键单击数据类型，然后选择重命名。

示例：

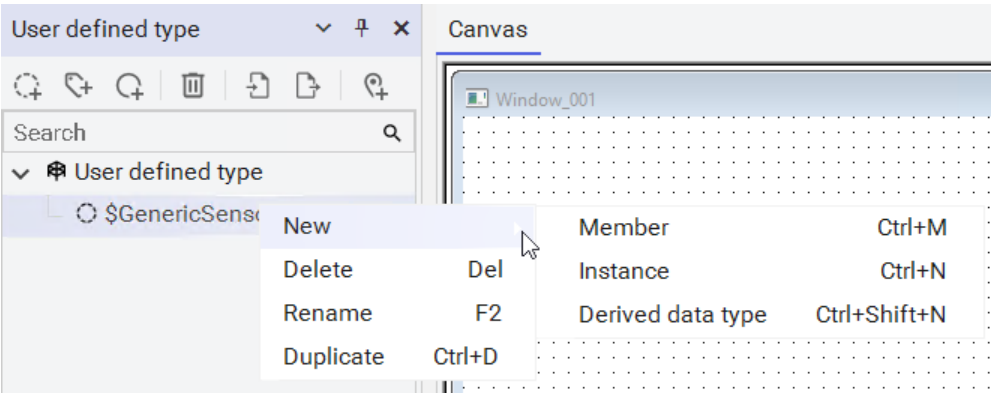
将新数据类型重命名为 GenericSensorType。



创建新成员

创建新成员

1. 在用户定义类型窗格中，使用鼠标右键单击数据类型，从上下文菜单中选择新建，然后选择成员，或选择工具栏上的新建成员图标。



或者

在工具栏中选择 “新建成员” 。

或者

使用快捷键 **Ctrl+M**。

即会创建新成员 **Member001**。成员名称可以使用字母数字字符加上 $、# 和 _（下划线）字符。成员名称的最大长度是 32 个字符。

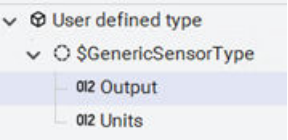
2. 要重命名成员，使用鼠标右键单击该成员，然后选择重命名。

示例：

您可以添加两个基本 InTouch 标记类型的成员，例如：

- 输出 - 类型：整数
- 单位 - 类型：消息（字符串）

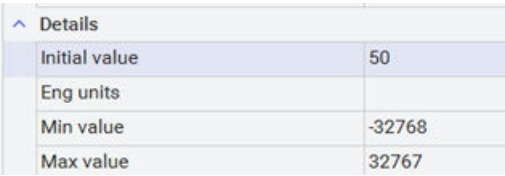
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您还可以在右窗格的属性网格中设置标记属性。

示例：

将输出成员“初始值”设置为 50。



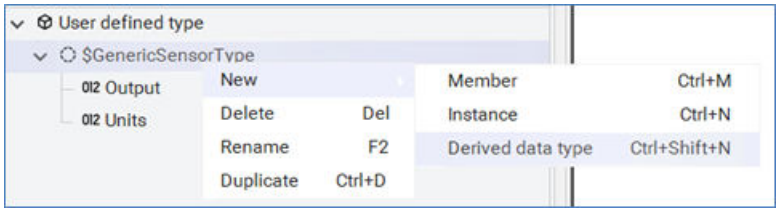
如需有关属性的详细信息，请参阅在“属性”网格中编辑用户定义数据类型。

创建新衍生数据类型

您可以衍生一种数据类型并继承它的所有成员。

创建衍生数据类型

1. 在用户定义类型窗格中，使用鼠标右键单击所需的数据类型，从上下文菜单中选择新建，然后选择衍生数据类型。



   或

   使用快捷键 **Ctrl+Shift+N**。

   此时会创建一个新的衍生数据类型。

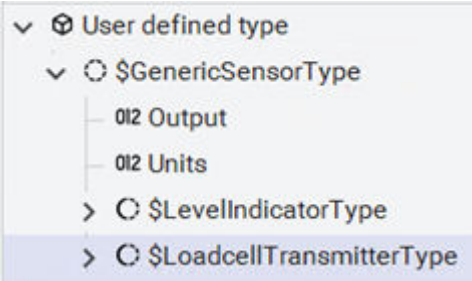
   衍生数据类型的缺省名称是 <父数据类型名称>_00*n*，其中 *n* 的初始值 = 1。例如，如果从 $Pump 创建衍生数据类型，那么第一个衍生数据类型的缺省名称是“$Pump_001”。衍生数据类型名称使用 $ 符号作为前缀，并且不能删除 $ 符号。衍生数据类型名称可以使用字母数字字符加上 $、# 和 _（下划线）字符。名称的最大长度是 32 个字符。

   基本数据类型中的成员将添加到衍生数据类型中。您可以修改父数据类型的成员属性（“名称”和“类型”属性除外）。您可以向衍生数据类型添加新成员。继承成员显示为灰色。您可以在属性窗格中修改衍生数据类型的属性。

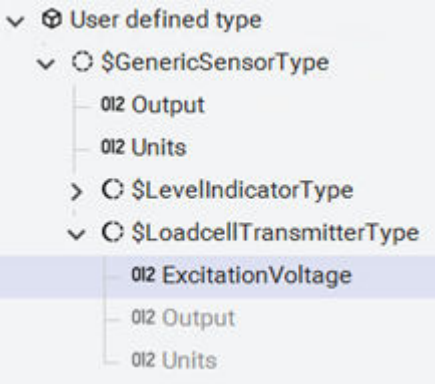
示例：

从 GenericSensorType 创建两个衍生数据类型，并将它们命名为 LevelIndicatorType 和 LoadcellTransmitterType。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



接下来，向 LoadcellTransmitterType 数据类型添加一个新成员，并将其命名为整数类型的 ExcitationVoltage。

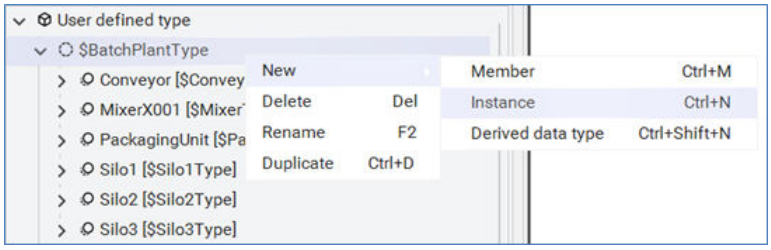


衍生数据类型继承父级的所有数据成员，并显示为灰色。

创建新实例

创建实例

1. 在**用户定义类型**窗格中，使用鼠标右**键单击**所需的数据类型，从上下文菜单中**选择新建**，然后**选择实例**。或者，可以**选择**工具栏上的**新建实例**图标。



   或

   在工具栏中选择“新建实例” 。

   或

   使用快捷键 **Ctrl+N**。

   此时会创建一个新实例。

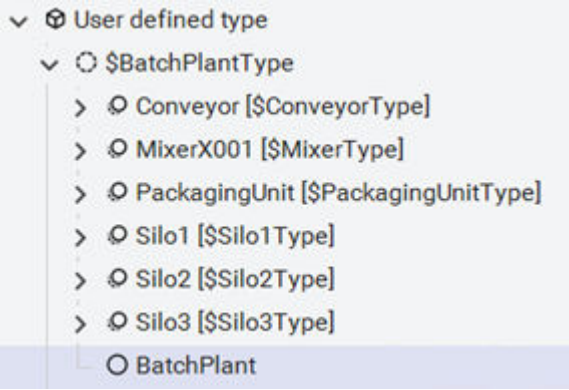
2. 重命名 UDT。缺省实例名称是 <数据类型名称>_00*n*，其中 *n* 的初始值 = 1。实例名称不使用 $ 字符作为前缀。实例名称可以使用字母数字字符加上 $、# 和 _（下划线）字符。名称的最大长度是 32 个字符。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

基本数据类型中的成员将添加到实例中。您无法向实例添加新成员。继承的成员将为灰色。您可以在属性窗格中修改实例的属性。您可以修改成员的属性（“名称”和“类型”属性除外）。数据类型或衍生数据类型下的所有成员都将属于同一数据类型或衍生数据类型下的实例。您可以从模型 – 标记名窗格中确认哪些成员属于哪个实例。所有实例及其成员都列在模型 – 标记名窗格的 **UDT** 下。

示例：

从 $BatchPlantType 创建一个实例并将其命名为 BatchPlant。

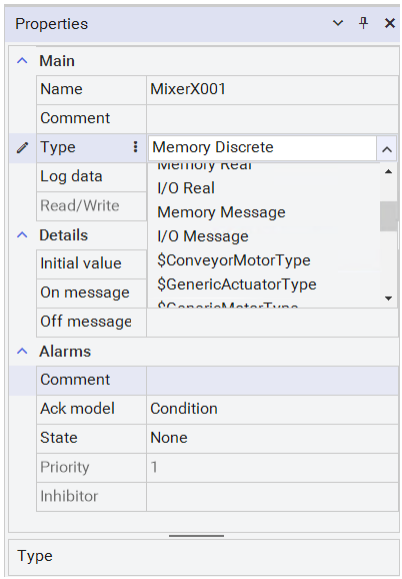


将成员嵌套在另一种数据类型下

UDT 嵌套是通过创建另一种数据类型的成员来完成的。您可以在属性窗格中将一种数据类型的成员指向另一种数据类型。嵌套可以减少创建标记所需的时间和精力。支持最多嵌套六层 UDT。

将成员嵌套在另一种数据类型下

1. 选择要嵌套的所需成员。
2. 在属性窗格的类型字段中，选择要在其下进行嵌套的所需数据类型或衍生数据类型。您也可以在类型字段中输入数据类型的名称，以过滤数据类型。



3. 该成员将嵌套在所选数据类型或衍生数据类型下。您可以对其进行重命名。例如，将嵌套成员重命名为“InletValve”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：此名称不是数据类型的另一个实例。它是用来引用数据类型实例的数据类型成员的另一种方式。

示例：

“InletValve”不是实际实例。它不会出现在标记名字典的根目录中，也不会出现在交叉引用实用程序中标记层次结构的根目录下。您可以将嵌套成员映射到实例，以在脚本和动画中引用它。

嵌套 UDT 级别的最大数为 6。请考虑以下示例：

Datatype1.Datatype2.Datatype3.Datatype4.Datatype5.Datatype6.Member1

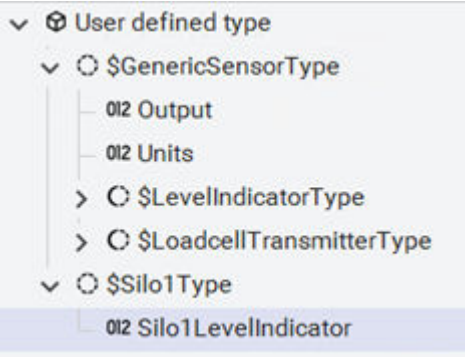
在本例中：

- “Datatype1.Datatype2.Datatype3.Datatype4.Datatype5.Datatype6”是 6 个级别的 UDT。
- “Member1”是一个基本 InTouch 标记类型，如内存整型。

注意：不允许循环引用。

示例：

创建一个新的数据类型并将其命名为 Silo1Type，然后添加一个名为 Silo1LevelIndicator 的成员。



现在，从属性网格中，可以将属性类型更改为另一种数据类型。假设您选择 LevelIndicatorType。

嵌套成员已添加。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

导入数据类型

您可以通过将包含先前创建的一组 UDT 的 .json 文件导入到您的应用程序中来有效地构建一组 UDT。

UDT 在 JSON 导入期间支持以下模式：

- **ASK**：在冲突期间显示弹出窗口以进行确认。此行为类似于 DBLoad/DBDump 操作中使用的 ASK 模式（来自 CSV 文件）。这是缺省模式。
- **SKIP**：跳过冲突的条目而不进行任何更改。功能类似于 DBLoad/DBDump 操作中的 IGNORE 模式。
- **OVERRIDE**：在冲突期间替换现有条目。行为类似于 DBLoad/DBDump 操作中的 REPLACE 模式。
- **ABORT**：停止 UDT 导入操作，就像 DBLoad/DBDump 操作中的 ABORT 操作一样。

缺省情况下，导出 UDT 时会在 JSON 文件中配置 ASK 模式。

注意：使用 OVERRIDE 模式可避免针对每个报警组显示弹出窗口。

在 JSON 文件中更改模式：

1. 打开要导入的所需 JSON 文件。（例如，udtinfo-20241108.json 或 udtinfo-20241105-DWTP Tags Import.json）
2. 更新“模式”字段，如下所示：

   Json 内容：

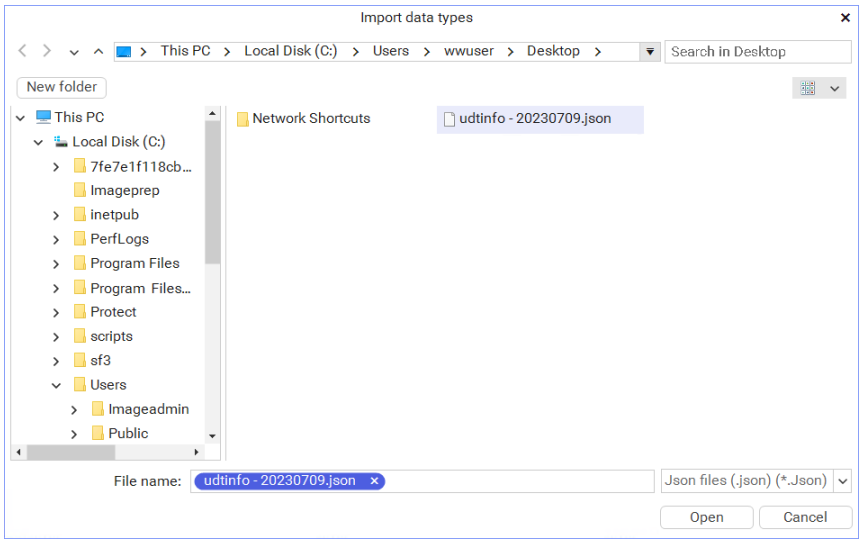
   {"Version":13,"Mode":"OVERRIDE"
3. 使用更新的 JSON 文件执行 UDT 导入操作。

导入数据类型

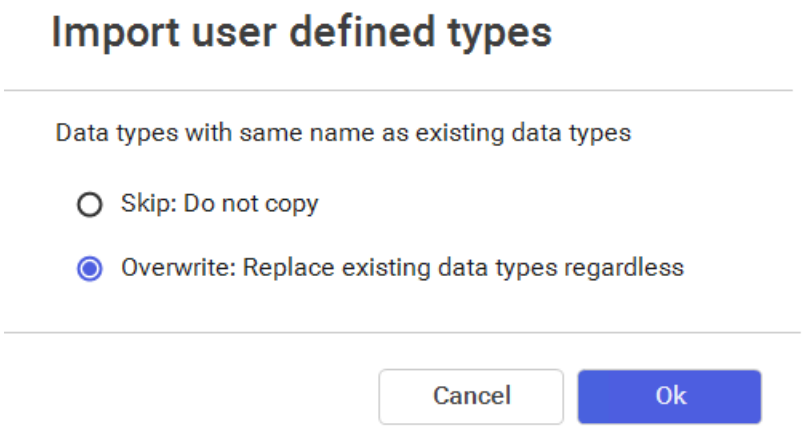
1. 打开用户定义类型窗格。
2. 使用鼠标右键单击用户定义类型并从上下文菜单中选择导入数据类型，然后选择工具栏上的导入图标。

3. 导航到并选择含有数据类型配置详细信息的 .json 文件，然后选择打开。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**导入用户定义类型对话**框中，
   - **选择跳过**：不**复制**，以跳过导入与现有数据类型同名的数据类型。
   - **选择覆盖**：**替换现有数据类型**，以覆盖并替换现有数据类型。



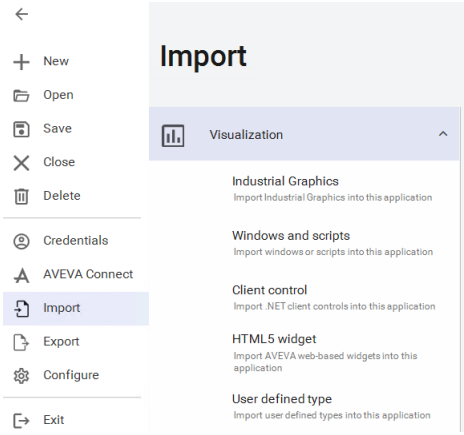
5. **选择确定**。

   .json 文件中可用的所有数据类型和其它项目将根据您在上一步中**选择**的**选项进**行显示。

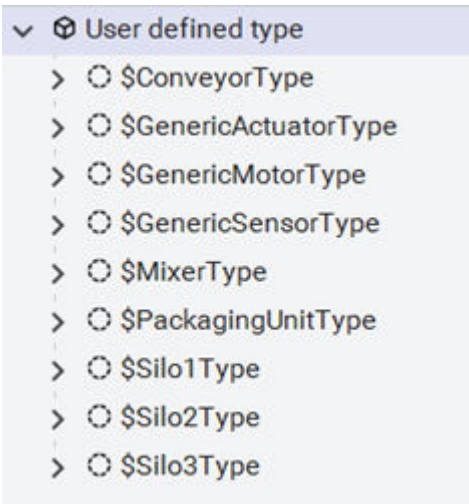
   注意：导入文件时不会保留覆盖的实例值。

您也可以使用**文件**菜单下的**导入** > **可视化** > **用户定义类型选项**导入数据类型。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



导入文件后，即会显示 UDT 层次结构。

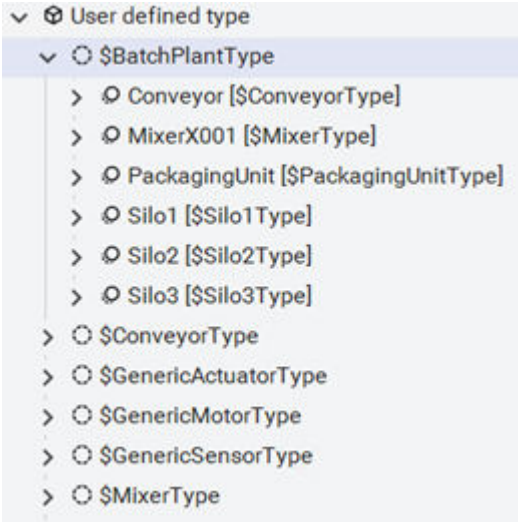


您可以向层次结构添加更多 UDT。为此，请创建一个名为“BatchPlantType”的新数据类型，添加成员，然后更改每个成员的属性网格中的类型，如下所示：

- Conveyor - 类型 $ConveyorType
- MixerX001 - 类型 $MixerType
- PackagingUnit - 类型 $PackagingUnitType
- Silo1 - 类型 $Silo1Type
- Silo2 - 类型 $Silo2Type
- Silo3 - 类型 $Silo3Type

生成的层次结构反映了新成员。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



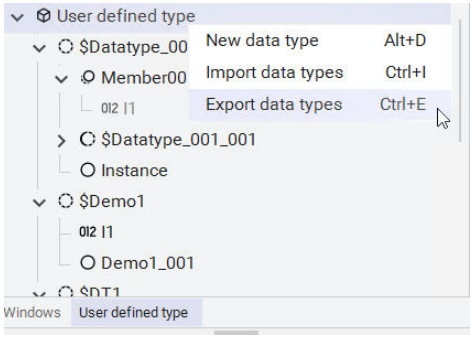
导出数据类型

您可以导出数据类型以在其它应用程序、系统中的其它计算机或其它系统中使用。

导出数据类型：

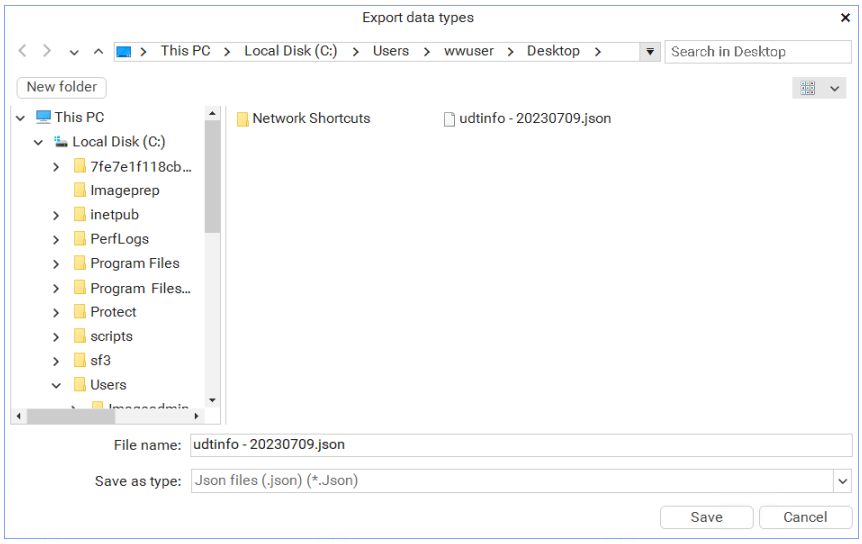
1. 打开用户定义类型窗格。

   使用鼠标右键单击用户定义类型并从上下文菜单中选择导出数据类型，然后选择工具栏上的导出图标。



   此时打开导出数据类型窗口。

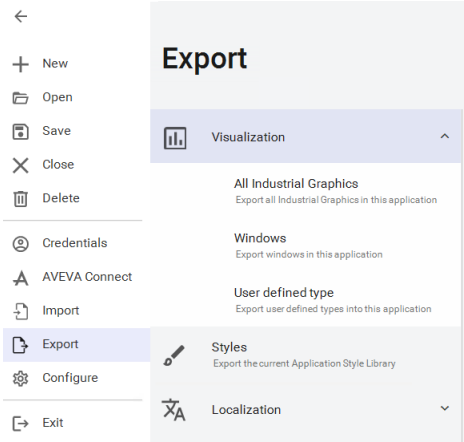
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在需要时修改文件名，然后**选择保存**。

   .json 文件将保存到所选的位置。您可以使用外部 json **编辑器编辑导**出的文件，然后再将其**导**回您的**应**用程序或另一个**应**用程序中。

您也可以使用**文件**菜单下的**导出** > **可视化** > **用户定义类型选项**导出数据类型。



注意：在**导**出期**间**，所有 UDT 都将包含在 JSON 文件中。

管理用户定义数据类型

除了前面主**题**中所述的**创**建和更新 UDT 的操作之外，您**还**可以**执**行基本的管理操作。

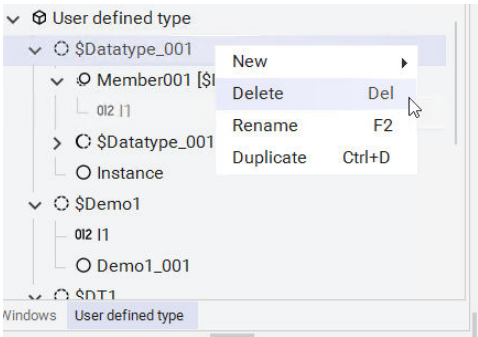
删除成员

- 右**键单击**想要**删**除的成**员**，然后从上下文菜**单**中 **选择删除。**您**还**可以**选择**工具**栏**上**显**示的**删除图标**删除成**员**。使用 **Ctrl 键选择**多个成**员**。

要除数据类型、衍生数据类型或实例

使用鼠**标**右**键单击**数据**类**型、衍生数据**类**型或**实**例，然后从上下文菜**单**中**选择删除，**或**选择**工具**栏**上的**删除图标。这**将**删**除数据**类**型及其所有成**员**。

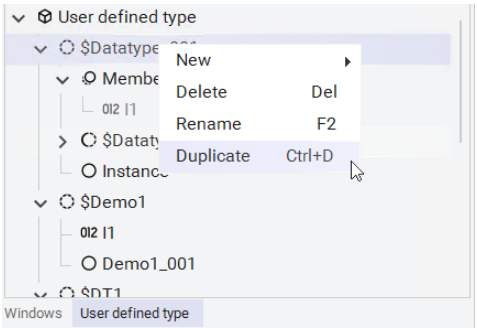
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



注意：除非删除数据类型下的实例和衍生数据类型，否则无法删除该数据类型。

复制数据类型

- 在**用户定义类型**窗格中，使用鼠标右**键单击**基本数据类型，然后从上下文菜单中**选择复制**。此时会创建基本数据类型及其所有成员的副本。



注意：您只能复制数据类型和衍生数据类型。无法复制实例和成员。

添加位置属性

1. 打开**用户定义类型**窗格。
2. 突出显示数据类型或衍生数据类型，然后**选择**工具栏上的**添加位置** 图标。

   "纬度"和"经度"的新属性将添加到数据类型或衍生数据类型中。如需有关使用位置属性的详细信息，请参阅本节后面的 MapApp 主题。

批量编辑用户数据定义类型

您可以使用 JSON 文件批量编辑 UDT。要编辑导出的 JSON 文件，请使用任何公开可用的流行 JSON 编辑器来添加、删除或编辑 UDT。

批量编辑或创建 UDT

1. 将现有的 UDT 导出到 JSON 文件中。如需有关如何导出 UDT 的详细信息，请参阅导出数据类型。
2. 在文本或 JSON 编辑器中打开 JSON 文件。
3. 通过复制现有 UDT 并将其粘贴回文件中来编辑或创建 UDT。
4. 根据需要重命名或编辑其它属性。
5. 保存 JSON 文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 将 JSON 文件导回到 WindowMaker 的用户定义类型窗格中。如需有关导入的详细信息，请参阅导入数据类型。

示例：


```
{
  Version : 12
  Templates : [ 1 item
    0 : {
      Name : DT1
      Comments : value
      Members : [ 1 item
        0 : {
          Name : IODisc
          TagType : AtomicTag
          Source : Reference
          TagComment : value
          AlarmGroup : 1
          LogData : False
          LogEvents : False
          LocalTag : false
          TypeOf : Discrete
          IoConfig : { 2 props }
          AlarmConfiguration : { 3 props }
        }
      ]
      Overrides : [ 0 items
      ]
    }
  ]
  Derived Templates : [ 0 items
  ]
  Instances : [ 3 items
    0 : {
      Name : DT1_001
      Comments : value
      TypeOf : DT1
      Overrides : [ 3 items
        0 : {
          Name : IOInt
          ContextName : value
          DataType : Int
          Val : {
            DataType : Int
            DefaultValue : value
          }
```


查看用户定义数据类型

您可以通过两种方式查看用户定义数据类型：

- 通过“用户定义类型”视图
- 通过“模型 - 标记名”视图

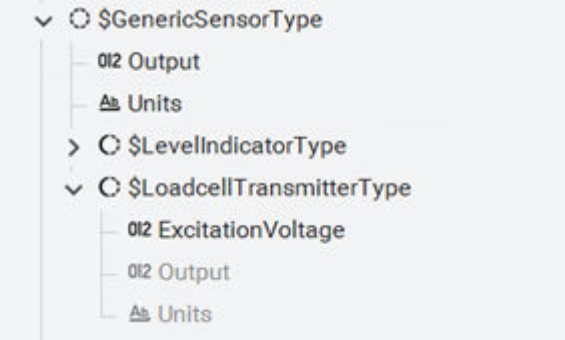
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

“用户定义类型”视图

用户定义类型窗格显示用户定义类型列表、成员及成员数据类型、衍生数据类型及实例。

- 数据类型使用“$”字符作为前缀。
- 成员数据类型会附加“[$<数据类型名称>]”。
- 所有成员和衍生数据类型都可以完全展开以显示成员。
- 实例名称无法在此视图中展开。
- 对于每种数据类型，所有子项都按如下所示的顺序进行排序。在组内，项目按字母顺序进行排序。
  a. 直接成员
  b. 继承成员
  c. 衍生数据类型。
  d. 实例。

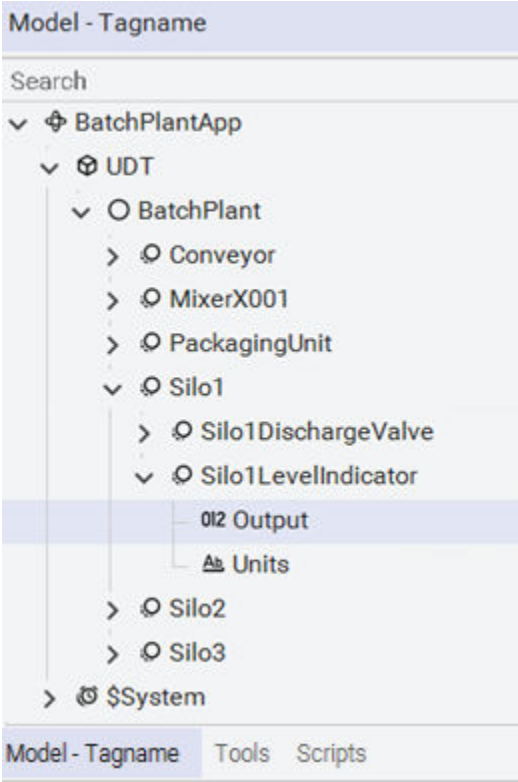
示例：



“模型 - 标记名”视图

模型 – 标记名视图是添加了 UDT 实例视图的 InTouch 标记视图。您可以查看 UDT 实例及其成员。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



在"属性"网格中编辑用户定义数据类型

注意： 您必须在属性网格中编辑 UDT。您无法在标记名字典中查看或编辑 UDT。

更新属性

您可以从属性网格中查看和修改 UDT 属性。**"用户定义类型"**窗格或**模型 - 标记名**窗格中所选项目的属性显示在 WindowMaker 右侧的属性窗格中。

- 突出显示每个属性可选择并更新该属性，或在属性窗格的底部查看相应的属性描述。
- 您所做的任何更改都会自动保存。
- 属性字段将根据在类型属性字段中选择的选项进行更改。
- 类型属性字段列出了 UDT 以及其它数据类型。
- 属性列表会更新以反映与所选类型相关联的属性。
- 您可以编辑定义成员的所有标记属性值（名称和类型属性除外）。

成员数据类型

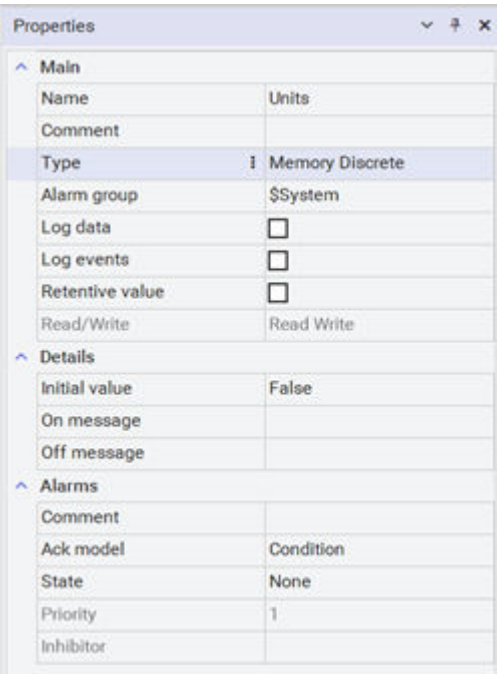
如果将成员更改为另一种数据类型，则该属性将仅显示以下属性。在示例中，您可以选择 $BatchPlantType > Silo1 成员。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

编辑成员

如果将成员类型更改为基本 InTouch 标记，则所有相关标记属性都将变为可编辑。

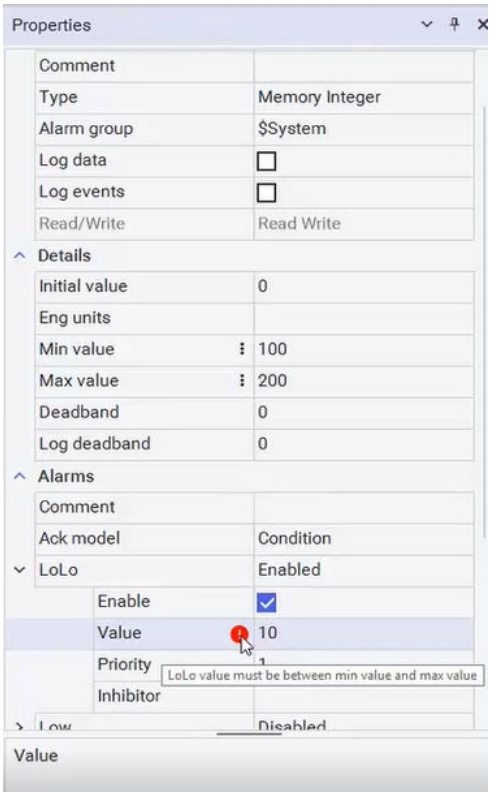
例如，选择一个名为 $GenericSensorType > Units 的 UDT，并将类型更改为“内存离散”。属性网格将刷新以显示此离散标记的所有相关标记属性。



无效值和必填字段

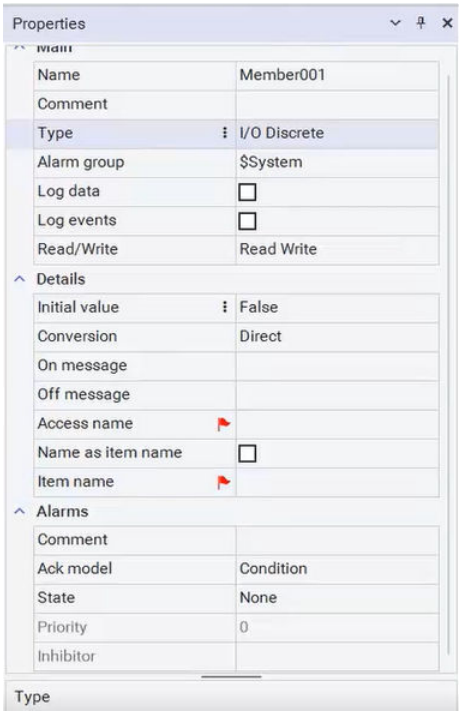
属性的无效值由红色图标指示。在更新属性值，然后选择任何其它用户定义类型之后，属性名称的前面即会显示一个红色的错误图标。如果将鼠标悬停在错误图标上，则会显示一个含有错误原因的工具提示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



需要值的属性由红色标帜指示。不强制您在数据类型级别输入值。但是建议您在实例级别输入或覆盖值。

示例：



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

值覆盖和更改传播

您可以在定义成员的数据类型级别编辑所有标记属性值（名称和类型属性除外）。

示例 1：

1. 选择 $GenericSensorType > Output。
2. 将初始值更改为 50。
3. 然后，选择 $LevelIndicatorType > Output。
4. 请注意，初始值也是 50。
5. 选择 $GenericSensorType > Output。
6. 将初始值更改为 60。

   返回 $LevelIndicatorType > Output。请注意，初始值现在也更新为 60。这是更改传播的一个示例。
7. 您可以选择覆盖初始值。将初始值设置为 75。

   请注意，初始值现在以粗体显示，表示它已被覆盖。

8. 选择 $GenericSensorType > Output 并将初始值更改为 65。

   在这种情况下，该值不会向下传播到 $LevelIndicatorType，因为它的值已被覆盖。

示例 2：

此示例是从前面的示例 1 扩展而来的。

1. 在模型 - 标记名视图中，选择 BatchPlant > Silo1 > Silo1LevelIndicator > Output。初始值为 75。

   这来自 $LevelIndicatorType 数据类型。
2. 输入覆盖值 25。
3. 选择 BatchPlant > Silo2 > Silo2LevelIndicator > Output 并将初始值覆盖为 55。
4. 选择 BatchPlant > Silo3 > Silo3LevelIndicator > Output。将初始值覆盖为 80。

更改传播

数据类型的更改会立即反映在用户定义类型视图中。

添加新成员

添加新成员会反映在衍生数据类型、成员数据类型和实例中。

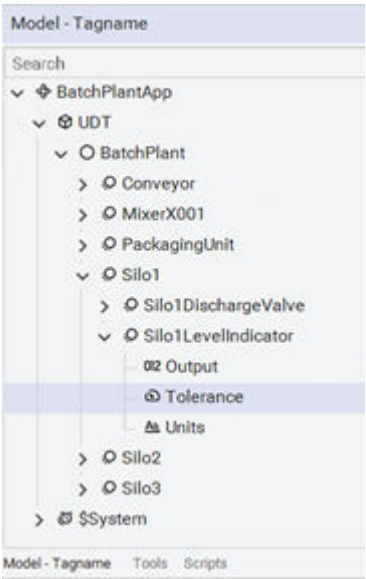
示例：

1. 选择 $GenericSensorType。
2. 添加一个名为“Tolerance”，类型为“内存实型”的新成员。

   这个新成员现在可以在其衍生数据类型 $LevelIndicatorType 中使用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

这个新成员在实例成员数据类型中也可用。



更改成员类型

更改现有成员类型也会立即反映在衍生数据类型、成员数据类型和实例中。

示例：

从前面的示例扩展而来：

1. 转到 $GenericSensorType > Tolerance。
2. 在属性网格中，将数据类型从“内存实型”更改为“内存消息”。

   $LevelIndicatorType > Tolerance 也更新为数据类型“内存消息”。

UDT 对现有功能的支持

注意：示例和图像中所示的 UDT 名称、成员名称和属性仅用于说明目的。您应该根据自己的应用程序适当地命名类型和成员。

在以下 InTouch 功能中支持 UDT：

- 智能感知
- 动画
- 脚本

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

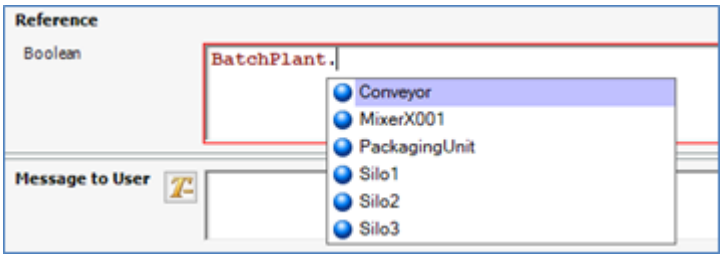
- 替换引用
- 浏览 UDT 成员
- 将实例的 UDT 成员拖放到工业图形编辑器画布中
- 动画和脚本
- 报警与报警客户端控件
- 历史和趋势客户端
- 将 UDT 成员配置为 I/O 标记
- 将 UDT 实例用作父对象
- Web 客户端
- MapApp
- 交叉引用
- 删除未使用标记
- 与 SuperTag 共存
- 许可
- 运行时标记查看器

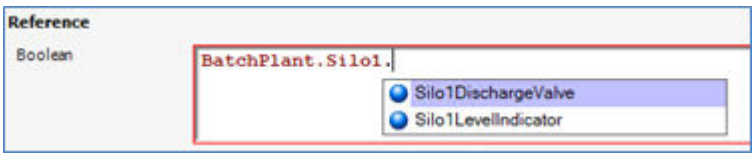
智能感知

UDT 实例的智能感知仅列出直接级别的所有成员和成员数据结构。“工业图形编辑器”支持脚本、动画和自定义属性中的智能感知。本机 InTouch 仅支持脚本中的智能感知。

- 工业图形：脚本、动画和自定义属性中的智能感知支持。
- 本机 InTouch：仅本机 InTouch 脚本中的智能感知支持。

示例：





© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

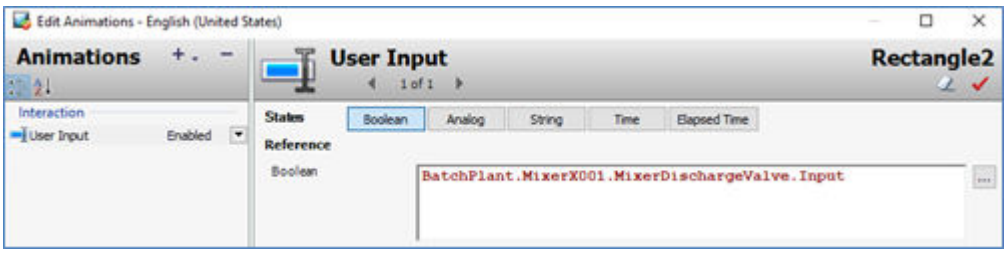
动画

您可以在工业图形动画和本机 InTouch 动画中配置 UDT。

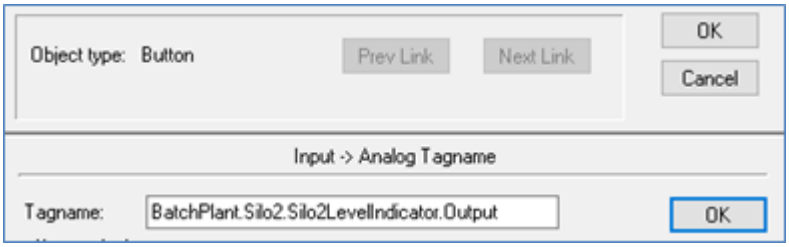
备注：具有值显示动画的 UDT 的替换标记的行为与远程标记相同。您需要将完整引用替换为 UDT 标记或带点域的 UDT 标记。

示例：

在工业图形动画中配置 UDT：



在本机 InTouch 动画中配置 UDT：

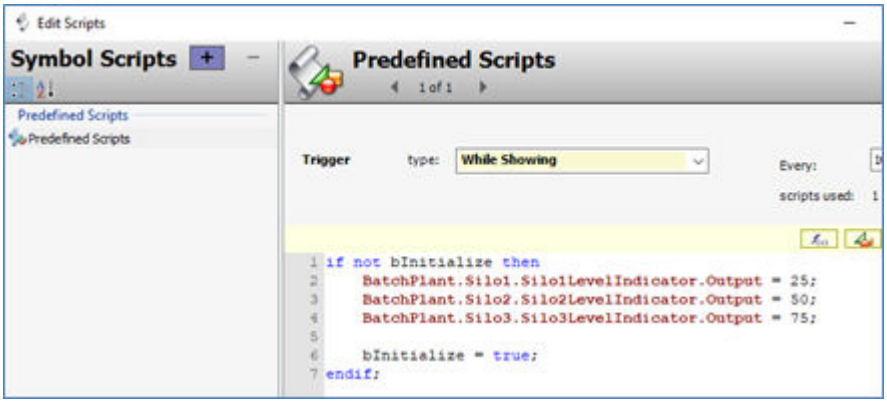


脚本

您可以在工业图形脚本和本机 InTouch 脚本中配置 UDT。

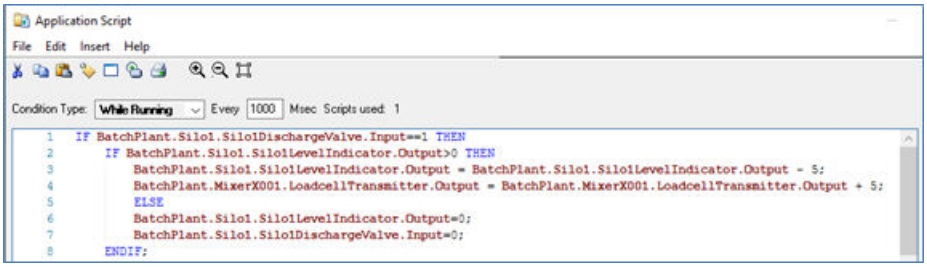
示例：

在工业图形脚本中配置 UDT：



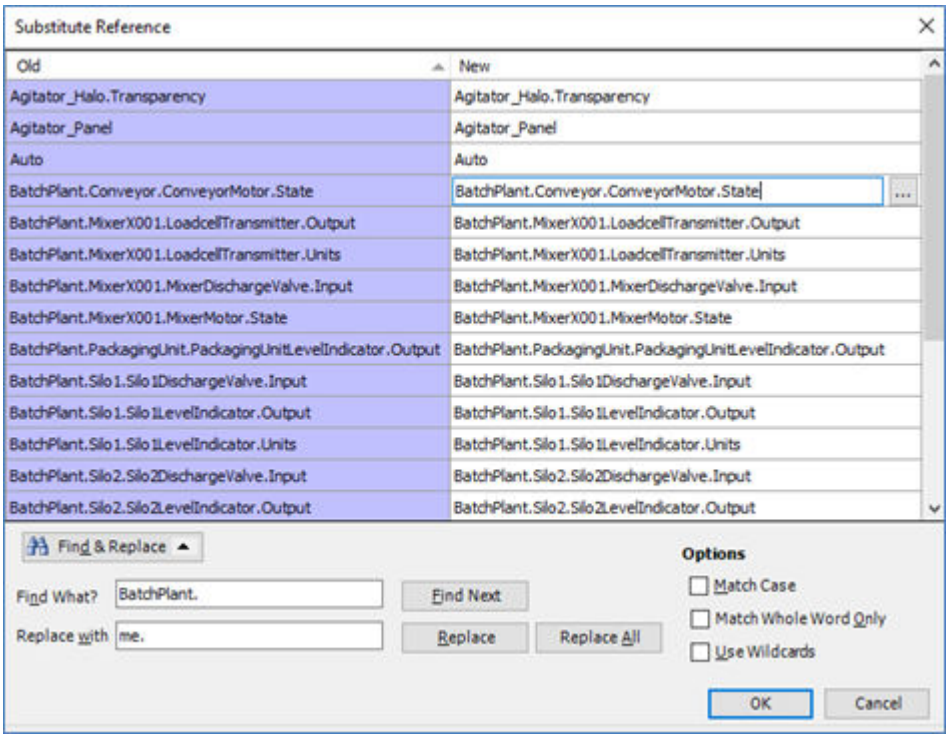
在本机 InTouch 脚本中配置 UDT：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



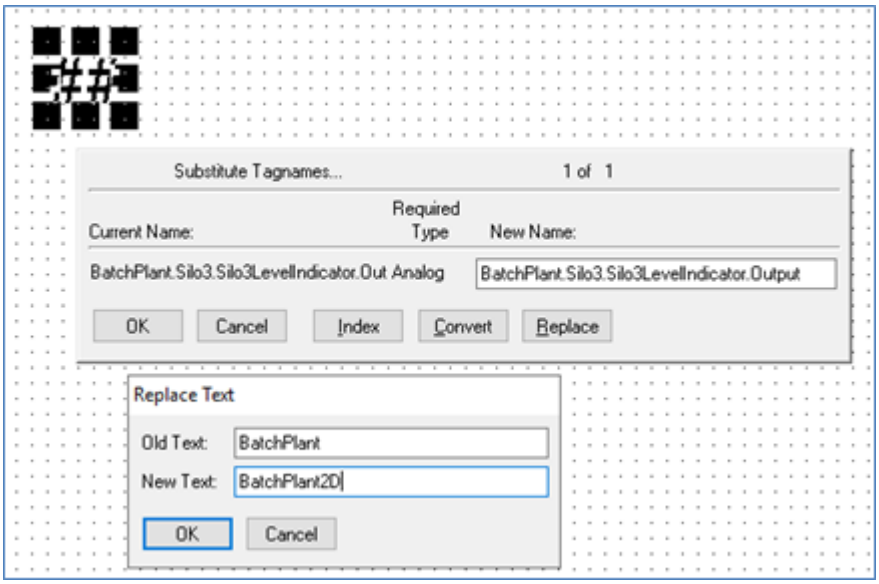
替换引用

在工业图形编辑器和本机 InTouch 中，您可以按照与其它 InTouch 标记相同的方式替换 UDT 实例成员引用。以下是工业图形中的替换引用示例：



以下是本机 InTouch 中的替换引用示例：

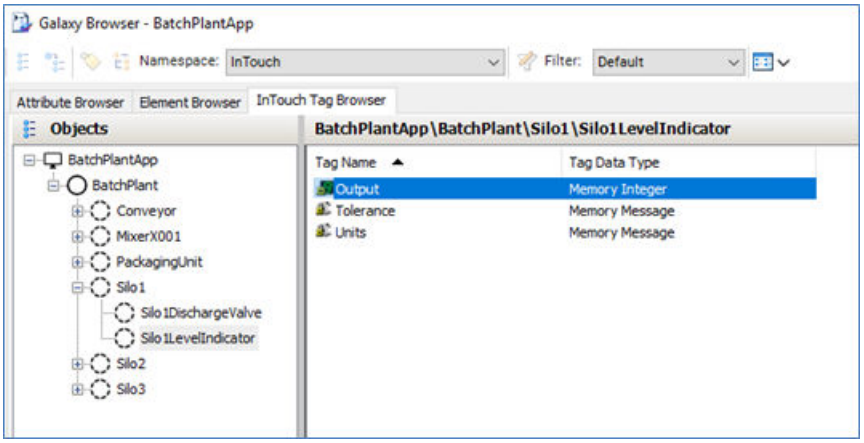
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



浏览 UDT 成员

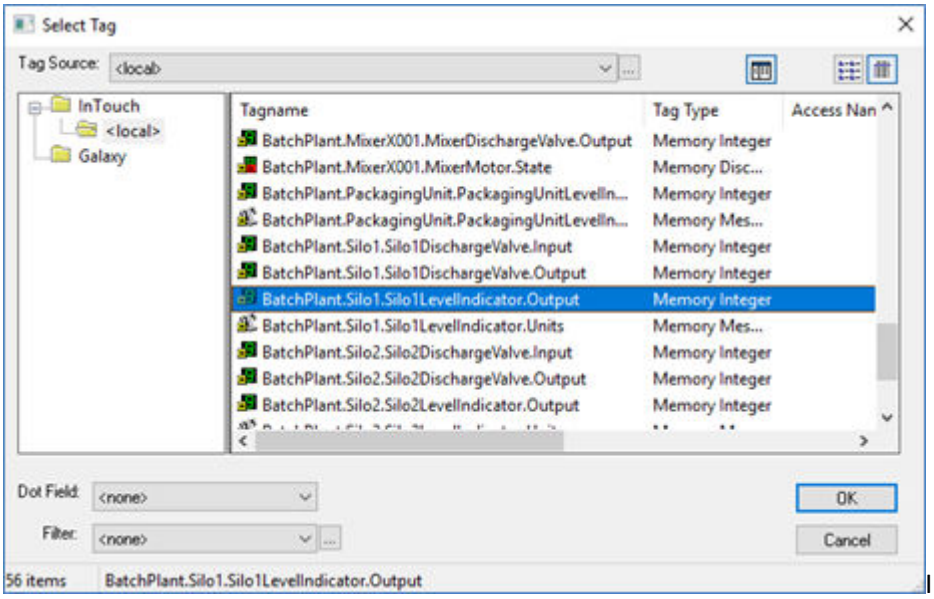
在“工业图形编辑器”中配置动画或使用脚本时，使用省略号按钮打开 **Galaxy 浏览器**来浏览 UDT 成员。在本机 InTouch 的动画或脚本选项中，打**开标记浏览器**来浏览 UDT 成员。

以下是在“工业图形编辑器”中浏览的示例：



以下是在本机 InTouch 中浏览的示例：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

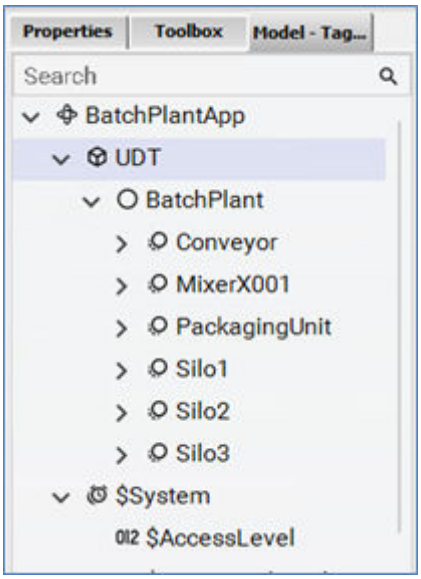


将实例的 UDT 成员拖放到工业图形编辑器画布中

在“工业图形编辑器”中，**模型 - 标记名**窗格还显示 UDT 实例。您可以将单个或多个 UDT 成员拖放到画布上，以在“工业图形编辑器”中创建动画。

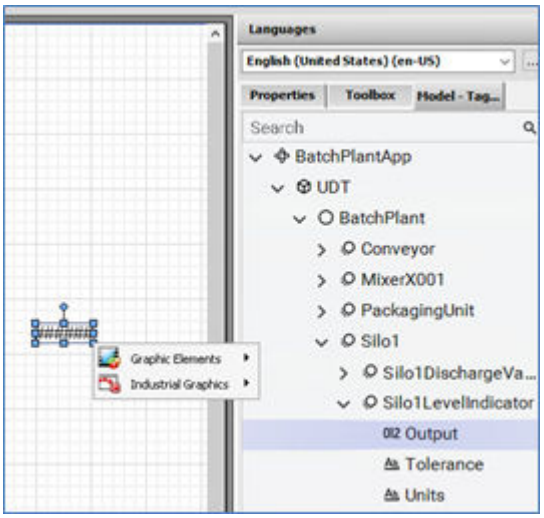
示例：

在**模型 - 标记名**中进行拖放：



在“工业图形编辑器”中进行拖放：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



动画和脚本

如果图形配置为以下内容，则 WindowViewer 中支持 UDT 实例成员：

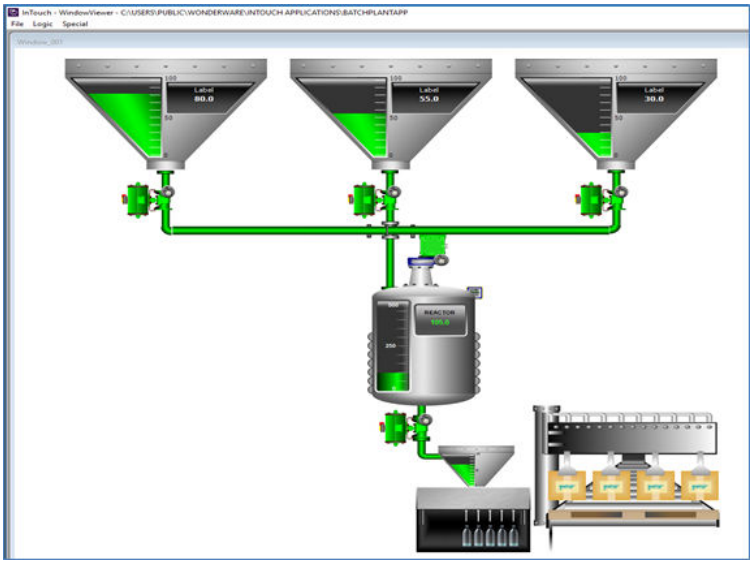
- 工业图形和本机 InTouch 图形中的动画和脚本

  或

- 本机 InTouch 中的脚本。

示例：

在 WindowViewer 中您可以看到以下内容：



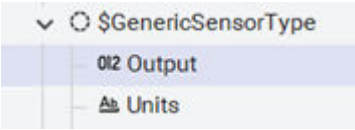
报警与报警客户端控件

实例的 UDT 成员以与其它 InTouch 标记相同的方式支持报警。您可以在属性网格中更新一个成员的报警属性。在 WindowViewer 中，可以查看生成报警的实例的成员并将它们显示在报警客户端控件中。

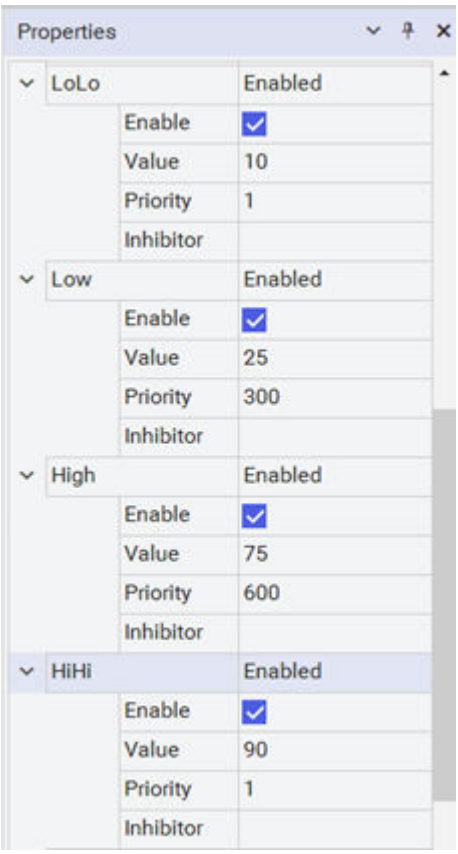
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例：为 UDT 成员配置报警

1. 在用户定义类型视图中，选择 $GenericSensorType > Output 成员。



2. 在属性网格中，更新此标记的报警属性，如下图中所示。

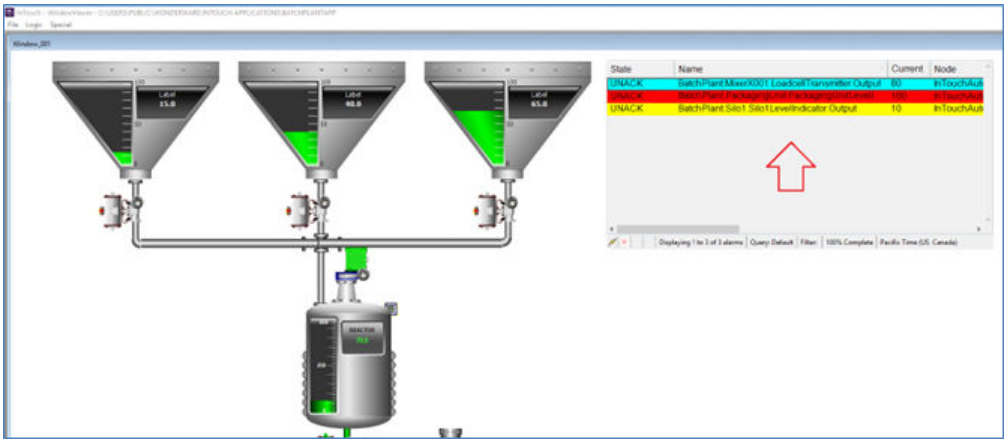


报警属性随后会向下传播到 UDT 实例级别。以下 UDT 成员可以配置报警，如前面的示例中所示。

- BatchPlant.Silo1.Silo1LevelIndicator.Output
- BatchPlant.Silo2.Silo2LevelIndicator.Output
- BatchPlant.Silo3.Silo3LevelIndicator.Output

下图是 WindowViewer 中 UDT 的报警支持的示例：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

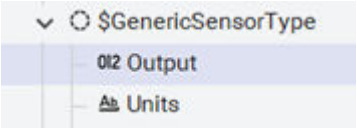


历史和趋势客户端

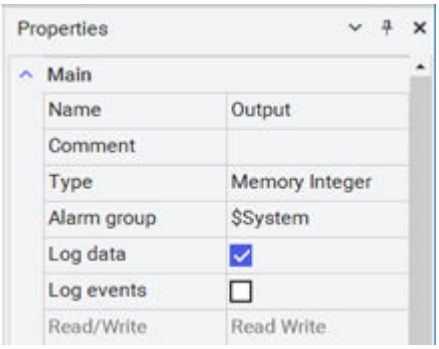
与其它 InTouch 标记一样，实例的 UDT 成员可以将数据记录到 LGH 文件或 Historian。

启用日志数据的示例

1. 在用户定义类型视图中，选择 $GenericSensorType > Output 成员。



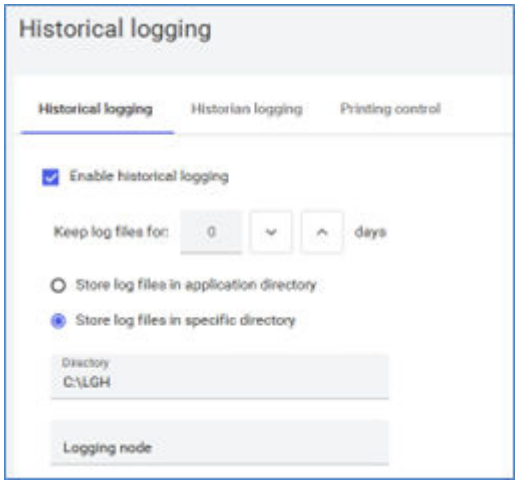
2. 在属性网格中，检查日志数据属性是否更新为 True（已选中），如下所示：



为 LGH 启用历史记录的示例：

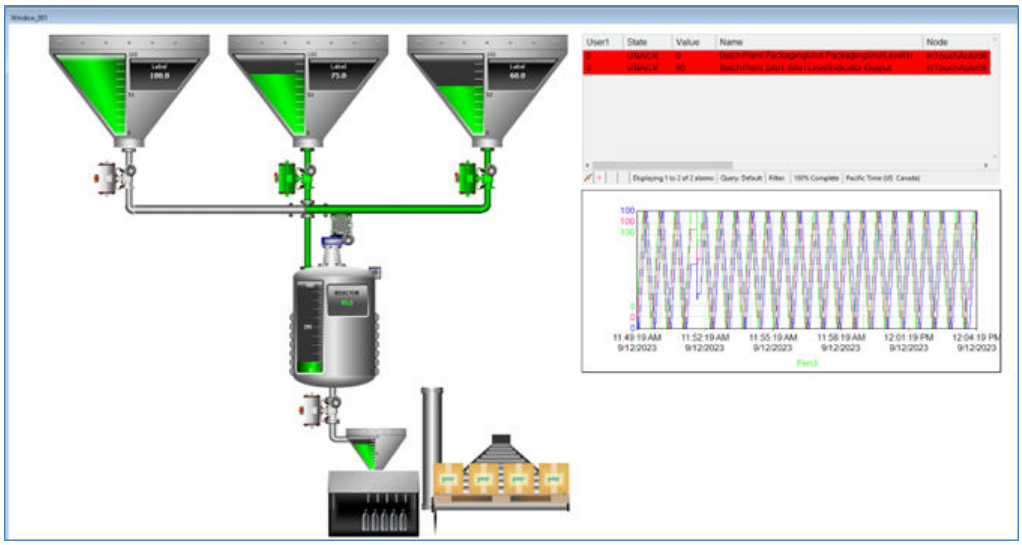
1. 转到后台：文件 > 配置 > 历史记录

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 如果指定的目录不存在，请使用文件资源管理器来创建它。
3. 打开 WindowViewer 以转到运行时并验证将历史数据写入 LGH 文件中。

   在下图中，显示 UDT 成员将历史数据记录到 LGH 文件中，并且趋势客户端正在绘制实时和历史数据。历史数据被回填到趋势中。



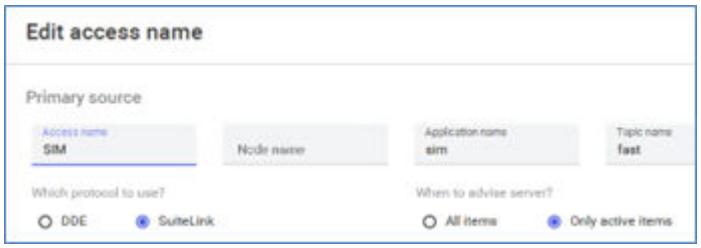
将 UDT 成员配置为 I/O 标记

您可以按照与其它 InTouch I/O 标记相同的方式将 UDT 成员配置为 I/O 标记。在属性网格中，配置类型、访问名和项目名字段，以将成员设置为 I/O 标记。

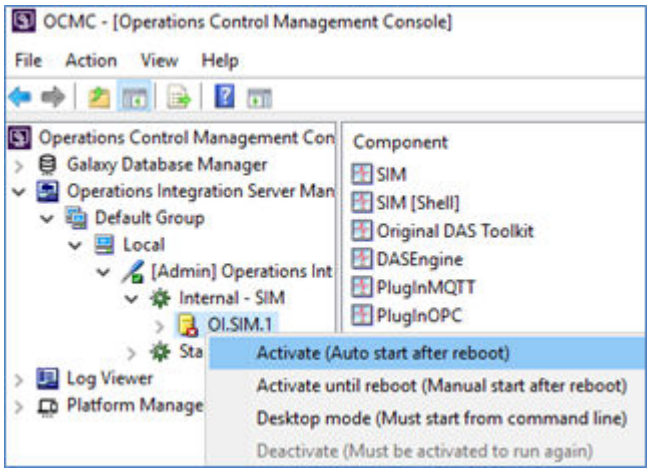
示例：将 UDT 成员配置为 I/O 标记

1. 为本地模拟器服务器创建访问名“SIM”。

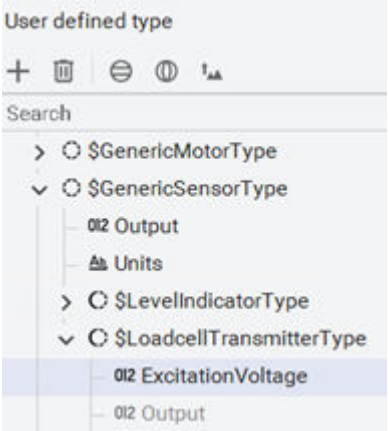
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 从 **Operations Control 管理控制台** (OCMC) 激活 SIM 服务器。

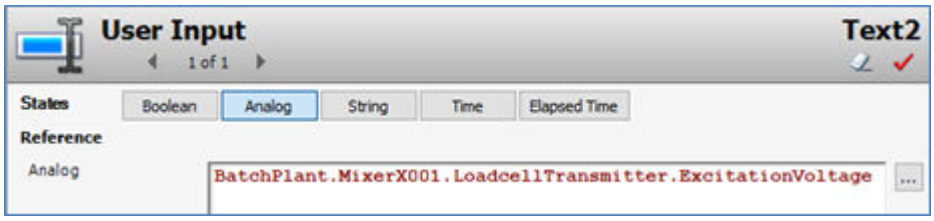


3. **选择**并配置 ExcitationVoltage 示例成**员**。

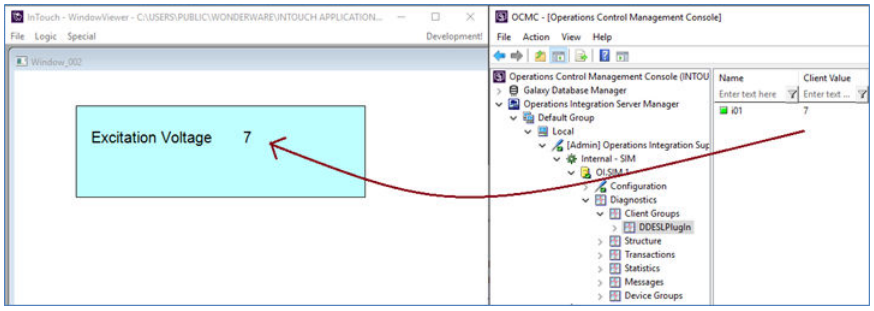


4. 配置属性以使成**员**成**为** I/O **标记**。您**还**可以在**实**例**级别**覆盖**访问**名和/或**项**目名。
   a. 将**类**型**设**置**为** I/O 整型
   b. 将**访问**名**设**置**为** SIM（您可以从下拉列表中**选择**）
   c. 将**项**目名**设**置**为** i01
5. 使用下**图**所示的引用**创**建**图**形并添加**动**画。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



6. 嵌入您刚刚创建的图形并在运行时查看它。值应绑定到 I/O 引用。



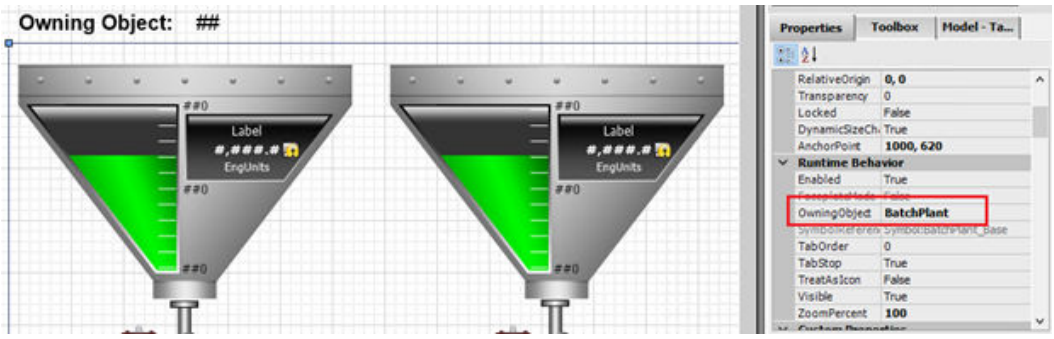
将 UDT 实例用作父对象

您可以配置嵌入图形的父对象并指向 UDT 实例。在运行时，将用 UDT 实例替换 **me.** 相对引用。您也可以在运行时将父对象更新为不同的 UDT 实例。

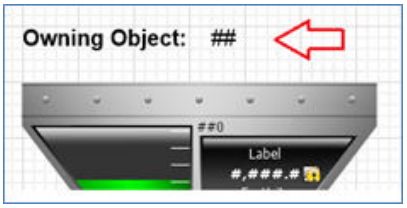
示例：将 UDT 实例用作父对象

1. 在可视化文件夹中，导航到 ReactorDemo Symbols > MainDisplays
2. 编辑 BatchPlant_OwningObj 图形。
3. 选择嵌入的图形 BatchPlant_RelativeRef1。

   在属性网格中，请注意 OwningObject 属性设置为 BatchPlant，它是 UDT 实例。



您可以从 BatchPlantType 创建另一个 UDT 实例，以便在运行时可以使用输入动画切换到不同的 UDT 实例。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例：在运行时将父对象切换到不同的 UDT 实例

1. 在另一个框架窗口中嵌入 BatchPlant_OwningObj 图形。
2. 如果 WindowViewer 正在运行，请关闭它。
3. 切换到 WindowViewer。
4. 验证“me.”相对引用是否绑定到 BatchPlant。

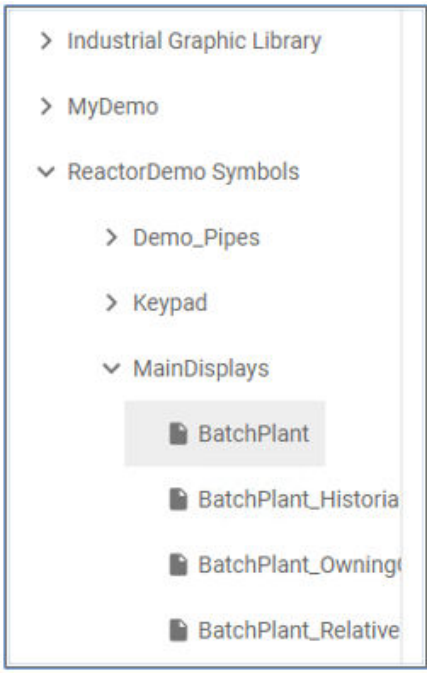
您可以将父对象更改为不同的实例。

Web 客户端

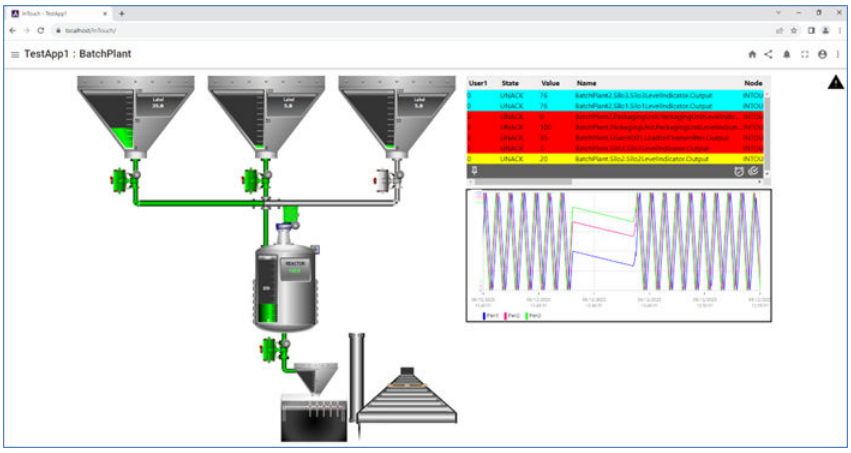
实例的 UDT 成员的工作方式与 Web 客户端中其它 InTouch 标记的工作方式相同。

示例：

当应用程序在 WindowViewer 中运行时，从 WindowMaker 启动 Web 客户端，然后从菜单中打开 BatchPlant 图形。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

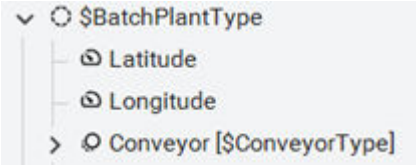


MapApp

实例的 UDT 成员的行为与 MapApp 中其它 InTouch 标记的行为相同。

添加位置成员的示例

1. 在此示例中选择“BatchPlantType”，然后使用工具栏中的添加位置选项添加新成员“纬度”和“经度”。

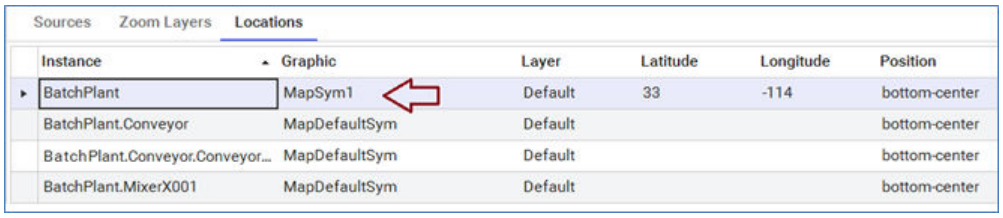


2. 在属性网格中设置以下值：
   - 纬度 = 33
   - 经度 = -114

配置地图设置的示例

1. 在可视化文件夹中，转到“小组件”并双击 Map_App。
2. 将图形“MapSym1”与“BatchPlant”相关联。MapSym1 是您想要在此 UDT 实例 BatchPlant 的 MapApp 中显示的图形。

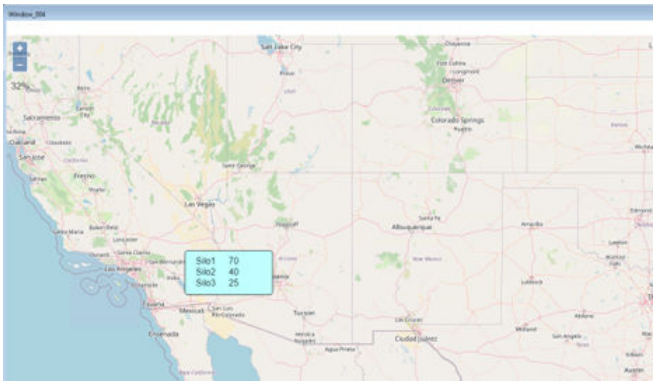
   MapApp 配置列出了实例以及该实例的所有成员（成员数据类型）



在运行时使用 MapApp 的示例

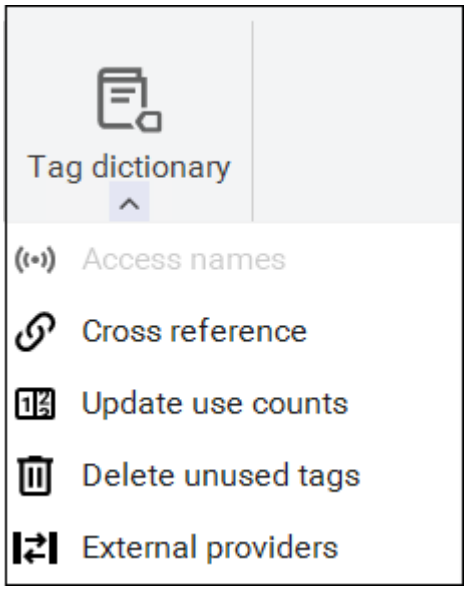
- 在另一个框架窗口中嵌入图形“MapAppMain”。如果 WindowViewer 已在运行，请将其关闭，然后切换到运行时。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



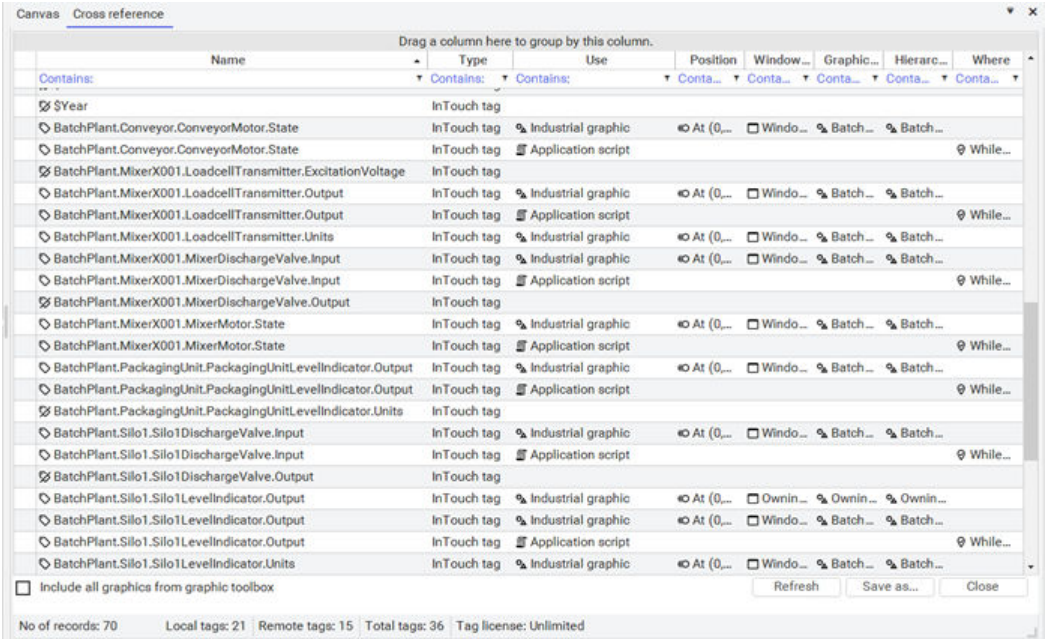
交叉引用

您可以使用交叉引用实用程序来查找实例的 UDT 成员的使用位置。选择 WindowMaker 上的**标记字典**，然后选择交叉引用。



交叉引用示例：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

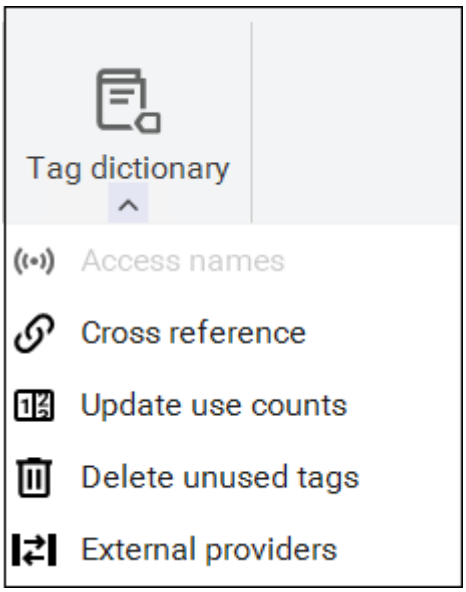


删除未使用标记

实例的未使用 UDT 成员显示在**删除未使用标记**页面中，您可以从那里删除它们。

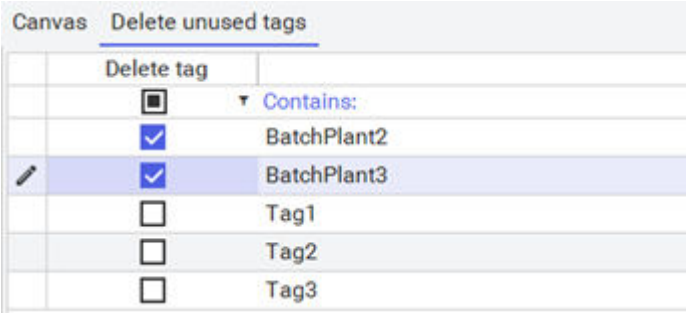
示例：

1. 创建 $BatchPlantType 的两个实例，并将它们命名为 BatchPlant2 和 BatchPlant3。
2. 选择**删除未使用标记**。



您可以看到可以删除未使用标记 BatchPlant2 和 BatchPlant3。BatchPlant 未显示，因为它正在使用中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



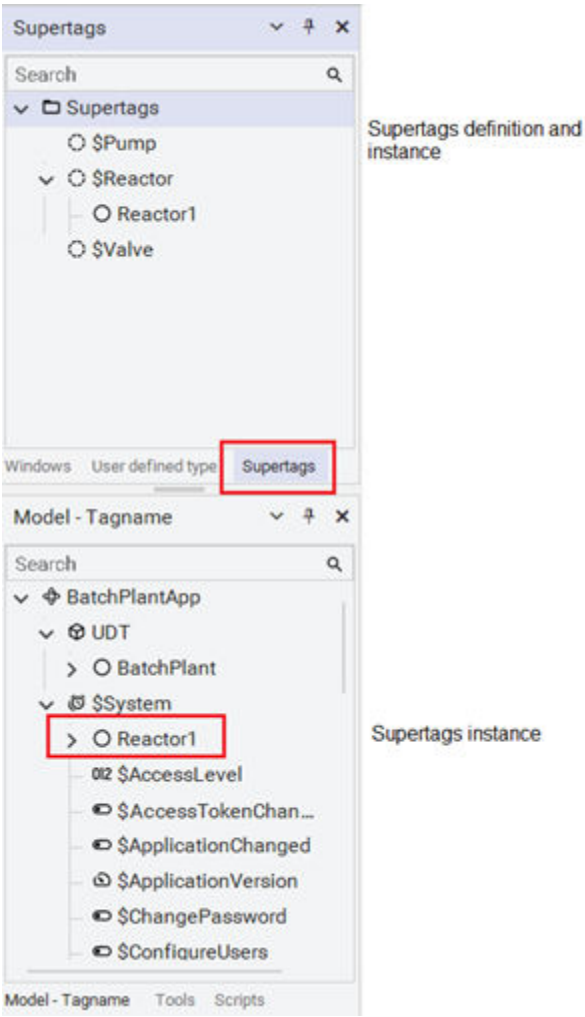
与 SuperTag 共存

UDT 可以与 SuperTag 共存。您可以创建具有相同名称的 SuperTag 和 UDT。SuperTag 实例和 UDT 实例可以在脚本、动画、报警和趋势中一起或单独工作。

共存行为如下所示：

- 缺省条件下，SuperTag 视图处于关闭状态。
  - 这适用于新应用程序以及迁移到最新发行版的应用程序。
  - 如果需要，您可以恢复到 Supertag 视图。
- SuperTag 继续使用“\”语法。
- SuperTag 的工作方式继续与 System Platform 2023 中的相同。
- SuperTag 定义和用户定义类型之间不存在名称冲突。
  - 您可以使用相同的名称创建 SuperTag 和 UDT，例如“Reactor”。
- SuperTag 实例和 UDT 实例可以在脚本、动画、报警和趋势中一起或单独工作。
- SuperTag 不支持父对象。这意味着 SuperTag 不支持“me.”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

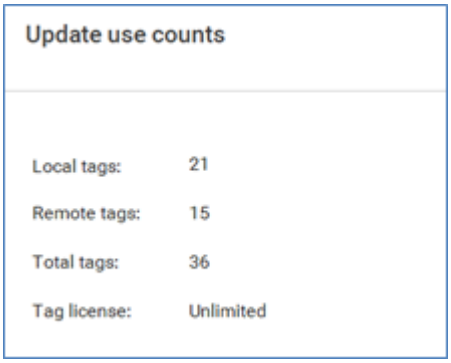


许可

实例的每个 UDT 成员都算作一个本地标记。UDT 数据类型或衍生数据类型不考虑使用任何本地标记。

示例：

在下面的示例中，本地标记的总数为 21。



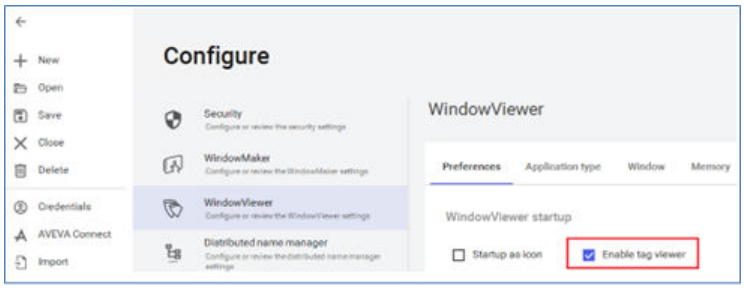
运行时标记查看器

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

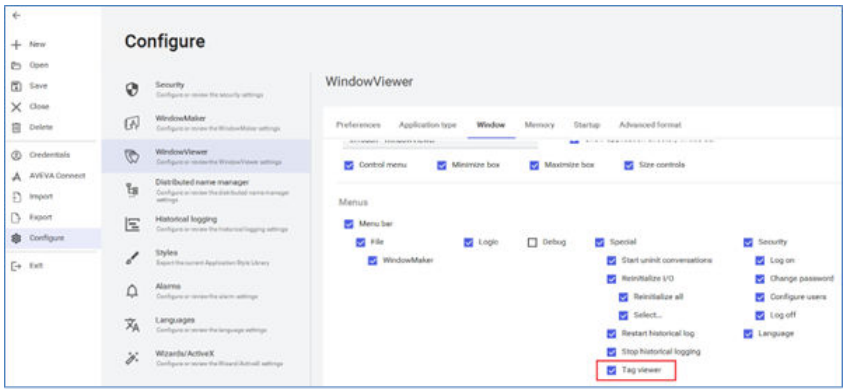
运行时标记查看器支持 UDT 成员。

在打开标记查看器之前，请确保：

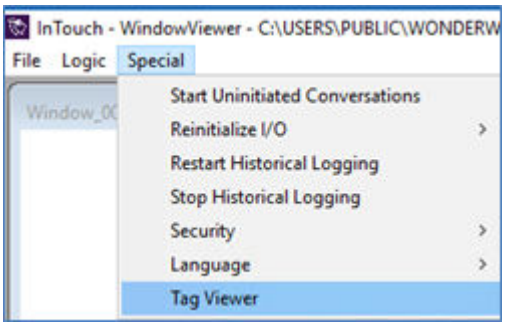
- 在文件 > 配置 > **WindowViewer** >首选项下选中启用标记查看器复选框。



- 在文件 > 配置 > **WindowViewer** >窗口下选中标记查看器复选框。

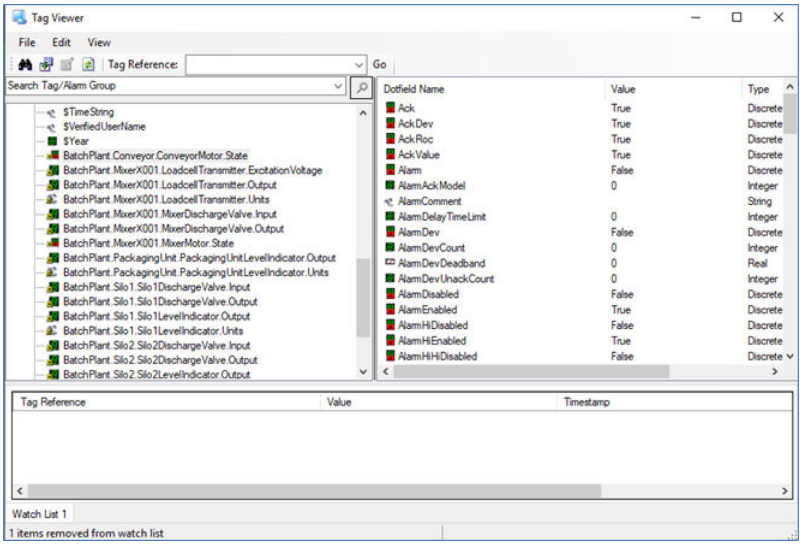


在 WindowViewer 中运行应用程序并启动标记查看器。



标记查看器显示 UDT 实例成员。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



UDT 限制

在当前发行版中，UDT 不支持以下功能：

- 在 ActiveX 控件属性中使用 UDT 标记。
- 在向导中使用 UDT 标记（按钮、时钟、框架等）。
- 在趋势中使用 UDT 标记（历史趋势）。
- 使用 UDT 标记覆盖智能符号的实例引用。
- 在 SQL Access Manager 绑定列表中使用 UDT 标记。
- 在 Recipe Manager 中使用 UDT 标记。
- 在本机用户输入动画的“最小值”和“最大值”字段中使用 UDT 标记。
- 来自 InTouch 代理对象的标记浏览器。
- 在报警注释中使用 UDT 标记。
- 在语言切换中使用 UDT 标记。
- 标记的保留属性。
- 在本机动画（如离散警报和模拟警报）中使用 UDT

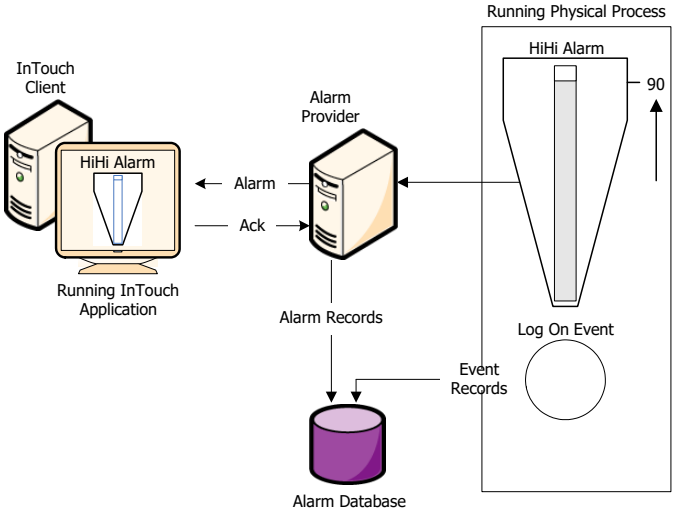
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 10 报警

通过创建可生成报警与事件的 InTouch 应用程序，可以通知操作员有关生产过程活动的状态。

- 报警向运行时操作员警告可能导致潜在问题的过程条件。通常，您设置一个在过程值超过定义的极限时触发的报警。操作员通常必须确认该报警。
- 事件代表正常的系统状态消息。通常事件在发生某种系统条件时触发，如操作员登录到 InTouch 应用程序。操作员不必确认事件。

下图显示 InTouch HMI 在应用程序运行期间如何处理报警与事件。报警与事件数据保存到报警数据库。



您可以配置任何标记来监视事件。每次标记值改变时，都有一个事件消息记录到报警系统。事件消息包含：值是如何改变的，是操作员、I/O、脚本还是系统促使了这个改变。

配置报警

要配置报警，只要给标记配置报警条件即可。

如果需要，您也可以：

- 定义报警层次结构。
- 禁用与约束报警。
- 配置报警注释。
- 配置其它报警与事件属性。

定义报警层次结构

每个 InTouch 报警都属于某个报警组。对相关报警进行分组可以便于操作员过滤、显示及确认报警。如需详细信息，请参阅报警组。

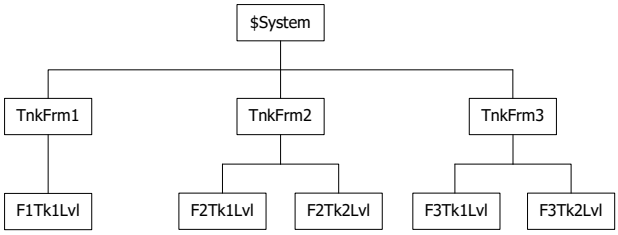
“分布式报警系统”会将报警组用作报警组列表的基础。如需有关创建“分布式报警系统”组列表的详细信息，请参阅创建报警组列表文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建报警组

在开始创建报警组之前，首先计划报警组的组织方式与所需的报警组名。通过采用统一的组命名惯例，可以给层次结构中的组实现一种逻辑上的顺序。

在下图中，对于指定给层次结构中相同级别的组的名称，注意它们的相似之处。



同样请注意，下属组名通过包含父名的一部分来引用它们的父组。例如，第三级报警组名 F1Tk1Lvl 通过在组名中包含 F1 前缀来引用它的父报警组 TnkFrm1。制订一种命名惯例，以显示层次结构中不同级别的报警组之间的父子关系。

注意：虽然在 InTouch 许可机制中，报警组不计为标记，但它们在数据库中却确实计为标记。因此，报警组加上实际标记的总数不能超出 InTouch 许可证设置的最大限制。

报警组名必须满足以下要求：

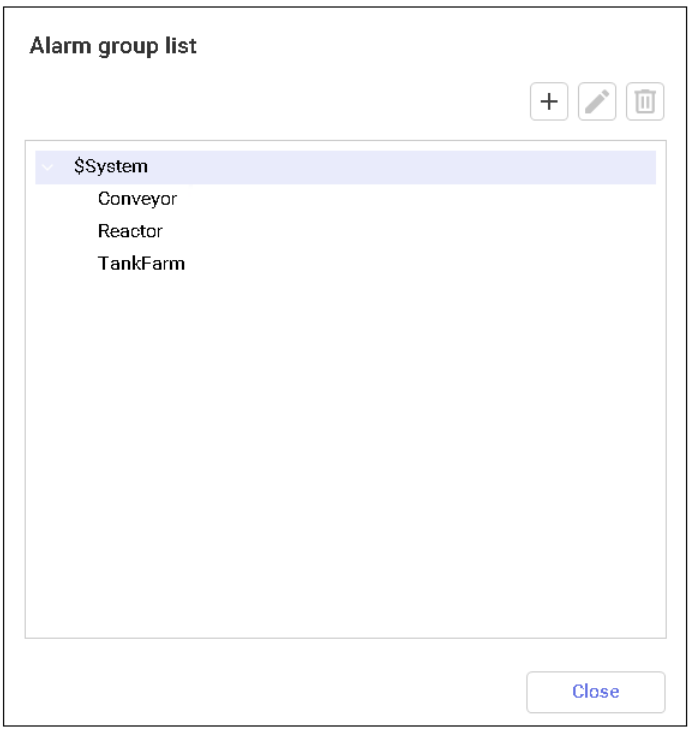
- 名称必须为 32 个字符或更少。
- 名称必须以字母数字字符（A-Z、a-z 或 0-9）开头。
- 名称从第二个字符位置开始，可以包含以下键盘字符 (@, #, $, %, &, -, _, ?, !, \)。
- 如果名称包含连字符 (-)，则必须以英文字符开头。
- 名称中不得包含空格。
- 名称至少必须包含一个英文字符。

要创建报警组

1. 在主页菜单上的报警组中，单击报警组。

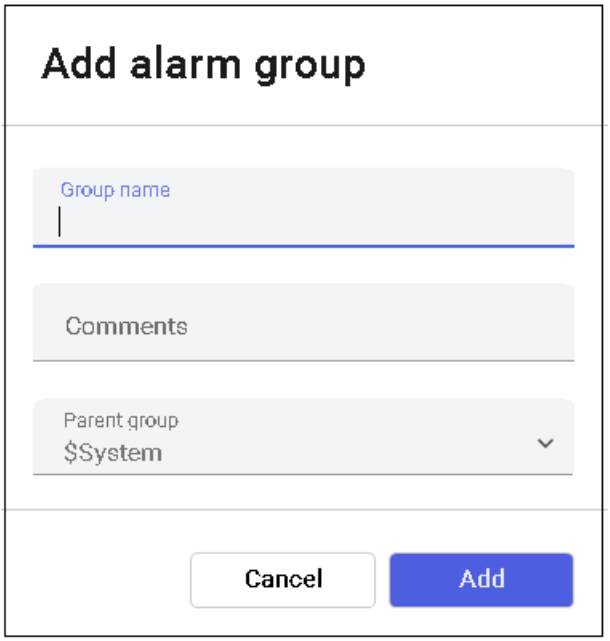
   此时出现报警组列表窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 单击 + 图标进行添加，或按 Alt+A。

   此时出现添加**报警组**对话框。



3. 在**组名**框中，输入新**报警组**的名称。
4. 在**注释**框中，为新**报警组**输入可选注释（最大长度是 49 个字符）。
5. 要将**报警组**重新指定给另一个父组：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

a. 从列表中选择父组名。如果这是为 InTouch 应用程序定义的第一个报警组，则该组自动指定给 $System 父组。

b. 从列表中选择一个新的父组。

6. 单击添加。

   报警组列表窗口会显示已添加到列表中的新报警组。

7. 单击关闭。

修改报警组

您可以修改报警组以：

- 给它重命名。
- 更改关联的注释。
- 将它重新指定给另一个组。

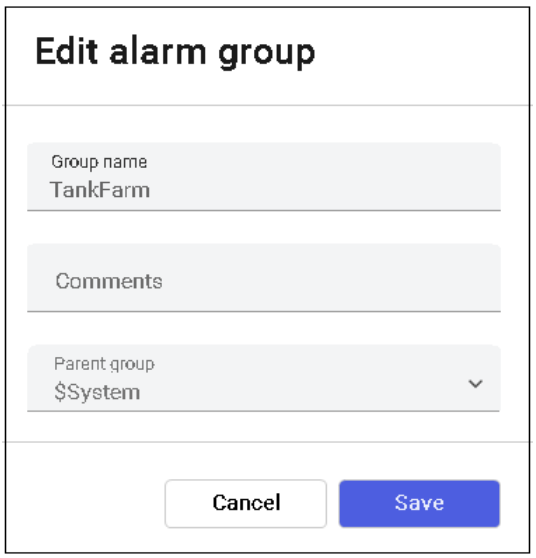
修改报警组

1. 在主页菜单上的报警组中，选择报警组。

   此时出现报警组列表窗口。

2. 突出显示要修改的报警组，然后选择“编辑”图标，或按 Alt+E。

   此时出现编辑报警组窗口。



3. 对报警组的名称或注释进行任何更改。
4. 要将报警组重新指定给另一个父组：
   a. 从父组下拉列表中选择一个新的父组。
5. 选择保存。
6. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

删除报警组

您可以删除某个报警组，并从层次结构中删除该组。属于所删除的报警组的报警与标记自动重新指定给层次结构中所删除的组的直接父组。同样，所删除的组的子组也将重新指定给所删除的组的直接父组。

删除报警组

1. 在主页菜单上的报警组中，选择报警组。

   此时出现报警组列表窗口。

2. 突出显示报警组，并选择删除图标或按 Alt+D。出现消息时，选择是。
3. 选择关闭。

为标记配置报警条件

通过指定报警类型与一个或多个报警阈值，您可以给任何标记配置报警。只要标记的值达到定义的阈值，便发生报警。标记值进入或脱离某种报警状态的任何转换都报告给“分布式报警系统”。

配置离散报警

离散报警对应于离散标记。您可以将报警的状态配置为对应于该离散标记的真（打开,是,1）状态还是假（关闭,否,0）状态。

为离散标记定义报警条件

1. 打开“标记名字典”。
2. 打开现有离散标记，或创建一个新的离散标记。
3. 在标记名字典对话框的顶部，单击报警或详细信息和报警，以显示离散报警详细资料对话框。

4. 在确认模型区域中，为标记选择报警确认模型。
   - 选择条件，确认会统计至确认时为止所有进入报警状态或子状态的转换次数。这是缺省的确认模型。
   - 选择面向事件，确认仅针对进入报警状态或子状态的特定转换；只有指最近一次事务时才会接受确认。
   - 选择扩展的摘要，确认只针对特定的转换，无论是转入报警状态、子状态，还是返回到正常状态。从正常状态的每次转换都标志着一个新的“返回正常”(RTN) 组的开始。RTN 组中的所有转换都必须单独确认，之后整个 RTN 组才会视为已确认。
5. 在报警注释框中，输入报警注释，最大长度为 131 个字符。

   注意：“报警注释”框不应包含双引号字符（"）。如果某过程使用双引号作为分隔符，那么它在获取包含双引号的报警注释时就会失败。

6. 在报警状态区域中，选择活动报警状态是离散标记的打开还是关闭值。
7. 在优先级框中，指定 1 到 999 之间的一个报警优先级。缺省优先级数字是 1，这是最高的报警优先级。
8. 作为可选项，还可以为离散报警指定一个报警约束标记。
   a. 在报警约束标记框中，单击按钮以显示选择标记对话框，其中包含一列已定义的标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

b. 在列表中突出显示标记，然后选择确定。选作约束标记的标记的名称出现在**报警约束标记**框中。

如需有关约束报警的详细信息，请参阅约束报警。

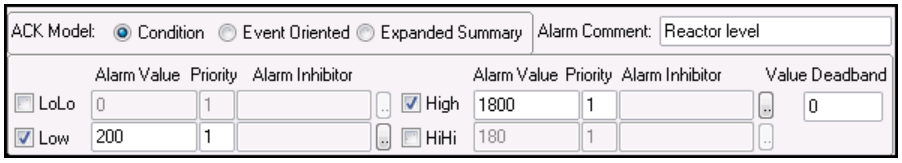
9. 选择保存。
10. 选择**关闭**以关闭**标记**名字典对话框。

配置值报警

值报警与整型或实型标记关联。您可以将报警设置成在标记值转换为超出一组预定义的阈值（从 LoLo 到 HiHi）时触发。您可以配置报警状态是否对应于该标记的任何值，以及与该报警关联的优先级。

配置值报警

1. 打开“标记名字典”。
2. 选择现有的实型或整型标记，或创建一个新的标记。
3. 在**标记名字典**对话框的顶部，单击**报警**或**详细信息和报警**，以显示报警详细资料对话框。



4. 在**确认模型**区域中，为标记选择报警确认模型。
   - **选择条件**，确认会统计至确认时为止所有进入报警状态或子状态的转换次数。这是缺省的确认模型。
   - **选择面向事件**，确认仅针对进入报警状态或子状态的特定转换。只有指最近一次事务时才会接受确认。
   - **选择扩展的摘要**，确认只针对特定的转换，无论是转入报警状态、子状态，还是返回到正常状态。从正常状态的每次转换都标志着一个新的“返回正常”(RTN) 组的开始。RTN 组中的所有转换都必须单独确认，之后整个 RTN 组才会视为已确认。
5. 在**报警注释**框中，输入缺省注释，最大长度为 131 个字符。注释指定给标记的 .AlarmComment 点域。

   注意：“报警注释”框不应包含双引号字符（"）。如果某过程使用双引号作为分隔符，那么它在获取包含双引号的报警注释时就会失败。

6. 选择报警类型（**LoLo**、**Low**、**High**、**HiHi**），以检测何时标记值超出绝对极限值。
7. 在**报警值**框中，为报警类型输入极限值。

   例如，对于 **LoLo** 与 **Low** 报警，只要标记值小于**报警值**，便存在报警条件。对于 **High** 与 **HiHi** 报警，只要标记值超出**报警值**，便发生报警。您可以给这些极限值使用实数。

8. 在**值死区**框中，输入工程单位数；标记值必须上升到报警值以上或下降到报警值以下这个数值，才能脱离报警状态。

   例如，要从报警条件下返回到正常状态，标记值不仅要返回到报警限之内，还需要返回到指定的“值死区”范围内。“值死区”可以防止由于报警的不断反复（即标记值在极限附近上下浮动，从而持续进出报警状态）造成过度报警。

9. 作为可选项，您可以为标记的报警类型（LoLo、Low、High、HiHi）指定报警约束标记。
   a. 在**报警约束标记**区域中，单击按钮以显示**选择标记**对话框，其中包含一列已定义的标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

b. 在列表中突出显示标记，然后选择确定。选作约束标记的标记的名称出现在**报警约束标记**框中。

如需有关约束标记详细信息，请参阅约束报警。

10. 选择保存。
11. 选择**关闭**以退出**标记**名字典对话框。

配置偏差报警

偏差报警与整型或实型标记关联。您可以通过以下方法来触发报警：将当前标记值与目标值进行比较，然后将差的绝对值与一个或多个极限进行比较，这些极限表示为标记值范围的百分比。

例如，以下这些值设置标记主、副偏差报警的条件：

最小值 = -1000

最大值 = 1000

副偏差百分比 = 10

主偏差百分比 = 15

目标值 = 500

将这些值用作示例，主、副偏差报警点按以下步骤进行计算：

1. 计算标记的整个值范围。

   1000 - (-1000) = 2000

2. 将标记的整个值范围乘以主、副偏差百分比。

   2000 x 0.10 = 200 = 副偏差限

   2000 x 0.15 = 300 = 主偏差限

3. 在目标值上加上与减去主、副偏差限。

   500 - 200 = 300 = 副偏差下限

   500 + 200 = 700 = 副偏差上限

   500 - 300 = 200 = 主偏差下限

   500 + 300 = 800 = 主偏差上限

配置偏差报警

1. 打开“标记名字典”。
2. 选择现有的实型或整型标记，或创建一个新的标记。
3. 在**标记名字典**对话框的顶部，单击**报警**或**详细信息和报警**，以显示报警详细资料对话框。

4. 选择要使用的偏差（主偏差与副偏差）报警类型，它用于检测何时模拟型标记的值在指定目标值的主偏差或副偏差范围内。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5. 在 **% 偏差**框中，输入触发主偏差或副偏差报警条件时模拟标记要偏离目标值的百分比。它表示为标记范围的百分比。对于 I/O 标记，范围由标记的详细资料对话框中输入的最小工程单位与最大工程单位值定义。对于内存标记，范围由最小值与最大值定义。
6. 在目标框中，输入标记参考值，主、副偏差百分比都基于这个参考值。
7. 在偏差死区百分比框中，输入偏差百分比；标记值必须下降到极限值以下这个百分比，标记才能脱离报警条件。
8. 选择保存。
9. 选择**关闭**以关闭**标记**名字典对话框。

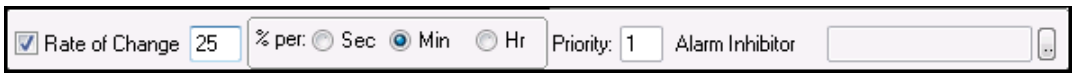
配置变化率报警

在测量间隔内报警的绝对值超过指定极限时，变化率报警能检测到这个情况。只要标记值发生变化，就会对标记进行测试以确定是否发出变化率报警。变化率使用前一个标记值、前一次变化的时间、当前标记值以及当前时间来计算。

计算出的一段时间内的变化率将与为标记指定的变化率百分比极限进行比较。如果计算出的变化率大于百分比极限，则为标记设置报警条件。变化率报警持续有效，直至标记变化的速率小于报警极限。

配置变化率报警

1. 打开“标记名字典”。
2. 选择现有的实型或整型标记，或创建一个新的标记。
3. 在**标记名字典**对话框的顶部，单击**报警**或**详细信息和报警**，以显示报警详细资料对话框。下图仅显示适用于变化率报警的那些选项。



4. 选择**变化率**框。
5. 在 **%** 框中，输入允许的最大变化百分比极限。
6. 选择秒、分或**时**作为时间间隔单位。
7. 在**优先级**框中，输入 1 与 999 之间的一个数字，以设置报警优先级。
8. 作为可选项，您可以为变化率报警指定一个报警约束标记。
   a. 在**报警约束标记**区域中，单击按钮以显示**选择标记**对话框，其中包含一列已定义的标记。
   b. 在列表中突出显示标记，然后选择确定。选作约束标记的标记的名称出现在**报警约束标记**框中。

   如需有关约束报警的详细信息，请参阅约束报警。
9. 选择保存。
10. 选择**关闭**以关闭**标记**名字典对话框。

禁用报警

您可以使用 .AlarmEnabled 或 AlarmDisabled 点域一次性禁用或启用标记的所有报警。对于具有子状态的报警，每个子状态都可以单独禁用。例如，模拟值报警可以启用 Hi 而禁用 HiHi。

在运行时，“报警供应器”不为禁用的报警或子状态生成报警。在运行时可以更改报警的禁用或启用状态。

只要报警从禁用转换为启用，检查逻辑就会确定“报警供应器”是否应将该项目置于报警状态。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果在项目处于报警状态时，报警变为禁用或有效约束状态，该项目将强制转换到一个不同的（有效）状态。具体的状态应取决于当前有哪些可用的状态，以及它们是否也被禁用。此活动由“报警供应器”根据报警类型与极限值进行处理。

约束报警

作为可选项，您可以为每个报警或其子状态指定一个报警约束标记，以防止报警转换为活动状态。

- 约束标记值变为且保持为“真”（非零或非空）时，报警便会被约束。
- 类似地，报警约束标记变为且保持为“假”（零或空）时，报警不会被约束。

您只能在 WindowMaker 中更改约束标记。在运行时，可以更改约束标记的值。

您可以为单独的报警子状态指定约束标记。每个子状态可以由不同的标记来约束，某些子状态也可以不指定约束标记。

受约束的报警（对应的标记值为“真”）不等待确认。如果报警有子状态，它仅可以在仍旧可用的子状态上等待确认。

报警或子状态可以单独禁用、约束，也可以同时进行禁用与约束。仅当报警既已启用且未被有效约束时，该报警才可以变为活动状态。

如果某个报警或子状态未指定约束标记，则效果相当于它有一个总是为“假”的约束标记。

只要转换导致报警脱离有效约束状态，检查逻辑便会确定 InTouch 是否应该将该项放入报警状态。

如果在项目处于报警状态时报警变为有效约束状态，该项目必须强制转换到一个不同的（有效）状态。具体的状态应取决于当前有哪些可用的状态，以及它们是否也被禁用或有效约束。此活动由 InTouch 根据报警类型、极限值等进行处理。

如果报警（或报警子状态）在等待确认时变为有效约束状态，则该项目必须强行转换到一个不同的（有效）状态。与确定项目是否报警一样，InTouch 还必须确定应该处于什么状态。

报警约束标记包含在使用计数与许可证限制中。

使用以下只读标记点域获取报警约束标记的名称：

- AlarmDscInhibitor
- AlarmLoLoInhibitor
- AlarmLoInhibitor
- AlarmHiHiInhibitor
- AlarmHiInhibitor
- AlarmMajDevInhibitor
- AlarmMinDevInhibitor
- AlarmRocInhibitor

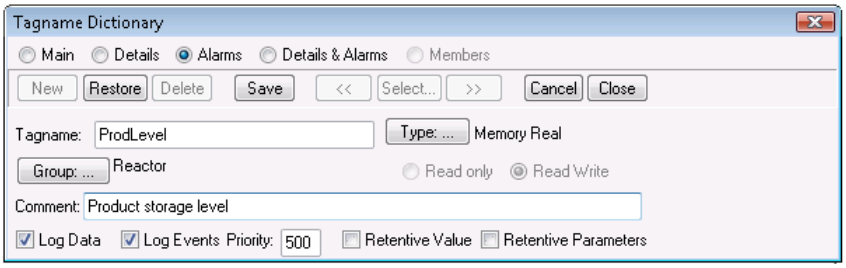
这些域返回标记的名称。因此，您可以在 InTouch QuickScript 的间接标记引用中使用该名称来获取报警约束标记的当前值，或更改报警约束标记的值。这样可以在运行时强制启用或有效约束报警组。

为单独的标记设置事件属性

定义标记以进行事件监视时，只要标记的值发生变化，便会有一条事件消息记录到报警系统。事件消息记录值如何改变。例如，是操作员、I/O、QuickScript，还是系统促使了改变。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1. 打开“标记名字典”。
2. 选择现有的标记，或创建一个新的标记，其数据将作为事件进行记录。
3. 选择记录事件。此时“优先级”框变为可用。为“优先级”输入的值确定标记的事件优先级别。



4. 在优先级框中，指定 1 到 999 之间的一个数字作为事件优先级。1 的事件优先级最高，999 的优先级最低。
5. 选择保存。
6. 选择关闭以关闭标记名字典对话框。

配置报警与事件的全局设置

您可以配置以下全局设置，这些设置应用于应用程序生成的所有报警与事件：

- 内部报警内存（缓冲区）大小。
- 报警返回到正常状态是否意味着确认。如需有关详细信息，请参阅标记值恢复到正常时使用自动确认。
- 事件记录。
- 报警启用状态在重新启动 WindowViewer 时是否保留。
- 是否将报警确认注释用作一般报警注释。如需有关详细信息，请参阅使用报警与确认注释。
- 是否在报警客户端控件网格中显示 LATCHED 报警。如需详细信息，请参阅启用锁存状态。

配置报警缓冲区大小

“分布式报警系统”中的通讯很大程度上是由在节点之间发送的报警查询与报警记录组成。在节点内部，报警查询与记录存储在 InTouch 内部报警内存（也称为报警缓冲区）中，以最大程度减少网络流量。报警缓冲区大小是节点可以为摘要或历史报警查询存储的最大报警数。报警缓冲区会删除最旧的记录以便为新记录腾出空间。

只有存储在内存中的报警事件才可以显示在应用程序窗口中。如果 InTouch 应用程序不显示任何报警状态，则可以将缓冲区大小设置为 1 以节省节点内存。

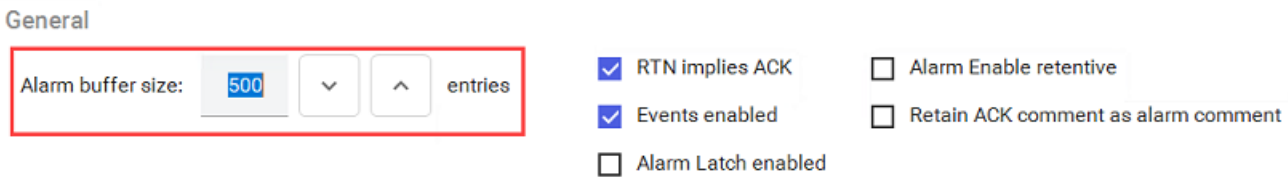
将一个很大的数值指定给报警缓冲区可能会影响节点性能。对于“分布式报警系统”，我们建议使用缺省值 500。

配置报警缓冲区大小

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择报警。

   此时出现报警配置屏幕。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**报警缓冲区大小**框中，输入可存储在内存报警缓冲区中用于摘要或历史查询的最大报警条目数。
4. 选择保存。

启用事件

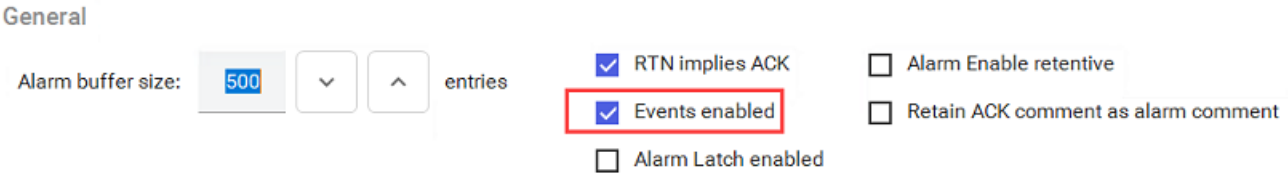
您可以在应用程序中启用事件记录功能。事件代表由操作员动作、QuickScript 或 I/O 引起且已被认识到的应用程序数据的变化。

在标记的相关事件可以存储到内部报警内存或记录到报警数据库之前，必须从“标记名字典”中设置标记的**记录事件**属性。如需有关为标记指定事件记录的详细信息，请参阅为单独的标记设置事件属性。

启用事件

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择报警。

   此时出现报警配置屏幕。



3. 选中已启用事件复选框，以记录 InTouch 应用程序运行期间发生的所有事件。
4. 选择保存。

保留报警启用功能

您可以选择保留指定给标记的 . **AlarmEnabled** 点域的当前值；在停止 InTouch 应用程序，然后再重新启动时，该值会保留下来。

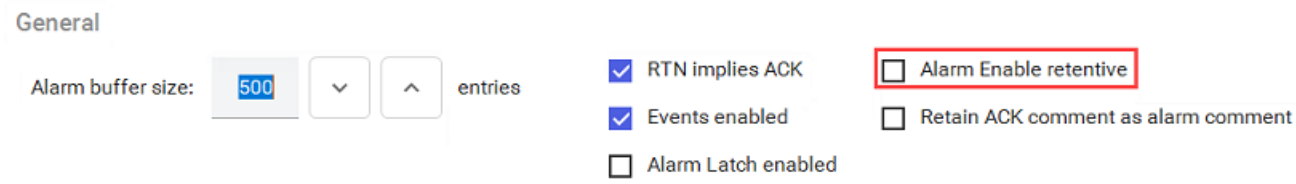
指定给. **AlarmEnabled** 点域的值可以打开或关闭报警与事件记录功能。.**AlarmEnabled** 点域可以指定给标记或报警组。.**AlarmEnabled** 点域指定给报警组时，它确定是否记录与指定的报警组中的标记关联的所有报警。

保留报警启用功能

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择报警。

   此时出现报警屏幕。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 选中**报警事件保留复选**框，以在重新启动 InTouch 应用程序时，保留 **.AlarmEnabled** 点域的当前状态作为初始值。
4. 选择保存。

启用锁存状态

您可以启用 LATCHED 状态，以在报警客户端控件网格中查看 LATCHED 报警。在以下情况下，报警将进入 LATCHED 状态：

- 确认 UNACK_RTN 状态的报警。

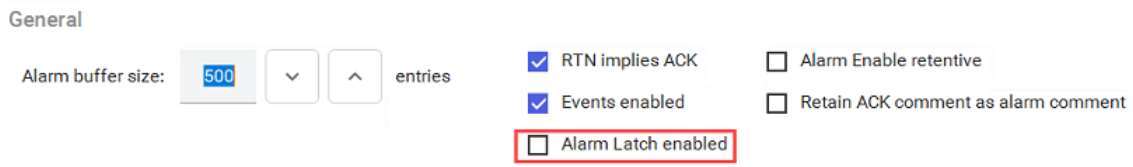
  或

- 报警从 ACK 状态返回到正常 (RTN) 状态。

启用锁存状态：

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择报警。

   此时出现报警配置屏幕。



3. 选中**启用报警锁存复选**框，以在报警客户端控件网格中查看 LATCHED 报警。
4. 选择保存。

创建报警组列表文件

您可以使用“分布式名称管理器”创建报警组列表。然后，将现有的报警组从本地与远程节点添加到该列表中。

下表显示从“分布式名称管理器”中指定报警组的语法。

节点	报警组语法
本地	\InTouch!Group_Name 或 Group_Name

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

节点	报警组语法
远程	\\Node_Name\InTouch!Group_Name 或 Node_Name.Group_Name

对于这些示例，Node_Name 是 InTouch 远程节点的名称。Group_Name 是报警组的名称。如果报警组是在定义报警组列表的本地节点上定义的，则只需输入报警组名，前面加上英文句点。例如，.Group_Name。

创建报警组列表

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择分布式名称管理器。

   此时出现分布式名称管理器配置屏幕。
3. 在分布式报警选项卡中，选择“添加”(+) 图标。

   此时出现添加分布式报警对话框。

4. 在成员框中，输入要包含在查询中的 InTouch 节点与报警组的列表。

   通过使用“标准的组输入”语法，或使用带英文句点的简短输入格式，可以输入节点名与报警组名。在保存报警组列表时，简短输入格式转换为“标准的组输入”格式。

   注意：Node.Group 与 .Group 语法仅可以用在此配置对话框中。在报警显示配置或任何报警 QuickScript 函数中，它是无效的。
5. 选择添加将列表添加到报警组文件中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时成员的语法会自动转换。

6. 选择保存。
7. 将报警组列表的名称添加到 Alarm Viewer 控件的查询中。此时，Alarm Viewer 控件显示列表中指定的所有组的报警。

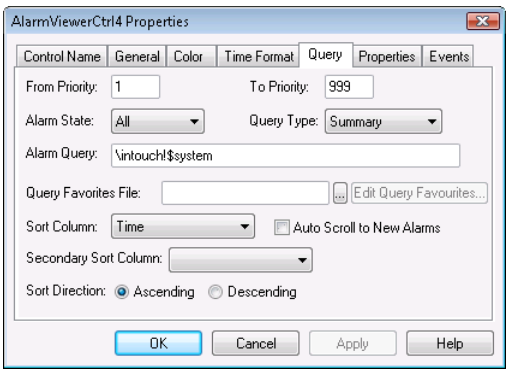
报警查询

报警查询检索以下内容之一：

- InTouch 内部报警内存或报警数据库中的报警与事件（历史报警）。
- InTouch 内部报警内存中的当前报警（摘要报警）。

配置 InTouch 报警 ActiveX 控件时，可以指定查询来源。您也可以选择用于过滤查询结果的查询选项。

下图显示 Alarm Viewer ActiveX 控件的查询选项卡。



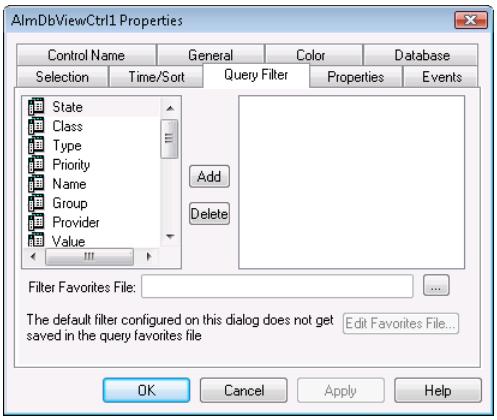
在本例中，您创建一个报警显示对象，显示根据以下准则选择的报警数据：

- 报警优先级 (1-999)
- 报警状态（全部、已确认或未确认）
- 查询类型（摘要或历史）
- 报警组（本地或远程数据源）

您可以将查询保存到名称为“query favorites”的 .xml 文件。在运行时，通过运行另一个查询（使用保存到文件中的选择准则），可以使用新的报警数据来更新报警显示。

其它 InTouch 报警 ActiveX 控件提供更全面的查询准则。下图显示 Alarm DB Viewer 控件的查询过滤器选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



通过从对话框左侧窗格显示的列表中选择报警或事件属性，您可以构建查询。然后，您给所选的属性指定一个值。最后，您可以通过使用布尔运算符将各个属性组合到一起来设置查询过滤器条件。

您可以编写包含查询函数或点域的 QuickScript 从报警内存中选择报警与事件记录。以下 Alarm Viewer 控件语句使用 ApplyQuery() 方法查询报警内存。

```
#AlarmViewerCtrl1.ApplyQuery ("\InTouch!$System",500,600,"All", "Historical");
```

此语句检索由“\InTouch!$System”查询指定的所有历史报警，优先级在 500 到 600 之间。所选的报警记录出现在 Alarm Viewer 控件显示中。

示例报警查询

本地节点上的报警查询遵循此语法：

\Provider!AlarmGroup

例如：

\InTouch!$System

远程节点使用以下查询语法：

\\节点名\Provider!AlarmGroup

例如，在 MyNode1 节点上：

\\MyNode1\InTouch!$System

如果在查询 Galaxy 的报警时选中了使用 **"Galaxy_<Galaxy name>"** 而非 **"Galaxy"** 注册复选框，请使用以下语法。此语法从特定计算机上特定区域中对象的特定报警名获取报警。报警名可以是属性名或原始的报警名。Galaxy 名显示在报警窗口的“供应器”列。

\\NodeName\Galaxy_GalaxyName!AreaName!ObjectName.AlarmName

以下语法从特定的区域获取所有报警：

\Galaxy_GalaxyName!AreaName

以下语法从两个区域获取报警：

\Galaxy_GalaxyName!Area1 \Galaxy_GalaxyName!Area2

以下语法从指定计算机节点（缺省情况下）的“平台”中的指定“区域”获取所有报警：

\\NodeName\Galaxy_GalaxyName!AreaName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您也可以使用单个通配符来匹配指定区域内的报警名。以下语法从“AreaName”区域中以字符“Tank”开头的所有对象中获取所有报警：

\Galaxy_GalaxyName!AreaName!Tank*

以下语法从“AreaName”区域的所有对象中获取名称为“Hi”的所有报警：

\Galaxy_GalaxyName!AreaName!*.Hi

注意：如果您要从 Alarmy 而非 InTouch 查询报警，则组名称允许的字符数为 128。

获取更多有关 InTouch 查询的信息

下表列出一些参考资料，供您获取有关每个 InTouch 查询来源的查询的详细信息。

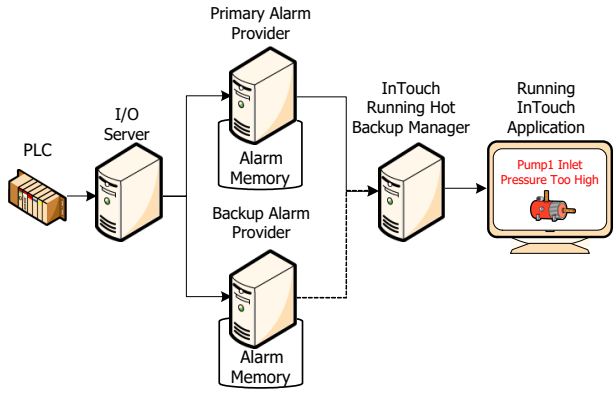
查询来源	请参阅 。
Alarm DB Logger Manager	配置所要记录的报警
Alarm Printer 实用程序	配置所要打印的报警
Alarm Viewer 控件	配置要显示的报警
Alarm Tree Viewer 控件	配置要显示的供应器与组
Alarm Pareto 控件	配置要分析的报警
分布式报警显示对象	配置要显示的报警
Hot Backup Manager	创建报警记录映射文件

通过冗余报警配置增强工厂安全性

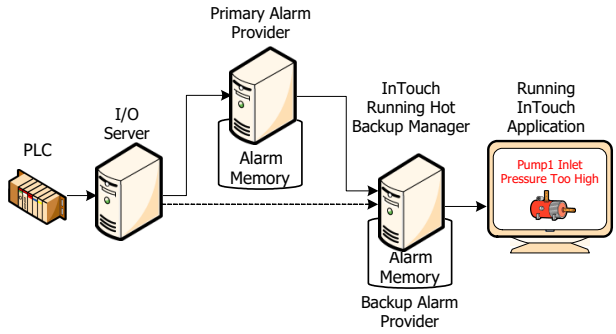
“InTouch 分布式报警”系统发出通知，并从网络中的远程节点上运行的应用程序接收报警确认。报警供应器应用程序将报警数据存储在它们的内存中。报警接收器应用程序作为客户端在其它节点上的运行，可以远程查询、显示及确认报警供应器中的报警。

您可以使用 Alarm Hot Backup Manager 来创建重复的报警供应器。下图显示 Hot Backup Manager 如何将当前的辅助报警储备库用作备份供应器。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您也可以在相同的节点上运行 Hot Backup Manager 与备份供应器，如下图所示：



理解热备份

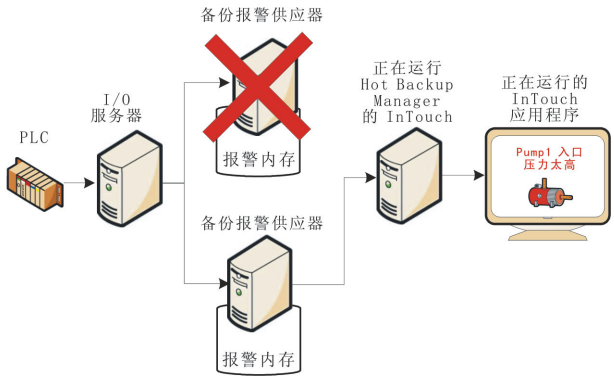
热备份提供一个名称（热备份对名），指向两个报警供应器，即主报警供应器与备份报警供应器（热备份对）。InTouch HMI 的报警接收器可以引用这个热备份对名，并从主报警供应器或备份报警供应器中检索报警。下列报警客户端支持查询热备份对：

- InTouch HMI Alarm DB Logger Manager
- InTouch HMI Alarm Printer
- InTouch HMI Alarm Viewer 控件
- 报警客户端控件

如果这两个供应器节点都正常工作，则报警接收器从主供应器接收报警数据。但如果主供应器发生故障，则报警接收器从备份供应器接收报警数据。您还可以配置 Hot Backup Manager 以继续使用备份供应器，即使主供应器可用也不例外。如果未选择此备份选项，则当主供应器可用时，Backup Manager 会自动切换到主供应器。如需详细信息，请参阅创建热备份对。

下图显示报警接收器如何在主报警供应器发生故障后继续接收报警。报警接收器仍然引用热备份对，但是由备份供应器提供报警数据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



指定热备份的报警供应器

您可以指定以下任何一个报警供应器：

- InTouch

  如果指定 InTouch 作为报警供应器，那么主节点和备份节点的供应器都必须是 InTouch。不过，备份节点的报警组可以与主节点的报警组不同。

- Galaxy

  要指定 Galaxy 或 Galaxy_<Galaxy 名> 报警供应器，您必须在 IDE 中配置 AppEngine 冗余。

- Galaxy_<Galaxy 名>

  Galaxy 名的有效字符为字母数字以及特殊字符 $、# 和 _。

  如果指定 Galaxy_<Galaxy 名>，那么主节点和备份节点的 Galaxy 名必须相同。

  如果指定 Galaxy 或 Galaxy_<Galaxy 名> 作为报警供应器，那么备份节点的供应器可以是 Galaxy 或 Galaxy_<Galaxy 名>。

  备份节点的报警组必须与主节点的报警组相同。

注意：系统不支持从两个不同 Galaxy 生成的报警热备份对。

关于 Hot Backup Manager

Hot Backup Manager 对主供应器与备份供应器之间的报警确认进行同步。如果主供应器上的某个报警得到确认，则备份供应器上的相同报警也同时会得到确认。

Hot Backup Manager：

- 提供用于创建备份对的配置实用程序。
- 提供某个配置实用程序，用于在热备份对的主供应器与备份供应器之间映射报警记录。
- 同步 InTouch 报警供应器备份对之间的报警确认。
- 在“分布式报警系统”启动与停止时，在属于一个“热备份”系统的所有节点之间建立通讯。

在 TSE 会话中使用 Alarm Hot Backup Manager

您只能在 WindowMaker 的不同 Terminal Services Edition (TSE) 中使用**工具 -> 应用程序**启动一个 Alarm Hot Backup Manager。

您可以在每个 TSE 会话中使用**开始 -> 程序**快捷方式启动一个 Alarm Hot Backup Manager。后保存的配置将会覆盖前面的配置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

TSE 会话中支持运行时查询报警热备份对。

配置热备份对

Alarm Hot Backup Manager 使用运行供应器应用程序的两个主机节点来创建一个备份对。您可以从 WindowMaker 中启动 Hot Backup Manager。要配置热备份对：

- 创建热备份对。
- 设置报警记录的关键码域。
- 映射报警记录关键码域。
- 将报警记录映射导入 Hot Backup Manager。

创建热备份对

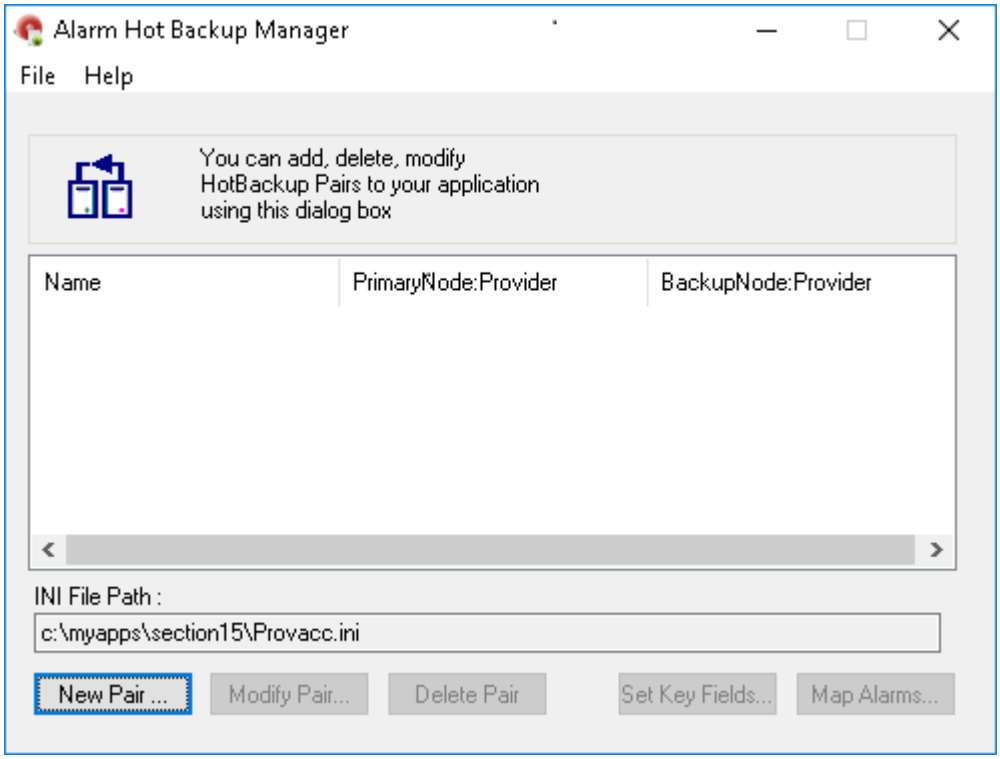
要创建热备份对，您可以：

- 给热备份对指定一个名称。
- 确定主报警供应器。
- 确定备份报警供应器。

您也可以指定一个包含配置信息的 Provacc.ini 文件。

配置热备份对

1. 打开 Alarm Hot Backup Manager。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm Hot Backup Manager**。

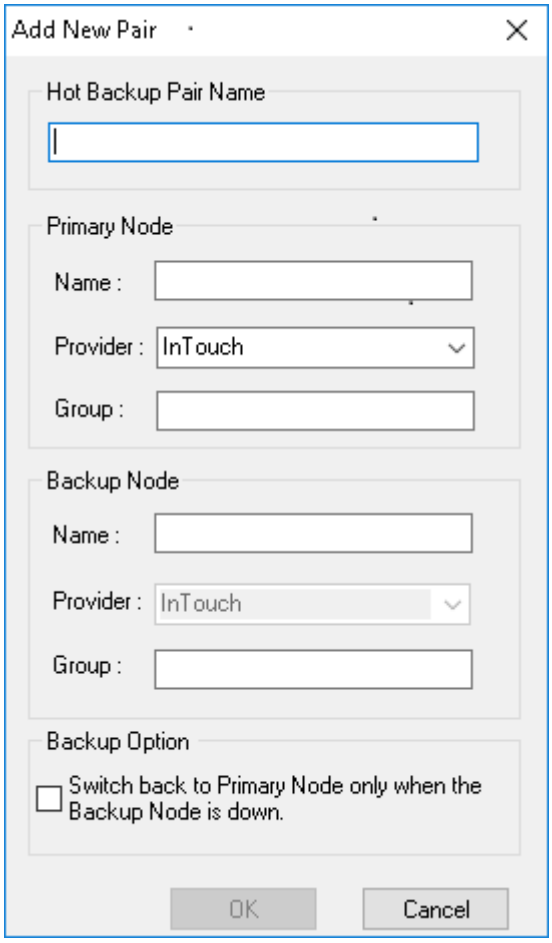


© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 在文件菜单上，选择打开。突出显示 Provacc.ini 文件，然后选择确定。

   缺省条件下，Alarm Hot Backup Manager 会在最近打开的 InTouch 应用程序文件夹中查找 Provacc.ini 文件。您应该使用 InTouch 应用程序文件夹中的 Provacc.ini 文件。否则，您可以在另一个指定的文件夹位置创建一份 Provacc.ini 文件，然后选择它供 Hot Backup Manager 使用。

3. 选择新对。此时出现添加新对对话框。



4. 在热备份对名框中，为新备份对输入唯一的名称。

   对名的最大长度为 32 个字母数字字符。您可以在对名中使用美元符号 ($)、井号 (#) 和下划线 (_)。

5. 在主节点区域中，配置主节点。执行以下操作：

   a. 在名称框中，输入运行主供应器应用程序的计算机的节点名。节点名对于 Hot Backup Manager 而言必须是唯一的。如果输入一个不存在的节点名，或节点已经用在其它热备份对中，则出现一条错误消息。

   b. 在供应器框中，从下拉列表中选择报警供应器。缺省值为 InTouch。

   c. 在组框中，输入从主供应器查询报警的报警组的名称。

6. 在备份节点区域中，配置备份节点。执行以下操作：

   a. 在名称框中，输入运行备份供应器应用程序的计算机的节点名。这可以是运行 Hot Backup Manager 的相同节点。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

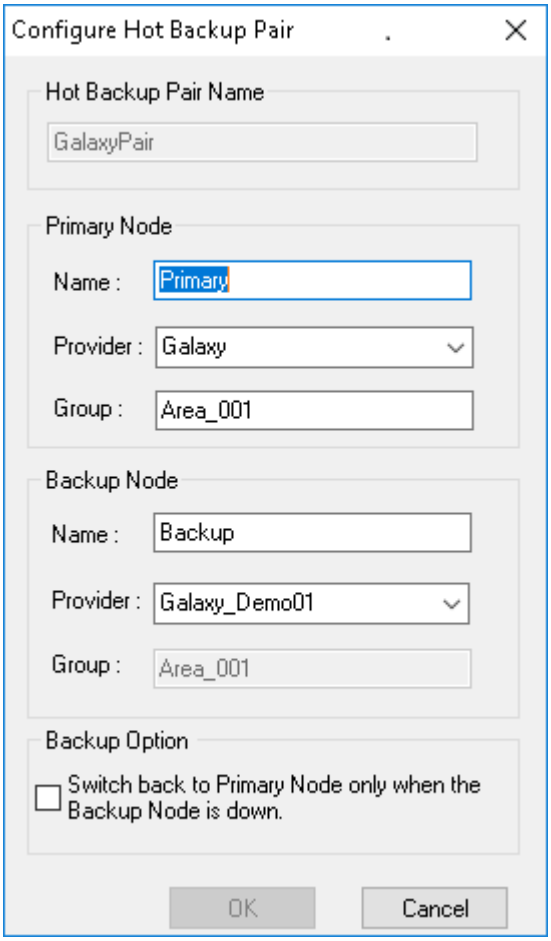
b. 在**供应器**框中，如果指定了 Galaxy 或 Galaxy_<Galaxy 名> **报警供应器**，那么需**选择备**份供应器。

如果指定了 Galaxy 或 Galaxy_<Galaxy 名> 作为主节点报警供应器，那么备份节点供应器必须是 Galaxy 或 Galaxy_<Galaxy 名>。

如果指定了 InTouch 作为主节点报警供应器，则备份节点供应器必须是 InTouch。

c. 在**组**框中，输入从备份供应器中查询报警的报警组的名称。

如果您指定了 Galaxy 或 Galaxy_<Galaxy 名> 报警供应器，那么备份节点组必须与主节点组相同，且无法编辑。



7. 要在主节点可用时继续使用备份节点，请选择仅当备份节点关闭时切换回主节点。（此选项尚未在非英语操作系统上进行测试。）

缺省条件下，该选项处于未选择状态。

8. 选择确定。
9. 在文件菜单上，选择保存。
10. 重新启动 WindowMaker。

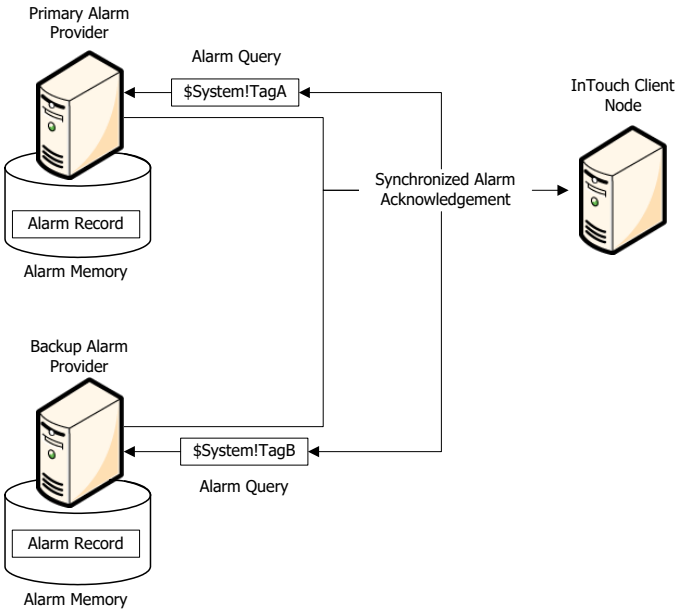
设置热备份对的报警关键码域

要同步主供应器与备份供应器之间的报警确认，必须确定标记报警记录字段的组合。此字段组合为每个供应器的当前报警储备库中存储的成对报警记录生成一个唯一的映射关键码。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您可以仅针对 InTouch 报警供应器设置报警关键码域并映射报警。

下图显示标准查询中基于报警记录字段的同步报警确认请求。



映射关键码可以是设计时与运行时报警记录的组合。设计时报警记录基于从“标记名字典”中定义标记时的报警属性。

例如，报警名字段在设计时是已知的，因为它采用主节点应用程序与备份节点应用程序中定义的标记名。QuickScript 或操作员动作定义或修改在应用程序运行期间作为记录来存储的报警属性。

您可以使用设计时或运行时报警记录字段的任意组合来创建映射关键码。映射关键码必须仅从每个供应器的当前报警储备库中选择一条记录。关键码域必须创建唯一的查询。

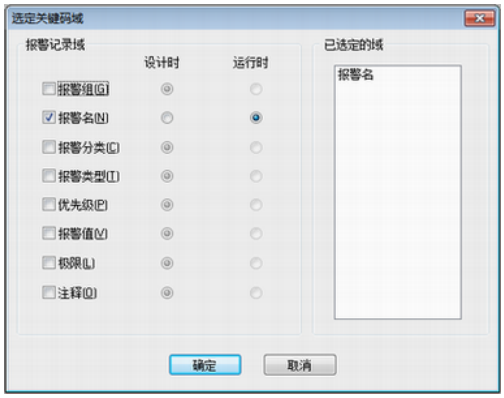
您可以使用 Hot Backup Manager 从报警记录字段中创建映射关键码列表。

注意：如果指定 Galaxy 或 Galaxy_<GalaxyName> 报警供应器，那么会禁用**设置关键码域**和**映射报警选项**。

为热备份对创建报警字段映射列表

1. 打开 Alarm Hot Backup Manager。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm Hot Backup Manager**。
2. 从列表中选择一个热备份对。
3. 选择设置关键码域。此时出现选定关键码域对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**报警记录域**区域中，**选择**要包含在映射**关键码**列表中的**报警记录**字段。

   所**选**的**报警记录**字段出**现**在**已选定的域**列表框中。

5. **为**所**选**的**报警记录**字段**选择设计时**或**运行时**。
6. **选择确定**。
7. 重新启**动** WindowMaker。

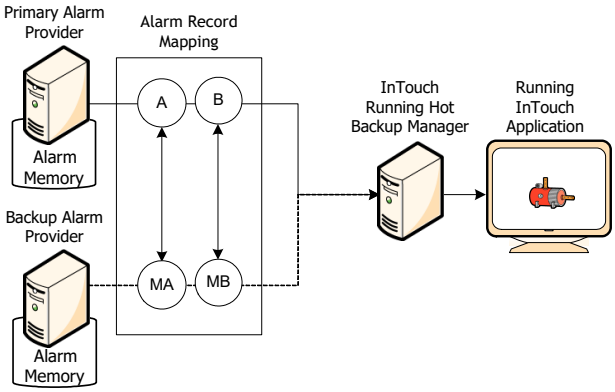
创建报警记录映射文件

只要主**报警供应**器与**备**份**报警供应**器在运行不同的**应**用程序，便必**须**映射**热备**份**对**的**报警记录**。**报警记录**映射在主供**应**器与**备**份供**应**器的不同**报警记录**之**间**建立起**对应关**系。例如，您可以基于指定的 InTouch **标记**名来映射**报警记录**。尽管它**们**的名称可能不同，但是在两个供**应**器**报警储备库**之**间**，**这**些**报警记录**在**逻辑**上是一致的。

注意：如果主供**应**器与**备**份供**应**器正在运行相同的**应**用程序，**则**不需要**创**建**报警记录**映射文件。如果没有提供映射文件，**则**“分布式**报警**系**统**”假**设**主供**应**器与**备**份供**应**器正在运行具有相同**报警记录**的相同**应**用程序。

映射可用于在运行不同**应**用程序的供**应**器之**间进**行**报警**确**认**的同**步**。“分布式**报警**系**统**”确**认**供**应**器上的**报警时**，它同**样**知道要确**认**另一个供**应**器上的哪个**报警**。

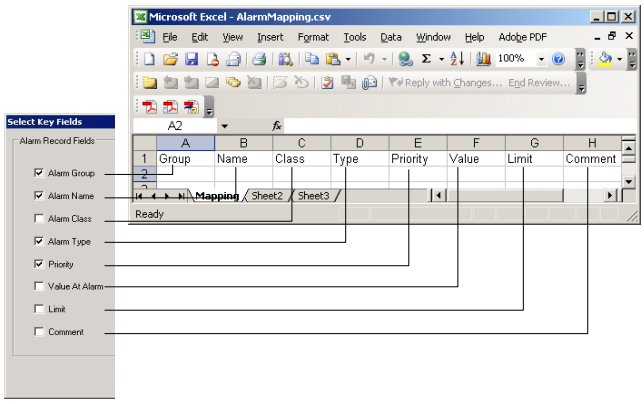
下**图显**示某个**热备**份**对**的两个供**应**器之**间**的**报警记录**映射。在本例中，主供**应**器的 A、B **报警记录**映射**为备**份供**应**器 MA 与 MB 中相**应**的**报警记录**。



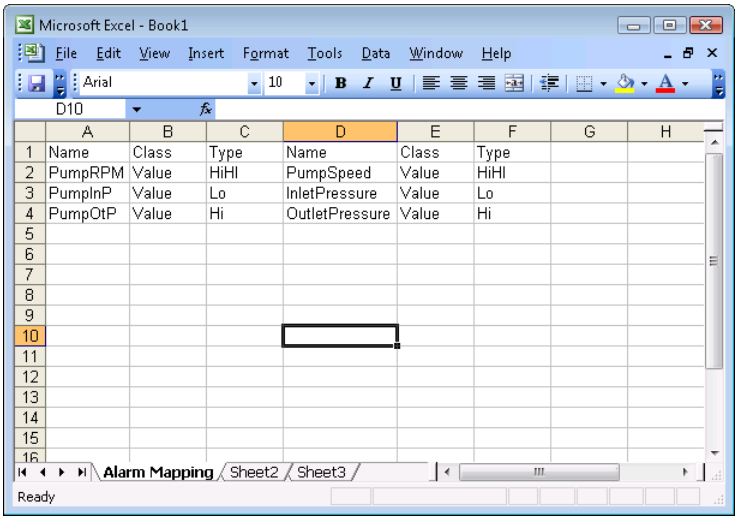
Hot Backup Manager 从使用 Microsoft Excel 或文本**编辑**器（如“**记**事本”）**创**建的逗号分隔**值** (CSV) 文件中**导入报警记录**映射。映射文件包含一个排序**过**的**报警记录**字段列表，**这**些字段**关联**到主供**应**器与**备**份供**应**器的相**应报警记录**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您必须将标记报警记录字段指定为映射文件的标题。文件中标题的顺序必须与选定关键码域对话框中显示的报警记录字段匹配。下图显示某个 Excel 文件的列标题，其顺序与选定关键码域对话框中报警记录字段的顺序匹配。



您可以创建一个映射文件，其中仅包含用于生成映射关键码的所选报警记录字段的标题。下图显示一个仅包含“名称”、“分类”以及“类型”等标题的 Excel 文件。添加标题时，它们的顺序必须总是与选定关键码域对话框中报警记录字段的顺序匹配。



在左侧的列集合中，指定主供应器的报警记录字段。同样地，在右侧的列集合中指定备份供应器的相同记录。

映射文件列标题	指定给报警记录字段的值
组	指定了该标记的报警组的名称。报警组名不得包含空格。
名称	映射其报警记录的标记的名称。标记名不得包含空格。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

映射文件列标题	指定给报警记录字段的值
类	指定给该标记的报警的分类。 可能的“分类”值有： • VALUE，代表值报警。 • DEV，代表偏差报警。 • ROC，代表变化率报警。 • DSC，代表离散报警。
类型	与报警类关联的报警条件的类型。 • 对于值报警，有 LOLO、LO、HI 以及 HIHI • 对于偏差报警，有 MinDev 与 MajDev • 对于变化率报警，有 ROC • 对于离散报警，有 DSC
优先级	指定给报警条件的优先级。优先级必须是 1 到 999 之间的一个数字。
值	请参阅以下备注。
限制	请参阅以下备注。
注释	请参阅以下备注。

“值”、“报警限”以及“注释”列：

- 特定节点中特定记录的“分类”或“类型”值未知时，“值”与“报警限”列的值可以是 Null 之外的任何值。
- 特定节点中特定记录的“分类”值已知为 Value、Dev 或 ROC 时，“值”与“报警限”列的值只能接受 1234567890.-+eE 等字符。
- 特定节点中特定记录的“类型”值已知为 LOLO、LO、HI、HIHI、MinDev、MajDev 或 ROC 时，“值”与“报警限”列的值只能接受 1234567890.-+eE 等字符。
- 特定节点中特定记录的“分类”或“类型”值中任何一个已知为 DSC 时，“值”与“报警限”列的值可以是 Null 之外的任何值。
- “注释”列的值没有任何限制。
- 映射文件中的所有记录都应该是唯一的。Hot Backup Manager 在导入过程中会跳过任何重复的记录。您可以在导入过程完成之后查看详细信息。

您可以合并报警记录的字段值 - 如“组”、“名称”以及“优先级” - 以生成唯一确定报警记录的“组合映射关键码”。

“InTouch 报警供应器”将“名称”字段的值赋给生成报警的标记名。因此，给定热备份对时，可以使用报警组名与标记名的组合来生成映射关键码。

例如：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

供应器节点	备份节点
$System!TagA	$System!TagB

如果某个供应器将名称字段与注释字段合到一起用作唯一的字段，则映射关键码可以是名称与注释的组合。

供应器节点	备份节点
tagA!CommentA	tagB!CommentB

对于第三个供应器的任何其它字段组合，此方法也适用。

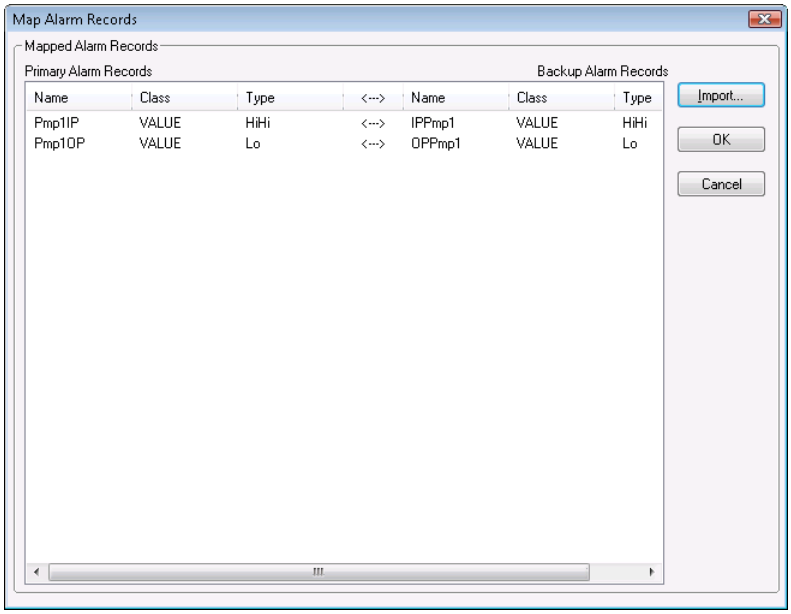
导入报警记录映射文件

您可以将报警记录映射文件的内容导入 Hot Backup Manager 中。

导入映射文件时，不会在“报警分类”与“报警类型”之间进行交叉验证。

从映射文件中导入报警记录

1. 打开 Alarm Hot Backup Manager。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm Hot Backup Manager**。
2. 从列表中选择热备份对。
3. 选择映射报警。此时出现映射报警记录对话框。



4. 选择导入。此时出现打开对话框。突出显示映射文件，然后选择打开。
   此时 Hot Backup Manager 开始从该文件中导入记录。
5. 导入所有映射记录之后，选择确定。
6. 在文件菜单上，选择保存。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

7. 重新启动 WindowMaker。

报警映射文件导入问题疑难排解

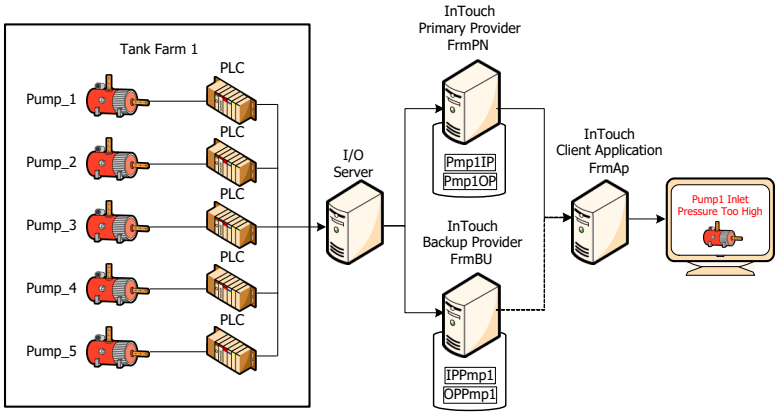
以下情况会导致无法导入文件：

- 对于导入文件中的所有记录，应该为要求填写的列填入一个值。任何记录都不得缺少或多出一些值。
- 导入文件的标题应该与选定关键码域对话框中的标题相同，并且顺序也应该相同。

如果导入的记录包含错误项，则提示您跳过这个特定的记录号或中止导入过程。

热备份对示例

本节介绍需要设置报警备份对的典型工作情形。下图显示 InTouch 贮油站应用程序的热备份对配置。在本例中，InTouch 应用程序监视泵浦压力，并将它用作一个报警条件。



所有三台计算机都在运行 InTouch HMI。热备份对包含作为主供应器的 FrmPN 以及作为备份供应器的 FrmBU。这两个节点用作“InTouch 分布式报警系统”中的报警供应器。

Hot Backup Manager 在 FrmAp 节点上运行。InTouch 客户端应用程序在 FrmAp 上运行，并且从热备份对的两个供应器中接收报警。

FrmPN 上运行的 InTouch 应用程序为泵浦入口压力与出口压力生成两个摘要报警。这两个报警属于 TnkFrm1 报警组。FrmBU 上运行的 InTouch 应用程序为泵浦压力生成逻辑上等价的两个报警。

设置冗余热备份对时，您可以：

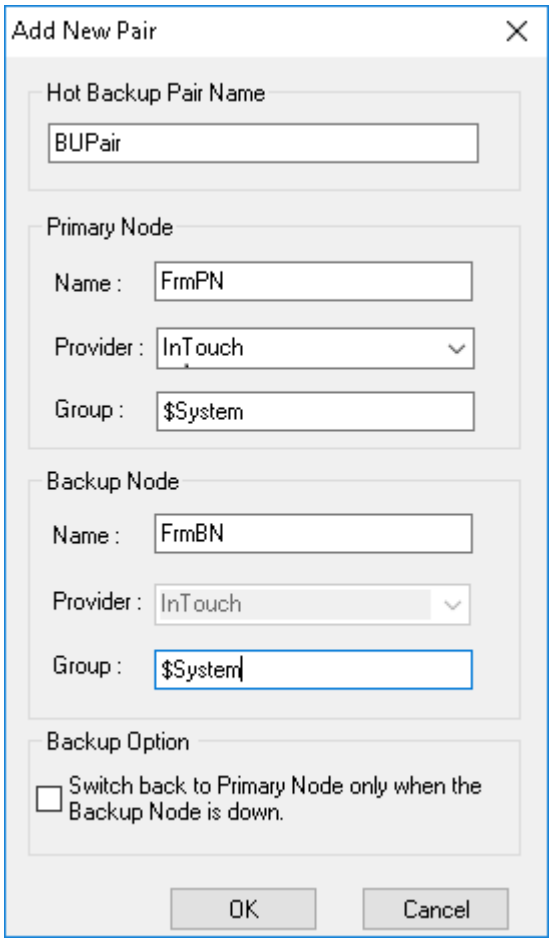
- 创建热备份对。
- 设置报警记录关键码域。
- 创建报警记录映射文件。
- 导入报警记录映射文件。

创建热备份对

1. 在 WindowMaker 中打开 InTouch 应用程序。在本例中，应用程序在 FrmAp 节点上运行。
2. 打开 Alarm Hot Backup Manager。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm Hot Backup Manager**。
3. 选择新对。此时出现添加新对对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 如下图所示，完成**添加新对**对话框中的选项。



5. 选择确定以返回 **Alarm Hot Backup Manager** 对话框。
6. 在 WindowMaker 中保持打**开 Alarm Hot Backup Manager** 对话框。

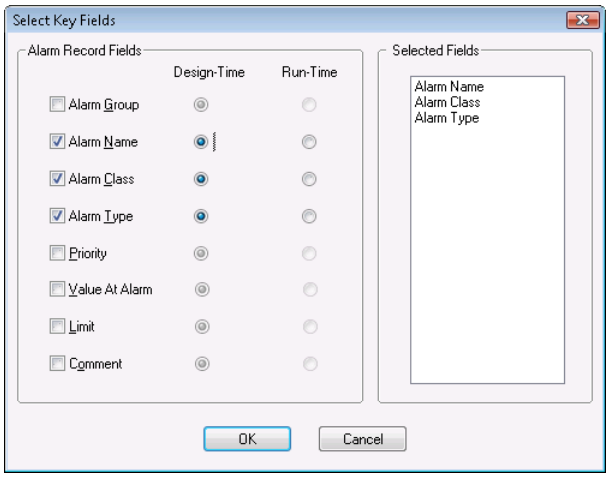
此时已完成第一个步骤，即创建热备份对。接下来要完成以下操作程序，以便为每个供应器的报警储备库中存储的成对报警记录生成唯一的映射关键码。

映射报警记录关键码域。

1. 选择列表中的热备份对。
2. 选择**设置关键码**域。此时出现**选定关键码**域对话框。

   如下图所示，完成**选定关键码**域对话框中的选项。主供应器与备份供应器之间的报警在逻辑上是一致的，但是指定了不同的名称并且属于不同的报警组。通过选择**报警组**、**报警名**、**报警分类**以及**报警类型**作为**设计时**选项，可以为每个供应器的报警储备库中存储的记录生成唯一的映射关键码。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 选择确定。出现消息时，选择是。

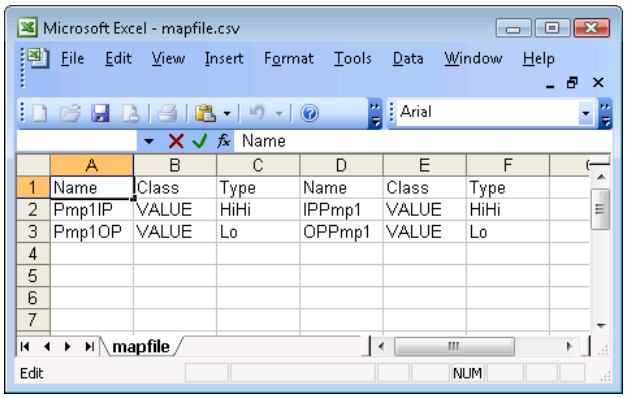
此时已完成第二个步骤，即创建报警记录映射关键码。

在此情形中，所有三个节点都在运行 InTouch 应用程序。两个供应器节点生成等价的报警，但使用不同的标记名。主供应器在泵浦的入口与出口压力过高时生成两个摘要报警。备份供应器为相同的泵浦压力报警条件生成两个逻辑上等价的报警。接下来要完成以下操作程序，以便创建映射文件，将每个供应器的报警储备库中存储的等价记录关联起来。

创建报警映射文件

1. 使用 Excel 或文本编辑器（如“记事本”）创建一个 .csv 文件。
2. 按选定关键码域对话框中报警记录字段选项的相同顺序输入文件标题的名称。

   在本例中，文件标题应该按报警组、报警名、报警分类以及报警类型的顺序排列。
3. 在文件的每一行上映射两个供应器之间的报警。
4. 下面的 Excel 文件示例显示应该如何为热备份对的两个供应器指定标题与报警条件。将映射文件保存到客户端节点上运行的 Hot Backup Manager 可以访问到的位置。



此时已完成第三个步骤，即创建报警记录映射文件。

在最后一个步骤，需要将报警映射文件的内容导入 Hot Backup Manager 中。在本例中，客户端应用程序知道要在两个报警供应器之间确认哪些泵浦压力报警记录。

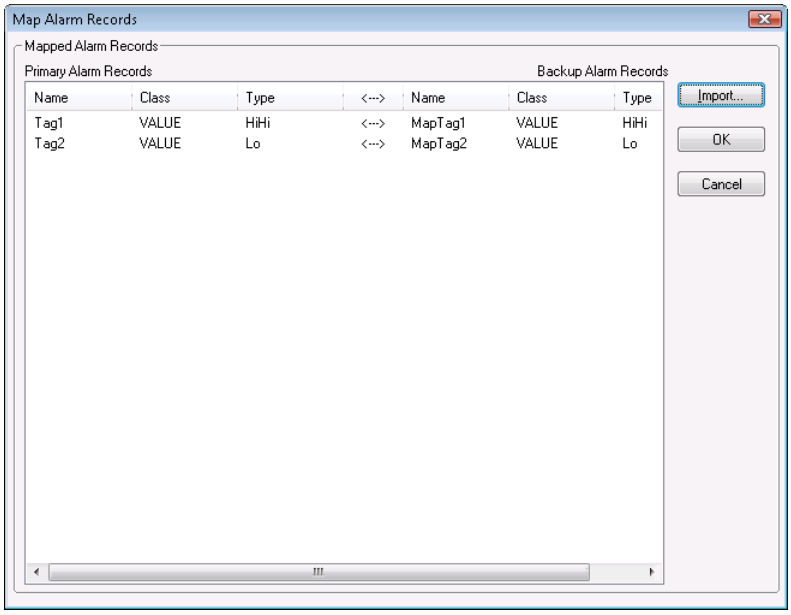
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

导入报警记录映射 .csv 文件

1. 打开 **Alarm Hot Backup Manager**。

   此时应列出先前创建的报警备份对。

2. 选择映射报警。此时出现映射报警记录对话框。
3. 选择导入。此时出现打开对话框。
4. 突出显示映射文件，然后选择打开。此时映射报警记录对话框列出文件中的报警映射记录。



5. 选择确定。此时 Hot Backup Manager 开始从该文件中导入记录。
6. 导入所有映射记录之后，选择确定。

现在您便可以运行热备份应用程序。

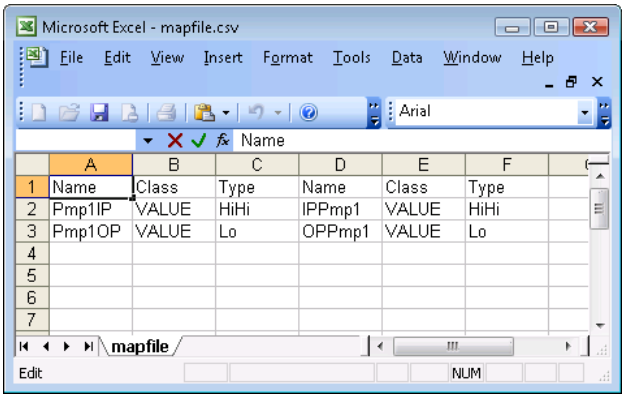
确认同步示例

在本例中：

- “报警名”、“报警分类”以及“报警类型”选作设计时关键码域。
- “报警组”选作运行时字段。

以下报警记录映射文件是使用 Microsoft Excel 创建的。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



	A	B	C	D	E	F
1	Name	Class	Type	Name	Class	Type
2	Pmp1IP	VALUE	HiHi	IPPmp1	VALUE	HiHi
3	Pmp1OP	VALUE	Lo	OPPmp1	VALUE	Lo
4						
5						
6						
7						

Pmp1IP 报警映射到 IPPmp1 报警。两者的“分类”都是 VALUE，“类型”都是 HIHI。

Pmp1OP 报警映射到 OPPmp1 报警。两者的“分类”都是 VALUE，“类型”都是 Lo。

- 只要两个节点上的报警组名保持相同，在主报警节点上确认 Pmp1IP 的 HiHi 报警时，便也会确认辅助供应器节点上 IPPmp1 的 HiHi 报警。
- 只要两个节点上的报警组名保持相同，在主报警节点上确认 Pmp1OP 的 Low 报警时，便也会确认辅助供应器节点上 OPPmp1 的 Low 报警。

仅当设计时与运行时映射相互匹配时，才会发生确认同步。

您可以选择设计时与运行时报警记录字段的任何组合来进行映射。不过请确保映射不会导致多次引用。

例如，如果选择了两个报警记录字段（如“分类”与“优先级”），则很可能有多个报警与这个准则匹配。在这种情况下，无法保证“热备份”同步能够正常工作。在传播确认的过程中，也可能会遇到符合该准则的随机报警，而其它匹配的报警则可能得不到确认。

有关热备份对的备注

- 只有对于 InTouch 7.11 及更高版本的客户端，才可以使用热备份。
- 如果“分布式报警显示”对象查询热备份对，然后再次单独查询主供应器，则“分布式报警显示”对象会显示重复的记录。
- 如果主供应器上的记录得到确认，而稍后才启动辅助供应器（确认发生时是停工的），则确认的记录的时间标签应该与主供应器中的记录相同。
- 查询热备份对的报警接收器显示供应器的单独节点名。
- 您可以选择设计时与运行时报警记录字段的任何组合来进行映射。
- 映射“值”与“报警限”关键码域时，值会舍入到小数点后四位，然后再进行映射。
- 没有特定的设计时与运行时映射组合的报警记录会使用缺省的运行时映射。
- “热备份”实用程序允许多个热备份对具有完全相同的节点名。此功能仅适用于供应器类型为 Galaxy 和 Galaxy_<name> 的供应器。

  例如，该实用程序现在可以添加以下两对：

示例 1

对名：Pair1

主节点：Node1

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

主供应器：Galaxy

主要组：A01/

备用节点：Node2

备份供应器：Galaxy

备用组：A01

示例 2

对名：Pair2

主节点：Node1

主供应器：Galaxy

主要组：A02/

备用节点：Node2

备份供应器：Galaxy

备用组：A02

实用程序中的错误报告已扩展为在尝试以这种方式添加对时向用户提供有关任何潜在问题的反馈。

注意： 使用非 InTouch 对时，输入的对不能与上一个对有相同的节点名称和组名称。如果您尝试添加的另一个对具有相同的节点名称，将会导致出现错误。例如：

对 1：Node1: n1 node2: n2 group: a1

对 2：Node1: n1 node2: n2 group: a1

创建报警审核跟踪

将 InTouch 报警供应器配置为使用操作系统或 ArchestrA 身份验证，并且发生报警时，假设操作员已登录，则报警显示对象会在“操作员全名”列中包含操作员的全名。

例如，如果某个用户注册在 PLANT_FLOOR 域中，用户 ID 为 JohnS，全名为 John Smith，则“操作员全名”列包含 John Smith 字样。如果随后确认了该报警，并且执行确认的节点设置为使用操作系统或 ArchestrA 安全性，则“操作员全名”列会进行更新，以显示确认操作员的全名。否则，报警显示对象显示与 $Operator 标记中的任何内容进行串联的计算机名。

InTouch 安全性可以在报警确认中包含操作员的全名。这在与报警检测有关的记录上也是可能的。在大多数机构中，登录 ID 并非某个人的全名，而是缩写或者是角色分类。

给供应器与接收器 InTouch 节点配置操作系统身份验证时：

- 报警显示对象在生成报警与执行确认时显示全名。
- Alarm Printer 在生成报警与执行确认时打印全名。
- Alarm DB Logger 随每个报警记录在 Operator 与 AckOperator 字段中记录域名、登录用户 ID 以及用户的全名。这样，即便某个机构中有两个员工的全名完全相同，也可以作出唯一的判断。
- “操作员”字段以“域名\用户名”的格式显示用户帐户。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

确认需要身份验证的报警

某些行业需要在确认特定类型的报警时执行显式“签名”或用户身份验证。InTouch 利用报警客户端控件或脚本功能来执行此类型的身份验证操作。报警优先级是用于确定要确认的报警是否需要特定用户签名的关键因素。

可以将报警配置成需要身份验证，以便用户可以确认这些报警。配置成需要身份验证的报警表示运行时用户必须输入其凭证，才能确认报警或报警组。如果系统配置成使用智能卡读取器，则用户可以输入智能卡凭证。

本节介绍如何在 WindowViewer 中确认需要身份验证的报警。

如需有关将报警客户端配置成需要身份验证进行报警确认的信息，请参阅《报警客户端控件文档》。

如需有关使用 SignedAlarmAck() 函数为确认报警配置身份验证行为的信息，请参阅*工业图形编辑器用户*文档中的“添加与维护图形脚本”。

关于在 WindowViewer 中确认需要身份验证的报警

确认针对身份验证进行了配置的报警具有以下要求：

- 必须为 Galaxy 启用安全性。
- 必须为部署的 InTouchViewApp 启用安全性。
- InTouchViewApp 必须至少包括一个报警客户端控件，该控件将配置为显示报警与事件，需要签名（身份验证）进行报警确认，并具有报警优先级范围的最小值和最大值。
- 运行时操作员必须拥有在 Galaxy 中配置的合适的用户名、密码和操作权限：操作员必须拥有足够权限才能运行 WindowViewer。

无论是谁当前作为 InTouch 应用程序的运行时用户登录，配置成需要签名进行报警确认的报警都始终需要用户身份验证。这不会影响已登录用户的会话。

报警确认基本运行时行为

当操作员选择一个或多个报警并尝试确认这些报警时：

- 报警客户端会检查它是否配置成需要确认签名。如果是，则它会检查是否任何所选报警都处于配置的优先级范围中。如果是，则它需要签名以确认所有所选报警。
- 如果不需要签名，则会显示一个弹出确认对话框。它可能具有用于确认注释的文本框（如果这样配置）。

- 如果操作员凭证有效，则报警客户端会尝试确认所有所选报警。
- 在报警客户端网格中，登录用户会标识为确认所有所选报警的人员。
- 报警客户端会在确认注释开头以文本“签名的确认”作为前缀，来指示这是签名的确认。
- 如果没有所选报警具有所需范围中的优先级，则报警客户端会在确认注释开头以文本“标准确认”作为前缀，以指示这是标准确认。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

确认需要身份验证的报警

1. 在 WindowViewer 中，选择一个或多个未确认的报警。右键单击并选择要执行的确认操作。如果任何报警都配置成需要签名，则会出现报警确认对话框。

   如果未启用智能卡，则用于选择智能卡或口令身份验证的模式按钮会禁用。

2. 如果配置成需要注释，请在注释框中添加注释。
3. 如果您使用网络用户帐户进行身份验证，则会显示该用户帐户选项。

   执行以下操作。

   a. 在用户名框中，输入您的用户名。缺省情况下会显示当前已登录用户的名称。如果当前还没有任何用户登录，则该框为空。
   b. 在密码框中，输入与用户名关联的密码。
   c. 在域框中，键入域名。如果安全模式是 ArchestrA Galaxy，则显示的域是 ArchestrA，并且您无法修改它。
   d. 选择确定。

4. 如果您使用智能卡进行身份验证，则会出现智能卡报警确认对话框。

   执行以下操作来使用智能卡进行身份验证。

   a. 在 **Certificate**（证书）列表中，选择您的智能卡证书。证书列表会以“域名/用户名”的形式显示。要使证书出现在该列表中，智能卡当前必须插入到附加到计算机的读取器中。缺省情况下会显示当前已登录用户的证书。如果您在安全写入对话框打开时插入或移除卡，则证书列表会自动更新。
   b. 在 **PIN** 框中输入所使用的智能卡的 PIN。
   c. 选择确定。

   要改为使用您的名称、口令和域进行身份验证，请选择“模式”按钮。转到第 3 步。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用分布式报警显示对象

此版本的 InTouch HMI 包含“分布式报警显示”对象，以支持使用 InTouch 7 及更早版本开发的应用程序。由于“分布式报警显示”对象是传统对象，所以您应该使用“Alarm Viewer 控件”，而不应使用更新版本的 InTouch HMI 来创建报警显示。

此版本的 InTouch HMI 包含“分布式报警显示”对象，以支持使用 InTouch 7 及更早版本开发的应用程序。由于“分布式报警显示”对象是传统对象，所以您应该使用“Alarm Viewer 控件”，而不应使用更新版本的 InTouch HMI 来创建报警显示。

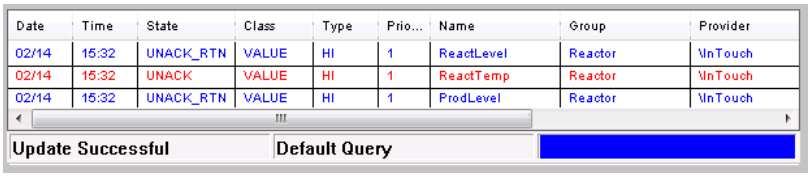
关于使用分布式报警显示对象

此版本的 InTouch HMI 包含“分布式报警显示”对象，以支持使用 InTouch 7 及更早版本开发的应用程序。由于“分布式报警显示”对象是传统对象，所以您应该使用“Alarm Viewer 控件”，而不应使用更新版本的 InTouch HMI 来创建报警显示。

此版本的 InTouch HMI 包含“分布式报警显示”对象，以支持使用 InTouch 7 及更早版本开发的应用程序。由于“分布式报警显示”对象是传统对象，所以您应该使用“Alarm Viewer 控件”，而不应使用更新版本的 InTouch HMI 来创建报警显示。

关于分布式报警显示对象

“分布式报警显示”对象提供单个显示对象来同时显示本地与远程报警。



此显示对象的功能包括内置滚动条、可调整大小的列、选择多个报警、状态栏、快捷菜单，以及基于报警优先级与状态的颜色。

“分布式报警显示”对象包含一些属性，可设置报警显示对象（包括显示的信息）的外观、用于各种报警条件的颜色，以及显示的报警组与报警优先级。

如需有关显示对象的详细信息，请参阅在运行时使用分布式报警显示对象。

分布式报警显示对象准则

使用“分布式报警显示”对象时，需要遵循以下准则：

- 每个显示对象都必须有一个标识符，以便任何关联的 QuickScript 函数知道要修改哪个显示对象。此标识符是在报警配置对话框的显示名框中输入的；对于每个显示对象，它都必须保持唯一。
- 显示对象不应该覆盖其它的 InTouch 对象，如窗口控件或图形对象等。通过选择 WindowMaker 中的“分布式报警显示”对象，并检查该显示对象的“手柄”，可以很方便地验证这点。这些手柄不应接触到屏幕上其它的对象。
- 这些显示对象应尽量少用。在一个屏幕上放置大量的显示对象可能会导致系统性能下降。如果可能，请限制窗口上的显示对象数，可以进一步调用包含其它显示对象的窗口。

在设计时配置分布式报警显示对象

您可以配置：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 常规功能，如状态栏、滚动条等。
- 列与排序顺序。
- 用于检索报警记录的报警查询。
- 显示报警记录的时间格式。
- 所显示的报警记录的字体与颜色。
- 运行时用户可以在显示对象中执行的操作，如调整列的大小、选择报警、访问快捷菜单等。

创建“分布式报警显示”对象

您可以像对待向导那样创建“分布式报警显示”对象。

创建“分布式报警显示”对象

1. 在**绘图**菜单上的**插入组**中，**选择向导**。
   此时出现**向导选择**对话框。
2. 从向导列表中选择**报警显示**。
3. 双击**分布式报警显示**向导。此时该对话框关闭，您的窗口再次出现，且光标处于“粘贴”模式。
4. 选择窗口以粘贴“分布式报警显示”对象。要调整向导大小，请拖曳选择手柄。

现在您便可以开始配置该显示对象。

配置网格的显示属性

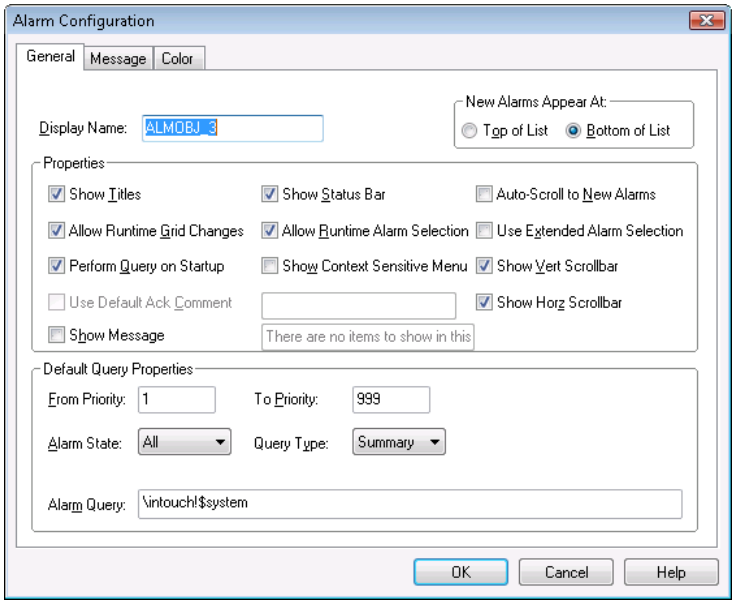
对于显示网格，您可以配置：

- 标题栏、状态栏及滚动条。
- 新报警出现在网格中的位置以及是否自动滚动到它们所在的位置。
- 要在没有报警记录可显示时显示的缺省消息。

配置网格外观

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现**报警配置**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**显示名**框中，**输入报警显示对象的名称。对于使用的每个报警显示对象，此名称必须保持唯一。此名称会在整个系统中用于引用此对象，以执行报警确认与查询之类的任务。**
3. 在**新报警出现在**区域中，配置希望新报警在对象中出现的位置：
   - **选择列表顶部**，以便在列表顶部显示最新的报警。
   - **选择列表底部**，以便在列表底部显示最新的报警。
4. 在**属性**区域中，配置**标题栏**、状**态栏**及**滚动**条。执行以下任意操作：
   - 选中**显示标题**复选框以显示报警消息标题栏。
   - 选中**显示状态栏**复选框以显示状态栏。
   - 选中**显示垂直滚动条**复选框，以显示垂直滚动条。
   - 选中**显示水平滚动条**复选框，以显示水平滚动条。
5. 选中**显示消息**复选框，以便在没有报警记录可显示时显示缺省消息。在方框中输入该消息。
6. 选中**自动滚动到新报警**复选框，以便让选择项自动跳到新报警处。新报警定义为当前未在显示对象中显示的那些报警。
7. 选择确定。

控制用户可以在运行时访问的功能

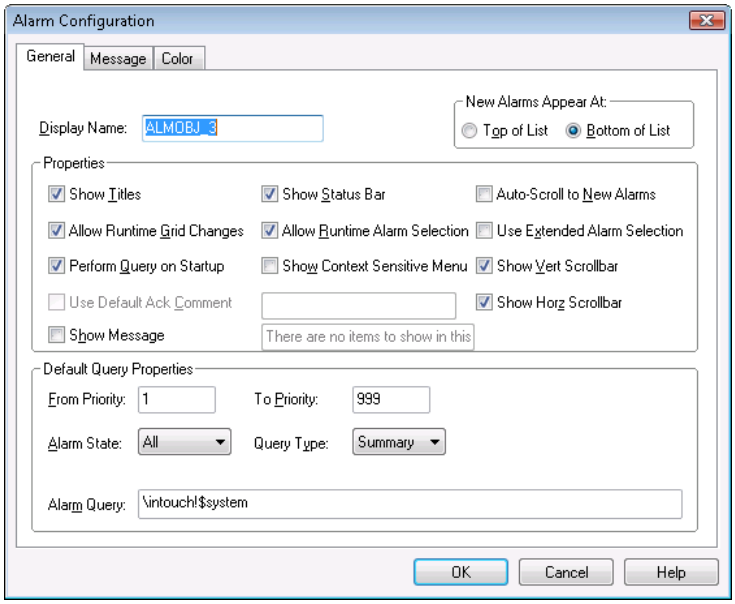
您可以允许运行时用户更改列设置、选择报警及打开快捷菜单。

注意：您可以使用脚本函数来运行快捷菜单上出现的命令。

配置运行时功能

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现报警配置对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 配置运行时可用的选项。执行以下任意操作：
   - 选中允许运行时改变网格复选框，以允许用户更改列设置。
   - 选中显示上下文相关菜单复选框以启用快捷菜单。
   - 选中允许运行时选择报警复选框以允许用户选择报警。
   - 选中使用扩展报警选择复选框，以便允许用户在按住 Ctrl 或 Shift 键的同时结合使用鼠标来选择多个报警。缺省情况是通过选择它们来切换选择的报警。
3. 选择确定。

配置要显示的报警

您可以配置“分布式报警显示”对象根据以下内容来显示报警：

- 优先级。
- 状态，如已确认或未确认。
- 类型，摘要或历史。

  配置报警查询时，只能使用文本。您无法使用标记。示例查询如下。

  “报警组”的完整路径：

  \\节点\InTouch!Group

本地“报警组”的完整路径

\InTouch!Group

另一个“组列表”：

GroupList

要执行多个查询，请使用空格分隔每个查询。例如：

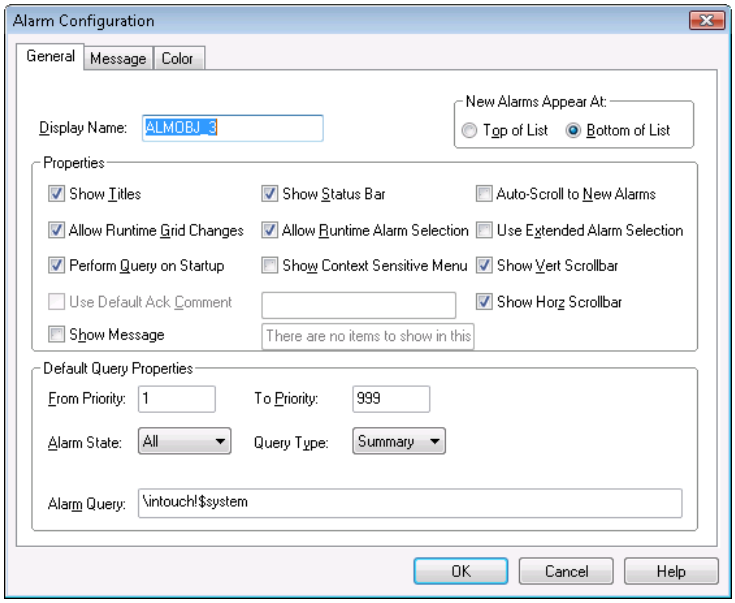
\InTouch!Group GroupList

仅在选中在启动时执行查询复选框，或执行 almDefQuery() 函数的情况下，才使用缺省的查询属性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置要显示的报警

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现报警配置对话框。



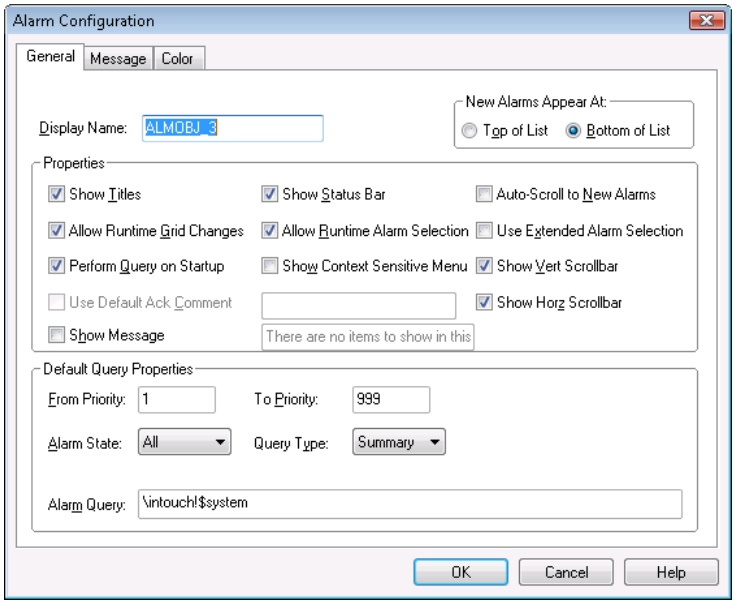
2. 选中在启动时执行查询复选框。
3. 在默认查询属性区域中，配置对象的缺省查询。
   - 在从优先级框中，输入缺省的最小报警优先级。
   - 在到优先级框中，输入缺省的最大报警优先级。
   - 在报警状态列表中，选择要查询的缺省报警状态（“全部”、“未确认”、“确认”）。
   - 在查询类型列表中，选择“摘要”或“历史”报警类型。
   - 在报警查询框中，输入初始报警查询。
4. 选择确定。

配置缺省报警注释

您可以配置要在操作员确认报警时使用的缺省注释。如果没有配置缺省注释，则在操作员确认报警时会出现一个对话框，要求操作员输入注释。该对话框可以填写，也可以留为空白。

配置缺省注释

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现报警配置对话框。



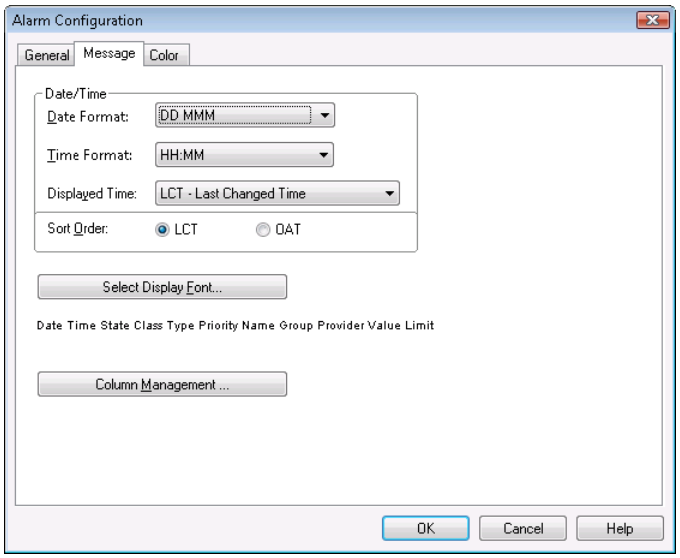
2. 选中**显示上下文相关菜单**复选框。
3. 选中**使用缺省确认注释**复选框，然后在框中**输入注释**文本。
4. 选择确定。

配置报警记录的时间格式

原始报警时间是报警发生时的日期/时间标签。如果标记是 I/O 标记，且 I/O 服务器能够传递时间标签，那么该标记就是来自该服务器的时间标签。

配置报警显示时间格式

1. 使用鼠标右键单击“分布式报警显示”对象，然后**选择属性**。此时出现**报警配置对话**框。
2. 选择消息选项卡。



3. 在**日期格式**列表中，**选择**日期的格式。可用的格式包括：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

所选部分	显示	所选部分	显示
DD MMM	28 二月	**MM/DD**	02/28
DD MMM YYYY	28 二月 2007	**MM/DD/YY**	02/28/07
DD/MM	28/07	**MMM DD**	二月 28
DD/MM/YY	28/02/07	**MMM DD YYYY**	二月 28 2007
YY/MM/DD	07/02/28	**YYYY/MM/DD**	2007/02/28

4. 在**时间格式**列表中，**单击时间**的格式。将此列表中的**值**用作指定**时间**格式的模板。例如，要将**时间**指定为 10:24:30 上午，**请**使用 AP HH:MM:SS。模板字符如下：

字符	描述
AP	选择“上午/下午”格式。例如，下午三点钟显示为下午 3:00。 未指定此项的时间缺省使用 24 小时军用时间格式。例如，下午三点钟显示为 15:00。
HH	显示报警/事件发生时的小时值。
MM	显示报警/事件发生时的分钟值。
SS	显示报警/事件发生时的秒钟值。
SSS	显示报警/事件发生时的毫秒值。

5. 在**排序方式**区域中，配置希望报警在对象中的排序方式：
   - 选择 OAT 以使用原始报警时间，即报警发生时的日期/时间标签。
   - 选择 **LCT** 以使用上次报警变化时间，即报警实例最近一次改变状态的日期/时间标签：报警发生、子状态改变、恢复到正常或确认。
6. 选择确定。

配置报警记录的字体

您可以设置控件中记录与标题的字体。

配置字体属性

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现报警配置对话框。
2. 选择消息选项卡。
3. 单击选择显示字体。此时出现标准的 Windows 字体对话框。
4. 配置字体，然后单击确定。

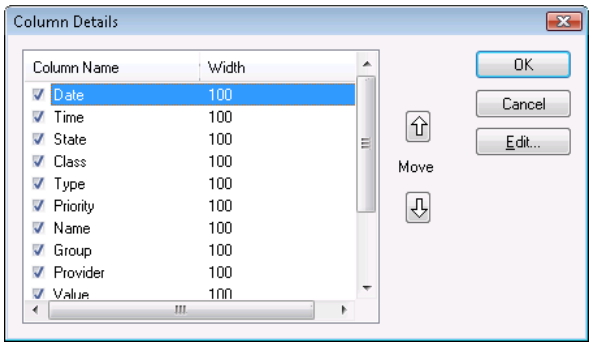
配置报警记录的列

您可以选择要显示的列、指定列的顺序，以及设置列名与宽度。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置显示列明细

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现报警配置对话框。
2. 选择消息选项卡。
3. 选择列管理。此时出现列明细对话框。

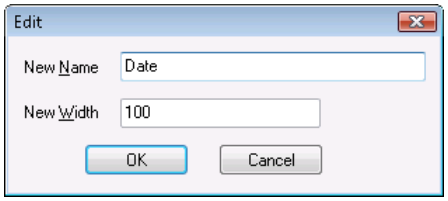


4. 选中列名列表中的复选框，以便在“分布式报警显示”对象中显示该列。您至少必须选择一列。下表介绍各个列：

列	描述
日期	按照从消息选项卡中选择的格式显示日期。
时间	按照从消息选项卡中选择的格式显示时间。
状态	显示报警的状态。
类	显示报警的类别。
类型	显示报警类型。
优先级	显示报警优先级。
姓名	显示报警/标记名。
组	显示“报警组”名。
供应器	显示报警供应器的名称。
值	显示报警发生时标记名的值。
限制	显示标记名的报警限值。
操作员	显示与报警条件关联的已登录操作员的 ID。
注释	显示标记名注释。此注释是在数据库中定义标记名的报警时，在报警注释框中输入的。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5. 要重新排列各个列，请选择列名，然后使用“上移”与“下移”箭头按钮。列明细对话框顶部出现的列名是在报警显示对象的最左侧显示的列。
6. 要编辑列名与宽度，请选择列名，然后选择编辑。此时出现该列的编辑对话框。



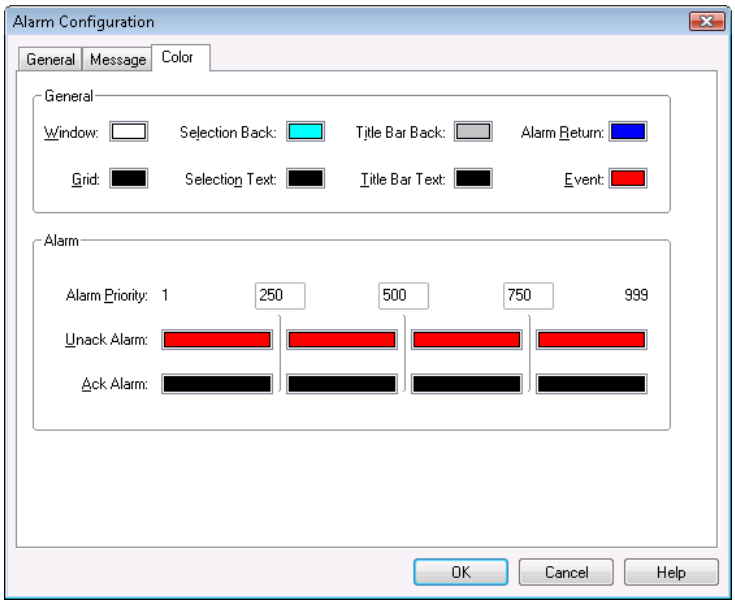
   a. 如果要显示与缺省列名不同的列名，请在新名称框中输入新的名称。
   b. 在新宽度框中，输入列宽。列宽可以在从 1 到 999 个像素的范围内。缺省列宽是 100 个像素。
   c. 选择编辑对话框中的确定。
7. 选择列明细对话框中的确定。
8. 选择应用。

配置报警记录的颜色

您可以为“分布式报警显示”对象的各个部分配置颜色，如背景颜色与所选记录的颜色。您也可以为不同范围的报警记录配置不同的颜色。

配置报警显示颜色

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现报警配置对话框。
2. 选择颜色选项卡。



3. 在常规区域中，选择每个颜色框以打开 InTouch 调色板。选择您想要为下列各项使用的颜色：

选项	描述
窗口	设置显示背景颜色。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

选项	描述
网格	设置网格颜色。
选择背景	设置突出显示的文本背景颜色。
选择文本	设置突出显示的文本颜色。
标题栏背景	设置标题栏背景颜色（仅当显示标题选项为开时设置的效果才可见）。
标题栏文本	设置标题栏文本颜色（仅当显示标题选项为开时设置的效果才可见）。
报警返回	设置返回的报警（未经确认即返回到正常的报警）的颜色。
事件	设置事件报警的颜色。

4. 在报警优先级框中，为报警显示对象输入各个断点值。
5. 选择未确认报警与确认报警颜色框以打开 InTouch 调色板。在调色板中选择您要使用的颜色。
6. 选择确定。

配置显示类型

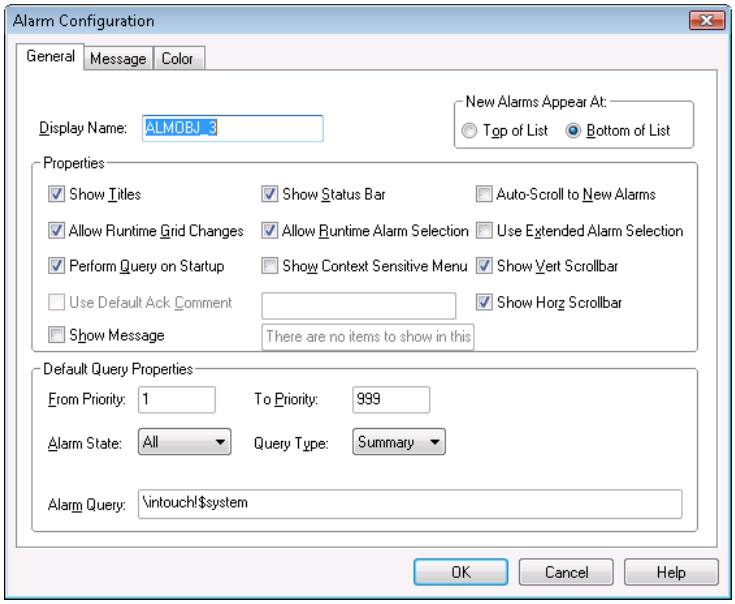
“分布式报警显示”对象可以显示活动报警的摘要或历史报警的列表。

您也可以在运行时动态地更改显示类型。例如，这可以通过运行 almQuery() 脚本函数来实现。almQuery() 脚本函数使用的参数可以用于将指定的“分布式报警显示”对象（例如："AlmObj_1"）设置为指定的显示类型（例如：“摘要”）。如需有关详细信息，请参阅 almQuery() 函数。

配置查询缺省值

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现报警配置对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**查询类型**列表中，**单击**希望用作运行**时**缺省**值**的**报警显示类型**。
3. **选择确定**。

使用分布式显示对象监视本地报警

您可以使用“分布式报警显示”对象来显示与确认本地及远程报警。

要设置仅监视本地报警的显示对象

1. 使用鼠标右键单击“分布式报警显示”对象，然后选择属性。此时出现**报警配置**对话框。
2. 在**报警查询**框中，输入：`\InTouch!$System`

   您可以将 $System 替换成任何有效的报警组。您也可以定义一个仅包含 \InTouch!$System 的报警组列表，然后使用这个组列表而不是直接引用。
3. 根据显示类型与应用程序要求的任何过滤设置来配置其它参数。

在运行时使用分布式报警显示对象

“分布式报警显示”对象使用网格来显示报警消息。此网格允许动态调整列宽，只要选择列手柄并将其拖动到所需的宽度即可。此功能仅在运行时提供。

网格列的更改不会保存；因此如果在更改网格列之后关闭包含该报警显示对象的窗口，然后再打开该窗口，则网格列又将使用它们的缺省宽度。您可以在 WindowMaker 中调整缺省列宽。

网格允许您在列表框中选择一个或多个报警。所选的报警可通过使用 almAckSelect() QuickScript 函数进行确认。在配置“分布式报警显示”对象时，也可以定义选择行为，以便允许切换单项选择（逐项）或多项选择（在按住 CTRL 或 SHIFT 键的同时，通过单击鼠标来选择多个报警）。您可以关闭运行时选择功能。

根据报警的优先级以及它是否已确认，您也可以为显示的每个报警消息配置多达八种不同的颜色。

如果在配置时打开滚动条，则可以使用滚动条。

根据控件的配置，报警显示对象可能有用于按页滚动报警记录的控件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

可调整大小的显示列

“分布式报警显示”对象使用网格来存放报警消息。在运行时，只要选择一个列并拖动它，便能动态调整列宽。您也可以双击垂直网格线来自动调整列宽。此功能仅在运行时提供。列宽调整功能必须在配置时打开。

关闭包含该报警显示对象的窗口时，网格列的更改不会保存。

多重选择

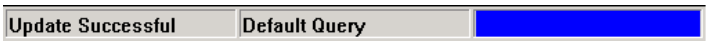
根据报警显示对象的配置，您可以在列表框中选择一个或多个报警。

报警消息颜色

根据报警的优先级以及报警是否已确认，最多可以给显示的每个报警消息使用八种不同的颜色。

状态栏

根据报警对象的配置，状态栏可以显示状态消息、当前报警查询及进度条。



状态栏的左侧部分显示控件的当前状态。

这些指示器提供显示查询当前状态的概况<1>，</1>并提供<2>“</2>分布式报警显示<3>”</3>对象中可用抑制的有关详细信息<4>。</4> 冻结生效时状态栏的右侧窗格是红色的；抑制生效时状态栏的左侧窗格是红色的。

抑制生效时左侧窗格中显示<1>“</1>抑制<2>”</2>字样<3>。</3>

功能	描述
状态消息	状态栏左端的状态消息提供更详尽的当前查询状态的说明。
报警查询	<1>“</1>报警查询<2>”</2>提供当前报警查询的可视化指示<3>。</3>
进度条	状态栏右端的更新进度条提供当前查询进度的可视化指示。

状态消息内容如下：

状态消息	状态/指示器	进度条
无	没有查询	无
未完全更新	查询未完成	蓝色/绿色
更新成功	查询完成	红色
抑制	查询名	纯蓝
冻结	查询名	红色

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

快捷菜单

根据报警对象的配置方式，您可以使用鼠标右键单击对象来打开常用命令的快捷菜单：

选择此命令	实现
确认已选项	确认所选的报警。
确认其它	确认显示对象中的所有报警，或仅确认可见的报警、所选组、所选标记名及优先级。
抑制已选项	抑制所选报警。
抑制其它	抑制显示对象中的所有报警，或仅抑制可见的报警、所选组、所选标记名及所选优先级。
查询收藏夹	打开报警查询对话框，在其中可以选择要使用的查询。
统计	打开报警统计对话框。
抑制	打开报警抑制对话框。
冻结	冻结当前显示对象。
解除已选项	解除所选报警。
解除其他	解除显示对象中的所有报警，或仅关闭可见的报警、所选组、所选标记名及所选优先级。

选择与配置报警查询收藏夹

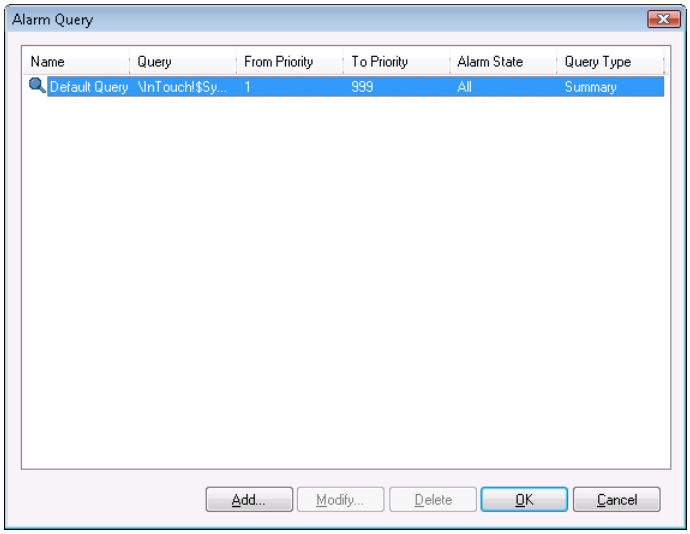
使用快捷菜单上的**查询收藏夹**命令从先前定义的报警查询列表中快速选择某个报警查询。您也可以创建新的命名查询、编辑或删除现有的查询。

注意：对于“分布式报警显示”中出现的多行报警查询，换行会出现“乱码”字符。但这不影响功能。

为显示对象选择报警查询

1. 在运行时，使用鼠标右键单击“分布式报警显示”，然后单击**查询收藏夹**。此时出现**报警查询**对话框。

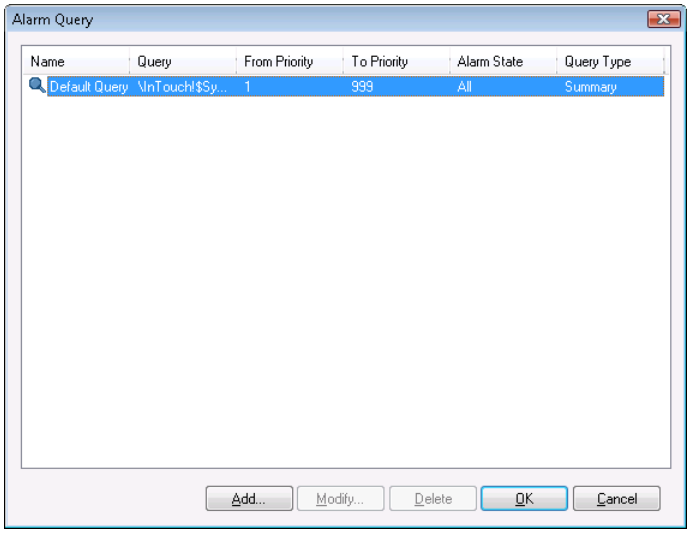
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 从当前定义的查询列表中，选择要使用的命名查询。
3. 选择确定。此时“分布式报警显示”对象显示所选查询的报警信息。

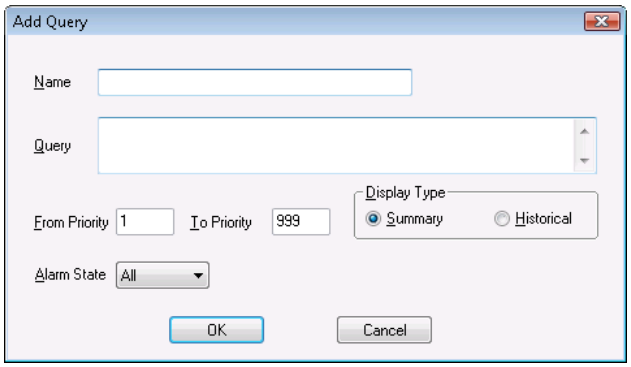
要添加新的命名查询

1. 在运行时，使用鼠标右键单击“分布式报警显示”，然后选择**查询收藏夹**。此时出现**报警查询**对话框。



2. 选择添加。此时出现添加**查询**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 配置查询。执行以下操作：
   a. 在名称框中，输入查询的名称。
   b. 在查询框中，输入希望执行的一组 InTouch 报警查询。您可以指定一个或多个报警供应器与组。
   c. 在从优先级框中，输入最小报警优先级值（1 到 999）。在到优先级框中，输入最大报警优先级值（1 到 999）。
   d. 在报警状态列表中，选择要在报警查询中使用的报警状态。
   e. 在显示类型区域中，选择要显示的报警类型。如需有关详细信息，请参阅摘要报警与历史报警。
4. 选择确定，关闭添加查询对话框。
5. 选择报警查询对话框中的确定。

要修改现有的命名查询

1. 在运行时，使用鼠标右键单击“分布式报警显示”，然后选择查询收藏夹。此时出现报警查询对话框。
2. 从当前定义的查询列表中，选择要修改的命名查询。
3. 选择 Modify。此时出现修改查询对话框。
4. 进行所需的修改，然后选择确定关闭修改查询对话框。
5. 在报警查询对话框中选择确定。

注意：对于正在使用被修改的报警查询的其它“分布式报警显示”对象，修改内容不会自动应用于它们。

要删除现有的命名查询

1. 在运行时，使用鼠标右键单击“分布式报警显示”，然后选择查询收藏夹。此时出现报警查询对话框。
2. 从当前定义的查询列表中，选择要修改的命名查询。
3. 选择删除。出现消息时，选择是。
4. 在报警查询对话框中选择确定。

注意：对于正在使用要删除的报警查询的其它“分布式报警显示”对象，删除操作不会自动应用于它们。

使用函数与点域控制分布式报警显示对象

您可以使用函数与点域在运行时控制“分布式报警显示”对象。

如需函数返回的错误号的列表，请参阅错误描述。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

获取或设置属性

属性可以通过 GetProperty*X*() 函数进行访问，其中 X 是数据类型（D 代表离散、I 代表整型、M 代表消息）。例如：

```
GetPropertyM(ControlName.Property, MsgTag)
```

如需有关 GetPropertyX() 函数的详细信息，请参阅内置函数。

运行包含 GetPropertyX() 函数的脚本时，属性值会保存到 MsgTag。如果选择了多个行，指定给 MsgTag 的属性是选择的多个行中第一行的标记值。

确认报警

“分布式报警显示”对象能够确认它可以查询到的任何报警（仅限于摘要显示对象）。“分布式报警显示”对象包含一些报警确认函数。对于用来确认本地报警与报警组的 **.Ack** 点域而言，这些函数可以起到补充作用。使用这些函数确认所有的报警、显示的报警以及所选的报警。

您也可以按照特性来确认它们。例如，组成员关系、优先级、应用程序名以及标记名。

- almAckAll() 函数
- 使用函数与点域控制分布式报警显示对象
- almAckGroup() 函数
- almAckPriority() 函数
- almAckRecent() 函数
- almAckTag() 函数
- almAckSelect() 函数
- almAckSelectedGroup() 函数
- almAckSelectedPriority() 函数
- almAckSelectedTag() 函数

almAckAll() 函数

确认当前查询中的所有报警，包括摘要模式下“分布式报警显示”对象中当前未显示的那些报警。

类别

报警

语法

```
[Result=]almAckAll(ObjectName,Comment);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

Comment
报警确认注释。

示例

```
MessageTag = "Acknowledge All by " + $Operator;
almAckAll("AlmObj_1",MessageTag);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

Ack(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()

almAckDisplay() 函数

仅确认摘要模式下“分布式报警显示”对象中当前可见的那些报警。

类别

报警

语法

```
[Result=]almAckDisplay(ObjectName,Comment);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

Comment
报警确认注释。

示例

```
almAckDisplay("AlmObj_1","Display Acknowledgement");
```

另请参阅

Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckRecent(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()

almAckGroup() 函数

确认命名的“分布式报警”对象中与特定的供应器及组名匹配的所有报警。

类别

报警

语法

```
[Result=]almAckGroup( “ObjectName”, ApplicationName, GroupName, Comment);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
“应用程序”的名称，例如 \\node1\Intouch

GroupName
InTouch 报警组的名称，如 $System。

Comment
报警确认注释。

示例

```
MessageTag = “Acknowledge group, Turbines, by ” + $Operator;
almAckGroup(“AlmObj_1”, “\Intouch”, “Turbine”, MessageTag);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

almAckPriority() 函数

确认指定的“分布式报警”对象中显示的所有报警，这些报警是上次的查询结果，该查询与报警的应用程序名、报警组及优先级范围匹配。

类别

报警

语法

```
[Result=]almAckPriority(ObjectName, ApplicationName, GroupName, FromPri, ToPri, Comment);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
“应用程序”的名称，例如 \\node1\Intouch

GroupName
“组”的名称，例如 $System

FromPri
报警优先级范围的起始数值。例如，100。

ToPri
报警优先级范围的结束数值。例如，900。

Comment
报警确认注释。

示例

```
almAckPriority(“AlmObj_1”, “\\node1\Intouch”, “Turbines”, 10, 100, “Range 10 to 100
acknowledged”);
```

almAckRecent() 函数

确认最近发生的报警。

语法

```
[Result=]almAckRecent(ObjectName, Comment)
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

Comment
报警确认注释。

示例

```
almAckRecent("AlmObj_1",$DateString);
```

almAckTag() 函数

确认指定的“分布式报警显示”对象中显示的所有报警，这些报警是上次的查询结果。报警必须与查询指定的应用程序名、组名、标记名及优先级范围匹配。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

语法

```
[Result=]almAckTag("ObjectName", ApplicationName, GroupName, TagName, FromPri, ToPri,
Comment);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
应用程序的名称。例如，\\node1\Intouch。

GroupName
报警组的名称。例如，$System。

TagName
值处于报警状态的标记的名称。

FromPri
报警优先级范围的起始数值。例如，100。

ToPri
优先级范围的结束数值。例如，900。

Comment
报警确认注释。

示例

```
almAckTag("AlmObj_1", "\\node1\Intouch", "Turbines", "Valve1", 10, 100, "Acknowledged for
Valve1");
```

另请参阅

Ack(), almAckAll(), almAckGroup(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()

almAckSelect() 函数

仅确认摘要模式下“分布式报警显示”对象中所选的那些报警。

类别

报警

语法

```
[Result=]almAckSelect(ObjectName,Comment);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

Comment
报警确认注释。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

本例仅确认白班或夜班中发生的那些报警。

```
IF ($Hour >= 0 and $Hour < 8) THEN
   AckTag = "Night Shift";
ELSE
   AckTag = "Day Shift";
ENDIF;
almAckSelect ("AlmObj_1",AckTag);
```

另请参阅

Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelectedGroup(), almAckSelectedPriority(), almAckSelectedTag()

almAckSelectedGroup() 函数

确认满足此条件的所有报警：这些报警具有相同的供应器名与组名，并且它们与指定的“分布式报警显示”对象中所选的一个或多个报警具有相同的组名。

类别

报警

语法

```
[Result=]almAckSelectedGroup(ObjectName,Comment);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

Comment
报警确认注释。

示例

```
MessageTag = "Acknowledge selected groups by " + $Operator;
almAckSelectedGroup ("AlmObj_1", MessageTag);
```

另请参阅

Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedPriority(), almAckSelectedTag()

almAckSelectedPriority() 函数

确认满足此条件的所有报警：这些报警具有相同的供应器名与组名，并且它们与指定的“分布式报警显示”对象中所选的一个或多个报警具有相同的优先级值。优先级是根据所选报警记录的最小与最大优先级计算得出的。

类别

报警

语法

```
[Result=]almAckSelectedPriority(ObjectName, Comment);
```

参数

ObjectName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

报警对象的名称。例如，AlmObj_1。

Comment
报警确认注释。

示例

```
MessageTag = "Acknowledge selected priorities by " + $Operator;
almAckSelectedPriority ("AlmObj_1", MessageTag);
```

另请参阅

Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedGroup(), almAckSelectedTag()

almAckSelectedTag() 函数

确认满足此条件的所有报警：这些报警具有相同的“标记名”、来自相同的供应器与组名，并且它们与指定的“分布式报警显示”对象中所选的一个或多个报警具有相同的优先级。仅当 InTouch 是报警供应器时，此函数才有效。

类别

报警

语法

```
[Result=]almAckSelectedTag(ObjectName,Comment);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

Comment
报警确认注释。

示例

```
MessageTag = "Acknowledge selected tagnames by " + $Operator;
almAckSelectedTag ("AlmObj_1", MessageTag);
```

另请参阅

Ack(), almAckAll(), almAckGroup(), almAckTag(), almAckDisplay(), almAckRecent(), almAckSelect(), almAckSelectedGroup(), almAckSelectedPriority()

选择报警

您可以创建脚本从“分布式报警显示”对象中选择报警。您可以选择所有的报警，也可以仅选择所选的报警，或者是获取当前的报警数。

- almSelectAll() 函数
- almUnselectAll() 函数
- almSelectionCount() 函数
- almSelectGroup() 函数
- almSelectItem() 函数
- almSelectPriority() 函数

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- almSelectTag() 函数

您也可以根据数据源、报警优先级以及 InTouch 标记来选择特定的报警。

almSelectAll() 函数

选择/取消选择指定的“分布式报警显示”对象中的所有报警。

类别

报警

语法

```
[Result=]almSelectAll(ObjectName);
```

参数

ObjectName
报警对象的名称。例如， AlmObj_1。

示例

```
If $AccessLevel > 8000 THEN
   almSelectAll("AlmObj_1");
   almAckSelect("AlmObj_1", "Ack Selected by a Manager");
ENDIF;
```

另请参阅

almSelectItem(), almSelectGroup(), almSelectPriority(), almSelectTag(), almUnSelectAll()

almUnselectAll() 函数

取消选择指定的“分布式报警显示”对象中所选的全部报警。

类别

报警

语法

```
[Result=]almUnselectAll(ObjectName);
```

参数

ObjectName
报警对象的名称。例如， AlmObj_1。

示例

```
If $AccessLevel == 9999 THEN
   almAckSelect("AlmObj_1", "Comment");{This alarm can be acknowledged by only
   Administrator}
   ELSE
   almUnselectAll("AlmObj_1");
ENDIF;
```

另请参阅

almSelectAll(), almSelectItem(), almSelectGroup(), almSelectPriority(), almSelectTag()

almSelectionCount() 函数

返回操作员在“分布式报警显示”对象中选择的报警的数量。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

语法

```
[Result=]almSelectionCount(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

示例

将操作员在“分布式报警显示”对象中选择的报警数指定给 AlarmCount 标记。

```
AlarmCount = almSelectionCount("AlmObj_1");
```

almSelectGroup() 函数

选择/取消选择指定的“分布式报警显示”对象中包含的所有报警，这些报警是上次的查询结果，并且产生的报警包含相同的报警组名。

类别

报警

语法

```
[Result=]almSelectGroup(ObjectName, ApplicationName,GroupName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
“应用程序”的名称。例如，\\node1\Intouch。

GroupName
“组”的名称。例如，$System。

示例

```
almSelectGroup("AlmObj_1","\InTouch","Turbine");
```

另请参阅

almSelectAll(), almSelectItem(), almSelectPriority(), almSelectTag(), almUnSelectAll()

almSelectItem() 函数

选择/取消选择报警显示对象中上次所选或取消选择的项目。

语法

```
[Result=]almSelectItem(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

示例

```
almSelectItem("AlmObj_1");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

almSelectAll(), almSelectGroup(), almSelectPriority(), almSelectTag(), almUnSelectAll()

almSelectPriority() 函数

选择/取消选择指定的“分布式报警显示”对象中的所有报警，这些报警是上次的查询结果，并且产生的报警在指定的优先级范围内。

类别

报警

语法

```
[Result=]almSelectPriority( "objectName", ApplicationName, GroupName, FromPri, ToPri );
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
“应用程序”的名称。例如，\\node1\Intouch。

GroupName
“组”的名称。例如，$System。

FromPri
报警的开始优先级。例如，100 或整型标记。

ToPri
报警的结束优先级。例如，900 或整型标记。

示例

```
almSelectPriority("AlmObj_1","\\node1\Intouch", "Turbines",10,100);
```

另请参阅

almSelectAll(), almSelectItem(), almSelectGroup(), almSelectTag(), almUnSelectAll()

almSelectTag() 函数

选择/取消选择指定的“分布式报警显示”对象中的所有报警，这些报警是上次的查询结果，并且它们具有指定的标记名。

类别

报警

语法

```
[Result=]almSelectTag (ObjectName, ApplicationName, GroupName, TagName, FromPri, ToPri);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
“应用程序”的名称。例如，\\node1\Intouch。

GroupName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

组的名称。例如，$System。

TagName
报警标记的名称。

FromPri
报警的开始优先级。例如，100 或整型标记。

ToPri
报警的结束优先级。例如，900 或整型标记。

示例

```
almSelectTag("AlmObj_1","\\node1\Intouch","Turbines","Valve1",10,100);
```

另请参阅

almSelectAll(), almSelectItem(), almSelectGroup(), almSelectPriority(), almUnSelectAll()

检索所选报警记录的有关信息

您可以创建脚本来返回所选报警的有关信息。在脚本中使用以下点域：

- .AlarmTime 点域
- .AlarmDate 点域
- .AlarmName 点域
- .AlarmValue 点域
- .AlarmClass 点域
- .AlarmType 点域
- .AlarmState 点域
- .AlarmLimit 点域
- .AlarmPri 点域
- .AlarmGroupSel 点域
- .AlarmAccess 点域
- .AlarmProv 点域
- .AlarmOprName 点域
- .AlarmOprNode 点域
- .AlarmComment 点域

.AlarmTime 点域

返回报警发生的时间。该报警必须在摘要模式下的“分布式报警显示”对象中处于选择状态。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmTime",TagName);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

在本例中，AlmObj_1 是“分布式报警显示”对象的名称，**almTime** 是内存消息标记。

```
GetPropertyM("AlmObj_1.AlarmTime",almTime);
```

如果用在“触动按钮”QuickScript 中，此语句将发生报警的时间返回到 **almTime** 标记中。

另请参阅

GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmType, .AlarmValue

.AlarmDate 点域

返回与所选的报警关联的日期。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmDate",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

如果用在“触动按钮”QuickScript 中，此语句将日期返回给 **almDate** 标记。

```
GetPropertyM("AlmObj_1.AlarmDate",almDate);
```

AlmObj_1 是“分布式报警显示”对象的名称，**almDate** 是内存消息标记，它用于检索与所选报警关联的标记的日期。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.AlarmName 点域

返回与所选报警关联的标记的名称。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmName",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

如果用在“触动按钮”QuickScript 中，此语句将报警名称返回给 **almName**。

```
GetPropertyM("AlmObj_1.AlarmName",almName);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almName** 是内存消息标记，它用于检索与所选报警关联的标记的名称。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmValue 点域

返回与所选报警关联的标记的报警值。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmValue,TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

此函数使用消息标记来检索数值。这是由于 GetProperty 函数不支持实数。您可以使用 StringToReal() 函数将结果指定给实型标记。

示例

如果用在“触动按钮”QuickScript 中，此语句将值返回给 **almValue**。

```
GetPropertyM("AlmObj_1.AlarmValue", almValue);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almValue** 是内存消息标记，它包含与所选报警关联的标记的报警值。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmClass 点域

返回与所选报警关联的标记的报警类。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmClass",Tagname);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

以下语句返回与所选报警关联的报警类。

```
GetPropertyM("AlmObj_1.AlarmClass",almClass);
```

AlmObj_1 是“分布式报警显示”对象的名称，**almClass** 是内存消息标记，它包含与所选报警关联的标记的报警类。

如果用在“触动按钮”QuickScript 中，此语句将报警的报警类返回给 **almClass** 标记。

另请参阅

GetPropertyM(), .AlarmAccess, .AlarmComment, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue

.AlarmType 点域

返回与所选报警关联的标记的报警类型。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmType",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

如果用在“触动按钮”QuickScript 中，则操作员确认报警时，此语句将所选报警的类型返回给 **almType** 标记。

```
GetPropertyM("AlmObj_1.AlarmType",almType);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almType** 是内存消息标记，它包含与所选报警关联的标记的报警类型。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmValue

.AlarmState 点域

返回所选报警的状态。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmState",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

如果用在“触动按钮”QuickScript 中，此语句将所选报警的状态返回给 almState 标记。

```
GetPropertyM("AlmObj_1.AlarmState",almState);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almState** 是内存消息标记，它包含与所选报警关联的标记的报警状态。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmTime、.AlarmType、.AlarmValue

.AlarmLimit 点域

返回与所选报警关联的标记的报警限。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM ("ObjectName.AlarmLimit",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

附注

此函数使用消息标记来检索数值。这是由于 GetProperty 函数不支持实数。您可以使用 StringToReal() 函数将结果指定给实型标记。

示例

如果用在触动按钮 QuickScript 中，此语句将所选报警的限制返回给 almLimit 标记。

```
GetPropertyM("AlmObj_1.AlarmLimit",almLimit);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almLimit** 是内存消息标记，它包含与所选报警关联的标记的报警限。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmPri 点域

返回与所选报警关联的标记的优先级 (1-999)。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM("ObjectName.AlarmPri", TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

如果用在“触动按钮”QuickScript 中，此语句将报警优先级返回给 almPrilvl 标记。

```
GetPropertyM("AlmObj_1.AlarmPri",almPrilvl);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almPrilvl** 是内存消息标记，它包含与所选报警关联的标记的优先级。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmGroupSel 点域

返回与所选报警关联的标记的报警组。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmGroupSel",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

如果用在“触动按钮”QuickScript 中，此语句将所选报警组的名称返回给 almGroup 标记。

```
GetPropertyM("AlmObj_1.AlarmGroupSel",almGroup);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almGroup** 是内存消息标记，它包含与所选报警关联的标记的报警组。

另请参阅

GetPropertyM()、.AlarmGroup、.AlarmName

.AlarmAccess 点域

返回与所选报警关联的标记的“访问名”。该报警记录必须从摘要模式下的“分布式报警显示”对象中进行选择。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

用法

```
GetPropertyM("Objectname.AlarmAccess",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

如果用在“触动按钮”QuickScript 中，此语句将与报警关联的标记的“访问名”返回给 almAccess 标记。

```
GetPropertyM("AlmObj_1.AlarmAccess",almAccess);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almAccess** 是内存消息标记，它包含与所选报警关联的标记的“访问名”。

另请参阅

GetPropertyM()、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType

.AlarmProv 点域

返回与所选报警关联的标记的报警供应器。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmProv",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

如果用在“触动按钮”QuickScript 中，此语句将供应器名返回给 almProv 标记。

```
GetPropertyM("AlmObj_1.AlarmProv", almProv);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

AlmObj_1 是“分布式报警显示”对象的名称，**almProv** 是内存消息标记，它包含与所选报警关联的标记的供应器名。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmOprNode、.AlarmPri、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmOprName 点域

返回确认所选报警且已登录的操作员的名称。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmOprName",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

```
GetPropertyM("AlmObj_1.AlarmOprName",almOprName);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almOprName** 是内存消息标记，它包含与标记关联的报警所对应的操作员的名称。

如果用在“触动按钮”QuickScript 中，此语句将操作员的名称返回给 almOprName 标记。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprNode、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType, .AlarmValue

.AlarmOprNode 点域

返回与所选报警关联的标记的操作员节点。该报警必须从摘要模式下的“分布式报警显示”对象中进行选择。

在“终端服务”环境下确认报警时，“操作员节点”是相应的操作员建立“终端服务”会话的客户端机器的名称。如果无法检索节点名，则使用节点的 IP 地址代替。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmOprNode",TagName);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

```
GetPropertyM("AlmObj_1.AlarmOprNode",almOprNode);
```

其中，AlmObj_1 是“分布式报警显示”对象的名称，**almOprNode** 是内存消息标记，它包含与所选报警关联的标记的操作员节点名。

如果用在“触动按钮”QuickScript 中，此语句将操作员节点返回给 almOprNode 标记。

另请参阅

GetPropertyM()、.AlarmAccess、.AlarmClass、.AlarmComment、.AlarmDate、.AlarmLimit、.AlarmName、.AlarmOprName、.AlarmPri、.AlarmProv、.AlarmState、.AlarmTime、.AlarmType、.AlarmValue

.AlarmComment 点域

返回报警注释，这是描述报警（而非标记）的可读写文本字符串。缺省情况下，此注释在新应用程序中为空。

不过，在将旧有的 InTouch 应用程序转换为 InTouch 7.11 或更高的版本时，为保证后向兼容性，会将标记注释复制到 .AlarmComment 点域中。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmComment",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

数据类型

字符串（只读）

示例

下例返回“分布式报警显示”对象 AlmObj_1 中所选标记的报警注释，并将它放入 almComment 标记中：

```
GetPropertyM("AlmObj_1.AlarmComment", almComment);
```

另请参阅

GetPropertyM(), .AlarmAccess, .AlarmClass, .AlarmDate, .AlarmLimit, .AlarmName, .AlarmOprName, .AlarmOprNode, .AlarmPri, .AlarmProv, .AlarmState, .AlarmTime, .AlarmType, .AlarmValue

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

设置报警查询

使用以下查询函数从报警内存中检索记录。

- almDefQuery() 函数
- almQuery() 函数
- almSetQueryByName() 函数

almDefQuery() 函数

使用缺省属性执行查询，以更新指定的“分布式报警显示”对象。

类别

报警

语法

```
[Result=]almDefQuery(ObjectName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

附注

缺省查询属性在 WindowMaker 中开发“分布式报警显示”对象时指定。

示例

```
almDefQuery("AlmObj_1");
```

另请参阅

almQuery(), almSetQueryByName()

almQuery() 函数

执行查询以更新指定的“分布式报警显示”对象，并使用指定的参数。

类别

报警

语法

```
[Result=]almQuery(ObjectName,AlarmList,FromPri,ToPri,State,Type);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

AlarmList
设置“报警查询/名称管理器”的别名以执行查询，例如 "\intouch!$System" 或消息标记。

FromPri
要显示的报警的开始优先级。例如，100 或整型标记。

ToPri
要显示的报警的结束优先级。例如，900 或整型标记。

State

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

指定要显示报警的类型。例如，“未确认”或“消息”标记。有效状态包括“全部”、“未确认”或“确认”。

Type
更新的显示对象中出现的报警记录的类型：
"Hist" = 历史报警
"Summ" = 摘要报警

示例

此语句检索 MyAlarmListGroup 中指定的、优先级为 500 到 600 的所有历史报警。这些报警出现在 AlmObj_1 报警显示对象中。

```
almQuery("AlmObj_1","MyAlarmListGroup",500,600,"All","Hist");
```

在本例中， MyAlarmListGroup 是从“名称管理器”设置中配置的报警列表。

另请参阅

almDefQuery(), almSetQueryByName()

almSetQueryByName() 函数

使用用户自定义查询收藏夹文件的参数，对“分布式报警显示”对象的指定实例启动新的报警查询。

类别

报警

语法

```
[Result=]almSetQueryByName(ObjectName, QueryName);
```

参数

ObjectName
报警对象的名称。例如， AlmObj_1。

QueryName
使用“查询收藏夹”创建的查询的名称。

附注

这是“分布式报警显示”对象特定实例的查询。屏幕上可能有多个这样的显示对象，每个都有自己的查询。

示例

本例使用名为“Turbine Queries”的查询中的参数开始新的查询。

```
almSetQueryByName("AlmObj_1","Turbine Queries");
```

另请参阅

almQuery(), almDefQuery()

检查当前查询属性

使用以下点域返回报警内存查询的状态。

- .AlarmGroup 点域
- .QueryType 点域
- .QueryState 点域
- .Successful 点域

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- .PriFrom 点域
- .PriTo 点域

.AlarmGroup 点域

包含用于填充“分布式报警显示”对象的当前查询。

类别

报警

用法

```
[ErrorNumber=]GetPropertyM( "ObjectName.AlarmGroup",TagName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

TagName
任何消息标记。

附注

此只读点域包含指定的“分布式报警显示”对象使用的当前报警查询。此查询可以是报警组的列表，也可以是报警供应器的直接引用。

数据类型

字符串（只读）

示例

此语句将“分布式报警显示”对象 AlmObj_1 使用的当前报警查询返回给 **CurrentQuery** 标记：

```
GetPropertyM("AlmObj_1.AlarmGroup",CurrentQuery);
```

另请参阅

GetPropertyM(), .AlarmGroupSel, .AlarmName

.QueryType 点域

显示当前报警查询的类型。

类别

报警

用法

```
[ErrorNumber=]GetPropertyI( "ObjectName.QueryType",TagName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

TagName
任何整型标记

附注

此只读点域包含指定的“分布式报警显示”对象当前使用的查询类型。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

整型（只读）

有效值

1 = 历史

2 = 摘要

示例

以下语句将“分布式报警显示”对象 AlmObj_1 的当前查询类型返回给 AlmQueryType 标记：

```
GetPropertyI("AlmObj_1.QueryType",AlmQueryType);
```

另请参阅

GetPropertyI(), .QueryState

.QueryState 点域

显示当前报警状态查询过滤器。

类别

报警

用法

```
[ErrorNumber=]GetPropertyI( "ObjectName.QueryState",TagName);
```

参数

ObjectName
报警对象的名称。例如， AlmObj_1。

TagName
任何整型标记

附注

此只读点域包含指定的“分布式报警显示”对象当前使用的查询过滤器。

数据类型

整型（只读）

有效值

0 = 全部

1 = 未确认

2 = 已确认

示例

以下语句将“分布式报警显示”对象 AlmObj_1 的当前查询过滤器返回给 AlmQueryState 标记：

```
GetPropertyI("AlmObj_1.QueryState", AlmQueryState);
```

另请参阅

GetPropertyI(), .QueryType

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.Successful 点域

指出当前查询是否成功。

类别

报警

用法

```
[ErrorNumber=]GetPropertyD( "ObjectName.Successful",TagName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

TagName
离散标记，处理函数时用于存放属性值。

附注

此只读点域包含指定的“分布式报警显示”对象使用的上一个查询的状态。

数据类型

离散（只读）

有效值

0 = 查询错误

1 = 查询成功

示例

以下语句将“分布式报警显示”对象 AlmObj_1 的上一个查询的状态返回给 AlmFlag 标记。

```
GetPropertyD("AlmObj_1.Successful",AlmFlag);
```

另请参阅

GetPropertyD()

.PriFrom 点域

返回当前查询使用的报警优先级范围的最小值。

类别

报警

用法

```
[ErrorNumber=]GetPropertyI("ObjectName.PriFrom", Tagname);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
任何整型标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

整型（只读）

示例

以下语句将“分布式报警显示”对象 AlmObj_1 所使用的查询的最小优先级值返回给 **MinPri** 整型标记：

```
GetPropertyI("AlmObj_1.PriFrom",MinPri);
```

另请参阅

GetPropertyI(), .PriTo, .AlarmPri

.PriTo 点域

包含当前查询使用的报警优先级范围的最大值。

用法

```
[ErrorNumber=]GetPropertyI("ObjectName.PriTo", Tagname);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
整型标记，处理函数时用于存放属性值。

数据类型

整型（只读）

示例

以下语句将“分布式报警显示”对象 AlmObj_1 所使用的查询的最大优先级值返回给 MaxPri 整型标记：

```
GetPropertyI("AlmObj_1.PriTo",MaxPri);
```

另请参阅

GetPropertyI(), .PriFrom, .AlarmPri

检查分布式报警显示对象的更新

使用以下点域找出“分布式报警显示”对象是否包含当前的所有报警，是否存在待处理的更新。

- .ListChanged 点域
- .PendingUpdates 点域

.ListChanged 点域

指出“分布式报警显示”对象是否有任何新的报警或更新。

类别

报警

用法

```
[ErrorNumber=]GetPropertyD("ObjectName.ListChanged",TagName);
```

参数

ObjectName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

报警对象的名称。例如，AlmObj_1。

TagName
离散标记，处理函数时用于存放属性值。

附注

此只读点域包含有关以下状态的信息：是否存在任何更改需要在“分布式报警显示”对象中进行更新。此属性在读取时自动重置。

数据类型

离散型（只读）

有效值

0 = 显示对象没有新的报警或更新

1 = 显示对象有新的更新

示例

以下语句将“分布式报警显示”对象 AlmObj_1 的任何新报警或更新的状态返回给 AlmDispStat 标记：

```
GetPropertyD("AlmObj_1.ListChanged",AlmDispStat);
```

另请参阅

GetPropertyD()

.PendingUpdates 点域

指出“分布式报警显示”对象待处理的更新数。冻结显示并创建新的报警记录时，通常有待处理的更新。这些更新不显示，但是待处理更新计数会增加。

类别

报警

用法

```
[ErrorMessage=]GetPropertyI( "ObjectName.PendingUpdates", TagName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

TagName
整型标记，处理函数时用于存放属性值。

附注

此只读点域包含指定的“分布式报警显示”对象中待处理的更新数。任何大于零的数值都表示“分布式报警显示”对象有新的报警数据。每次刷新对象时，此值都会重置。

数据类型

整型（只读）

示例

如果“分布式报警显示”对象 AlmObj_1 有任何待处理的更新，则以下语句会将整数 1 或更大的数值返回给 AlarmPendingUpdates 标记：

```
GetPropertyI( "AlmObj_1.PendingUpdates",AlarmPendingUpdates);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

GetPropertyI()

抑制报警

在与排除标准匹配的报警接收器上，“分布式报警显示”对象可以抑制一个或多个报警。如果某个报警与排除标准匹配，则它不会出现在显示对象的实例中。

您可以使用 QuickScript 函数来抑制报警。

- almSuppressAll() 函数
- almUnsuppressAll() 函数
- almSuppressDisplay() 函数
- almSuppressGroup() 函数
- almSuppressPriority() 函数
- almSuppressTag() 函数
- almSuppressSelected() 函数
- almSuppressSelectedGroup() 函数
- almSuppressSelectedPriority() 函数
- almSuppressSelectedTag() 函数
- almSuppressRetain() 函数
- .SuppressRetain 点域

almSuppressAll() 函数

抑制显示当前查询中的当前以及将来要发生的这些报警，包括摘要模式下“分布式报警显示”对象中当前未显示的那些报警。

语法

```
[Result=] almSuppressAll(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

附注

此函数类似于 almAckAll()，通过确定希望在显示对象中查看的所有报警，并上下滚动来查看所有这些报警，以确定要抑制哪些报警。

示例

```
almSuppressAll("AlmObj_1");
```

另请参阅

almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

almUnsuppressAll() 函数

撤销抑制所有被抑制的报警。

语法

```
[Result=] almUnSuppressAll(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

示例

```
almUnSuppressAll("AlmObj_1");
```

另请参阅

almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag()

almSuppressDisplay() 函数

抑制显示摘要模式下“分布式报警显示”对象中当前的可见报警以及将来要发生的这些报警。

语法

```
[Result=]almSuppressDisplay(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

附注

此函数类似于相应的 almAckDisplay() 函数，通过确定当前显示的所有报警来确定要抑制的报警。

示例

```
almSuppressDisplay("AlmObj_1");
```

almSuppressGroup() 函数

抑制显示当前以及将来要发生的任何报警，它们具有指定的供应器与组名。

语法

```
[Result=]almSuppressGroup(ObjectName, ApplicationName,GroupName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
应用程序的名称。例如，\\node1\InTouch

GroupName
报警组的名称。例如，$System

示例

```
almSuppressGroup( "AlmObj_1","\InTouch","Turbines");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

almSuppressAll(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()

almSuppressPriority() 函数

抑制显示当前以及将来要发生的任何特定报警：它们在指定的优先级范围内，并且具有相同的供应器名与“组名”。

语法

```
[Result=]almSuppressPriority(ObjectName, ApplicationName, GroupName, FromPri, ToPri);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
应用程序的名称。例如，\\node1\InTouch

GroupName
“组”的名称。例如，$System

FromPri
报警的开始优先级。例如，100 或整型标记。

ToPri
报警的结束优先级。例如，900 或整型标记。

示例

```
almSuppressPriority("AlmObj_1","\\node1\Intouch", "Turbines",10,100);
```

另请参阅

almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()

almSuppressTag() 函数

抑制显示当前以及将来要发生的任何报警：它们属于指定的标记名，并且具有相同的供应器名、组名以及优先级范围。

类别

报警

语法

```
[Result=]almSuppressTag(ObjectName, ApplicationName, GroupName, TagName, FromPri, ToPri,
AlarmClass, AlarmType);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

ApplicationName
应用程序的名称。例如，\\node1\InTouch

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

GroupName
“组”的名称。例如，$System

TagName
报警标记的名称。

FromPri
报警的开始优先级。例如，100 或整型标记。

ToPri
报警的结束优先级。例如，900 或整型标记。

AlarmClass
报警的类。例如，“Value”。

AlarmType
报警的类型。例如，“HiHi”。

示例

```
almSuppressTag("AlmObj_1","\\node1\Intouch", "Turbines","Valve1",10,100,"Value","LoLo");
```

另请参阅

almSuppressAll(), almSuppressGroup(), almSuppressDisplay(), almSuppressPriority(), almSuppressRetain(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()

almSuppressSelected() 函数

抑制显示摘要模式下“分布式报警显示”对象中当前所选的报警以及将来要发生的这些报警。

类别

报警

语法

```
[Result=]almSuppressSelected(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

附注

此函数类似于 **almAckSelect()** 函数，根据显示对象中所选的报警来确定要抑制的报警。

示例

```
almSuppressSelected("AlmObj_1");
```

另请参阅

almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()

almSuppressSelectedGroup() 函数

抑制显示当前以及将来要发生的特定报警：它们与一个或多个所选的报警属于相同的组，并且在指定的“分布式报警显示”对象中具有相同的供应器名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

语法

```
[Result=]almSuppressSelectedGroup(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

附注

此函数类似于 almAckSelectedGroup() 函数，首先确定所选的报警，然后确定这些报警所属的组，并抑制那些组中的报警将来的实例。

示例

```
almSuppressSelectedGroup("AlmObj_1");
```

另请参阅

almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressSelected(), almSuppressSelectedPriority(), almSuppressSelectedTag()

almSuppressSelectedPriority() 函数

抑制显示当前以及将来要发生的特定报警：它们与一个或多个所选的报警属于相同的优先级，并且在指定的“分布式报警显示”对象中具有相同的供应器名与“组”标记。

类别

报警

语法

```
[Result=]almSuppressSelectedPriority(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

附注

优先级是根据所选报警记录的最小与最大优先级计算得出的。

此函数类似于 almAckSelectedPriority() 函数，首先确定显示对象中的所选报警，然后确定那些报警相应的优先级，并抑制将来要发生的具有相同优先级的报警。

示例

```
almSuppressSelectedPriority("AlmObj_1");
```

另请参阅

almSuppressAll(), almSuppressGroup(), almSuppressTagName(), almSuppressDisplay(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedTag(), almAckSelectedPriority()

almSuppressSelectedTag() 函数

抑制显示当前以及将来要发生的任何特定报警：它们与一个或多个所选的报警具有相同的“标记名”，并且具有相同的供应器名、组名以及优先级范围。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

语法

```
[Result=]almSuppressSelectedTag(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

示例

```
almSuppressSelectedTag("AlmObj_1");
```

另请参阅

almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressSelectedAlarm(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almAckSelectedTag(), almUnSuppressAll()

almSuppressRetain() 函数

抑制后续查询产生的所有报警。

类别

报警

语法

```
[Result=]almSuppressRetain(ObjectName,SuppressionRetainFlag);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

SuppressionRetainFlag
任何离散或模拟标记、0 或非零值。如果为后续查询保留抑制信息，则为 TRUE；否则为 FALSE。

附注

如果标帜为 0，则报警查询改变时，会删除抑制过滤器。

示例

```
almSuppressRetain("AlmObj_1", 0);
```

另请参阅

almSuppressAll(), almSuppressGroup(), almSuppressTag(), almSuppressDisplay(), almSuppressPriority(), almSuppressSelected(), almSuppressSelectedGroup(), almSuppressSelectedPriority(), almSuppressSelectedTag(), almUnSuppressAll()

.SuppressRetain 点域

读取/写入“分布式报警显示”对象保持抑制的功能的状态。

类别

报警

用法

```
[ErrorNumber=]GetPropertyD( "ObjectName.SuppressRetain",TagName);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
[ErrorNumber=]SetPropertyD( "ObjectName.SuppressRetain",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

Tagname
离散标记，处理脚本时用于存放属性值。

数据类型

离散（可读写）

有效值

0 = 关闭保持

1 = 打开保持

示例

以下语句使用 SupRtn 离散标记设置“AlmObj_1”抑制保持器的状态。

```
SetPropertyD("AlmObj_1.SuppressRetain", SupRtn);
```

另请参阅

GetPropertyD(), SetProperty()

滚动报警显示

使用以下函数与点域垂直或水平滚动“分布式报警显示”对象中的报警列表。您也可以冻结显示对象。

- almMoveWindow() 函数
- .Freeze 点域
- .PrevPage 点域
- .NextPage 点域

almMoveWindow() 函数

垂直或水平滚动“分布式报警显示”对象的报警列表。

类别

报警

语法

```
[Result=]almMoveWindow(ObjectName,Option,Repeat);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

Option
要执行的滚动操作的类型：

类型	描述
LineDn	下移一行。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类型	描述
LineUp	上移一行。
PageDn	下翻一页。
PageUp	上翻一页。
Top	滚动到列表顶部。
Bottom	滚动到列表底部。
PageRt	右移一页。
PageLf	左移一页。
Right	滚动到列表末尾（右侧）。
Left	滚动到列表开头（左侧）。

Repeat
应重复此操作的次数。

示例

```
almMoveWindow("AlmObj_1", "Bottom", 0);
almMoveWindow("AlmObj_1", "LineDn", 3);
almMoveWindow("AlmObj_1", "PageUp", 0);
```

.Freeze 点域

.Freeze 点域读取冻结状态，或冻结/撤销冻结“分布式报警显示”对象。

类别

报警

用法

```
[ErrorNumber=]GetPropertyD("ObjectName.Freeze", TagName);
[ErrorNumber=]SetPropertyD("ObjectName.Freeze", TagName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

TagName
离散标记，处理函数时用于存放属性值。

附注

包含或更改“分布式报警显示”对象冻结状态的可读写点域。冻结报警显示对象时，显示的报警无法更新，也无法添加新的报警。冻结对报警是否闪烁没有影响。

数据类型

离散（可读写）

有效值

0 = 关闭冻结

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1 = 打开冻结

示例

以下语句使用 AlmFreeze 离散标记设置“AlmObj_1”的 Freeze 属性。

```
SetPropertyD("AlmObj_1.Freeze",AlmFreeze);
```

另请参阅

GetPropertyD(), SetPropertyD()

.PrevPage 点域

将“分布式报警显示”对象向上滚动一页（一整屏的报警）。

类别

报警

用法

```
[ErrorNumber=]SetPropertyD("ObjectName.PrevPage",0);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

附注

设置此属性时，“分布式报警显示”对象显示上一页。显示上一页之后，变量自动设置为 1，除非已到达列表顶部。这种情况下，值保持为 0。

数据类型

离散（可读写）

另请参阅

GetPropertyD(), SetPropertyD(), .NextPage, .PageNum, .TotalPages

.NextPage 点域

将“分布式报警显示”对象向下滚动一页（一整屏的报警）。

类别

报警

用法

```
[ErrorNumber=]SetPropertyD("ObjectName.NextPage",0);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

附注

设置此属性时，“分布式报警显示”对象显示上一页。显示下一页之后，变量自动设置为 1，除非已到达列表底部。这种情况下，值保持为 0。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

离散（可读写）

另请参阅

GetPropertyD(), SetPropertyD(), .PrevPage, .PageNum, .TotalPages

显示报警统计与计数

使用以下函数与点域显示当前“分布式报警显示”对象的有关统计信息。

- almShowStats() 函数
- .PageNum 点域
- .TotalPages 点域
- .NumAlarms 点域
- .ProvidersReq 点域
- .ProvidersRet 点域

almShowStats() 函数

显示指定的"分布式报警显示"对象的**报警统计**对话框。

类别

报警

语法

```
[Result=]almShowStats(ObjectName);
```

参数

ObjectName
报警对象的名称。例如，AlmObj_1。

示例

```
almShowStats("AlmObj_1");
```

.PageNum 点域

包含报警对象中显示的当前页码。

类别

报警

语法

```
[ErrorNumber=]GetPropertyI("ObjectName.PageNum",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
整型标记，存放“分布式报警显示”对象中当前显示的页面的编号。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

此只读点域返回指定的“分布式报警显示”对象中当前显示的页面的编号。

数据类型

整型（只读）

示例

以下语句将“分布式报警显示”对象 AlmObj_1 中当前显示的页面的编号返回给 **AlarmPage** 整型标记：

```
GetPropertyI("AlmObj_1.PageNum",AlarmPage);
```

另请参阅

GetPropertyI(), .NextPage, .PrevPage, .TotalPages

.TotalPages 点域

包含“分布式报警显示”对象中的总页数。

类别

报警

语法

```
[ErrorNum=]GetPropertyI("ObjectName.TotalPages", TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如， AlmObj_1。

TagName
整型标记，检索指定的“分布式报警显示”对象中包含的总报警页面数。

附注

此点域返回指定的“分布式报警显示”对象中包含的总报警页面数。一个页面即任何给定时刻屏幕上的对象中显示的所有报警。

数据类型

整型（只读）

示例

以下语句将“分布式报警显示”对象 AlmObj_1 中包含的总页数返回给 Alm**TotalPages** 整型标记：

```
GetPropertyI( "AlmObj_1.TotalPages",AlmTotalPages);
```

另请参阅

GetPropertyI(), NextPage, PrevPage, PageNum

.NumAlarms 点域

包含“分布式报警显示”对象中的报警数。

类别

报警

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
[ErrorNum=]GetPropertyI("ObjectName.NumAlarms", Tagname);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
整型标记，存放指定的“分布式报警显示”对象中注册的当前报警数。这不仅包括那些已显示的报警，还包括已注册的所有报警。

附注

此只读点域返回“分布式报警显示”对象中的总报警数。

数据类型

整型（只读）

示例

以下语句将“分布式报警显示”对象 AlmObj_1 所使用的当前报警数返回给 **AlarmCount** 整型标记：

```
GetPropertyI("AlmObj_1.NumAlarms",AlarmCount);
```

另请参阅

GetPropertyI()

.ProvidersReq 点域

包含指定的“分布式报警显示”对象所使用的当前查询所需的报警供应器数。

类别

报警

语法

```
[ErrorNumber=]GetPropertyI( "ObjectName.ProvidersReq",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
整型标记，存放指定的“分布式报警显示”对象中注册的当前报警供应器数。这不仅包括那些已显示的报警，还包括已注册的所有报警。

数据类型

整型（只读）

示例

以下语句返回“分布式报警显示”对象 "AlmObj_1" 所使用的当前查询所需的报警供应器数。此值写入 TotalProv 整型标记：

```
GetPropertyI("AlmObj_1.ProvidersReq",TotalProv);
```

另请参阅

GetPropertyI(), .ProvidersRet

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.ProvidersRet 点域

包含指定的“分布式报警显示”对象所使用的当前查询返回的报警供应器数。

类别

报警

用法

```
[ErrorNumber=]GetPropertyI ("ObjectName.ProvidersRet",TagName);
```

参数

ObjectName
“分布式报警显示”对象的名称。例如，AlmObj_1。

TagName
整型标记，存放已经成功将其报警返回给指定的“分布式报警显示”对象的报警供应器数。

附注

此只读点域包含指定的“分布式报警显示”对象所使用的当前查询返回的报警供应器数。

数据类型

整型（只读）

示例

以下语句返回特定的报警供应器的数量，这些供应器已成功将其报警返回给“分布式报警显示”对象 AlmObj_1。此值写入 **RetProv** 整型标记：

```
GetPropertyI("AlmObj_1.ProvidersRet",RetProv);
```

另请参阅

GetPropertyI(), .ProvidersReq

错误描述

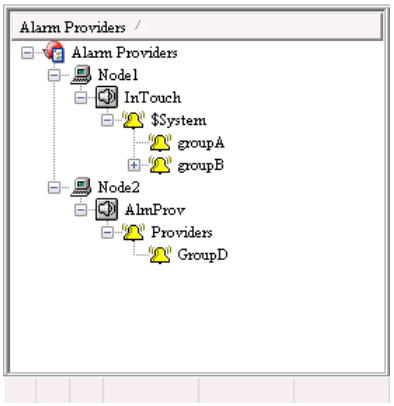
下表介绍各个错误号。如果返回此表中没有的某个编号，则该错误是未知错误。

错误号	描述
0	成功
-1	一般错误
-2	可用内存不足
-3	属性为只读
-4	指定的项目已经存在
-5	未知对象名
-6	未知属性名

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

查看报警层次结构

Alarm Tree Viewer ActiveX 控件显示报警查询所选的报警供应器的报警组层次结构。Alarm Tree Viewer 控件中出现的项目包括报警供应器、节点以及组。



您可以通过使用 Alarm Tree Viewer 控件来提高 Alarm Viewer 控件的有用性。您可以创建一个脚本，使操作员在 Alarm Tree Viewer 控件中选择报警供应器时，Alarm Viewer 控件会查询新的报警供应器。

您可以配置 Alarm Tree Viewer 控件的显示方式以及显示的数据。如需详细信息，请参阅配置 Alarm Tree Viewer 控件。

完成 Alarm Tree Viewer 控件的配置时，可以通过以下操作来修改要查看的数据：

- 按名称给数据排序。
- 更新目录树。
- 执行其它查询。

如需有关 ActiveX 控件的详细信息，请参阅 ActiveX 控件。

配置 Alarm Tree Viewer 控件

您可以为 Alarm Tree Viewer 控件配置以下属性：

- 一般控件外观，包括颜色
- 文本字体
- 自动刷新
- 用户可以在运行时访问的功能
- 要显示的供应器与组
- 自定义的保存查询
- 报警组的排序顺序

您可以从 WindowMaker 中以及在 Alarm Tree Viewer 控件运行期间配置这些选项。

配置外观与颜色

配置 Alarm Tree Viewer 控件的视觉外观时，可以：

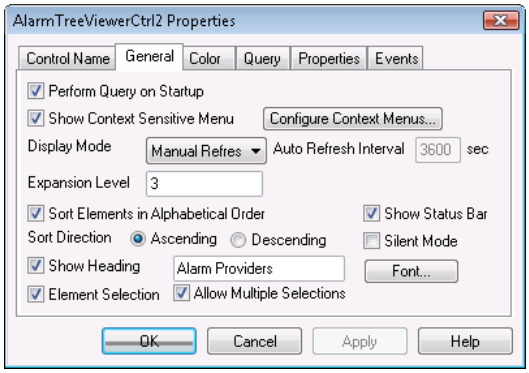
- 包含状态栏。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 包含列标题。
- 设置可视化元素的颜色。

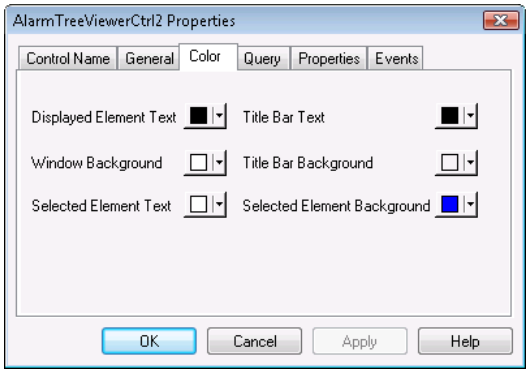
配置外观

1. 使用鼠标右键单击 Alarm Tree Viewer 控件，然后单击属性。此时出现 **AlarmTreeViewerCtrl** 属性对话框。
2. 选择常规选项卡。



3. 配置 Alarm Tree Viewer 控件呈现给运行时用户的方式。执行以下任意操作：
   - 选中在启动时执行查询复选框，以使用缺省查询属性自动更新目录树。否则用户必须运行刷新命令才能更新目录树。
   - 选中显示上下文相关菜单复选框以激活快捷菜单。选择“配置上下文菜单”以配置要在菜单上出现的命令。如需详细信息，请参阅控制用户可以在运行时访问的功能。
   - 在显示模式列表中，选择所需的目录树刷新方式。对于自动刷新，请在自动刷新间隔框中输入刷新间隔。范围是 5 到 32767 秒。
   - 在展开级别框中，输入目录树的展开级数。这确定手工刷新控件时报警目录树打开的报警组分支级别。值为 1 时仅显示供应器，值为 2 时显示供应器的直接报警组，依此类推。
   - 选中按字母顺序排列元素复选框，以便按字母顺序给目录树元素排序。选择升序或降序作为排序方向。
   - 选中显示标题复选框，以便在层次结构上显示标题。在方框中，输入标题栏文本。
   - 选中显示状态栏复选框，以便在 Alarm Tree Viewer 控件底部显示状态栏。
   - 选择“字体”以配置目录树的字体属性。此时出现标准的 Windows 字体对话框。
   - 选中选择元素复选框，以允许用户选择目录树中的元素。
   - 选中允许多重选择复选框，以允许用户使用 CTRL 与 SHIFT 键选择一个或多个元素。
   - 选中“无提示模式”复选框，以防止 Alarm Tree Viewer 控件显示运行时错误消息。错误消息总是发送到 Logger 中。
4. 选择应用。
5. 选择颜色选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



1. 选择调色板按钮，以便给 Alarm Tree Viewer 控件的可视化元素指定颜色。

   您可以设置标题栏文本、窗口背景、所选元素文本以及所选元素背景的颜色。

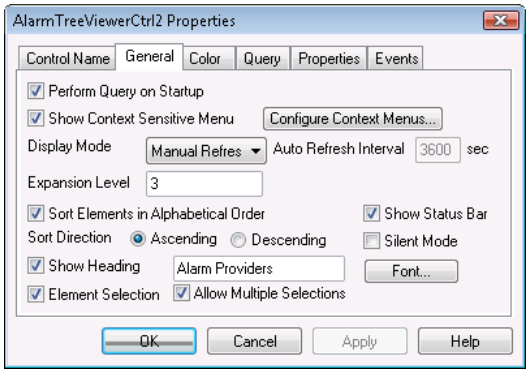
2. 选择应用。

配置字体

您可以为 Alarm Tree Viewer 控件配置文本外观。

配置字体

1. 使用鼠标右键单击 Alarm Tree Viewer 控件，然后单击属性。此时出现 **AlarmTreeViewerCtrl** 属性对话框。
2. 选择常规选项卡。



3. 选择字体。此时出现标准的 Windows 字体对话框。配置字体，然后单击确定。
4. 选择确定。

配置自动刷新

您可以将 Alarm Tree Viewer 控件配置为在运行时自动刷新。否则，操作员必须手工刷新 Alarm Tree Viewer 控件。

配置自动刷新

1. 使用鼠标右键单击 Alarm Tree Viewer 控件，然后单击属性。

   此时出现 **AlarmTreeViewerCtrl** 属性对话框。

2. 选择常规选项卡。
3. 在显示模式列表中，选择所需的目录树刷新方式，即手工刷新或自动刷新。对于自动刷新，请在自动刷新间隔框中输入目录树刷新间隔。范围是 5 到 32767 秒。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 选择应用。

调整 Alarm Tree View 刷新

刷新 Alarm Tree View 时，该目录树中列出的报警供应器可能与查询中所包括的报警供应器不匹配。如果远程报警供应器有许多报警组层次结构，或者如果与报警供应器的链接很慢，可能会发生这种情况。

要确保 Alarm Tree View 正常刷新，可以在 InTouch.ini 文件中的 [InTouch] 部分中输入和调整三个设置。这些设置指定等待查询完全响应的时间以及在该期间内重新尝试报警查询的频率。

缺省情况下，InTouch.ini 文件中不包含这些设置，必须手动输入并进行调整。InTouch.ini 文件中没有明确输入这些设置的时候，将使用其缺省值。

AlarmTreeFastRetryMax

此设置决定查询提交后多长时间。执行快速重试（每秒 1 次）目录树数据检索。这些值以秒为单位。

允许的值为：1 到 32767

缺省值：10

示例：

```
[InTouch]
AlarmTreeFastRetryMax=5
```

AlarmTreeSlowRetryInterval

快速重试完成后，此设置决定执行其它重试目录树数据检索的频率（以秒为单位）。

允许的值为：1 到 32767

缺省值：5

示例：

```
[InTouch]
AlarmTreeSlowRetryInterval=10
```

AlarmTreeTotalRetryMax

此设置指定执行两种类型重试的最大期间（以秒为单位）。

允许的值为：1 到 32767

缺省值：30

示例：

```
[InTouch]
AlarmTreeTotalRetryMax=60
```

使用缺省设置的重试行为

例如，使用缺省重试设置，以及 30 秒的最大缺省重试期间：

- 在快速重试间隔期间，将会重试检索 10 秒，每秒一次。
- 在慢速重试间隔期间，将会每 5 秒重试检索一次，再重试 20 秒。

刷新视为完成时

显示将继续重试，直到：

- 完全连接到报警查询中指定的所有供应器，以及
- 显示所有所需供应器的层次结构目录树

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

每次提交查询时，显示刷新计时器会重置为其修改值或缺省值。

管理运行时对功能的访问权

Alarm Tree Viewer 控件包含一个快捷菜单，通过使用鼠标右键单击控件，操作员可以在运行时打开它。您可以配置要在菜单上出现的命令。

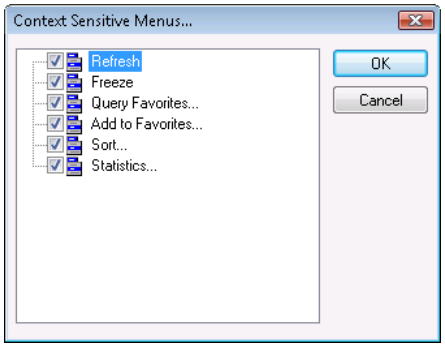
配置运行时快捷菜单

1. 使用鼠标右键单击 Alarm Tree Viewer 控件，然后单击属性。

   此时出现 **AlarmTreeViewerCtrl** 属性对话框。

2. 选择常规选项卡。
3. 选择显示上下文相关菜单复选框以激活快捷菜单。
4. 选择配置上下文菜单。

   此时出现上下文相关菜单对话框。



5. 选中要在快捷菜单中出现的每个命令对应的复选框。您至少必须选择一个快捷菜单命令。

命令	描述
刷新	刷新 Alarm Tree Viewer 控件中显示的数据。
冻结	可用于切换目录树的冻结/撤销冻结模式。
查询收藏夹	显示**报警查询**对话框，以便从可用列表中选择一个查询收藏夹。
添加到收藏夹	可用于从**添加查询**对话框中添加新的查询。
排序	显示**排序**对话框，以按照升序或降序给 Alarm Tree Viewer 控件数据排序。
统计	显示**报警统计**对话框，列出 Alarm Tree Viewer 控件中显示的内容占当前所检索的报警供应器的百分比。

6. 选择确定以关闭上下文相关菜单对话框。
7. 选择应用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置要显示的供应器与组

您可以为属于 Alarm Tree Viewer 控件的报警供应器与组配置报警查询。报警查询是由空格分隔的一个或多个报警供应器的列表。报警供应器的有效语法如下。

报警供应器的完整路径：

\\节点\供应器名

本地报警供应器的路径：

\供应器名

对于多个查询，请使用空格分隔每个查询。例如：

\InTouch \\Node17\InTouch \\MyNode\InTouch

缺省的报警查询是 \InTouch。不得将标记用于报警查询。

如果查询多个报警组，并且稍后取消部署了一个或多个组，则 Alarm Tree Viewer 控件不自动更新以便从视图中删除这些组。您必须停止并重新启动报警供应器，才能取消它的注册。

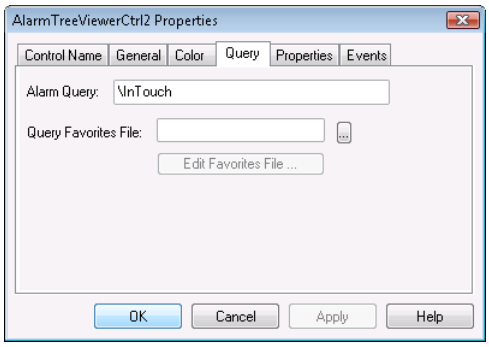
如果通过在报警查询中指定 \Galaxy 来查询 Galaxy, 则会显示 Galaxy 中部署的所有 InTouch 报警供应器。例如：

\\Node\Galaxy!Area[name]

如果从具有多个报警供应器的节点查询信息，并且这些供应器包含同名的组，则目录树中显示最后一个报警供应器的记录。

配置报警查询

1. 使用鼠标右键单击 Alarm Tree Viewer 控件, 然后选择属性。此时出现 **AlarmTreeViewerCtrl** 属性对话框。
2. 选择查询选项卡。



3. 在报警查询框中，输入初始报警查询的路径。
4. 选择应用。

使用查询收藏夹创建自定义的保存查询

您可以配置一个查询收藏夹列表，供操作员从快捷菜单中进行选择。

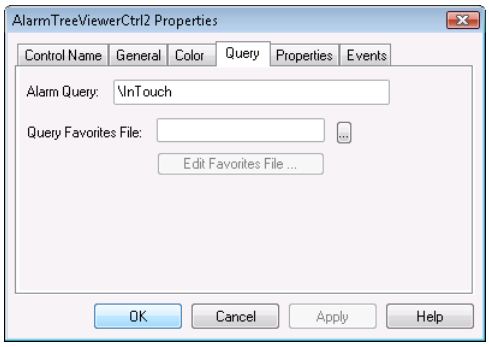
查询文件可以保存到任何文件夹，而不必保存到 InTouch 应用程序文件夹中。报警查询文件是 .xml 文件。

配置查询收藏夹文件

1. 使用鼠标右键单击 Alarm Tree Viewer 控件, 然后选择属性。此时出现 **AlarmTreeViewerCtrl** 属性对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 选择**查询**选项卡。



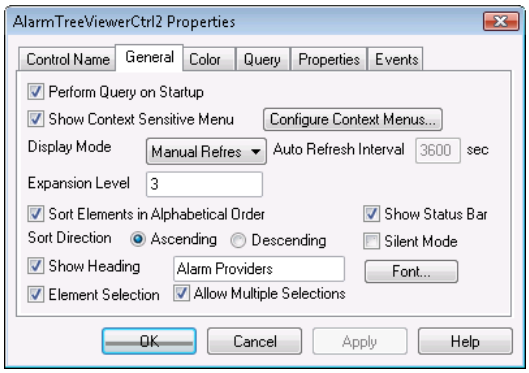
3. 配置**查询**收藏**夹**文件。
   a. 在**查询收藏夹文件**框中，**输**入网**络**路径与文件名，或**选择**省略号按**钮**以**浏览**文件。
   b. 要**编辑查询收藏夹**文件，请选择**编辑收藏夹文件**按**钮**。此**时**打**开报警查询**窗口，可供您在收藏**夹**文件中添加、修改或**删**除**过滤**器。完成**时选择确定**，以保存更改并**关闭**窗口。
4. 选择确定。

配置报警组的排序顺序

在 Alarm Tree Viewer 控件中，**节**点与**报**警**组**可以按字母**顺**序的升序或降序**显**示。

配置报警组的排序顺序

1. 使用鼠**标**右**键单击** Alarm Tree Viewer 控件，然后**选择**“属性”。此**时**出**现 AlarmTreeViewerCtrl 属性**对话框。
2. 选择常**规**选项卡。

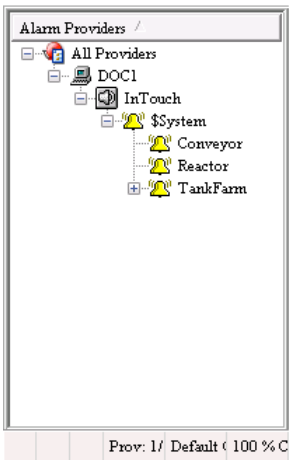


3. 选中**按字母顺序排序元素复选**框，以便按字母**顺**序列出**报警组**。
4. 选择**升序**或**降序**以指定排序方向。
5. 选择确定。

在运行时使用 Alarm Tree Viewer 控件

Alarm Tree Viewer 控件可用于**浏览报**警供**应**器与**报警组**的**层**次**结**构。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



Alarm Tree Viewer 控件可以显示多个节点与报警供应器。

- 节点由计算机图标表示。
- 报警供应器由扬声器图标表示。
- 报警组由钟型图标表示。

选择一个或多个报警组时，可以生成报警查询，以用在 Alarm Tree Viewer 控件与 Alarm DB View 控件中。要选择多个报警组，请在按住 Shift 键的同时选择某个组。要取消选择所有的组，请选择一个空白区域。

根据配置控件的方式，运行时快捷菜单上会出现一个或多个以下命令：

- **刷新** – 强制手工更新报警。
- **冻结** – 停止更新报警。
- **查询收藏夹** – 打开**报警查询**对话框，在其中可以从先前定义的报警查询列表中选择某个报警查询。
- **添加到收藏夹** – 打开**添加查询**对话框，其中已根据所选的“组”（如果有）输入了查询字符串。
- **排序** – 打开**排序**对话框，其中包含一些选项，可以按字母顺序的升序或降序给报警组排序。
- **统计** – 打开**报警统计**对话框，以显示检索的报警供应器的百分比。

了解 Alarm Tree Viewer 控件状态栏信息

Alarm Tree Viewer 控件状态栏出现在窗口底部，显示以下信息：

- 当前查询的名称
- 当前查询的完成百分比状态

使用查询收藏夹

通过使用 Alarm Tree Viewer 控件快捷菜单上的查询收藏夹命令，可以从先前定义的报警查询列表中快速选择并运行报警查询。您也可以创建新的命名查询、编辑或删除现有的查询。

选择并运行报警查询

1. 在运行时使用鼠标右键单击 Alarm Tree Viewer 控件。
2. 选择**查询收藏夹**此时出现“报警查询”对话框。
3. 从当前定义的查询列表中，选择要显示的命名查询。
4. 选择确定。此时 Alarm Tree Viewer 控件显示所选查询的报警组信息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用 Alarm Tree Viewer 控件 ActiveX 属性

您可以使用脚本直接设置 Alarm Tree Viewer 控件属性的值，或是将它指定给 InTouch 标记或 I/O 引用。如需有关设置属性的详细信息，请参阅编写 ActiveX 控件脚本。

下表列出 Alarm Tree Viewer 控件的所有属性。如需有关设置颜色值的详细信息，请参阅配置 ActiveX 控件的颜色。

属性名	用途
AddtoFavoritesMenu	启用或禁用添加到收藏夹快捷菜单命令。
AlarmQuery	显示初始报警查询并允许更改查询。有效语法为 \\<节点>\<供应器> 或 \<供应器>。
ElementSelection	控制操作员在运行时是否可以选择目录树中的元素。
ExpansionLevel	设置手工刷新控件时报警目录树打开的分支级别。值为 1 时仅显示供应器，值为 2 时显示供应器的直接报警组，依此类推。
字体	获取或设置控件中显示的记录与标题的字体。
FreezeMenu	启用或禁用冻结菜单命令。
HeaderText	获取或设置出现在 AlarmTreeViewer 控件的标题中的文本。
MultiSelection	允许在 AlarmTreeViewer 控件中选择多个元素。
QueryFavoritesFile	获取或设置查询收藏夹文件名。
QueryFavoritesMenu	启用或禁用查询收藏夹菜单命令。
QueryStartup	如果选择此属性，则使用缺省查询属性自动更新 AlarmTreeViewer 控件。如果没有选择，则必须重新查询才能更新 Alarm Tree Viewer 控件。
RefreshInterval	获取或设置以秒为单位的控件自动刷新间隔。
RefreshMenu	获取或设置一个值，确定快捷菜单中是否出现刷新命令。
SelTextBackColor	获取或设置所选元素的背景色。
SelTextColor	获取或设置所选元素的文本颜色。
ShowContextMenu	启用或禁用快捷菜单。
ShowHeading	显示或隐藏 Alarm Tree Viewer 控件的标题栏。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性名	用途
ShowStatusBar	获取或设置一个值，确定是否显示状态栏。
SilentMode	获取或设置一个值，确定控件是否处于“无提示”模式。
SortElements	启用或禁用 Alarm Tree Viewer 控件中的排序功能。
SortMenu	启用或禁用排序菜单命令。
SortOrder	获取或设置排序方向。可能值有“升序”与“降序”，分别用 0 与 1 表示。
StatsMenu	启用或禁用统计菜单命令。
TextColor	获取或设置 Alarm Tree Viewer 控件的文本颜色。
标题背景色	获取或设置标题栏背景色。仅在设置了 ShowHeading 属性时可用。
标题前景色	获取或设置标题栏前景色。仅在设置了 ShowHeading 属性时可用。
窗口颜色	获取或设置 Alarm Tree Viewer 控件的窗口背景色。

使用 Alarm Tree Viewer 控件 ActiveX 方法

您可以在脚本中使用 Alarm Tree Viewer 控件方法来：

- 检索有关控件的信息。
- 检索有关报警层次结构中特定项目的信息。
- 冻结控件。
- 创建查询字符串。
- 运行查询。

如需有关调用方法的详细信息，请参阅编写 ActiveX 控件脚本。

检索有关 Alarm Tree Viewer 控件的信息

您可以使用这些方法检索有关 Alarm Tree Viewer 控件的信息。

- AboutBox() 方法, AboutBox() 方法, AboutBox() 方法
- GetElementCount() 方法, GetElementCount() 方法

AboutBox() 方法

显示 Alarm Tree Viewer 的“关于”对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

GetElementCount() 方法

获取目录树中元素的总数。

语法

```
Object.GetElementCount()
```

示例

控件名为 AlarmTreeViewerCtrl1， nTag1 是整型或实型标记。

```
nTag1 = #AlarmTreeViewerCtrl1.GetElementCount();
```

检索有关特定 Alarm Tree Viewer 项目的信息

您可以使用一组方法来检索 Alarm Tree Viewer 控件窗口中所显示的元素的有关信息。

- CheckElementMembership() 方法
- GetElementCount() 方法, GetElementCount() 方法
- GetElementName() 方法
- GetElementPath() 方法
- GetSelectedElementCount() 方法
- GetSelectedElementName() 方法
- GetSelectedElementPath() 方法
- GetSubElementCount() 方法
- GetSubElementName() 方法
- GetSubElementPath() 方法

CheckElementMembership() 方法

检查子目录树元素是否为父目录树元素的一部分。

语法

```
Object.CheckElementMembership(PathName, DescendantElementName, AncestorElementName)
```

参数

PathName
路径的名称。例如， \InTouch 或 \\NodeName。

DescendantElementName
子元素的名称。例如， GroupA。

AncestorElementName
父元素的名称。例如,GroupB。

GetElementCount() 方法

获取目录树中元素的总数。

语法

```
Object.GetElementCount()
```

示例

控件名为 AlarmTreeViewerCtrl1， nTag1 是整型或实型标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
nTag1 = #AlarmTreeViewerCtrl1.GetElementCount();
```

GetElementName() 方法

获取与索引对应的元素名。

语法

```
Object.GetElementName(ElementIndex)
```

参数

ElementIndex
元素的索引。

示例

控件名为 AlarmTreeViewerCtrl1，StrTag 是消息标记。

```
StrTag = #AlarmTreeViewerCtrl1.GetElementName(3);
```

GetElementPath() 方法

获取索引对应的元素路径，一直到指定的展开级别。

语法

```
Object.GetElementPath(ElementIndex, ExpansionLevel)
```

参数

ElementIndex
元素的索引。

ExpansionLevel
展开的级别。

示例

控件名为 AlarmTreeViewerCtrl1，StrTag 是消息标记，返回索引为 17 的元素的路径，最多 4 个展开级别。

```
StrTag = #AlarmTreeViewerCtrl1.GetElementPath(17, 4);
```

GetSelectedElementCount() 方法

获取目录树中所选元素的数量。

语法

```
Object.GetSelectedElementCount()
```

示例

控件名为 AlarmTreeViewerCtrl1，nTag1 是整型或实型标记。

```
nTag1 = #AlarmTreeViewerCtrl1.GetSelectedElementCount();
```

GetSelectedElementName() 方法

获取 Alarm Tree Viewer 控件上所选元素的名称。

语法

```
Object.GetSelectedElementName()
```

示例

控件名为 AlarmTreeViewerCtrl1，StrTag 是消息标记。

```
StrTag = #AlarmTreeViewerCtrl1.GetSelectedElementName();
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

GetSelectedElementPath() 方法

获取到指定的展开级别的所选元素的路径。

语法

```
Object.GetSelectedElementPath(ExpansionLevel)
```

参数

ExpansionLevel
展开的级别。

示例

控件名为 AlarmTreeViewerCtrl1，StrTag 是消息标记。

```
StrTag = #AlarmTreeViewerCtrl1.GetSelectedElementPath(3);
```

GetSubElementCount() 方法

获取指定的元素中的子元素总数。

语法

```
Object.GetSubElementCount(Path, ElementName)
```

参数

Path
路径的名称。例如：
\\节点名\InTouch
如果 Path 参数为空，则 Alarm Tree Viewer 控件查找与指定的元素名匹配的第一个目录树元素。

ElementName
元素的名称。例如，Group1。

示例

控件名为 AlarmTreeViewerCtrl1，nTag1 是整型或实型标记。

```
nTag1 = #AlarmTreeViewerCtrl1.GetSubElementCount("", "Group1" );
nTag1 = #AlarmTreeViewerCtrl1.GetSubElementCount( "\\NodeName", "Group1" );
nTag1 = #AlarmTreeViewerCtrl1.GetSubElementCount( "\InTouch", "Group1" );
nTag1 = #AlarmTreeViewerCtrl1.GetSubElementCount( "\\NodeName\InTouch", "Group1" );
```

GetSubElementName() 方法

对于指定的元素，获取相应索引上的子元素的名称。

语法

```
Object.GetSubElementName(Path, ElementName, ElementIndex)
```

参数

Path
路径的名称。例如：
\\节点名\InTouch
如果 Path 参数为空，则 Alarm Tree Viewer 控件查找与指定的元素名匹配的第一个目录树元素。

ElementName
元素的名称。例如，Group1。

ElementIndex

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素的索引。

示例

控件名为 AlarmTreeViewerCtrl1，StrTag 是消息标记。

```
StrTag = #AlarmTreeViewerCtrl1.GetSubElementName("", "Group1", 1);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementName("\\NodeName", "Group1", 1);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementName("\InTouch", "Group1", 1);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementName("\\NodeName\InTouch", "Group1", 1);
```

GetSubElementPath() 方法

根据元素名的索引来获取子元素的路径（到指定的展开级别）。

语法

```
Object.GetSubElementPath(Path, ElementName, ElementIndex, ExpansionLevel)
```

参数

Path
路径的名称。例如：
\\节点名\InTouch
如果 Path 参数为空，则 Alarm Tree Viewer 控件查找与指定的元素名匹配的第一个目录树元素。

ElementName
元素的名称。例如，Group1。

ElementIndex
元素的索引。

ExpansionLevel
展开的级别。

示例

控件名为 AlarmTreeViewerCtrl1，StrTag 是消息标记。

```
StrTag = #AlarmTreeViewerCtrl1.GetSubElementPath("", "Group1", 1, 3);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementPath("\\NodeName", "Group1", 1, 3);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementPath("\InTouch", "Group1", 1, 3);
StrTag = #AlarmTreeViewerCtrl1.GetSubElementPath("\\NodeName\InTouch", "Group1", 1, 3);
```

冻结目录树

您可以使用 Freeze() 方法防止使用任何进一步的更改来更新 Alarm Tree Viewer 控件目录树。

Freeze() 方法

冻结 Alarm Tree Viewer 控件目录树。

语法

```
Object.Freeze(Frozen)
```

参数

Frozen
控制是否可以更新目录树。

1 = 冻结目录树。

0 = 撤销冻结目录树。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

Tag1 定义为内存离散标记，控件名为 AlarmTreeViewerCtrl1。

```
Tag1 = 1;
#AlarmTreeViewerCtrl1.Freeze(Tag1);
```

从所选元素中创建查询字符串

您可以使用 GetAlarmQueryFromSelection() 方法从 AlarmTreeViewer 控件的所选元素中检索查询字符串。查询字符串随后可以在 Alarm Viewer 控件中动态使用。

GetAlarmQueryFromSelection() 方法

从 Alarm Tree Viewer 控件的所选元素中返回报警查询字符串。

语法

```
Object.GetAlarmQueryFromSelection()
```

示例

控件名为 AlarmTreeViewerCtrl1，StrTag 是消息标记。例如：StrTag 设置为 \\NodeName\InTouch\GroupA。

```
StrTag = #AlarmTreeViewerCtrl1.GetAlarmQueryFromSelection();
```

运行查询

您可以使用一些方法为 Alarm Tree Viewer 控件运行查询，这些方法可以检索查询收藏夹文件中保存的现有查询，或设置指定新报警供应器集合的字符串。

- SetQueryByName() 方法, SetQueryByName() 方法
- SetQueryByString() 方法

SetQueryByName() 方法

将当前查询设置为传递的查询名所指定的查询。查询必须在查询收藏夹文件中。

语法

```
Object.SetQueryByName(QueryName)
```

参数

QueryName
使用查询收藏夹创建的查询的名称。例如，Turbine Queries。

示例

控件名为 AlarmTreeViewerCtrl1。

```
#AlarmTreeViewerCtrl1.SetQueryByName("Turbine Queries");
```

SetQueryByString() 方法

将当前查询设置为一个新字符串，以指定新的“报警供应器”集合。

语法

```
Object.SetQueryByString(NewQuery)
```

参数

NewQuery
包含报警查询的字符串。例如：
\\RepositoryNode\InTouch

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

控件名为 AlarmTreeViewerCtrl1。

```
#AlarmTreeViewerCtrl1.SetQueryByString("\\RepositoryNode\InTouch");
```

使用方法与属性时的错误处理

所有 Alarm Tree Viewer 控件错误消息都发送到 Logger。如果将 Alarm Tree Viewer 控件配置成在无提示模式下运行，则不显示运行时错误。

使用 Alarm Tree Viewer 控件 ActiveX 事件触发脚本

您可以将 QuickScript 指定给 Alarm Tree Viewer 控件事件（如鼠标单击或双击）。该事件发生时，此 QuickScript 便会运行。

Alarm Tree Viewer 控件支持以下事件：

- 单击
- DoubleClick
- ShutDown
- StartUp

Click 事件有一个参数 ClicknElementID，它可以确定在运行时所单击的目录树中的元素。

DoubleClick 事件有一个参数 DoubleClicknElementID，它可以确定在运行时所双击的目录树中的元素。

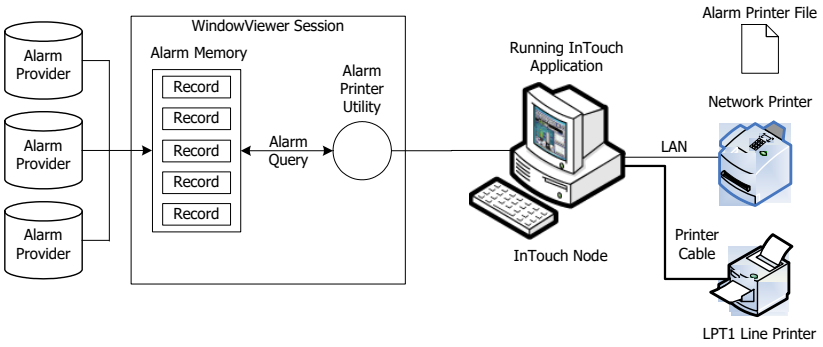
对于 Click 与 DoubleClick 事件，对于“全部供应器”节点，返回的 ElementID 是 -1。

注意：Alarm Tree Viewer 控件从 StartUp 事件调用方法时会忽略用户界面方法，原因在于控件尚不可见。这些方法包括：AboutBox()、CheckElementMembership()、Freeze()、GetAlarmQueryFromSelection()、GetElementCount()、GetElementName()、GetElementPath()、GetSelectedElementCount()、GetSelectedElementName()、GetSelectedElementPath()、GetSubElementCount()、GetSubElementName()、GetSubElementPath() 以及 Refresh()。

如需有关编写 ActiveX 事件脚本的详细信息，请参阅编写 ActiveX 控件脚本。

打印报警

InTouch Alarm Printer 实用程序可以打印来自多个节点的报警。您可以使用专门的行式打印机或网络打印机来按照事件打印报警内存中存储的报警记录。同样，您也可以使用 Alarm Printer 将报警记录保存到文件。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您可以将“分布式报警”系统配置成发生特定的事件时在行式打印机上打印它们。通常您会立即打印报警，以便在万一发生灾难性故障时将信息记录下来。通常您会使用点阵打印机，它通过串行或并行端口直接连接到运行 InTouch 应用程序的计算机上。由于 Windows 网络打印机与激光打印机在实际打印页面之前将整个页面保存在内存中，因此它们一般不适合灾难性事件的数据采集。

配置报警打印与记录

您可以运行多个 Alarm Printer 实例。Alarm Printer 的每个实例都必须配置成打印到不同的打印机，并且必须配置成使用单独的报警查询。

您可以配置 Alarm Printer 各个单独的实例，以打印特定优先级范围内的报警。例如，一个 Alarm Printer 实例仅打印高优先级的报警，而另一个实例则仅打印低优先级的报警。与此类似，您也可以使用 Alarm Printer 的一个实例来打印工厂某个区域的报警，而使用另外一个实例来打印另一个区域的报警。

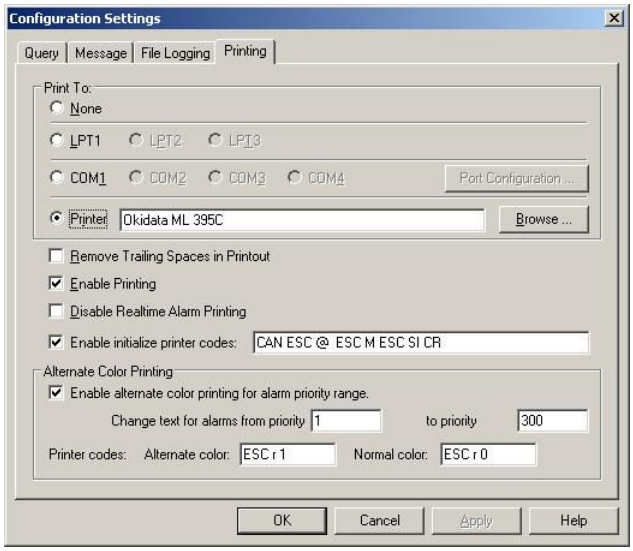
您可以将 Alarm Printer 配置保存到扩展名为 .alc 的文件中。您可以根据需要创建任意多个配置文件。对于每个正在运行的 Alarm Printer 实例，Alarm Printer 均使用单独的配置文件。

配置打印机设置

您必须指定供 Alarm Printer 使用的打印机。还必须选择发生报警时是否打印报警。打印报警时应该使用专用打印机。打印机可以连接到本地机器或网络上。任何 Windows 打印机都可以用作 Alarm Printer 的输出设备。

配置打印机设置

1. 打开 Alarm Printer 实用程序。执行以下操作：
   a. 在 WindowMaker 工具视图中，展开应用程序。
   b. 双击 **Alarm Printer**。
2. 在菜单栏上，选择配置。此时出现配置设置对话框。
3. 选择打印控制选项卡。



4. 在打印到区域中，选择与 Alarm Printer 的连接。
   - 选择无以使用打印机。
   - 选择 **LPT1-3** 所使用的打印机通过并行端口连接到运行 InTouch 应用程序的计算机。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 选择 **COM1-4** 所使用的打印机通过串行端口连接到运行 InTouch 应用程序的计算机。选择端口配置以显示 **COM 属性对话**框，并更改指定给所选 COM 端口的缺省值。
- 选择打印机所使用的打印机是通过网络连接到运行 InTouch 应用程序的计算机。在方框中，输入打印机的名称，或选择浏览来选择可用的打印机。

注意：如果显示的打印机都不符合你的要求，请使用“Windows 添加打印机”向导添加打印机。

5. 选中打印时删除尾部空格复选框，以防止打印机打印空行或空页。
6. 选中启用打印复选框以打印报警。
7. 选中禁用实时报警打印复选框，以防止 Alarm Printer 在报警发生时打印它们。
8. 也可以选择配置打印命令序列以使用特定设置初始化打印机，并在指定的优先级范围内更改报警的打印机特性。如需详细信息，请参阅配置报警打印命令。
9. 选择确定。

配置报警打印命令

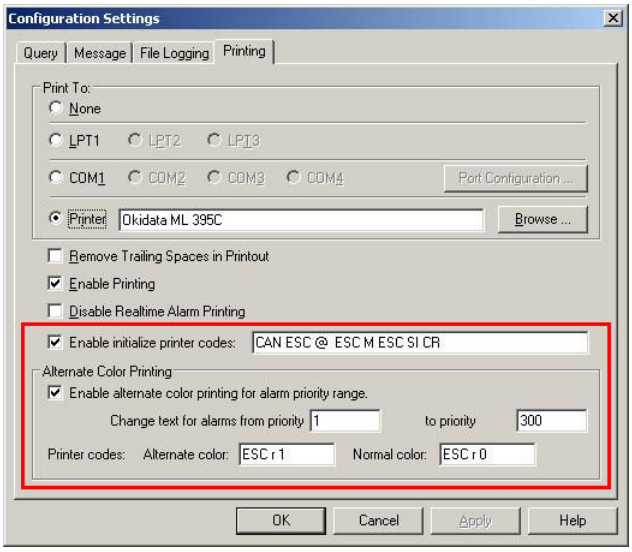
你可以将“报警打印机”配置为，第一次打开打印机（即，每次查询启动和停止时）时，发送打印机命令初始化序列。打印机命令初始化序列可以用于配置打印机设置，例如，字符间距和打印的左右边距。

您还可以配置替代和正常的打印颜色，这样打印颜色可以指示报警是否在指定的报警优先级范围之内。替代和正常颜色命令不一定指定颜色。还可以用来启用/禁用双重打印、强调印刷、下划线、斜体或其他各种打印特性，以使一组报警优先级明显区别于其它报警优先级。

注意：对于黑白打印机，您可能会发现双重打印比强调打印快得多。

打印命令设置

配置设置对话框中的打印选项卡中包含用于指定打印机命令序列的设置。



这些设置在下面的列表中描述。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

设置	描述
启用初始化打印机代码：	每次启动和停止查询时，将附加框中输入的命令序列发送到打印机。
启用报警优先级范围的替代颜色打印：	选择此选项，可传递替代和正常颜色命令以用于指定的报警优先级范围。
将报警的文本从优先级... 更改为优先级	从较低数字（代表较高优先级报警）到较高数字（代表较低优先级报警）输入报警优先级范围。
替代颜色：	当报警优先级在指定的范围中时，每次将报警文本行发送到打印机时将发送的打印机命令序列。
正常颜色：	当报警优先级不在指定的范围中时，每次将报警文本行发送到打印机时将发送的打印机命令序列。

样本打印机命令序列

例如，以下打印机命令序列用于 Okidata Microline 395C 颜色点阵打印机。如需打印机的命令序列，请参阅打印机附带的文档。

打印机初始化命令：

```
CAN ESC @ ESC M ESC SI CR
```

其中的命令如下所示：

- CAN – 取消打印机内部缓冲区中剩余的所有数据。
- ESC @ – 将打印机重置为菜单缺省值。
- ESC M – 将打印设置为 12 CPI (Elite)。
- ESC SI – 将打印设置为压缩以获得 20 CPI。
- CR – 将打印头移回到左侧边距。

将打印颜色设置为红色的替代颜色命令：

```
ESC r 1
```

将打印颜色设置为黑色的正常颜色命令：

```
ESC r 0
```

打印机命令换码序列语法

每个打印机命令换码序列由一个或多个字节构成。有关这些字节的信息可真打印机的文档中找到，也可以通过打印机分析 ESC/P 或 ESC/P2 语法来了解。有关这些语法的信息公开发布在 Internet 上。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

通常，打印机命令换码序列的文本长度没有限制。分析器会解释文本标志流并将这些文本标志转换为一系列连续的字节值。然后，这些字节值会在适当的时间（例如，打印机初始化时，或者必须修改并恢复报警的打印机设置时）完全不变地流送到打印机。

将打印机命令换码序列指定为文本时，必须以空格字符进行分隔每个字节字符。空格字符为分析器指示标志的开始和结束位置。例如，如果打印机的草稿模式可由 ESCx0 激活，那么命令序列可输入为：

```
ESC x 0
```

如果需要将空格字符发送到打印机，请使用缩写 SP 来代表该字节。

注意：分析器是区分大小写的，因此所有缩写都必须以大写字母输入。例如，使用 ESC，而不能使用 esc 或 Esc。

命令序列中可以包括以下类型的项目：

- 控件字符的缩写名称
- 数字、字母或其他可打印的字符
- ASCII 字符的十进制数
- ASCII 字符的十六进制数

有关控件和可打印字符的十六进制或十进制数列表，请参阅公开发布的 ASCII 表。

输入控件字符的名称

由于控件字符不易通过键盘进行输入，所以您可以输入其名称。有关控件字符名称的列表，请参阅公开发布的 ASCII 表的缩写列。唯一一个通常不会列出在 ASCII 表中的字符是空格字符，必须输入为 SP。

注意：分析器不会处理常见的 C 语言换码字符（例如，\r、\n 和 \t）。

输入可打印的字符

通过在框中输入来输入任意可打印的字符。唯一的例外是空格字符（十六进制 0x20，十进制 32），必须输入为 SP。该字符的值解释为在 ASCII 表中代表该值的数字，也是发送到打印机的值。

有关可打印字符的列表，请参阅公开发布的 ASCII 表。

输入字符的十进制数

通过在框中输入来输入字符的十进制数字。最大可输入的十进制值是 255 （十六进制 0xFF）。如果输入的十进制数太大（例如，497）不会发出警告；会返回 255。不支持负数。

输入字符的十六进制数

通过在框中输入来输入字符的十六进制数字。允许的值范围是 0x00 到 0xFF。

配置所要打印的报警

Alarm Printer 会查询报警内存来选择要打印或保存到日志文件的记录。查询基于以下条件从内部报警内存中选择记录：

- 报警优先级
- 当前报警状态（未确认/确认）
- 报警组成员关系

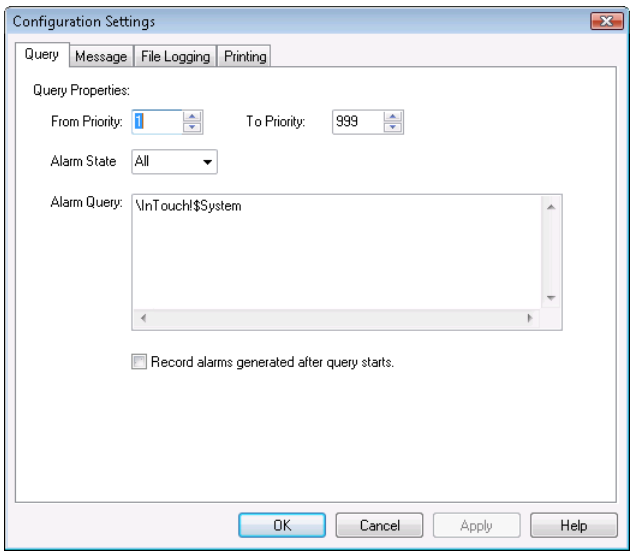
每个报警都有一个指定的优先级编号，代表报警的严重程度。报警优先级的范围是从 1 到 999。报警严重程度最高的优先级为 1。报警严重程度最低的优先级为 999。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果网络或打印机连接失败，Alarm Printer 不会重新打印所有的报警。Alarm Printer 仅打印连接失败之前尚未打印的那些报警。

配置所要打印的报警

1. 打开 Alarm Printer 实用程序。执行以下操作：
   a. 在 WindowMaker 工具视图中，展开应用程序。
   b. 双击 **Alarm Printer**。
2. 在菜单栏上，选择配置。此时出现配置设置对话框。
3. 选择查询选项卡。



4. 在从优先级框中，输入最高优先级的报警值（1 到 999）。
5. 在到优先级框中，输入最低优先级的报警值（1 到 999）。
6. 在报警状态列表中，单击报警状态。

选择	显示
全部	所有报警。
确认	仅限已确认的报警。
未确认	仅限未确认的报警。

7. 在报警查询框中，输入一个或多个报警查询。您可以指定一个或多个报警供应器与组。使用空格分隔多个查询。
8. 选中记录查询开始后产生的报警复选框，以便仅包含查询开始之后发生的报警。如此，Alarm Printer 会忽略那些在其开始查询之前已触发、且目前存在于报警内存中的报警记录。
9. 选择确定。

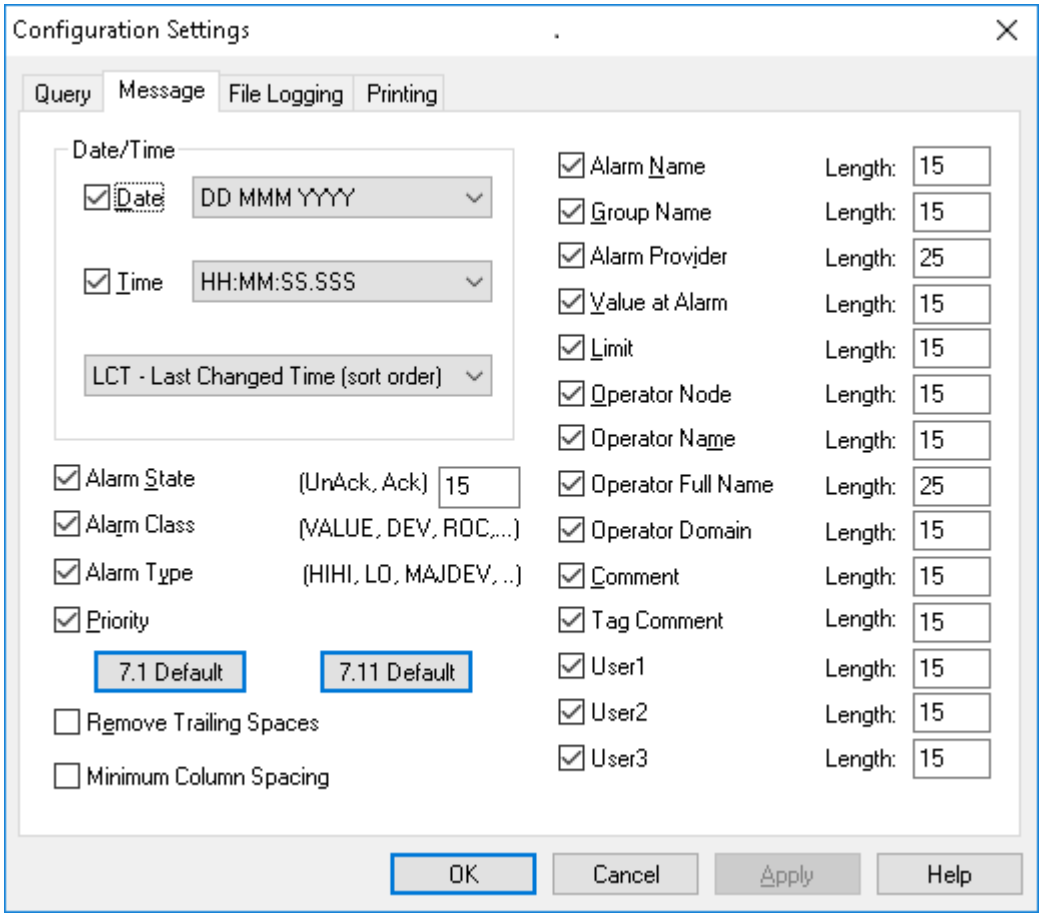
配置打印与文件输出的格式

选择的每个选项在打印输出中都显示为一个单独的字段。包含数据的字段超出指定的最大字段长度时，将按照其各自的限制进行截断。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置报警打印与文件输出的格式

1. 打开 Alarm Printer 实用程序。执行以下操作：
   a. 在 WindowMaker 工具视图中，展开应用程序。
   b. 双击 **Alarm Printer**。
2. 在菜单栏上，选择配置。此时出现配置设置对话框。
3. 选择消息选项卡。



4. 在日期/时间区域中，选中日期复选框，然后从列表中选择一种日期格式。列出的日期格式包含以下组成部分：

选项	描述
DD	两位数字日期值 (01-31)
MM	两位数字月份值 (01-12)
YY	年份的后两位数字
YYYY	四位数字年份值
MMM	三个英文字符的月份缩写

5. 选中时间复选框，然后从列表中选择一种时间格式。列出的格式包含时间的以下组成部分：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

选项	描述
AP	选择“上午/下午”时间格式。例如，下午 3:00 显示为“下午 03:00”。未指定此项的时间缺省使用 24 小时军用时间格式。例如，下午 3:00 显示为 15:00。
HH	发生报警/事件时的两位数字小时值（00-23 或 01-12）。
MM	发生报警/事件时的两位数字分钟值 (00-59)。
SS	发生报警/事件时的两位数字秒值 (00-59)。
SSS	发生报警/事件时的三位数字毫秒值。

6. 根据报警发生的时间选择报警出现在报警记录中的顺序：

选项	描述
OAT	（原始报警时间）发生报警时的日期/时间标签。
LCT	（上次更改时间）报警实例最近一次状态改变的日期/时间标签：报警发生、子状态改变、返回正常，或者是确认。
LCT 但确认后为 OAT	（“上次更改时间”，但确认时使用“原始报警时间”）报警未确认时使用上次更改时间，确认报警之后使用原始报警时间。

7. 选择要打印的报警信息。

选项	描述
报警状态	打印报警状态。例如，未确认、确认。
报警类	打印报警类。例如，VALUE、DEV、ROC。
报警类型	打印报警类型。例如，HIHI、LO、MAJDEV。
优先级	打印报警优先级 (1-999)。
7.1 缺省值（按钮）	选择 InTouch 7.1 版 Alarm Printer 实用程序的缺省设置。
7.11 缺省值（按钮）	选择 InTouch 7.11 版 Alarm Printer 实用程序的缺省设置。
删除尾部空格	实际字段值的长度小于为该字段配置的值时，从打印的字段中删除额外的尾部空格。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

选项	描述
最小列间距	减少各个列的间距，使打印的页面上可以显示更多的字段。
报警名	打印报警名（标记）。在“长度”框中，输入报警名的字符数（最多 64 个字符）。
组名	打印报警组的名称。在“长度”框中，输入报警组名的字符数（最多 64 个字符）。
报警供应器	打印报警供应器的名称。在“长度”框中，输入报警供应器名的字符数（最多 64 个字符）。
报警值	打印标记的值。在“长度”框中，输入报警值的字符数（最多 32 个字符）。
限制	打印标记的报警限。在“长度”框中，指定可输入的报警限的字符数（最多 32 个字符）。此数字应该足够大，以提供所需级别的精度。
操作员节点	打印与报警条件关联的操作员节点。在“长度”框中，输入操作员节点可使用的字符数（最多 64 个字符）。在“终端服务”环境中，这是相应的操作员建立“终端服务”会话的客户端计算机的名称。如果无法检索节点名，则使用节点的 IP 地址代替。
操作员姓名	打印与报警条件关联的操作员姓名。在“长度”框中，指定可输入的操作员姓名的字符数（最多 16 个字符）。
操作员全名	打印已登录操作员的全名。
操作员域	打印与报警条件关联的已登录操作员的域。
注释	打印与标记关联的报警注释。在“长度”框中，指定可输入的注释的字符数（最多 131 个字符）。
标记注释	打印标记的注释。
用户 1	打印与报警对应的“用户自定义数字 1”的数值。
用户 2	打印与报警对应的“用户自定义数字 2”的数值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

选项	描述
用户 3	打印与报警关联的用户自定义字符串属性的字符串值。最多 131 个字符。

8. 选择应用。

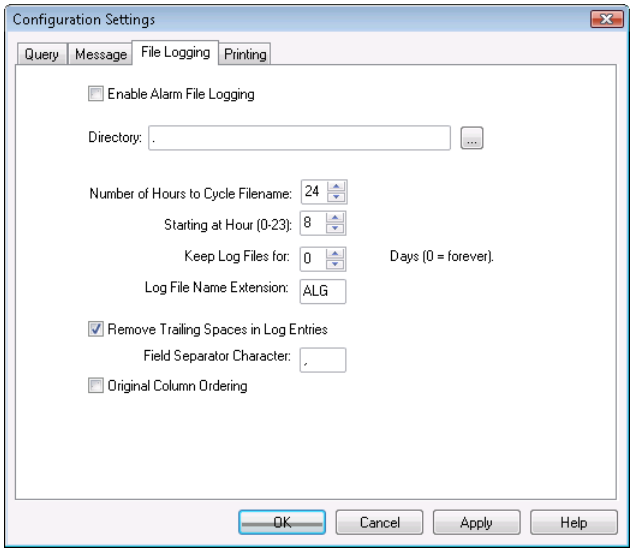
配置报警的日志文件

您可以将报警记录保存到文件中。缺省条件下，Alarm Printer 会根据以下命名惯例为日志文件指定名称：YYMMDDHH.ALG

记号	描述
YY	创建日志文件时的年份的后两位数字。
MM	创建日志文件时的两位数字月份值 (01-12)。
DD	创建日志文件时的两位数字日期值 (01-31)。
HH	创建日志文件时的两位数字小时值 (00-23)。

配置报警记录日志

1. 打开 Alarm Printer 实用程序。执行以下操作：
   a. 在 WindowMaker 工具视图中，展开应用程序。
   b. 双击 **Alarm Printer**。
2. 在菜单栏上，选择配置。此时出现配置设置对话框。
3. 选择日志文件选项卡。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 选中启用报警文件记录复选框，以便将报警记录保存到日志文件。
5. 在目录框中，输入要保存报警日志文件的路径，或浏览到某个文件夹位置。
6. 在循环文件名小时数框中，输入要在单独的日志文件中保存多少个小时的报警记录。

   有效输入为 1 到 24。如果 InTouch 应用程序持续运行，请选择一个适当的文件记录间隔，以便创建一组时间长度相等的日志文件。例如，将循环文件名小时数设置为 6 时，可以创建 4 个时间长度相等的日志文件。
7. 在开始小时值框中，输入每天开始将报警记录保存到文件的起始小时值。

   有效输入为 0 到 23。

   例如，某个炼油厂按照每天三班的方式运转。第一班从 6:00 开始。管理人员应该记录每一班的报警记录。要做到这一点，请为循环文件名小时数选项输入 8。为开始小时值 (0-23) 选项输入 6。Alarm Printer 从 06:00 到 14:00 创建一个日志文件，从 14:00 到 22:00 创建另一个，从 22:00 到 06:00 创建第三个。
8. 在保留日志文件时间框中，输入要保留日志文件的天数。

   Alarm Printer 将保存日志记录文件指定的天数（包括当天）。Alarm Printer 删除超过保留期限的日志文件。要无限期保存日志文件，请输入 0。
9. 在日志文件扩展名框中，接受缺省的 ALG 文件扩展名，或者为日志文件另外指定一个三个英文字符的扩展名。

   如果使用 .csv 扩展名，则可以将日志文件直接导入 Excel 或“记事本”。
10. 要删除日志文件中每个项目末尾的空格，请选中日志条目中删除尾部空格复选框。

    您也可以指定一个字段分隔符，放到日志文件中每个记录的末尾。
11. 选中按原始列排序复选框，以便让日志文件记录保持报警显示中的相同顺序。
12. 选择应用。

保存与加载配置文件

从 WindowMaker 中启动 Alarm Printer 时，**Alarm Printer** 对话框显示缺省的配置设置。这些配置设置保存在报警配置文件 (.alc) 中。

您可以将打印机配置设置保存到某个文件，在打印报警之前可以加载该文件。

如果在运行时从命令提示符中或通过双击 *.alc 文件名来打开特定的报警配置文件，则可以显示它。

创建新的 Alarm Printer 配置文件

1. 打开 Alarm Printer 实用程序。执行以下操作：
   a. 在 WindowMaker 工具视图中，展开应用程序。
   b. 双击 **Alarm Printer**。
2. 在文件菜单上，单击新建以显示带缺省值的 Alarm Printer。
3. 在菜单栏上，选择配置。此时出现配置设置对话框。
4. 配置报警设置。
5. 在文件菜单上，选择保存。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

编辑现有的 Alarm Printer 配置文件

1. 在文件菜单上，选择打开。
2. 选择要编辑的 Alarm Printer 配置文件。
3. 编辑该文件。
4. 在“文件”菜单上，选择保存。选择另存为将更改保存到新文件，而不更改现有文件。

使用报警打印机打印报警

每个查询记录当前打开的 Alarm Printer 配置文件 (.alc) 中指定的所有报警。如果未指定文件，则使用配置 Alarm Printer 期间当前所选的设置。

您可以使用 Alarm Printer 运行多个查询。每个查询都使用不同的参数，并且与单独的 Alarm Printer 实例关联。如果 Alarm Printer 的两个实例正在运行相同的查询，则查询结果将完全相同。

Alarm Printer 正在运行时，可以手工启动或停止查询。务必确保启用了打印。

启动报警查询

- 在查询菜单上，选择开始/停止。

停止报警查询

- 在查询菜单上，选择开始/停止。

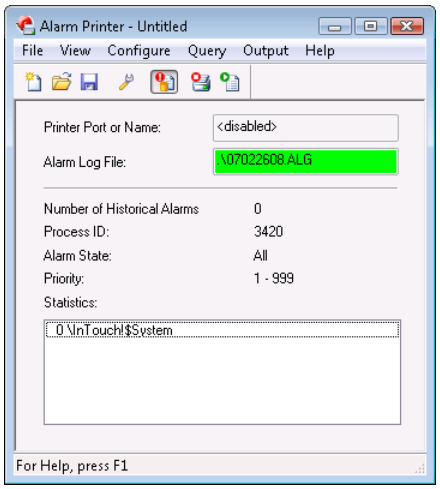
将报警记录到文件

您可以使用 Alarm Printer 将报警记录保存到文件中。您应该已经从配置设置对话框中配置好记录功能。

将报警记录到文件

1. 打开 Alarm Printer 实用程序。
   a. 在 WindowMaker 工具视图中，展开应用程序。
   b. 双击 **Alarm Printer**。
2. 根据需要配置查询以采集要记录的报警记录。
3. 选择文件记录按钮。
4. 运行报警查询。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



此时由查询采集的报警记录写入所配置的日志文件中。

使用特定配置启动 Alarm Printer

通过在批处理文件中使用以下命令，可以在启动系统时自动启动 Alarm Printer 并打开特定的文件。

```
ALMPRT.EXE MYQUERY.ALC
```

其中，MYQUERY.ALC 是打开的 Alarm Printer 配置文件的名称。指定 .exe 文件扩展名属于可选项。

确保批处理文件切换到安装 InTouch HMI 的文件夹。

要防止因系统意外关机并重新启动而导致丢失任何查询数据，可以通过从批处理文件中运行以下命令来自动启动 Alarm Printer 并自动运行特定的查询。

```
ALMPRT.EXE -q MYQUERY.ALC
```

通过在命令中使用 -q，查询会在系统启动时自动运行。

使用脚本控制 Alarm Printer

您可以在脚本中使用 Alarm Printer 函数来控制报警打印。

Alarm Printer 函数返回一个整型错误码，指出函数是否成功完成。下表显示各个错误码。

错误码	错误消息
0	成功
1	实例未找到或未在运行
2	接口未初始化
3	无法访问虚拟内存
4	错误码无效
5	已在运行的实例太多
6	结果字符串太长
7	传递给函数的实例索引无效

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

错误码	错误消息
12	无法将消息发送到 Alarm Printer 应用程序
9	未等到 Alarm Printer 应用程序的响应
20	“到优先级”必须大于或等于“从优先级”
21	优先级值无效
22	报警状态无效
23	由于查询正在运行，无法执行命令
24	查询字符串无效
25	查询处理状态无效
26	打印状态选择器无效
27	Alarm Printer 窗口收到的命令无法识别
28	查询无法启动

停止与启动 Alarm Printer 实例或查询

使用以下函数启动与停止 Alarm Printer 实例与报警查询。

- APUStartInstance() 函数
- APUStartQuery() 函数
- APUStopInstance() 函数
- APUStopQuery() 函数

APUStartInstance() 函数

使用配置文件中指定的值在最小化状态下启动 Alarm Printer 的实例。

类别

查看

语法

```
[Result=] APUStartInstance(sFilePath, iTagInstance);
```

参数

sFilePath
配置文件的完整路径（输入字符串）。

iTagInstance
整型标记。如果函数成功执行，则给它返回实例号。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

您最多可以启动十六个 Alarm Printer 实例。此函数将实例号 (0 - 15) 写入 iTagInstance 参数。实例号在启动新的 Alarm Printer 实例时进行递增。

此实例号可以由其它 Alarm Printer 函数用于识别 Alarm Printer 实例。

此函数返回 0；如果发生错误，则返回错误码。

Alarm Printer 程序不自动开始处理报警内存中的报警。使用 **APUStartQuery()** 函数开始处理该实例报警内存中的数据。

示例

```
Status = APUStartInstance("c:\MyAlarmCfg\Area1Alarms.alc", Inst);
```

另请参阅

APUStartQuery(), APUStopInstance(), APUStopQuery()

APUStartQuery() 函数

设置要从报警内存中处理的记录的日期与时间限制，然后启动查询。

类别

查看

语法

```
[Result=] APUStartQuery(iInstance, iYear, iMonth, iDay, iHour, iMinute);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iYear
年份数字。

iMonth
月份数字。

iDay
日期值。

iHour
小时值。

iMinute
分钟值。

附注

如果在查询已经运行时试图启动另一个查询，则会发生错误。

如果将所有的日期与时间值都设置为 0，则打印所有报警。这是由于 0 被解释为 1900 年 1 月 1 日午夜。指定的日期与时间采用本地时间格式。年份值为 -1 时，将日期设置为处理命令时的当前时间。

返回整型错误码。

示例

```
Status = APUStartQuery(Inst,2007,4,16,22,12);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

APUStartInstance(), APUStopInstance(), APUStopQuery()

APUStopInstance() 函数

停止指定的 Alarm Printer 实例。此时将停止进一步添加任何要打印的记录，终止当前正在执行的任何打印查询，并关闭程序的实例。

类别

查看

语法

```
[Result=] APUStopInstance(iInstance);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

附注

返回整型错误码。

示例

```
Status = APUStopInstance(5);
```

另请参阅

APUStartInstance(), APUStartQuery(), APUStopQuery()

APUStopQuery() 函数

请求指定的实例停止运行查询。应用程序保持运行状态，但是不处理任何查询。调用 **APUStartQuery()** 可以使实例启动查询。

类别

查看

语法

```
[Result=] APUStopQuery(iInstance);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

附注

返回整型错误码。

示例

```
Status = APUStopQuery(5);
```

另请参阅

APUStartInstance(), APUStartQuery(), APUStopInstance()

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

查询报警查询信息

使用以下函数根据 Alarm Printer 配置文件中设置的标记优先级与数值来检索查询参数。

- APUGetAlarmGroupText() 函数
- APUGetQueryFromPriority() 函数
- APUGetQueryToPriority() 函数
- APUGetConfigurationFilePath() 函数
- APUGetPrinterJobCount() 函数
- APUGetQueryAlarmState() 函数
- APUGetQueryProcessingState() 函数

APUGetAlarmGroupText() 函数

获取“报警查询”报警组文本。

类别

查看

语法

```
[Result=] APUGetAlarmGroupText(iInstance, sTagGroup);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

sTagGroup
文本 - 报警组

附注

初始报警组文本从指定的 Alarm Printer 实例正在使用的 .alc 配置文件中读取。报警组文本作为 sTagGroup 参数传递到 InTouch 消息标记中。

返回整型错误码。

示例

TagGroup 消息标记在运行函数之后可能包含以下值：\intouch!$system

```
Status = APUGetAlarmGroupText(Inst,TagGroup);
```

另请参阅

APUGetConfigurationFilePath(), APUGetPrinterJobCount(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority()

APUGetQueryFromPriority() 函数

获取查询的“从优先级”值。

类别

查看

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
[Result=] APUGetQueryFromPriority(iInstance, iTagPriority);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iTagPriority
整型标记，用于接收“从优先级”值。

附注

初始优先级从指定的 Alarm Printer 实例正在使用的 .alc 配置文件中读取。“从优先级”值作为 iTagPriority 参数传递给某个 InTouch 整型标记。

返回整型错误码。

示例

在本例中，FromPri 是包含“从优先级”值（如 1）的整型标记。

```
Status = APUGetQueryFromPriority(Inst, FromPri);
```

另请参阅

APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetPrinterJobCount(), APUGetQueryAlarmState(), APUGetQueryProcessingState(), APUGetQueryToPriority()

APUGetQueryToPriority() 函数

获取查询的“到优先级”。

类别

查看

语法

```
[Result=] APUGetQueryToPriority(iInstance,iPriority);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iPriority
整型标记，用于接收“到优先级”值。

附注

其它查询不得与包含 APUGetQueryToPriority() 函数的脚本同时运行。“到优先级”值写入函数的 iPriority 参数，它是一个整型标记。

返回整型错误码。

示例

ToPri 整型标记接收“到优先级”值，如 999。

```
Status = APUGetQueryToPriority(Inst,ToPri);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetPrinterJobCount(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState()

APUGetConfigurationFilePath() 函数

返回用于查询的 .alc 配置文件的完整文件路径。

类别

查看

语法

```
[Result=] APUGetConfigurationFilePath(iInstance, sTagFilePath);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

sTagFilePath
消息标记，用于检索 Alarm Printer 实例正在使用的配置文件的文件路径名。

附注

文件路径文本返回给一个消息标记，该标记可以指定为此函数的 sTagFilePath 参数。

返回整型错误码。

示例

CfgFilePath 消息标记接收特定配置文件的文件路径，该配置文件与 Inst 整型标记中包含的 Alarm Printer 实例关联。例如：c:\MyAlarmCfg\Area1Alarms.alc

```
Status = APUGetConfigurationFilePath(Inst, CfgFilePath);
```

另请参阅

APUGetAlarmGroupText(), APUGetPrinterJobCount(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority()

APUGetPrinterJobCount() 函数

为此实例使用的打印机返回最新的作业计数这个 Windows 打印机状态。

类别

查看

语法

```
[Result=] APUGetPrinterJobCount(iInstance, iTagCount);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iTagCount
整型标记，用于接收计数值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

此函数在 iTagCount 参数中返回计数值，该参数是一个整型标记。

除非查询正在运行，否则结果不是最新的。除非报警已经打印，否则结果不是最新的 - 作业计数通常在初次打开打印机时更新，然后在每打印一行报警时更新。

返回的作业计数值仅对 Windows 打印机有效，对于与并行或串行端口关联的打印机则没有多少意义。

返回整型错误码。

示例

PJCount 是一个整型标记，用于接收指定的实例的计数值。

```
Status = APUGetPrinterJobCount(Inst, PJCount);
```

另请参阅

APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority()

APUGetQueryAlarmState() 函数

返回查询的报警状态。

类别

查看

语法

```
[Result=] APUGetQueryAlarmState(iInstance, iTagState);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iTagState

整型标记，用于接收同指定的实例关联的查询的报警状态。值的含义如下：

0 = 全部

1 = 已确认

2 = 未确认

附注

初始报警状态从 .alc 文件中读取。此函数在 iTagState 参数中返回它，该参数是一个整型标记。

返回整型错误码。

示例

AlmState 是包含 0、1 或 2 的整型标记。

```
Status = APUGetQueryAlarmState(Inst,AlmState);
```

另请参阅

APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetPrinterJobCount(), APUGetQueryFromPriority(), APUGetQueryProcessingState(), APUGetQueryToPriority()

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

APUGetQueryProcessingState() 函数

返回报警查询的处理状态。

类别

查看

语法

```
[Result=] APUGetQueryProcessingState(iInstance, iTagState);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iTagState

整型标记，用于从函数中接收处理状态。它的含义如下：

0 = 停止

1 = 启动

附注

此函数给 iTagState 参数返回一个整数值，该参数是一个整型标记。

返回整型错误码。

示例

ProcState 是一个整型标记，查询不在运行时设置为 0，查询正在运行时设置为 1。

```
Status = APUGetQueryProcessingState(Inst, ProcState);
```

另请参阅

APUGetAlarmGroupText(), APUGetConfigurationFilePath(), APUGetPrinterJobCount(), APUGetQueryAlarmState(), APUGetQueryFromPriority(), APUGetQueryToPriority()

查询实例信息

使用以下函数返回 Alarm Printer 实例的当前状态。

- APUFindAlarmGroupInstance() 函数
- APUFindFileInstance() 函数
- APUFindPrinterInstance() 函数
- APUGetInstanceCount() 函数
- APUIsInstanceUsed() 函数

APUFindAlarmGroupInstance() 函数

返回使用指定的报警组字符串的第一个 Alarm Printer 实例。

类别

查看

语法

```
[Result=] APUFindAlarmGroupInstance(sGroup, iInstance);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

sGroup
要在各实例中查找的报警组的名称。

iInstance
整型标记，对于查找到的使用指定组名的实例，此标记接收它的值。

附注

此函数将实例号作为 iInstance 参数返回给某个整型标记。初始报警组字符串从 .alc 文件中读取。如果未找到任何实例，则函数返回 1 作为错误码（没有可用的实例），并将 0 写入整型标记参数。

返回整型错误码。

示例

FoundInstance 是一个整型标记，用于接收所找到的第一个实例的编号，该实例将 $System 用作其查询。

```
Status = APUFindAlarmGroupInstance("$System", FoundInstance);
```

另请参阅

APUFindFileInstance()、APUFindPrinterInstance()、APUGetInstanceCount()、APUIsInstanceUsed()

APUFindFileInstance() 函数

查找使用指定的 .alc 配置文件的第一个 Alarm Printer 实例。

类别

查看

语法

```
[Result=] APUFindFileInstance(sFilePath, iInstance);
```

参数

sFilePath
要查找的实例的 .alc 配置文件路径。

iInstance
整型标记，用于接收实例编号。

附注

此函数将实例号作为 iInstance 参数返回给某个整型标记。使用此函数获取所需的 Alarm Printer 实例。匹配文件路径字符串时不区分大小写。

如果未找到任何实例，则函数返回 1 作为错误码（没有可用的实例），并将 0 写入整型标记参数。

返回整型错误码。

示例

InstFound 是一个整型标记，它接收使用 c:\MyAlarmCfg\Area1Alarms.alc 配置文件的第一个实例的编号。

```
Status = APUFindFileInstance("c:\MyAlarmCfg\Area1Alarms.alc", InstFound);
```

另请参阅

APUFindAlarmGroupInstance()、APUFindPrinterInstance()、APUGetInstanceCount()、APUIsInstanceUsed()

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

APUFindPrinterInstance() 函数

查找使用指定的打印机名称或端口的第一个 Alarm Printer 实例。

类别

查看

语法

```
[Result=] APUFindPrinterInstance(sPrinter, iInstance);
```

参数

sPrinter
要查找的实例的打印机名称。

iInstance
整型标记，用于接收实例编号。

附注

此函数将实例号作为 iInstance 参数返回给某个整型标记。使用此函数获取所需的 Alarm Printer 实例。打印机名存储在 .alc 文件中，并可以从中读取。匹配打印机名字符串时不区分大小写。如果未找到任何实例，则函数返回 1 作为错误码（没有可用的实例），并将 0 写入整型标记参数。

返回整型错误码。

示例

FoundInst 是一个整型标记，对于在关联的 .alc 文件中将 LPT1 用作打印机名的第一个实例，它用于接收该实例的编号。

```
Status = APUFindPrinterInstance("LPT1", FoundInst);
```

另请参阅

APUFindAlarmGroupInstance()、APUFindFileInstance()、APUGetInstanceCount()、APUIsInstanceUsed()

APUGetInstanceCount() 函数

返回正在运行的 Alarm Printer 实例的数量，最多 16 个。

类别

查看

语法

```
[Result=] APUGetInstanceCount(iCount);
```

参数

iCount
整型标记，用于接收实例编号。

附注

此函数将实例数量作为参数返回给某个整型标记。前十六个同时运行的实例之外的任何其它实例都不是动态控制的，也无法获取其状态。

返回整型错误码。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

ICount 是一个整型标记。运行时的值为 7 时，表示当前有七个 Alarm Printer 实例正在运行。

```
Status = APUGetInstanceCount( iCount );
```

另请参阅

APUFindAlarmGroupInstance(), APUFindFileInstance(), APUFindPrinterInstance(), APUIsInstanceUsed()

APUIsInstanceUsed() 函数

返回一个离散值，指出当前是否正在使用某个实例。

类别

查看

语法

```
[Result=] APUIsInstanceUsed(iInstance);
```

参数

iInstance
Alarm Printer 实例的编号（0 到 15）。

附注

结果为 0 或 1：

0 = 实例未使用。

1 = 实例在使用。

示例

InUse 是一个离散标记，如果实例 5 正在使用，则该标记设置为真 (1)；如果实例 5 不在使用，则设置为假 (0)。

```
InUse = APUIsInstanceUsed(5);
```

另请参阅

APUFindAlarmGroupInstance(), APUFindFileInstance(), APUFindPrinterInstance(), APUGetInstanceCount()

查询打印机信息

使用以下函数返回 Alarm Printer 的状态。

- APUGetPrinterName() 函数
- APUGetPrinterStatus() 函数

APUGetPrinterName() 函数

返回此实例使用的打印机的 Windows 打印机名或端口名。

类别

查看

语法

```
[Result=] APUGetPrinterName(iInstance, sTagPrinter);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

iInstance
Alarm Printer 实例编号（0 到 15）。

sTagPrinter
消息标记，它接收与指定实例关联的配置的打印机名或端口名。

附注

如果未给某个实例配置打印机，则此函数返回 NONE。打印机名存储在 .alc 文件中，并可以从中读取。此函数将打印机名或端口名作为 sTagPrinter 参数返回给某个消息标记。

返回整型错误码。

示例

PrtName 是一个消息标记，它接收与 Alarm Printer 实例 3 关联的配置的打印机名或端口名。

```
Status = APUGetPrinterName(3,PrtName);
```

另请参阅

APUGetPrinterStatus()

APUGetPrinterStatus() 函数

返回此实例使用的 Windows 打印机的最新状态。

类别

查看

语法

```
[Result=] APUGetPrinterStatus(iInstance, iSelector, iTagStatus);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iSelector
整数值，指定如下内容：

0 = 获取“Alarm Printer 错误”的状态

1 = 获取“Alarm Printer 缺纸”的状态

2 = 获取“Alarm Printer 离线”的状态

3 = 获取“Alarm Printer 溢出”的状态

iTagStatus
整型或实型标记，它接收某个打印机的状态，该打印机与指定的实例号以及 iSelector 参数所选择的类型关联。

附注

此函数将打印机状态作为 iTagStatus 参数返回给某个整型或实型标记。除非查询正在运行并且已经打印报警，否则结果不是最新的。状态通常在初次打开打印机时更新，然后在每打印一行报警时更新。

此状态信息是根据 Microsoft 或 Windows 的驱动程序标准向打印机查询的。并非所有的打印机制造商都遵循这些标准。因此，并非所有打印机都会返回状态信息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

返回整型错误码。

示例

PrtStat 是一个整型标记，它从同实例 5 关联的打印机接收“打印机离线”状态。

```
Status = APUGetPrinterStatus(5, 2, PrtStat);
```

另请参阅

APUGetPrinterName()

设置报警查询信息

使用以下函数设置 Alarm Printer 查询中使用的参数。

- APUSetAlarmGroupText() 函数
- APUSetQueryAlarmState() 函数
- APUSetQueryFromPriority() 函数
- APUSetQueryToPriority() 函数
- APUSetTimeoutValues() 函数

APUSetAlarmGroupText() 函数

设置“报警查询”报警组文本。

类别

查看

语法

```
[Result=] APUSetAlarmGroupText(iInstance, sGroup);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

sGroup
报警组文本。

附注

查询不在运行时，此函数才能成功。

返回整型错误码。

示例

本例将 Alarm Printer 实例 1 的查询设置为 \InTouch!GroupA。

```
Status = APUSetAlarmGroupText(1, "\InTouch!GroupA");
```

另请参阅

APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues()

APUSetQueryAlarmState() 函数

设置查询的报警状态。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

查看

语法

```
[Result=] APUSetQueryAlarmState(iInstance, iState);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iState
具有以下可能值的整数：

0 = 全部

1 = 已确认

2 = 未确认

附注

运行使用 **APUSetQueryAlarmState()** 函数的脚本时，无法同时运行查询。

返回整型错误码。

示例

本例设置 Alarm Printer 实例 3 的查询，使其仅查询已确认的报警。

```
Status = APUSetQueryAlarmState(3, 1);
```

另请参阅

APUSetAlarmGroupText(), APUSetQueryFromPriority(), APUSetQueryToPriority(), APUSetTimeoutValues()

APUSetQueryFromPriority() 函数

设置报警查询的下边界或“从优先级”。

类别

查看

语法

```
[Result=] APUSetQueryFromPriority(iInstance, iPriority);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iPriority
表示“从优先级”的整数值（1 到 999）。

附注

查询不在运行时，此函数才能成功。

返回整型错误码。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

本例设置某个查询的“从优先级”值，该查询与 Inst 整型标记的实例值以及 FromPri 整型标记的值关联。

```
Status = APUSetQueryFromPriority(Inst, FromPri);
```

另请参阅

APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryToPriority(), APUSetTimeoutValues()

APUSetQueryToPriority() 函数

设置查询的“到优先级”。

类别

查看

语法

```
[Result=] APUSetQueryToPriority(iInstance, iPriority);
```

参数

iInstance
Alarm Printer 的实例（0 到 15）。

iPriority
表示“到优先级”的整数值，范围从 1 到 999。它应该设置为大于或等于指定的实例中相同查询的“从优先级”值。

附注

“到优先级”必须大于或等于“从优先级”，才能设置有效的报警优先级范围。包含 **APUSetQueryToPriority()** 函数的脚本无法与查询同时运行。

返回整型错误码。

示例

本例将同实例 0 关联的查询的“到优先级”值设置为 240。

```
Status = APUSetQueryToPriority(0,240);
```

另请参阅

APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetTimeoutValues()

APUSetTimeoutValues() 函数

APUSetTimeoutValues() 函数设置以秒为单位的超时间隔。超时间隔控制：程序正在运行时，观察到由于内存访问或无法获取有效的响应所导致的错误数量。

缺省内存访问超时为两秒钟。缺省短响应等待时间为 10 秒，缺省长响应等待时间为 20 秒。

类别

查看

语法

```
[Result=] APUSetTimeoutValues(iMemory, iShort, iLong);
```

参数

iMemory

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

整型 - 访问超时

iShort
短响应等待时间（整型）

iLong
长响应等待时间（整型）

示例

```
Status = APUSetTimeoutValues(iMemory,iShort,iLong);
```

另请参阅

APUSetAlarmGroupText(), APUSetQueryAlarmState(), APUSetQueryFromPriority(), APUSetQueryToPriority()

处理 Alarm Printer 错误

使用 APUTranslateErrorCode() 函数将 Alarm Printer 错误码转换为错误消息。

APUTranslateErrorCode() 函数

将 APU 函数之一返回的错误码转换为简要描述错误码的中文字符串。

类别

查看

语法

```
[Result=] APUTranslateErrorCode(iErrorCode, sTagMessage);
```

参数

iErrorCode
整型错误码，通常由其它大多数 APU 函数返回。

sTagMessage
消息标记，用于接收错误消息。

附注

此函数将错误消息作为 sTagMessage 参数返回给某个消息标记。如果传递了未知的错误码，此函数可能会失败。

返回整型错误码。

示例

如果 Alarm Printer 的实例 15 当前不在运行，则本例将消息标记 errmsg 设置为“没有可用的实例”。

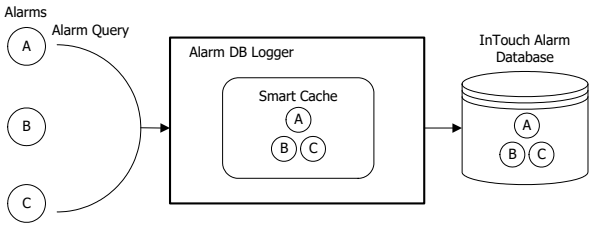
```
Status = APUTranslateErrorCode(APUSetAlarmGroupText(15,"$system"), ErrMsg);
```

将报警记录到报警数据库

“InTouch 分布式报警系统”包含 Alarm DB Logger 实用程序，它可以将报警与事件记录到报警数据库。

Alarm DB Logger 是一个报警接收器。您可以给它配置一个或多个查询，以便从 InTouch 报警供应器中选择报警。这些查询选择的报警存储在临时内存缓冲区（也称为“智能缓存”）中。Alarm DB Logger 按周期性的间隔将“智能缓存”中的内容作为报警与事件记录写入报警数据库。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



Alarm DB Logger 可以自动重新连接。丢失与数据库的连接时，Logger 按固定的间隔检查数据库连接。重新建立与报警数据库的连接之后，记录功能恢复执行。

无论是作为服务还是作为 Logger 的一般应用程序来运行，Alarm DB Logger 都会报告所有错误。

Alarm DB Logger Manager 的 SQL Server 帐户

Alarm DB Logger Manager 在 SQL Server 数据库中创建并使用以下帐户：

帐户	密码
wwAdmin	wwadmin
wwPower	wwpower
wwUser	wwuser

使用 Alarm DB Logger Manager

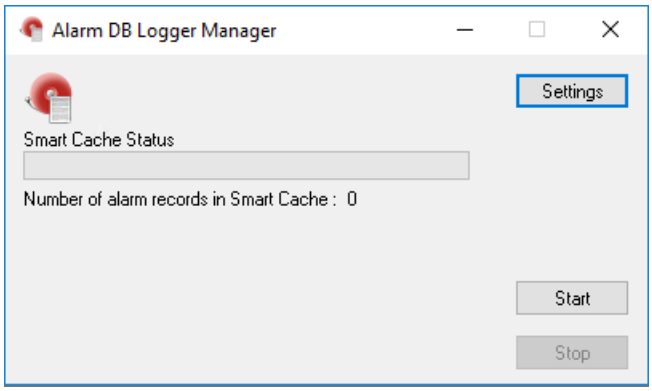
使用 Alarm DB Logger Manager 启动与停止记录操作。Alarm DB Logger Manager 可以作为服务或普通的应用程序来启动。

您可以使用 Alarm DB Logger 配置向导来配置报警记录功能，该向导可以从 Alarm DB Logger Manager 中启动。

Alarm DB Logger Manager 显示报警记录占用“智能缓存”的百分比。SQL Server 连接丢失或报警发生速度比 Alarm DB Logger 可以将它们记录到报警数据库的速度更快时，这些报警会进行缓存。

打开 Alarm DB Logger Manager。

1. 在工具视图中，展开应用程序。
2. 双击 **Alarm DB Logger Manager**。此时出现 Alarm DB Logger Manager。



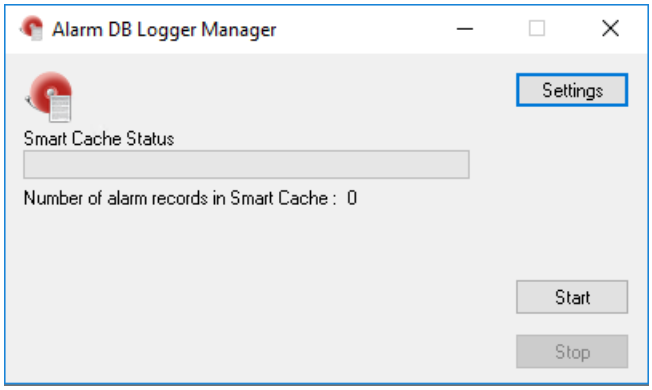
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

启动与停止报警数据库记录功能

Alarm DB Logger 将记录写入报警数据库。它既可以作为服务，也可以作为普通的应用程序来启动与运行（具体取决于配置 Alarm DB Logger 时选择的运行模式）。

启动或停止报警记录功能

1. 在工具视图中，展开应用程序。
2. 双击 **Alarm DB Logger Manager**。此时出现 Alarm DB Logger Manager。



   智能缓存状态显示报警记录占用内存缓冲区的百分比。
3. 单击开始以开始报警记录过程。
4. 单击停止以结束报警记录过程。

配置报警数据库记录功能

要将 InTouch 报警与事件数据记录到报警数据库，请从 Alarm DB Logger Manager 中执行以下操作：

- 配置与报警数据库的连接
- 选择要记录到报警数据库的报警
- 设置将记录保存到报警数据库的间隔
- 选择运行 Alarm DB Logger 的方法

建立数据库连接

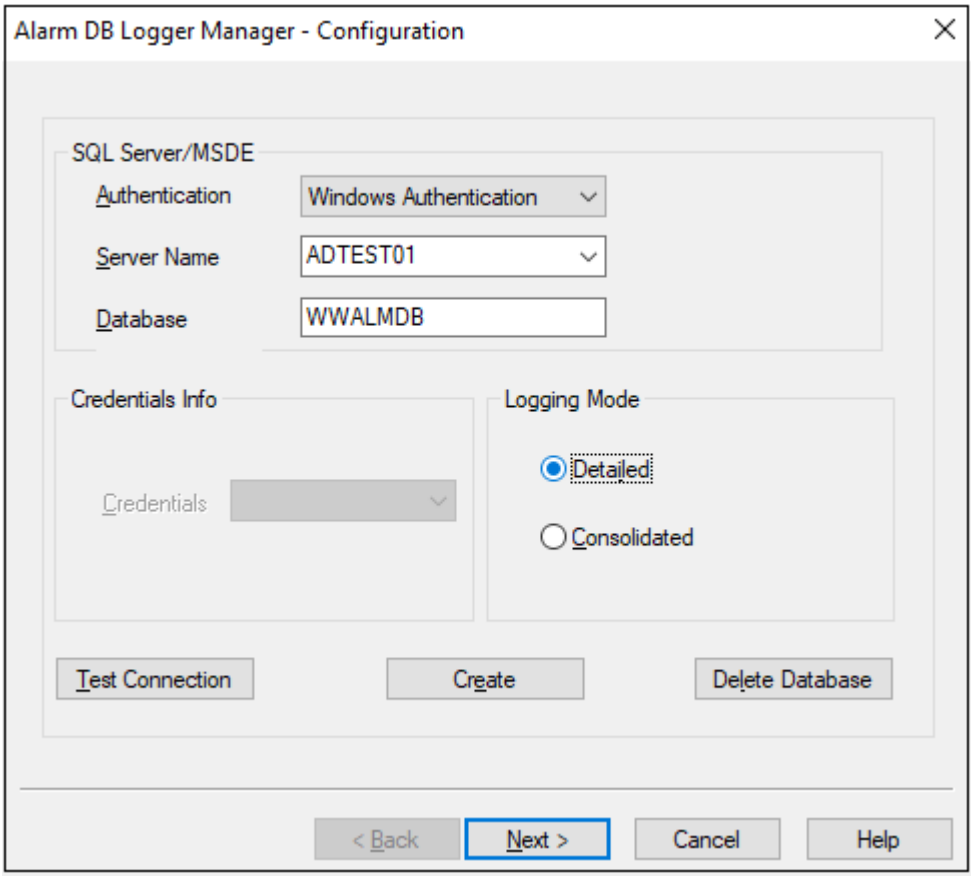
您必须在 Alarm DB Logger 与报警数据库之间建立连接。

Alarm DB Logger 使用 SQL Server 和 Windows 身份验证。SQL Server 安装必须设置为混合模式。

配置数据库连接

1. 打开 Alarm DB Logger Manager。执行以下操作：
   a. 在“工具”视图中，展开应用程序。
   b. 双击 **Alarm DB Logger Manager**。
2. 选择设置。此时出现 **Alarm DB Logger Manager - 配置**向导。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在 **SQL Server/MSDE** 部分中，**执**行以下操作：
   a. 在**身份验证**列表中，**选择**身份**验证**方法：SQL Server 身份**验证**或 Windows 身份**验证**（缺省）。

   注意：与 SQL 身份**验证**相比，Windows 身份**验证**可提供更好的**应**用程序安全性。因此，如果从 Windows 身份**验证**切**换为** SQL 身份**验证**，**则**将**显**示一个**对话**框，建**议**您使用 Windows 身份**验证**。如果您**选择**忽略此警告并**继续**使用 SQL 身份**验证**，**请选择确定**。在 OCMC Log Viewer 中将**记录**一条**类**似的消息。

   b. 在**服务器名**框中，**输**入安装了**报**警数据**库**的**计**算机的**节**点名。
   c. 在**数据库**框中，**输**入 InTouch **报**警数据**库**的名称。
4. 在**凭据信息**部分的**凭据**下拉列表中，**选择**用于身份**验证**的凭据。

   备注：
   - 仅当**选择**“SQL Server 身份**验证**”**类**型**时**，才会启用“凭据”下拉列表。Windows 身份**验证**方法使用当前登**录**用**户**的凭**证**，并禁用用**户**名和密**码**字段。
   对于独立 InTouch **应**用程序，从“**应**用程序管理器”中**检**索凭据。**对**于托管 InTouch **应**用程序，从 Application Server 的“凭据管理器”中**检**索凭据。如需**详细**信息，**请**参**阅**《AVEVA™ InTouch HMI **应**用程序**开发**》中的使用凭据管理器。
5. 在**记录模式**区域中，配置存**储记录**的方式。**执**行以下任一操作：
   a. **选择详细**，以便**为每**个**报**警条件（**处**于**报**警状**态**、已确**认**、已返回正常）存**储**一条**单**独的**记录**。
   b. **选择合并**，以便将**报**警的所有状**态**（**处**于**报**警状**态**、已确**认**、已返回正常）存**储**在一条**记录**中，并包含**每**次**转换**的**时间标签**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 如果需要，选择**创建**以创建数据库。
7. 选择**测试连接**以验证与报警数据库的连接。此时出现一条消息，指出已成功连接到数据库。
8. 选择下一步以配置所要记录的报警。如需详细操作说明，请参阅配置所要记录的报警。

配置所要记录的报警

在 Alarm DB Logger Manager 配置向导的第二个页面中，可以定义一个或多个查询，以便从 InTouch 或 Galaxy 报警供应器中选择报警。您也可以选择作为数据库查询一部分的报警优先级值范围。

远程节点使用以下查询语法：

\\节点名\Provider!AlarmGroup

本地节点使用以下查询语法：

\Provider!AlarmGroup

示例：

\\ProdSvr\InTouch!$System

以下是使用 Galaxy 报警供应器的示例。此查询提供特定区域的所有报警。

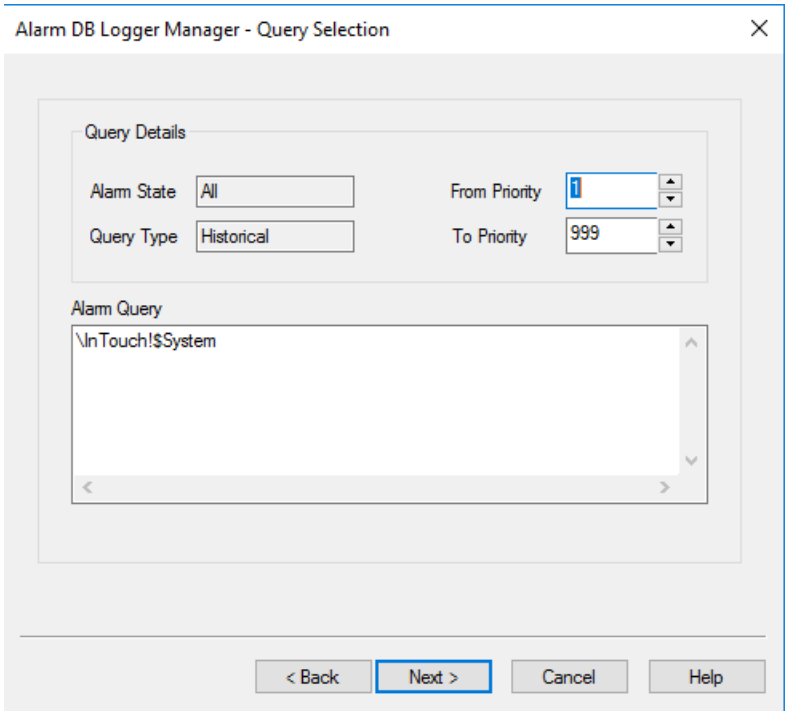
\Galaxy!Area

注意：Alarm DB Logger Manager 仅支持 50 个查询，3072 个字符。

配置所要记录的报警

1. 打开 Alarm DB Logger Manager 并启动配置向导。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Logger Manager**。
   c. 选择设置。此时出现 **Alarm DB Logger Manager - 配置**向导。
2. 选择下一步。此时出现 **Alarm DB Logger Manager - 查询选择**页面。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



只读的**报警状态**框显示用于记录的报警状态。只读的**查询类型**框显示查询类型。

3. 在“从优先级”框中，输入报警优先级范围的起始值。
4. 在“到优先级”框中，输入报警优先级范围的结束值。
5. 在“报警查询”框中，输入要用于在报警数据库中存储或检索数据的报警查询。

注意：Historian 不支持包含 128 个字符的区域名称。

6. 选择下一步以配置记录间隔。请参阅配置记录间隔。

配置记录间隔

在 Alarm DB Logger Manager 配置向导的第三个页面中，可以配置高级查询设置。您可以将事件记录到报警数据库中。通过设置将报警记录从内部报警内存写入数据库的频率，您也可以调整报警记录功能的性能。

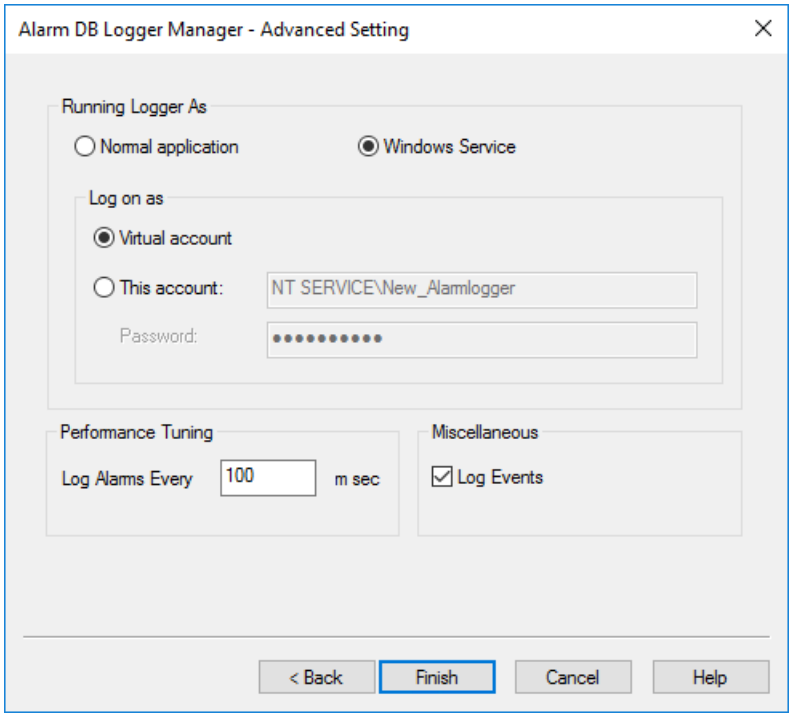
记录间隔不同于 SQL Server 的重新连接速率。记录间隔是读取报警的间隔。重新连接速率取决于尝试与 SQL Server 建立连接的超时值。

将记录间隔设置得过低会影响系统性能。

配置记录间隔

1. 打开 Alarm DB Logger Manager 并启动配置向导。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Logger Manager**。
   c. 选择设置。此时出现 **Alarm DB Logger Manager - 配置**向导。
2. 选择下一步。此时出现 **Alarm DB Logger Manager - 查询选择**页面。
3. 选择下一步。此时出现 **Alarm DB Logger Manager - 高级设置**页面。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



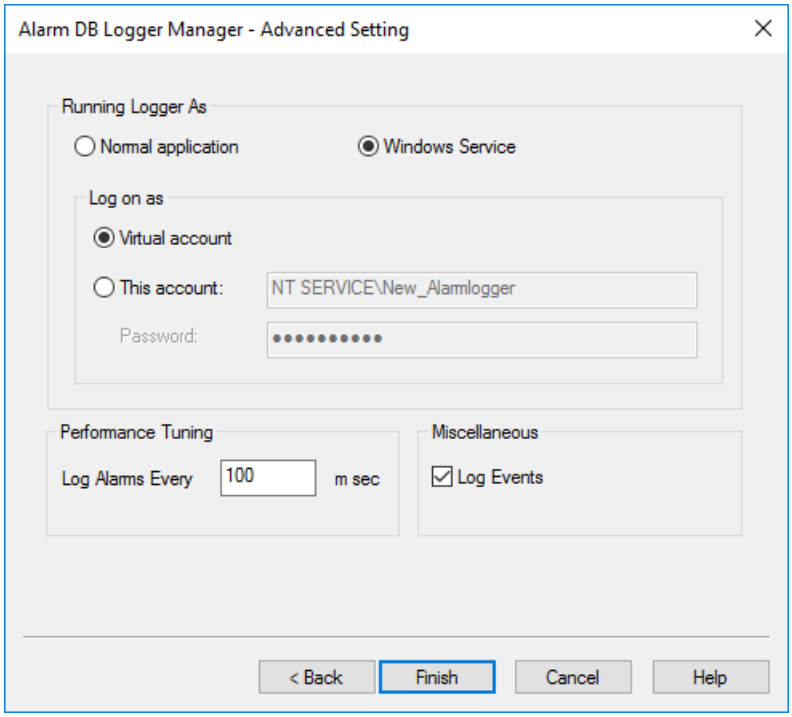
4. 如果希望将 InTouch 事件记录存储到报警数据库，请选中**记录事件复选**框。您还可以储存来自 Application Server Galaxy 的事件。
5. 在**性能调整**区域中，输入将报警记录写入报警数据库的间隔（以毫秒为单位）。
6. 选择完成。

将 Alarm DB Logger 配置成服务

您可以将 Alarm DB Logger 配置为 Windows 服务来运行。为减少以管理员权限运行 Alarm DB Logger 时的安全隐患，该用户帐户权限已设置为非交互式，以将 Alarm DB Logger 作为服务运行。

1. 作为管理员登录到计算机。
2. 打开 Alarm DB Logger Manager 并启动配置向导。执行以下操作：
   a. 在**工具视图**中，展**开应用程序**。
   b. 双击 **Alarm DB Logger Manager**。
   c. 选择**设置**。此时出现 **Alarm DB Logger Manager - 配置**向导。
3. 选择下一步。此时出现 **Alarm DB Logger Manager - 查询选择**页面。
4. 选择下一步。此时出现 **Alarm DB Logger Manager - 高级设置**页面。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 在**记录器运行方式为**区域中，选择正常应用程序或 **Windows 服务**。

   凭据将取决于在 **Alarm DB Logger Manager - 配置** 向导页中选择的身份验证方法。

6. 在**登录身份**区域中，根据以下条件选择登录帐户，然后选择完成。

身份验证方法	记录器运行方式为	登录身份	用户名	密码
SQL Server 身份验证	Windows 服务	虚拟账户	--	--
Windows 身份验证	Windows 服务	虚拟账户	--	--
Windows 身份验证	Windows 服务	此帐户	<用户名>	<口令>

注意：如需有关虚拟帐户的详细信息，请参阅使用虚拟帐户。

缺省情况下，**选择此帐户**将显示当前登录用户的用户名，此时不需要口令。如果更改用户名，则必须提供口令。

必须为用户帐户提供适当的权限才能完成配置。

下次启动 Alarm DB Logger 时，它将预先填充配置详细信息。

使用虚拟帐户

选择虚拟帐户，就可以添加额外的安全层。选择此选项，将以名为 New_AlarmLogger 的 NT 服务帐户身份启动 Alarm DB Logger。

虚拟帐户既支持 SQL，也支持 Windows 身份验证。对于使用“LocalSystem”配置的较早的系统，该帐户将做如下迁移：

- 如果先前的系统是使用“LocalSystem”在“此帐户”或“本地系统帐户”选项中配置的，那么“此帐户”将使用“LocalSystem”作为用户进行预先填充。
- 如果先前的系统是使用“此帐户”作为“NT 服务\New_AlarmLogger”配置的，那么它将迁移到“虚拟帐户”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

报警数据库视图

您可以使用一组 SQL Server 数据库视图轻松查询过去与当前发生的报警与事件。

数据库视图只是一个逻辑表，它将多个底层数据库表中的信息合并到一起。数据库视图通常称为虚拟数据库表，原因在于它们可以使用标准的 SQL 语句来查询，好像它们是真实存在的表格那样。查询视图时，它返回一组记录（或称作行）。每行有多列信息，包含的是记录数据。

您可以使用报警数据库视图来执行复杂的查询。例如，您可以检索工厂的特定区域在特定的时间跨度内发生的所有 HiHi 报警的报警记录。或者，您也可以检索某个特定的报警供应器节点所记录的所有数据改变事件。

视图中的所有字符串都采用 Unicode 编码。

报警历史视图

v_AlarmHistory 视图包含所选时间范围内发生的所有历史报警与报警转换事件。查询指定开始和结束日期与时间（EventStamp 或 EventStampUTC 列）。返回的记录包括报警发生、报警确认、报警启用、报警禁用以及报警返回正常等事件。

下表介绍视图中的各个列：

列名	数据类型	描述
EventStamp	日期时间	报警事件的日期与时间（采用数据库本地时间）
AlarmState	nChar	报警状态：UNACK、UNACK_RTN、ACK、ACK_RTN、LATCHED 之一。
TagName	nChar	生成报警的对象的名称，如 TIC101。
描述	nVarchar	报警的描述字符串。缺省情况下，可为对象描述（或 InTouch HMI 中的注释）。 或者是已确认的记录的确认注释。
区域	nChar	报警的“区域”或“组”的名称。
类型	nChar	报警类型，例如 Hi、HiHi、ROC、PV.HiAlarm。
值	nChar	报警时报警变量的值。
CheckValue	nChar	报警时报警限的值。
优先级	整型	报警优先级。
类别	nChar	报警类或报警类别。例如“值”、“偏差”、“变化率”、“过程”、“批次”、“系统”，等等。
供应器	nChar	报警的供应器：节点/InTouch，或 GalaxyName。
操作员	nChar	操作员的姓名。
DomainName	nChar	域的名称。
UserFullName	nChar	操作员用户的全名 （例如，Joseph P. Smith）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

列名	数据类型	描述
UNACKDuration	浮点	最近一次报警转换（报警或子状态）与确认之间的时间（如果有）。
用户 1	浮点	用户自定义字段编号 1。
用户 2	浮点	用户自定义字段编号 2。
User 3	nChar	用户自定义字段，为字符串类型。
EventStampUTC	日期时间	报警事件的 UTC 日期/时间。
Millisec	Small integer	事件标签秒值的小数部分，以 0.1 毫秒为增量。
操作员节点	nvarchar(32)	操作员在其中确认报警的节点的名称。

v_AlarmHistory2 视图是 v_AlarmHistory 视图的变体。

列名	数据类型	描述
EventStamp	日期时间	报警事件的日期与时间（采用数据库本地时间）
AlarmState	nChar	报警状态：UNACK、UNACK_RTN、ACK、ACK_RTN、ACK_ALM、UNACK_ALM、LATCHED 之一
TagName	nChar	生成报警的对象的名称，如 TIC101。
描述	nVarchar	报警的描述字符串。缺省情况下，可以为对象描述（或 InTouch 中的注释）。或者是已确认的记录的确认注释。
区域	nChar	报警的“区域”或“组”的名称。
类型	nChar	报警类型，例如 Hi、HiHi、ROC、PV.HiAlarm。
值	nChar	报警时报警变量的值。
CheckValue	nChar	报警时报警限的值。
优先级	整型	报警优先级。
类别	nChar	报警类或报警类别。例如“值”、“偏差”、“变化率”、“过程”、“批次”、“系统”，等等。
供应器	nChar	报警的供应器：节点/InTouch，或 GalaxyName。
操作员	nChar	操作员名称：JoeR（如果有）。
DomainName	nChar	域的名称。
UserFullName	nChar	操作员用户的全名（例如，Joseph P. Smith）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

列名	数据类型	描述
AlarmDuration	浮点	报警发生与返回正常之间的时间。
用户 1	浮点	用户自定义字段编号 1。
用户 2	浮点	用户自定义字段编号 2。
User 3	nChar	用户自定义字段，为字符串类型。
EventStampUTC	日期时间	报警事件的 UTC 日期/时间。
Millisec	Small integer	事件标签秒值的小数部分， 以 0.1 毫秒为增量。

报警历史视图示例查询

本例从“报警历史”视图中选择所有记录：

```
SELECT * FROM v_AlarmHistory
```

本例从“报警历史”视图中选择优先级大于 100 的所有记录。

```
SELECT * FROM v_AlarmHistory WHERE Priority >100
```

Event history view

v_EventHistory 数据库视图列出所选时间范围内发生的所有历史事件。查询客户端指定开始和结束日期与时间范围。返回的记录包括所有的非报警事件。

列名	数据类型	描述
EventStamp	日期时间	事件的日期与时间。
TagName	nChar	生成事件的对象的名称，如 Pump1。
描述	nVarChar	事件的描述字符串。缺省情况下，可以为对象描述，或 InTouch 中的注释。
区域	nChar	事件的“区域”或“组”的名称。
类型	nChar	事件类型，如“操作员数据改变”、“启动时”，等等。
值	nChar	新值（如果有）。
CheckValue	nChar	旧值（如果有）。
类别	nChar	事件类别或类，如“值”、“过程”、“批次”、“系统”，等等。
供应器	nChar	事件发生器，如节点/InTouch，或用于更改用户的“视图引擎”的名称。
操作员	nChar	operator1 的名称：JoeR（如果有）。
DomainName	nChar	域的名称。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

列名	数据类型	描述
UserFullName	nChar	操作员用户的全名 （例如，Joseph P. Smith）。
用户 1	浮点	用户自定义字段编号 1。
用户 2	浮点	用户自定义字段编号 2。
User 3	nChar	用户自定义字段，为字符串类型。
EventStampUTC	日期时间	事件的 UTC 日期/时间。
Millisec	Small integer	事件标签秒值的小数部分，以 0.1 毫秒为增量。

报警事件历史视图

v_AlarmEventHistory 数据库视图提供所选时间范围内发生的所有事件与报警的历史列表。查询客户端指定开始和结束日期与时间。返回的记录包括所有的报警与事件。该视图将报警视图与事件视图合而为一，代表这两个视图中的记录的集合。

列名	数据类型	描述
EventStamp	日期时间	事件的日期与时间。
AlarmState	nChar	报警状态：UNACK、UNACK_RTN、ACK、ACK_RTN、LATCHED 之一。不适用于事件。
TagName	nChar	生成报警的对象的名称，如 TIC101。
描述	nVarchar	报警/事件的描述字符串。缺省情况下，可以为对象描述（或 InTouch 中的注释）。 或者是已确认的记录的确认注释。
区域	nChar	报警的“区域”或“组”的名称。
类型	nChar	报警或事件的类型，例如 Hi、HiHi、ROC、PV.HiAlarm、“操作员数据更改”，等等。
值	nChar	报警时报警变量的值。
CheckValue	nChar	报警时报警限的值，或事件的旧值。
优先级	整型	报警优先级。
类别	nChar	报警或事件的类，或报警类别，如“值”、“过程”、“批次”、“系统”，等等。
供应器	nChar	报警的供应器，如节点/InTouch，或 GalaxyName。
操作员	nChar	确认操作员或数据改变操作员的名称。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

列名	数据类型	描述
DomainName	nChar	域的名称。
UserFullName	nChar	操作员用户的全名 （例如，Joseph P. Smith）。
UNACKDuration	浮点	从最近一次报警转换到“确认”所经过的毫秒数。
用户 1	浮点	用户自定义字段编号 1。
用户 2	浮点	用户自定义字段编号 2。
User 3	nChar	用户自定义字段，为字符串类型。
EventStampUTC	日期时间	事件的 UTC 日期/时间。
Millisec	Small integer	事件标签秒值的小数部分， 以 0.1 毫秒为增量。

v_AlarmEventHistory2 视图是 v_AlarmEventHistory 视图的变体。

列名	数据类型	描述
EventStamp	日期时间	事件的日期与时间。
AlarmState	nChar	报警状态：UNACK、UNACK_RTN、ACK、ACK_RTN、LATCHED 之一。不适用于事件。
TagName	nChar	生成报警的对象的名称，如 TIC101。
描述	nVarchar	报警/事件的描述字符串。缺省情况下，可以为对象描述（或 InTouch 中的注释）。 或者是已确认的记录的确认注释。
区域	nChar	报警的“区域”或“组”的名称。
类型	nChar	报警或事件的类型，例如 Hi、HiHi、ROC、PV.HiAlarm、“操作员数据更改”，等等。
值	nChar	报警时报警变量的值。
CheckValue	nChar	报警时报警限的值，或事件的旧值。
优先级	整型	报警优先级。
类别	nChar	报警或事件的类，或报警类别，如“值”、“过程”、“批次”、“系统”，等等。
供应器	nChar	报警的供应器，如节点/InTouch，或 GalaxyName。
操作员	nChar	确认操作员或数据改变操作员的名称。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

列名	数据类型	描述
DomainName	nChar	域的名称。
UserFullName	nChar	操作员用户的全名 （例如，Joseph P. Smith）。
AlarmDuration	浮点	从报警发生到返回到正常 (RTN) 所经过的毫秒数。
用户 1	浮点	用户自定义字段编号 1。
用户 2	浮点	用户自定义字段编号 2。
User 3	nChar	用户自定义字段，为字符串类型。
EventStampUTC	日期时间	事件的 UTC 日期/时间。
Millisec	Small integer	事件标签秒值的小数部分，以 0.1 毫秒为增量。

报警事件历史视图示例查询

对于标记名为 MyTag1、报警状态为 ACK_RTN 或 ACK 的所有记录，下例选择这些记录中 TagName、Area 以及 Type 列中的数据。结果按报警供应器排序。

```
SELECT TagName, Area, Type FROM v_AlarmEventHistory
   WHERE TagName='MyTag1'
      AND (AlarmState='ACK_RTN' OR AlarmState='ACK')
      ORDER BY Provider
```

报警数据库存储过程

您可以使用一组 SQL Server 存储过程轻松检索信息，以便用于分析以及报警与事件的报告。

存储过程是 SQL 语句的集合，用于执行特定的功能并从数据库返回数据。存储过程可以接受一些输入参数，控制它们如何工作，如何以结果集的形式返回数据。

返回的结果是一组记录，以非常类似于 SQL Server 数据库视图的 SQL 记录集的面目出现。存储过程在数据库中以内嵌 SQL 语句的形式存在，可减少同客户端之间的往返，因此可以提高性能。

如需有关存储过程的详细信息（包括如何查看它们的定义），请参阅 Microsoft SQL Server 文档。

调用存储过程

使用 EXECUTE 语句调用存储过程。

```
EXECUTE sp_AlarmCounter @StartDate='2007-01-01', @EndDate='2007-03-31', @Tagname = 'tag1',
@Type = 'LO', @Provider = 'WW21353\InTouch', @Comment = 'SSAADD'
```

在本例中，StartDate 与 EndDate 参数是必需的。其余参数属于可选项。如果不提供某个参数的值，则它不会用于过滤结果集。

如果存储过程包含日期/时间变量，则可以使用 SQL Server 文档中指定的任何有效格式。

AlarmCounter 数据库存储过程

此存储过程返回一定时间间隔内发生独特报警的次数。独特的报警由 TagName、Provider、AlarmType 及 Category 来标识。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此计数器仅适用于报警发生次数（不适用于确认或返回到正常之类的转换）。因此举例而言，如果发生某个报警，然后确认该报警，再然后该报警返回到正常状态，则该报警的计数只有 1（而不是 3）。

查询必须指定开始和结束日期与时间。它有五个可选参数：TagName、Class、Type、Provider 以及 Comment。

此视图可回答的问题的例子有：供应器 Node1|InTouch（供应器）上的对象 TIC101（标记名）在某个时间跨度上发生了多少次 HiHi（类型）值报警（类别）？

注意：AlarmCounter 存储过程仅适用于“详细”记录模式，不受“合并”记录模式的支持。

sp_AlarmCounter

列名	数据类型	描述
TagName	Nchar	生成报警的对象的名称，如 TIC101。
组名	Nchar	报警的区域或组的名称。
AlarmType	Nchar	报警类型，例如 Hi、HiHi、ROC、PV、HiAlarm。
AlarmClass	Nchar	报警类或报警类别，如“值”、“过程”、“批次”等。
AlarmCount	整型	一定时间范围内报警发生的次数。 如果报警发生在开始日期与时间之前，则它不包括在计数内。
优先级	整型	报警优先级。
供应器	Nchar	报警的供应器，如节点/InTouch，或 GalaxyName。
注释	Nchar	报警注释。

示例查询：

```
EXECUTE sp_AlarmCounter @StartDate=’2007-01-01 23:23:23’, @EndDate=’2007-03-31 23:23:23’,
@Tagname = ’$NewAlarm’
```

EventCounter 数据库存储过程

此数据库存储过程提供一定时间间隔内某个特定标记上发生的某一特定类型的事件数。查询客户端必须指定开始和结束日期与时间。它有三个可选参数：TagName、Provider 以及 Comment。此计数器仅适用于非报警事件。此视图的用途是显示每个事件的频率。例如，泵浦打开了多少次？TagName、Provider、Category 及 Type 用于唯一确定事件并执行计数。

sp_EventCounter

列名	数据类型	描述
TagName	NChar	生成报警的对象的名称，如 TIC101
区域	NChar	报警的“区域”或“组”的名称。
类型	NChar	事件类型。
类别	NChar	报警类或报警类别，如“值”、“过程”、“批次”等。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

列名	数据类型	描述
EventCount	整型	在指定的时间范围内 TagName（标记名）此 Type（类型）的事件发生的次数。
供应器	NChar	事件的供应器：节点/InTouch，或 GalaxyName。
注释	NChar	事件注释。

示例查询：

<1>EXECUTE sp_EventCounter @StartDate=</1>'<2>2007-01-01 23:23:23</2>'<3>, @EndDate=</3>'<4>2007-03-31 23:23:23</4>'<5>, @Tagname = </5>'<6>$NewAlarm</6>'

查看记录的报警

您可以使用 Alarm DB View ActiveX 控件以可视化方式显示报警数据库中的数据。使用此控件可以显示 InTouch 应用程序在运行时生成的所有报警与事件信息。

对于此控件，您可以配置：

- 上下文相关菜单功能
- 显示模式
- 列表控制选项
- 不同属性的颜色
- 字体类型、样式和大小
- 数据库规格（服务器名、用户 ID 和口令）
- 查询过滤器
- 列管理
- 排序

在应用程序运行期间，运行时用户可以更改一些选项来选择要查看的数据。它们可以：

- 给列中的信息排序
- 更新显示对象
- 执行查询
- 调整列宽

如需有关 ActiveX 控件的详细信息，请参阅 ActiveX 控件。

配置 Alarm DB View 控件

配置 Alarm DB View 控件时，可以：

- 配置与报警数据库的连接。
- 配置网格外观。
- 选择显示字体。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 配置视觉外观。
- 设置用户在应用程序运行期间可以访问的功能。
- 配置要显示的报警。
- 创建自定义过滤器。
- 设置各类报警记录的颜色。
- 配置如下所示的时间格式。
- 配置显示的报警记录的排序顺序。

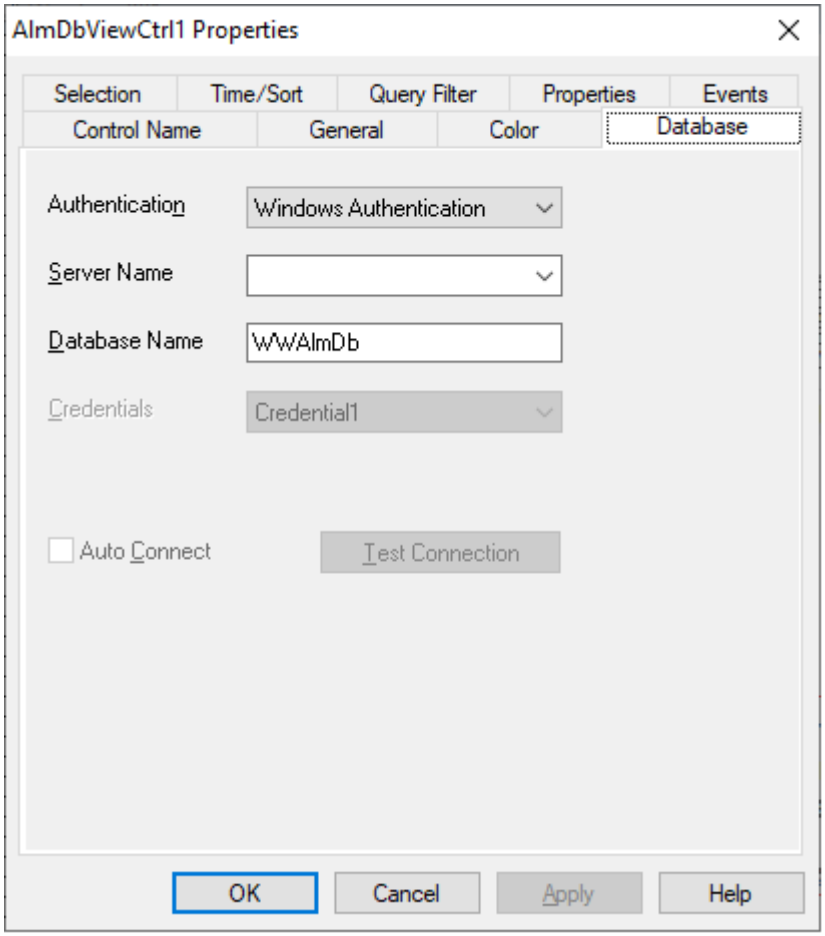
将 Alarm DB View 连接到报警数据库

您必须配置 Alarm DB View 控件与报警数据库之间的连接。

使用对报警数据库有适当的只读访问权限的帐户，而不是系统管理员帐户。

配置与报警数据库的连接

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl** 属性对话框。
2. 选择数据库选项卡。



3. 配置与报警数据库的连接。执行以下操作：
   a. 在身份验证列表中，选择身份验证方法：**SQL Server 身份验证**或 **Windows 身份验证**（缺省）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：与 SQL 身份验证相比，Windows 身份验证可提供更好的应用程序安全性。因此，如果从 Windows 身份验证切换为 SQL 身份验证，则将显示一个对话框，建议您使用 Windows 身份验证。如果您选择忽略此警告并继续使用 SQL 身份验证，请选择确定。在 SMC Log Viewer 中将记录一条类似的消息。

b. 在服务器名框中，输入安装了报警数据库的计算机的节点名。

c. 在数据库名框中，输入数据库名。

d. 在凭据框中，选择用于身份验证的凭据。

注意：
- 仅当选择“SQL Server 身份验证”类型时，才会启用“凭据”下拉列表。Windows 身份验证方法使用当前登录用户的凭证，并禁用用户名和密码字段。
对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发》中的使用凭据管理器。

4. 选中自动连接复选框，可以使 Alarm DB View 控件在 InTouch 应用程序开始运行时自动连接到报警数据库。

   如果不选中自动连接复选框，则必须通过明确调用 Connect() 方法将 Alarm DB View 控件配置为连接到报警数据库。如需有关 Connect 方法的详细信息，请参阅 Connect()。

5. 选择测试连接以验证与报警数据库的连接。此时出现一条消息，指出连接成功。

6. 选择应用。

配置网格的外观属性

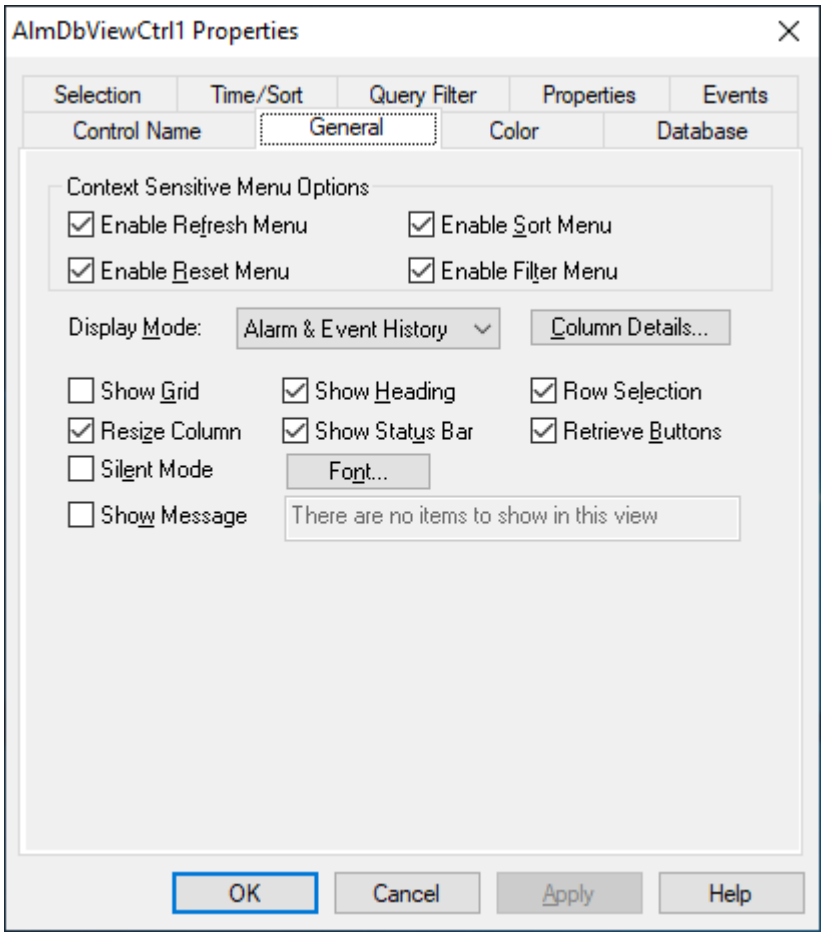
您可以配置用于确定 Alarm DB View 控件视觉外观的属性。

配置网格外观

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl** 属性对话框。

2. 选择常规选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 配置常规外观。执行以下操作：
   - 选中**显示网格**复选框，以显示网格。
   - 选中**无提示模式**复选框，以防止控件在运行时显示任何错误消息。
   - 选中**显示消息**复选框，以便在报警查询未返回任何记录的情况下显示一条消息。在方框中输入要显示的消息。
   - 选中**显示标题**复选框以显示控件标题。
   - 选中**显示状态栏**复选框以显示状态栏。
4. 选择**应用**。

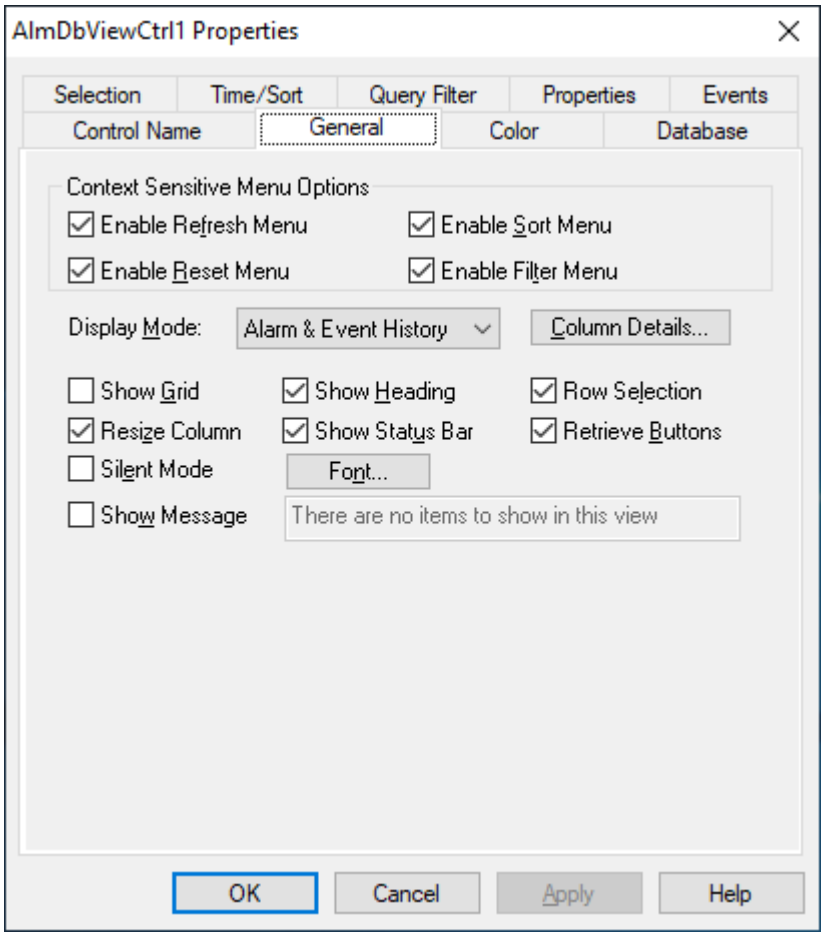
配置显示字体

您可以选择 Alarm DB View 控件中出现的所有文本的字体。

配置字体属性

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl 属性**对话框。
2. 选择常规选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 选择字体。此时出现标准的 Windows“字体”对话框。配置字体，然后单击确定。
4. 选择应用。

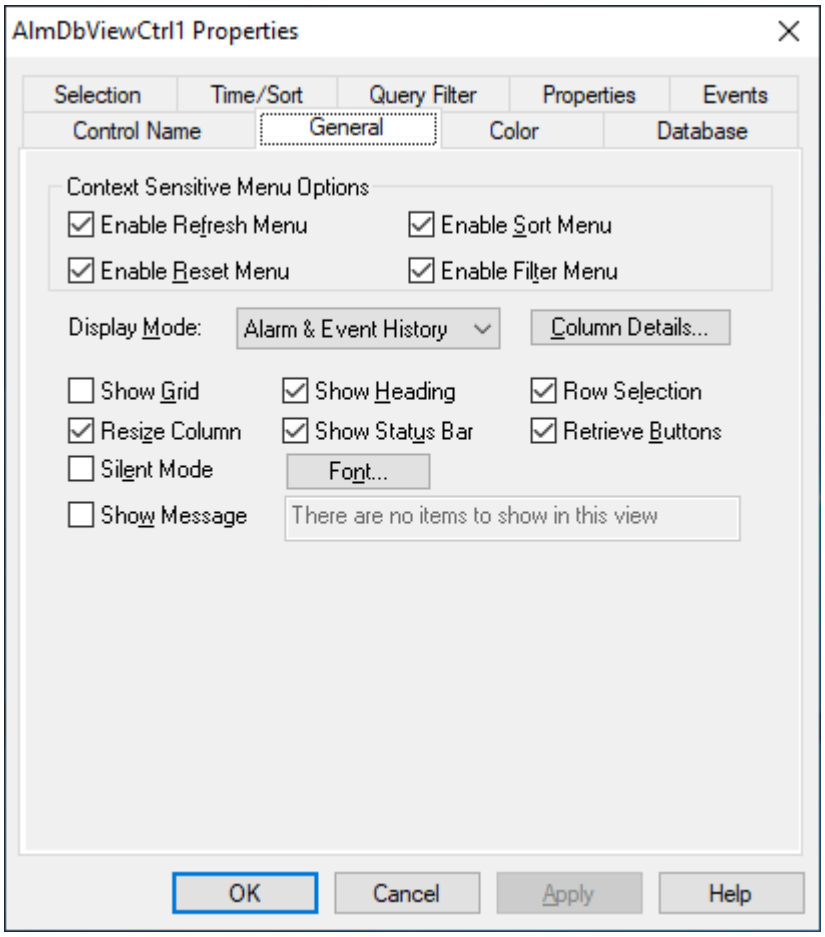
选择报警或事件数据

您可以配置在 Alarm DB View 控件中显示报警记录、事件记录，或同时显示这两者。

选择数据类型

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl** 属性对话框。
2. 选择常规选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**显示模式**列表中，**选择历史记录的类型**：
   a. **选择报警和事件历史**，以同时显示报警与事件的历史数据库记录。
   b. **选择报警历史**以便仅显示历史报警记录。
   c. **选择事件历史**以便仅显示历史事件记录。
4. **选择应用**。

选择与配置显示列

对于 Alarm DB View 控件，可以：

- **选择**各个列并进行排序。
- 以像素为单位设置列宽。
- 重命名列。

您至少必须选择一个列。

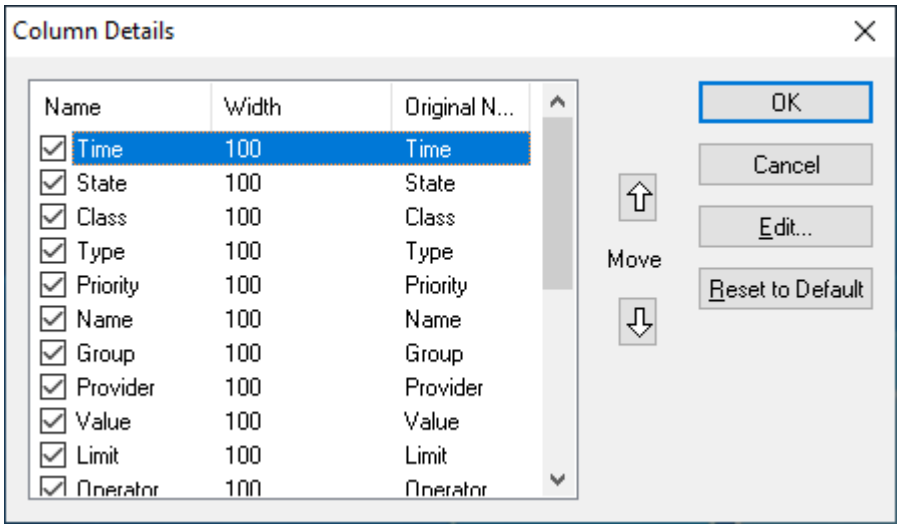
要点：如果显示模式发生更改，则列名会重置为缺省的列名。在更改列名之前，最好先选择显示模式。

配置显示列明细

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl 属性对话**框。
2. **选择常规选项**卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 选择列明细。此时出现列明细对话框。



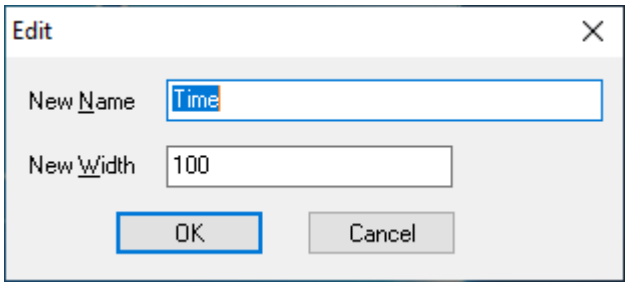
4. 在名称列中，对于希望出现在 Alarm DB View 控件中的列，选择其名称旁边的复选框。您至少必须从列表中选择一个列。

列名	描述
时间	显示报警发生的时间。
状态	显示报警的状态。
类	显示报警类别。
类型	显示报警类型。
优先级	显示报警优先级。
姓名	显示触发报警或事件的标记或源。
组	显示报警组名。
供应器	显示报警供应器的名称。
值	显示报警发生时标记的值。
限制	显示标记的报警限值。
操作员	显示与报警条件关联的已登录操作员的 ID。
操作员全名	显示已登录操作员的全名。
操作员节点	显示与报警条件关联且已登录的操作员的节点。 在“终端服务”环境中，这是操作员建立“终端服务”会话的客户端计算机的名称。如果无法检索节点名，则使用节点的 IP 地址代替。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

列名	描述
操作员域	显示与报警条件关联且已登录的操作员的域。
报警注释	显示标记的注释。此注释是在数据库中定义标记的报警时，在**报警注释**框中输入的。 为报警引入某个确认注释时，新的注释更新到这个注释列中。
用户 1	显示与报警对应的用户自定义数字 1 的数值。
用户 2	显示与报警对应的用户自定义数字 2 的数值。
用户 3	显示与报警关联的用户定义字符串属性的字符串值。
Duration	显示未确认持续时间或报警持续时间，具体取决于操作员所作的选择。
UTC 时间	显示 UTC 格式的报警时间。

5. 通过选择列名并使用上、下箭头，重新排列各个列。列明细对话框顶部显示的列名是报警控件最左侧的列。
6. 要更改某个列的名称或其宽度，请选择该列并选择**编辑**。此时出现“编辑”对话框。



   a. 在新名框中，输入新的列名。
   b. 在新宽度框中，输入列宽。列宽可以在从 1 到 999 个像素的范围内。
   c. 选择确定。
7. 选择重置为缺省值，以恢复缺省的列明细设置。
8. 选择列明细对话框中的确定。
9. 选择应用。

控制用户可以在运行时访问哪些功能

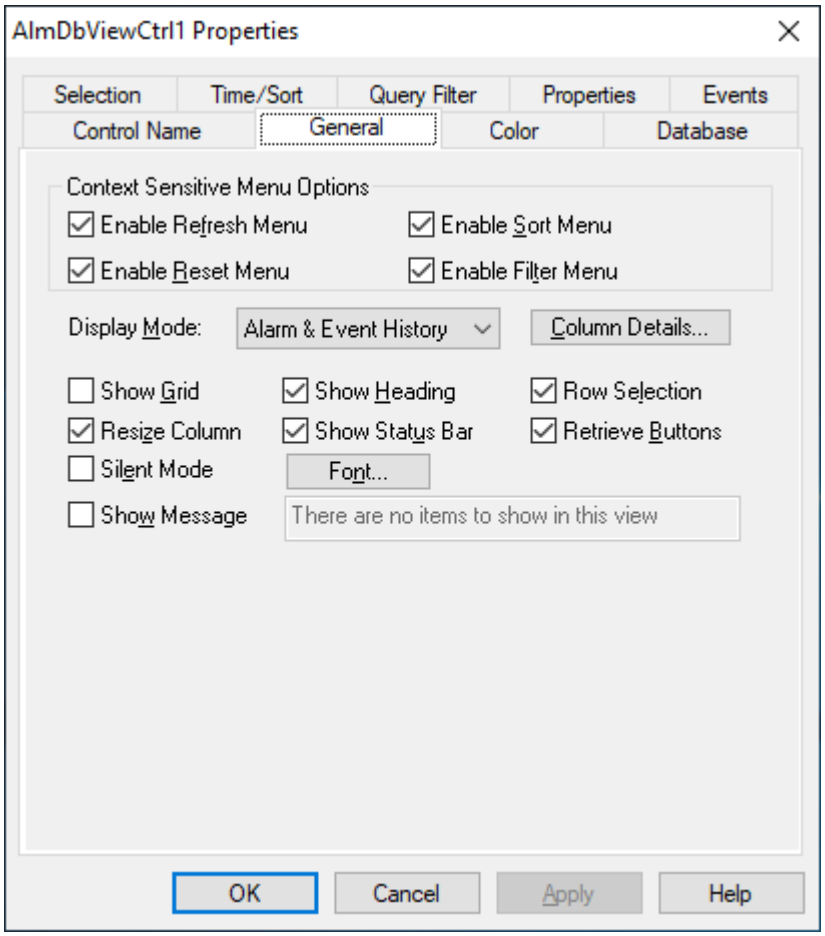
您可以启用与禁用此控件中可用的快捷菜单功能。

配置运行时功能

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl** 属性对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 选择常规选项卡。



3. 在上下文相关菜单选项区域中，配置在运行时可用的菜单命令。
   - 选中启用刷新菜单复选框，以在运行时启用控件快捷菜单中的刷新菜单选项。刷新菜单根据数据库来刷新控件，如果连接成功，它显示 1 到 MaxRecords 属性所定义的数字这个范围内的一组记录。
   - 选中启用复位菜单复选框，以在运行时启用控件快捷菜单中的重置菜单选项。重置菜单将所有的列还原为设计时保存的设置。
   - 选中启用排序菜单复选框，以在运行时启用控件快捷菜单中的排序菜单选项。此菜单显示第二位排序菜单，用于设置用户自定义的列排序方式。
   - 选中启用过滤器菜单复选框，以在运行时启用控件快捷菜单上的过滤器菜单选项。此菜单显示过滤器菜单，用于设置用户自定义的过滤准则。
4. 选中改变列尺寸复选框以调整列宽。
5. 选中行选择复选框，以允许选择报警记录。
6. 选中检索按钮复选框，以在控件右侧显示检索按钮。
7. 选择应用。

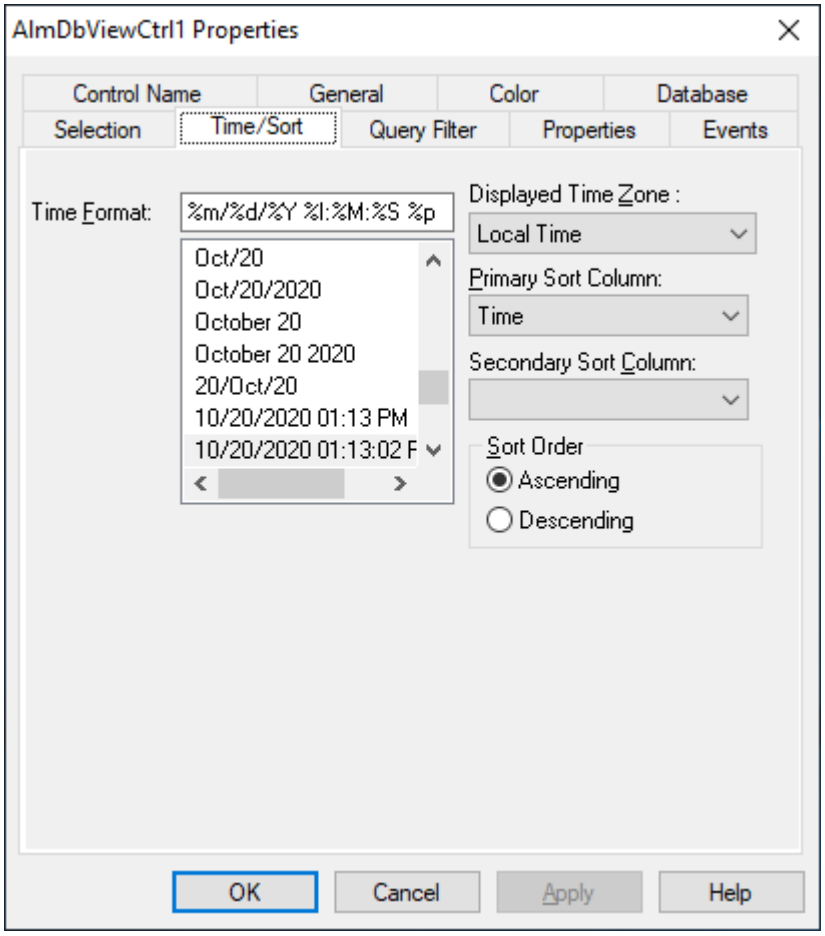
配置为报警记录显示的时间格式与时区

您可以为 Alarm DB View 控件中显示的报警记录配置时间格式与时区。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置时间格式

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl** 属性对话框。
2. 选择时间/排序选项卡。



3. 在时间格式列表中，选择某个时间格式：

字符串字符	描述	示例
d	两位数字日期值	31
b	三个英文字母的月份缩写	八月
是	四位数字年份值	2007
m	两位数字月份值	11
/	日期分隔符	06/2007
y	两位数字年份值	07
#x	完整的日期与星期	2002 年 8 月 9 日星期五

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串字符	描述	示例
B	完整的月份名	九月
-	连字符日期分隔符	07-06
。	句点日期分隔符	07.06
、	逗号日期分隔符	八月 09, 2007
H	24 小时制时间	22:15
:	时间分隔符，如 4:41	
M	分钟	00:41
p	显示的“上午”或“下午”	
S	秒钟	16:41:07
s	秒钟的小数部分	16:41:07.390
I	指定 AM/PM 的 12 小时制时间	04:41 下午

您也可以使用自定义文本与格式字符在列表框中手动输入自己的格式字符串。下面是一些时间格式字符串的示例：

时间格式字符串	显示
%d %b	09 八月
%m/%d/%Y	08/09/2002
%#x	2002 年 8 月 9 日星期五
%Y-%m-%d	2002-08-09
%Y/%m/%d %H:%M %p	2002/08/09 16:56 下午
%Y/%m/%d %H:%M:%s %p	2002/08/09 16:56:38 下午
%I:%M %p	04:56 下午

4. 在显示时区列表中，选择时区：

时区	描述
GMT	报警时间标签使用“格林尼治标准时间”。
本地时间	报警时间标签显示时使用 Alarm DB View 控件所在客户端的本地时间。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

时区	描述
原始时间	报警时间标签显示时使用报警供应器的本地时间。

5. 选择应用。

选择报警数据库中数据的时间段

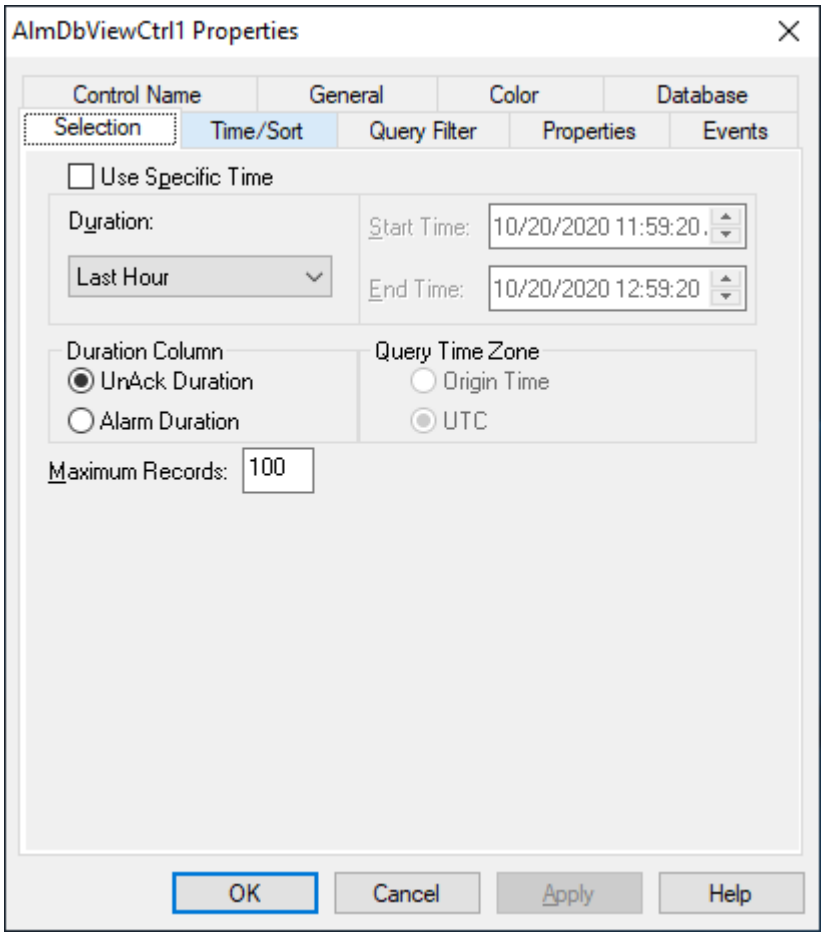
您可以设置查询值来根据所选的时间选择记录。您也可以配置要查看的最大记录数、报警查询的开始与结束时间，以及查询时区。

通过报警数据库查询获取的记录数会影响在 Alarm DB View 控件中显示记录所需的时间。要显示 50000 条从报警数据库获取的记录，Alarm DB View 控件需要大约 30 秒。

要提高 Alarm DB View 控件的性能，请减少最大记录数值以更快显示记录。此外，还可以在持续时间字段中选择较短的记录获取间隔，以提高 Alarm DB View 控件的性能。

选择数据的时段

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl** 属性对话框。
2. 选择选择选项卡。



3. 要使用预定义的时间间隔（查询数据时总是使用 UTC 时区），请选择持续时间列表中的某个间隔。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 要使用特定的开始时间与结束时间，请选择使用指定时间，然后配置详细资料。
   a. 在**开始时间**框中，输入检索报警记录的开始时间。字符串必须采用 YYYY/MM/DD HH:MM:SS 格式。使用任何时区中自 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 之间的任何日期。
   b. 在**结束时间**框中，输入停止检索报警记录的结束时间。字符串必须采用 YYYY/MM/DD HH:MM:SS 格式。使用任何时区中自 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 之间的任何日期。
   c. 在**查询时区区域**中，单击 **UTC** 或**原始时间**。UTC 时间是“格林尼治标准时间”，也称作“通用协调时间”或 Zulu。原始时间指操作员所在时区的当前时间。如需详细信息，请参阅查询时区主题。
5. 在**持续时间列**区域中，配置显示哪种类型的持续时间。持续时间以毫秒为单位计算。
   - 选择**未确认持续时间**，以显示最近的报警转换（报警或子状态）与确认（如果有）之间的时间。
   - 选择**报警持续时间**，以显示报警最初的实例与它返回到正常的时间之间经过的时间量。

     如需有关计算 LATCHED 状态的未确认持续时间和报警持续时间的详细信息，请参阅 LATCHED 报警的报警持续时间和 UNACK 持续时间 主题。
6. 在**最大记录数**框中，输入在一个实例上可以从控件中查看的记录数。最大记录数的有效范围是从 1 到 1000。
7. 选择**应用**。

查询时区

“查询时区”选项有**原始时间**与 **UTC**。

- 原始时间指操作员所在时区的本地时间。
- UTC 时间是“格林尼治标准时间”，也称作“通用协调时间”或 Zulu。

如果选择“使用指定时间”，则可以在两个查询时区选项之间进行选择。如果未选择“使用指定时间”，则所选的“持续时间”预定义的时间间隔查询数据时总是使用 UTC 时区。

为了避免“夏令时”转换期间的问题，建议在“查询时区”中总是使用 UTC。如果改为使用“本地时间”，则从“夏令时”到“正常时间”的转换期间的报警记录可能丢失。

如果运行多台使用不同时区设置的计算机，并且它们都记录到相同的报警数据库，则每条记录都将获得 UTC 时间标签，加上所需的时区偏差以及夏令时调整，将该时间标签转换成相应的“原始时间”时需要这样。结果，数据库中的每一项都有两个时间标签：UTC 时间与执行记录的计算机的“原始时间”。这使检索速度更快。在表格项中，UTC 时间标识为“转换时间”；而“原始时间”标识为 "EventStamp"。

LATCHED 报警的报警持续时间和 UNACK 持续时间

本节介绍 LATCHED 状态的报警持续时间和 UNACK 持续时间的计算。报警持续时间是指活动报警和非活动报警之间的持续时间。UNACK 持续时间是指 UNACK 报警和 ACK 报警之间的持续时间。可以通过两种方式来计算 LATCHED 状态的报警持续时间和 UNACK 持续时间：

选项 1：确认报警，然后将其返回

报警状态	表示为	报警持续时间	UNACK 持续时间
UNACK	Tt	-	-
ACK	Ta	-	Tf-Tt
LATCHED	Tf	Tf-Tt	Tf-Tt

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ACKRTN	Td	Tf-Tt	-

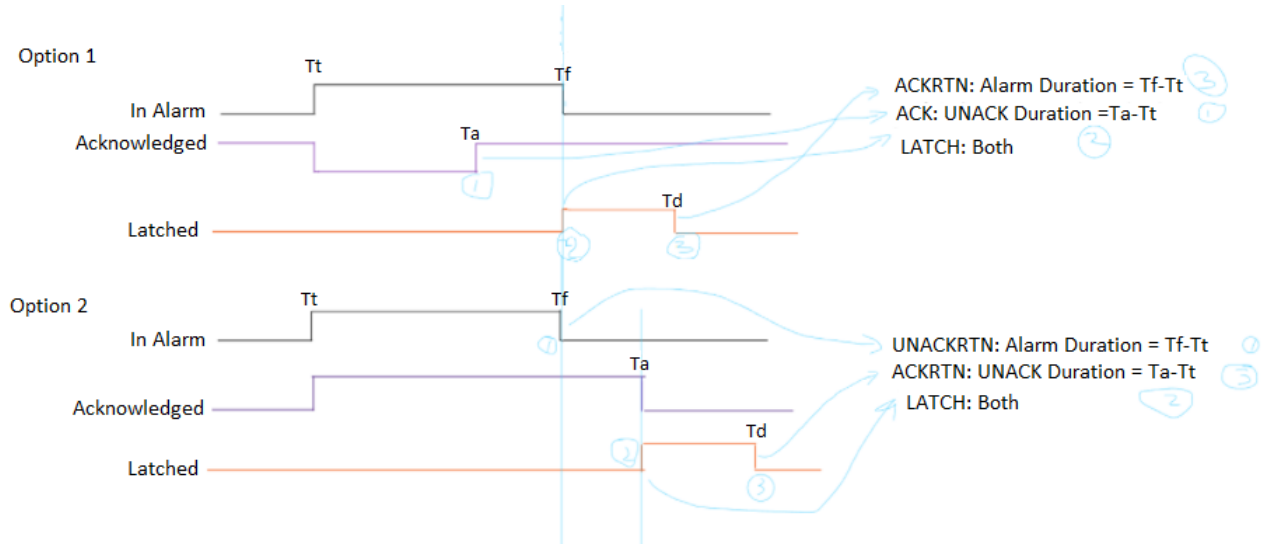
当确认报警时，UNACK 持续时间将是 ACK 和 LATCHED 状态之间的时间差，即 Ta-Tt。不会有任何报警持续时间。当报警状态更改为 LATCHED 时，报警持续时间将随着 LATCHED 和 UNACK 的时间差（即 Tf-Tt）进行更新。UNACK 持续时间保持不变。当 LATCHED 状态更改为 ACKRTN 时，报警持续时间将相同，即 Tf-Tt，并且不会有任何 UNACK 持续时间。

选项 2：返回报警，然后再确认

报警状态	表示为	报警持续时间	UNACK 持续时间
UNACK	Tt	-	-
UNACKRTN	Tf	Tf-Tt	-
LATCHED	Ta	Tf-Tt	Tf-Tt
ACKRTN	Td	-	Tf-Tt

当返回报警时，报警持续时间将是 UNACKRTN 和 UNACK 之间的时间差，即 Tf-Tt。当报警状态更改为 LATCHED 时，报警持续时间将相同。UNACK 持续时间将是 LATCHED 和 UNACK 之间的持续时间，即 Ta-Tt。当报警更改为 ACKRTN 时，UNACK 持续时间将相同，并且不会有任何报警持续时间。

下图显示了 LATCHED 报警的报警持续时间和 UNACK 持续时间选项的时序图：



创建自定义过滤器与使用过滤器收藏夹

您可以选择要在查询结果中包含哪些记录。例如，您可以按记录日期或报警状态来选择过滤器。您可以选择多个字段来限制或扩展查询结果。

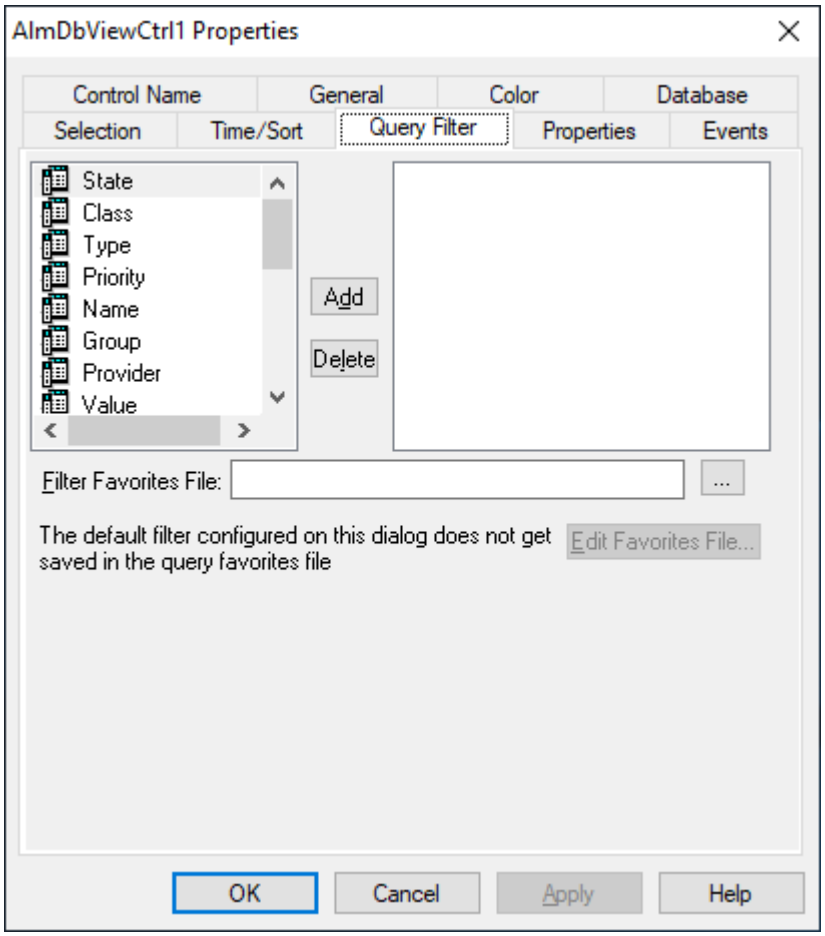
如果没有定义自定义的过滤器，则返回所有的记录。

如果已翻译语言切换字符串，便无法将这些字符串用作过滤器准则。翻译的字符串储存在报警数据库之外的地方。

创建自定义过滤器

1. 使用鼠标右键单击 **Alarm DB View** 控件，然后选择属性。此时出现 **AlmDBViewCtrl** 属性对话框。

2. 选择**查询过滤器**选项卡。



3. 在左侧窗格中，**选择过滤器**字段，然后**选择添加**，以便将它们包含到右侧窗格显示的**过滤器**中。这些**过滤器**字段在下表中介绍：

字段名	**查询过滤准则：**
状态	报警状态。 如需详细信息，请参阅状态列的值。
类	报警类。
类型	报警类型。
优先级	报警优先级。
名称	报警名。
组	报警组名。
供应器	报警供应器。
值	报警值。“值”列中的值显示为字母数字值。在“查询过滤器”中，这些值的比较是当作字符串比较来完成的。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字段名	查询过滤准则：
限制	报警限。这些值是字母数字字符。在“查询过滤器”中，这些值的比较是当作字符串比较来完成的。
操作员	操作员。
操作员全名	操作员的全名。
操作员节点	与报警关联的操作员的节点名。
操作员域	与报警关联的操作员的域名。
Comment	报警注释。
用户 1	用户自定义的报警数值 1。
用户 2	用户自定义的报警数值 2。
用户 3	用户自定义的报警字符串值。
持续时间	未确认与报警持续时间。“持续时间”列设置为零时，不会在查询中产生任何包含空值的记录。

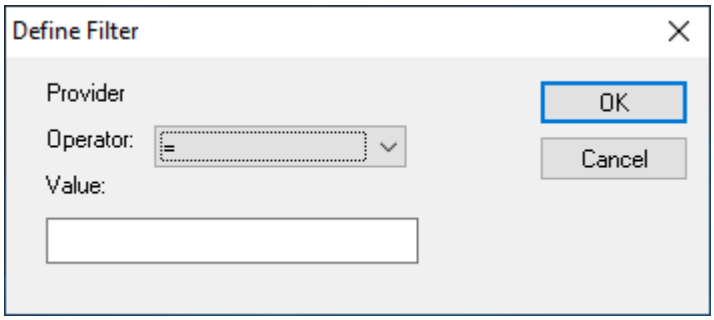
4. 要从过滤器窗格中删除某个字段，请选择要删除的字段，然后选择删除。删除过滤器的操作无法撤消。出现消息时，选择是。
5. 配置每个过滤器字段的准则。如需详细信息，请参阅定义列过滤器准则, 定义列过滤器准则。
6. 配置运算符与过滤器的组合。如需详细信息，请参阅组合多个报警列, 组合多个报警列。
7. 配置查询收藏夹文件。执行以下操作：
   a. 在过滤器收藏夹文件框中，输入网络路径与文件名，或选择省略号按钮以浏览文件。
   b. 要编辑过滤器收藏夹文件，请选择编辑收藏夹文件按钮。此时打开过滤器收藏夹窗口，可供您在收藏夹文件中添加、修改或删除过滤器。完成时选择确定，以保存更改并关闭窗口。
8. 选择应用。

定义列过滤器准则

对于查询中包含的每个列过滤器，都必须配置过滤器准则。例如，您可能只希望查看特定操作员的报警。

定义列过滤器

1. 使用鼠标右键单击字段，然后选择编辑过滤器。此时出现定义过滤器对话框。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 在运算符列表中，选择所需的运算符。
3. 在值框中，输入要匹配的过滤器准则。值框不接受所选的查询无法处理的值。给字母数字列名使用 **Like** 与 **Not Like** 过滤器运算符时，值框接受以下通配符：

字符	查找
%	包含零个或多个字符的任何字符串
_	任何单个字符
[ ]	指定的范围（如 [a-f]）或集合（如 [abcdef]）内的任何单个字符。
[^]	指定的范围（如 [^a-f]）或集合（如 [^abcdef]）之外的任何单个字符。

“值”框的以下限制适用于不同的字段：

字段	限制
所有字段	除“用户 1”、“用户 2”以及“优先级”之外，接受所有的字母数字字符。
优先级	接受从 1 到 999 的整数值。
用户 1、用户 2	仅接受负数、正数或小数。

4. 选择确定。

组合多个报警列

定义多个字段时，各个列使用布尔运算符进行合并。

- AND 运算符返回满足全部所选字段值的记录。
- OR 运算符返回满足任何所选字段值的记录。

要使用 AND/OR 运算符来设置过滤器选择准则，必须将相应的字段组合到一起。对于过滤器窗格中的某一项，只能创建一个过滤器表达式。如果需要多个表达式，则必须将该项目再次添加到过滤器窗格中。

缺省条件下，组合的字段会使用 AND 运算符。

AND 与 OR 运算符是父节点。每个父节点下的所选字段都是子节点。您无法将字段从父节点拖放到子节点上。

组合多个报警列

1. 使用鼠标右键单击字段，然后选择组合。
2. 将一个字段拖放到另一个字段上。

复制或移动查询过滤器

如果有多个 Alarm Pareto 控件的实例，并且希望为多个实例使用相同的过滤器，则可以将定义的过滤器从一个实例复制或剪切到另一个过滤器中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

复制过滤器

1. 在 Alarm Pareto 控件的第一个实例中定义过滤器。
2. 使用鼠标右键单击这些过滤器，然后选择复制。要移动这些过滤器，请选择剪切。
3. 关闭 Alarm Pareto 控件的第一个实例。
4. 打开 Alarm Pareto 控件的下一个实例，然后选择查询过滤器选项卡。
5. 将箭头放在右侧窗格中。使用鼠标右键单击所选的过滤器，然后选择粘贴。

状态列的值

将状态列添加到过滤器查询时，可以从定义过滤器对话框的值菜单中指定值。可用的值在下表中介绍：

值	描述
已确认	查询所有的系统确认。
已确认报警	查询所有已确认的报警。
未确认报警	查询所有未确认的报警。
已确认且已恢复正常	查询已返回到正常状态且已确认的所有报警。
未确认但已恢复正常	查询已返回到正常状态但尚未确认的报警。
所有 UNACK 记录	查询未确认的所有记录。
所有 ACK 记录	查询所有已确认的记录。
所有报警记录	查询所有的报警记录。
所有 RTN 记录	查询已返回到正常状态的所有报警。

注意：扩展的摘要报警模式中的标记用于创建某个报警（该报警在确认主报警之后返回到正常状态）时，将创建两条记录。第一条记录是“已确认并返回到正常状态”记录，因为新报警已经在返回到正常状态下。第二条记录是已确认记录，它对应于确认主报警。以前对 ACK_ALM 的实现已经更改为 ACK。

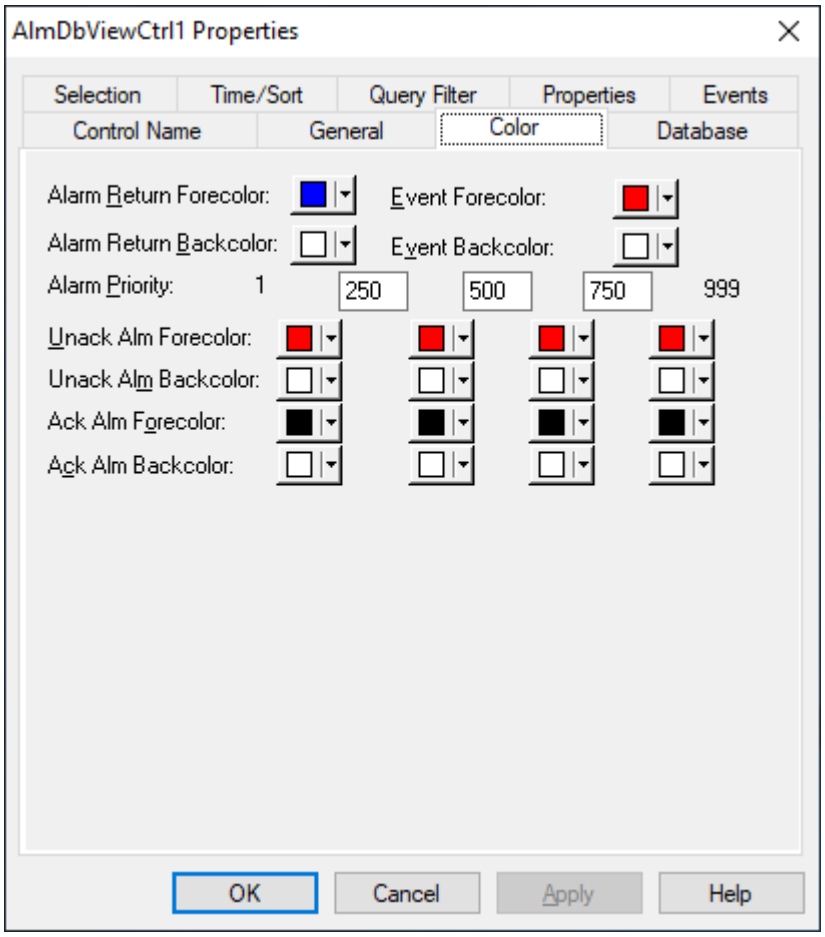
配置不同类型报警记录的颜色

您可以为报警与事件记录设置颜色。

配置报警显示颜色

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl** 属性对话框。
2. 选择颜色选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 通过选择颜色框以打开调色板，为以下每一个选项配置颜色。

选项	描述
报警返回前景色	设置返回到正常的报警记录的前景色。
报警返回背景色	设置返回到正常的报警记录的背景色。
事件前景色	设置事件记录的前景色
事件背景色	设置事件记录的背景色

4. 在“报警优先级”框中，为报警显示对象输入各个断点值。您可以指定多个断点值，这样报警便会根据“报警优先级”显示为不同的颜色。缺省的最小与最大报警优先级值分别是 1 与 999。

   例如，您将未确认报警前景色的优先级范围 250 到 499 设置为橙色，优先级范围 1 到 249 设置为红色。如果所发生的报警的优先级是 254，则它以橙色字体显示。如果所发生的报警的优先级是 12，则它以红色字体显示。

5. 通过选择颜色框以打开调色板，为以下每一个选项配置颜色。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

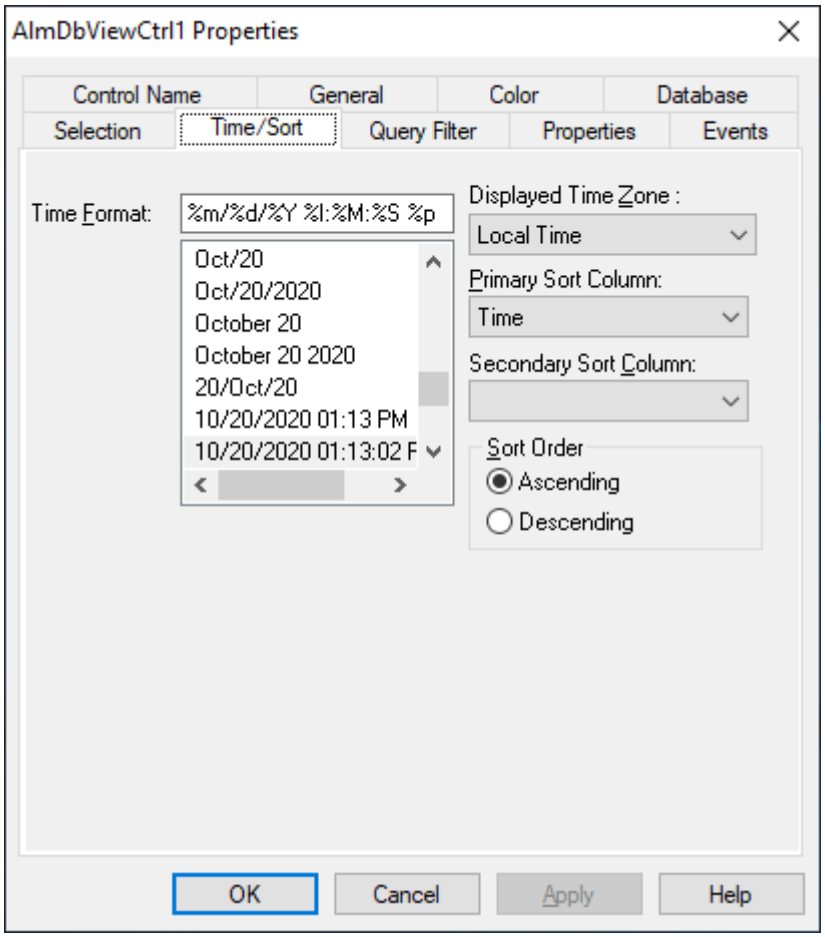
选项	描述
未确认报警前景色	给未确认的报警的每个颜色优先级范围设置前景色。
未确认报警背景色	给未确认的报警的每个颜色优先级范围设置背景色。
确认报警前景色	给已确认的报警的每个颜色优先级范围设置前景色。
确认报警背景色	给已确认的报警的每个颜色优先级范围设置背景色。

6. 选择**应**用。

配置报警记录的排序顺序

您可以控制报警记录在控件中的排序方式。

1. 使用鼠标右键单击 Alarm DB View 控件，然后单击属性。此时出现 **AlmDBViewCtrl** 属性对话框。
2. 选择**时间/排序**选项卡。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 在主排序列列表中，选择主排序列的名称。只有可见的列才出现在排序列列表中。如果没有看到所需的列，请选择常规选项卡，然后从“列明细”中选择所需的列。
4. 在第二位排序列列表中，选择第二位排序列的名称。
5. 选择升序或降序作为排序方向。
6. 选择确定。

在运行时使用 Alarm DB View 控件

根据控件的配置，您可以：

- 检索与刷新数据。
- 给数据排序。
- 过滤数据。
- 选择行。
- 通过拖动列来更改它们的顺序。
- 将所有的列还原为设计时保存的设置。

为记录排序

您可以给显示对象中的记录排序。选择标题对所有的行进行排序。

使用鼠标右键单击控件，然后选择排序以打开第二位排序对话 框，从中可以按升序或降序对单个及多个列进行排序。

要指定要排序的列，请选中列名旁边的复选框。使用排序顺序方向键对列重新排序。

如果多个报警事件具有相同的时间标签，则它们可能不会按期望的顺序出现。

例如，如果所需的排序方式是先按照报警状态的降序进行排序，则需要：

1. 选中日期与状态复选框。
2. 选择状态行。
3. 选择向上排序顺序箭头。
4. 在排序类型区域中，选择降序。
5. 选择确定。

理解状态栏信息

状态栏显示控件的当前状态。

- 左侧框架显示服务器名与连接的数据库。
- 中间框架显示查询返回的总记录数中显示了多少条记录。
- 右侧框架显示与服务器的连接状态。

使用 Alarm DB View ActiveX 属性

您可以使用脚本直接设置 Alarm DB View 控件属性的值，或者将它指定给一个 InTouch 标记或 I/O 引用。如需有关设置属性的详细信息，请参阅编写 ActiveX 控件脚本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如需有关设置颜色值的详细信息，请参阅配置 ActiveX 控件的颜色。

AckAlmBackColor 属性

获取或设置已确认报警的背景色。此设置会覆盖已确认报警的各个单独范围的颜色设置（AckAlmBackColorRange1 到 AckAlmBackColorRange4）。

类型

整型

缺省值

白色

语法

```
Object.AckAlmBackColor [= color]
```

值

color
确定背景色的值或常数。

AckAlmBackColorRange1 属性

获取或设置已确认报警的背景色。此颜色应用于控件中显示的特定记录，其状态为 ACK_ALM，且优先级在 1 到 ColorPriorityRange1 范围之间。

类型

整型

缺省值

白色

语法

```
Object.AckAlmBackColorRange1 [= color]
```

值

color
确定背景色的值或常数。

AckAlmBackColorRange2 属性

获取或设置已确认报警的背景色。此颜色应用于控件中显示的特定记录，其状态为 ACK_ALM，且优先级在 ColorPriorityRange1 到 ColorPriorityRange2 范围之间。

类型

整型

缺省值

白色

语法

```
Object.AckAlmBackColorRange2 [= color]
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

值

color
确定背景色的值或常数。

AckAlmBackColorRange3 属性

获取或设置已确认报警的背景色。此颜色应用于控件中显示的特定记录，其状态为 ACK_ALM，且优先级在 ColorPriorityRange2 到 ColorPriorityRange3 范围之间。

类型

整型

缺省值

白色

语法

```
Object.AckAlmBackColorRange3 [= color]
```

值

color
确定背景色的值或常数。

AckAlmBackColorRange4 属性

获取或设置已确认报警的背景色。此颜色应用于控件中显示的特定记录，其状态为 ACK_ALM，且优先级在 ColorPriorityRange3 到 999 范围之间。

类型

整型

缺省值

白色

语法

```
Object.AckAlmBackColorRange4 [= color]
```

值

color
确定背景色的值或常数。

AckAlmForeColor 属性

获取或设置已确认报警的前景色。此设置会覆盖已确认报警的各个单独范围的颜色设置（AckAlmForeColorRange1 到 AckAlmForeColorRange4）。

类型

整型

缺省值

黑色

语法

```
Object.AckAlmForeColor [= color]
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

值

color
确定文本颜色的值或常数。

AckAlmForeColorRange1 属性

获取或设置已确认报警的前景色。此颜色应用于控件中显示的特定记录，其状态为 ACK_ALM，且优先级在 1 到 ColorPriorityRange1 范围之间。

类型

整型

缺省值

黑色

语法

```
Object.AckAlmForeColorRange1 [= color]
```

值

color
确定文本颜色的值或常数。

AckAlmForeColorRange2 属性

获取或设置已确认报警的前景色。此颜色应用于控件中显示的特定记录，其状态为 ACK_ALM，且优先级在 ColorPriorityRange1 到 ColorPriorityRange2 范围之间。

类型

整型

缺省值

黑色

语法

```
Object.AckAlmForeColorRange2 [= color]
```

值

color
确定文本颜色的值或常数。

AckAlmForeColorRange3 属性

获取或设置已确认报警的前景色。此颜色应用于控件中显示的特定记录，其状态为 ACK_ALM，且优先级在 ColorPriorityRange2 到 ColorPriorityRange3 范围之间。

类型

整型

缺省值

黑色

语法

```
Object.AckAlmForeColorRange3 [= color]
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

值

color
确定文本颜色的值或常数。

AckAlmForeColorRange4 属性

获取或设置已确认报警的前景色。此颜色应用于控件中显示的特定记录，其状态为 ACK_ALM，且优先级在 ColorPriorityRange3 到 999 范围之间。

类型

整型

缺省值

黑色

语法

```
Object.AckAlmForeColorRange4 [= color]
```

值

color
确定文本颜色的值或常数。

AckRtnBackColor 属性

获取或设置返回到正常的已确认报警 (ACK_RTN) 的背景色。

类型

整型

缺省值

白色

语法

```
Object.AckRtnBackColor [= color]
```

值

color
确定背景色的值或常数。

AckRtnForeColor 属性

获取或设置返回到正常的已确认报警 (ACK_RTN) 的文本颜色。

类型

整型

缺省值

蓝色

语法

```
Object.AckRtnForeColor [= color]
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

值

color
确定文本颜色的值或常数。

AlmRtnBackColor 属性

获取或设置返回到正常的报警的背景色。此颜色应用于显示的状态为 ALM_RTN 的记录。

类型

整型

缺省值

白色

语法

```
Object.AlmRtnBackColor [= color]
```

值

color
确定背景色的值或常数。

AlmRtnForeColor 属性

获取或设置已返回的报警的前景色。此颜色应用于显示的状态为 ALM_RTN 的记录。

类型

整型

缺省值

蓝色

语法

```
Object.AlmRtnForeColor [= color]
```

值

color
确定文本颜色的值或常数。

身份验证属性

返回在“数据库”选项卡中所选的“身份验证类型”。

类型

消息

语法

```
AType = AlarmDBCtrl1.Authentication;
```

值

AType
消息值；为“SQL 身份验证”或“Windows 身份验证”。

AuthenticationMode 属性

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

设置身份验证模式。

类型

整型

语法

```
Object.AuthenticationMode = [Int]
```

值

Int
整数值 0 或 1。0 表示 SQL 身份验证，1 表示 Windows 身份验证。

AutoConnect 属性

获取或设置一个值，确定控件在运行时是否自动连接到数据库。

数据类型

整型

缺省值

False

语法

```
Object.AutoConnect [= Integer]
```

值

Integer
整型表达式，指定控件在运行时是否连接到数据库。
True = 连接到数据库。
False =（缺省值）不连接到数据库。

附注

您必须明确调用 Connect() 方法才能连接到数据库。

ColorPriorityRange1 属性

设置要显示的报警的优先级范围边界。此属性的值必须大于一且小于 ColorPriorityRange2 的值。

类型

整型

缺省值

250

语法

```
Object.ColorPriorityRange1 [= integer or priority]
```

ColorPriorityRange2 属性

设置要显示的报警的优先级范围边界。此属性的值必须大于 ColorPriorityRange1 的值且小于 ColorPriorityRange3 的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类型

整型

缺省值

500

语法

```
Object.ColorPriorityRange2 [= integer or priority]
```

ColorPriorityRange3 属性

设置要显示的报警的优先级范围边界。此属性的值必须大于 ColorPriorityRange2 的值且小于 999。

类型

整型

缺省值

750

语法

```
Object.ColorPriorityRange3 [= integer or priority]
```

ColumnResize 属性

返回或设置一个值，确定是否可以调整列宽。

类型

离散型

缺省值

True

语法

```
Object.ColumnResize [= Discrete]
```

值

Discrete
True = （缺省值）列宽在运行时可以调整。
False = 列宽无法调整。

ConnectStatus 属性

返回连接的状态。此属性为只读。

数据类型

消息

语法

```
Object.ConnectStatus
```

值

已连接 = 控件已连接到数据库。
未连接 = 控件未连接到数据库。
进行中 = 控件正在连接到数据库。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

控件名为 AlmDbView1，Tagname 是一个消息标记。

```
Tagname = #AlmDbView1.ConnectStatus;
```

CustomMessage 属性

获取或设置报警数据库中检索不到报警记录时 Alarm DB View 控件显示的消息。

类型

消息

缺省值

此视图中没有项目可显示。

语法

```
Object.CustomMessage [= string]
```

DatabaseName 属性

指定要连接的数据库。

类型

消息

语法

```
Object.DatabaseName [= text]
```

DisplayMode 属性

返回控件的显示模式，确定仅显示报警、事件，还是两者都显示。此属性为只读。

类型

字符串型

缺省值

报警与事件历史

语法

```
Object.DisplayMode
```

附注

可能的值有：
报警与事件历史
报警历史
事件历史

示例

控件名为 AlmDbView1，tag 是一个消息标记。

```
tag = #AlmDbView1.DisplayMode;
```

DisplayedTimeZone 属性

获取或设置显示的时区。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类型

字符串型

缺省值

本地时间

语法

```
Object.DisplayedTimeZone [= message]
```

附注

可能的值有：

GMT - 报警时间标签使用“格林尼治标准时间”。

本地时间 - 报警时间标签显示时使用 Alarm DB View 控件所在客户端的本地时间。

原始时间 - 报警时间标签显示时使用报警供应器的本地时间。

Duration 属性

获取或设置用于设置开始与结束时间的持续时间。

类型

消息

缺省值

"最近 1 小时"

语法

```
Object.Duration [= text]
```

值

text
包含持续时间的字符串表达式。此属性必须包含以下字符串之一：
最近 1 分钟
最近 5 分钟
最近 15 分钟
最近 30 分钟
最近 1 小时
最近 2 小时
最近 4 小时
最近 8 小时
最近 12 小时
最近 1 天
最近 2 天
最近 3 天
上星期
最近 2 星期
最近 30 天
最近 90 天

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

EndTime 属性

返回或设置结束时间。

类型

消息

语法

```
Object.EndTime [= text]
```

值

text
为结束时间赋值的字符串表达式。返回的字符串总是采用 (YYYY/MM/DD HH:MM:SS) 格式。设置字符串值时也要求使用相同的格式。此属性可以处理任何时区中自 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 的日期。

EventBackColor 属性

获取或设置事件报警的背景色。此颜色应用于控件中显示的状态为 EVT_EVT 的记录。

类型

整型

缺省值

白色

语法

```
Object.EventBackColor [= color]
```

值

color
确定背景色的值或常数。

EventForeColor 属性

获取或设置事件报警的前景色。此颜色应用于控件中显示的状态为 EVT_EVT 的记录。

类型

整型

缺省值

红色

语法

```
Object.EventForeColor [= color]
```

值

color
确定文本颜色的值或常数。

FilterFavoritesFile 属性

获取或设置过滤器收藏夹文件。此文件由**过滤器收藏夹**对话框用于读取或写入过滤器收藏夹。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类型

字符串型

缺省值

Null

语法

```
Object.FilterFavoritesFile [= String]
```

FilterMenu 属性

获取或设置一个值，确定是否在快捷菜单中显示过滤器菜单项。

类型

离散型

缺省值

True

语法

```
Object.FilterMenu [= Discrete]
```

值

True = 显示过滤器菜单项（缺省值）。

False = 不显示过滤器菜单项。

FilterName 属性

返回当前过滤器的名称（如果有）。

类型

字符串（只读）

缺省值

Null

语法

```
Object.FilterName [= String]
```

FromPriority 属性

获取或设置控件的“从优先级”值。

类型

整型

缺省值

1

语法

```
Object.FromPriority [= integer]
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

您可以使用此属性来过滤要显示哪些报警记录。例如，如果将此属性设置为 760，则仅显示优先级范围在 760 到 ToPriority 属性值之间的报警。

GroupExactMatch 属性

GroupExactMatch 属性为真时，仅显示报警组名与 GroupName 属性值完全匹配的报警。当它为假时，组名只需指定要过滤的报警组名的一部分。

类型

离散型

缺省值

False

语法

```
Object.GroupExactMatch [= discrete]
```

附注

将此属性与 GroupName 属性配合使用来过滤 Alarm DB View 控件。

示例

例如：

```
#AlarmDBViewCtrl1.GroupName = "Group"
#AlarmDBViewCtrl1.GroupExactMatch = 0;
#AlarmDBViewCtrl1.Refresh();
```

GroupName 属性

获取或设置当前 Alarm DB View 控件报警组名过滤器。

类型

字符串型

缺省值

（无）

语法

```
Object.GroupName [= GroupName]
```

附注

将此属性设置为“GroupA”，重新查询时，显示对象中仅出现属于 GroupA 报警组的标记。

MaxRecords 属性

返回或设置要检索的最大记录数。

类型

整型

缺省值

100

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
Object.MaxRecords [=integer]
```

值

integer
整型表达式，指定要在给定的时间检索的记录数。最大记录数可以在从 1 到 1000 的范围内。为获取最佳性能，应根据需要尽可能将此值设置得小一些。

Password 属性

返回或设置检索数据的 SQL Server 口令。

类型

消息

语法

```
Object.Password [= text]
```

值

text
为口令赋值的字符串表达式。

PrimarySort 属性

获取或设置用于给报警显示对象排序的主列名。

类型

消息

缺省值

（无）

语法

```
Object.PrimarySort [= message]
```

ProviderExactMatch 属性

ProviderExactMatch 属性为真时，仅显示报警供应器名与 ProviderName 属性值完全匹配的的报警。当它为假时，供应器名只需指定要过滤的报警供应器名的一部分。

类型

离散型

缺省值

False

语法

```
Object.ProviderExactMatch [= discrete]
```

附注

将此属性与 ProviderName 属性配合使用来过滤 Alarm DB View 控件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

例如：

```
#AlarmDBViewCtrl1.ProviderName = "Provider"
#AlarmDBViewCtrl1.ProviderExactMatch = 0;
#AlarmDBViewCtrl1.Refresh();
```

ProviderName 属性

获取或设置当前 Alarm DB View 控件的报警供应器名过滤器。

类型

字符串型

缺省值

（无）

语法

```
Object.ProviderName [= ProviderName]
```

附注

如果某个标记属于 Provider1 报警供应器，则将此属性设置为“Provider1”并重新查询时，显示对象中仅出现属于 Provider1 报警供应器的标记。

QueryTimeZoneName 属性

获取或设置给查询使用特定时间时的时区。

类型

离散型

缺省值

False

语法

```
Object.QueryTimeZone [= Discrete]
```

值

True = GMT

False = 原始时间，即报警供应器的本地时间。

RefreshMenu 属性

获取或设置一个值，确定是否在快捷菜单中显示刷新菜单项。

类型

离散型

缺省值

True

语法

```
Object.RefreshMenu [= Discrete]
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

值

True = 显示刷新菜单项（缺省值）。

False = 不显示刷新菜单项。

ResetMenu 属性

获取或设置一个值，确定是否在快捷菜单中显示重置菜单项。

类型

离散型

缺省值

True

语法

```
Object.ResetMenu [= Discrete]
```

值

True = 显示重置菜单项（缺省值）。

False = 不显示重置菜单项。

RowCount 属性

返回控件中显示的记录数。此属性为只读。

类型

整型

语法

```
Object.RowCount
```

示例

控件名为 AlmDbView1，tagname 是一个整型标记。

```
tagname = #AlmDbView1.RowCount;
```

RowSelection 属性

返回或设置一个值，确定在运行时是否可以选择行。

类型

离散型

缺省值

True

语法

```
Object.RowSelection [= Discrete]
```

值

离散型

True =（缺省值）允许选择行。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

False = 不允许选择行。

附注

如果不允许选择行，则不会生成 Click 或 DoubleClick 事件。

SecondarySort 属性

获取或设置用于给报警显示对象排序的辅助列名。

类型

消息

缺省值

（无）

语法

```
Object.SecondarySort [= text]
```

ServerName 属性

返回或设置控件检索数据时连接的服务器的名称。

类型

消息

语法

```
Object.ServerName [= text]
```

ShowFetch 属性

返回或设置一个值，确定是否显示检索按钮。

类型

离散型

缺省值

True

语法

```
Object.ShowFetch [= Discrete]
```

值

离散型
True =（缺省值）显示检索按钮。
False = 不显示检索按钮。

ShowGrid 属性

返回或设置一个值，确定是否显示网格线。

类型

离散型

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

缺省值

False

语法

```
Object.ShowGrid [= Discrete]
```

值

离散型
True = 显示网格线。
False =（缺省值）不显示网格线。

ShowHeading 属性

返回或设置一个值，确定是否显示列标题。

类型

离散型

缺省值

True

语法

```
Object.ShowHeading [= Discrete]
```

值

Discrete
True =（缺省值）显示列标题。
False = 不显示列标题。

ShowMessage 属性

确定报警数据库中没有记录时是否显示自定义消息“此视图中没有项目可显示”。

类型

离散型

缺省值

False

语法

```
Object.ShowMessage [= discrete]
```

ShowStatusBar 属性

返回或设置一个值，确定是否显示状态栏。

类型

离散型

缺省值

True

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
Object.ShowStatusBar [= Discrete]
```

值

Discrete
True = （缺省值） 显示状态栏。
False = 不显示状态栏。

SilentMode 属性

获取或设置一个值，确定控件是否处于“无提示”模式。

类型

离散型

缺省值

False

语法

```
Object.SilentMode [= Discrete]
```

值

True = 无提示模式打开。

False = 无提示模式关闭 （缺省值）。

SortMenu 属性

返回或设置一个值，确定是否在快捷菜单中显示排序菜单项。

类型

离散

缺省值

True

语法

```
Object.SortMenu [= Discrete]
```

值

离散表达式。

True = （缺省值） 显示排序菜单项。

False = 不显示排序菜单项。

SortOrder 属性

获取或设置报警的排序顺序，这取决于要排序的列（主排序列）。

类型

离散型

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

缺省值

True

语法

```
Object.SortOrder [= discrete]
```

值

离散表达式。

True = 升序。

False = 降序。

SpecificTime 属性

返回或设置一个值，确定控件是使用 StartTime 与 EndTime 属性，还是根据 Duration 属性的值来计算开始时间与结束时间。

类型

离散

缺省值

False

语法

```
Object.SpecificTime [= Discrete]
```

值

True = 使用 StartTime 与 EndTime 属性。

False =（缺省值）根据“Duration”属性计算 StartTime 与 EndTime。

StartTime 属性

返回或设置开始时间。

类型

消息

语法

```
Object.StartTime [= text]
```

值

text
为 StartTime 赋值的字符串表达式。返回的字符串总是采用 (YYYY/MM/DD HH:MM:SS) 格式。设置字符串值时也要求使用相同的格式。此属性可以处理任何时区中自 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 的日期。

Time 属性

获取与设置要在显示对象中使用的时间格式。

类型

消息

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

缺省值

%m/%d/%Y %I:%M:%S %p

语法

```
Object.Time [= message]
```

附注

如需有关时间格式字符串的详细信息，请参阅配置为报警记录显示的时间格式与时区。

ToPriority 属性

获取或设置控件的“到优先级”值。

类型

整型

缺省值

999

语法

```
Object.ToPriority [= integer]
```

附注

使用此属性过滤要显示哪些报警记录。例如，如果将此属性设置为 900，则仅显示优先级范围从 FromPriority 属性值到 900 之间的报警。

TotalRowCount 属性

返回当前查询的总记录数。此属性为只读。

类型

整型

语法

```
Object.TotalRowCount
```

附注

行计数指当前查询中返回的行数，通常与 MaxRecords 属性相同，除非检索到的记录数小于 MaxRecords 属性。例如，如果符合特定准则的记录有 950 条，而 MaxRecords 属性为 100，则最后一页有 50 条记录，行数为 50。在相同的示例中，TotalRowCount 属性始终是 950。

示例

控件名为 AlmDbView1，tagname 是一个整型标记。

```
tagname = #AlmDbView1.TotalRowCount;
```

UnAckAlmBackColor 属性

获取或设置未确认的报警的背景色。此颜色应用于控件中显示的状态为 UNACK_ALM 的所有记录。它会覆盖 UnAckAlmBackColorRange1 到 UnAckAlmBackColorRange4 属性值所作的任何设置。

类型

整型

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

缺省值

白色

语法

```
Object.UnAckAlmBackColor [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckAlmBackColorRange1 属性

获取或设置未确认的报警的背景色。此颜色应用于控件中显示的特定记录，其状态为 UNACK_ALM，且优先级在 1 到 ColorPriorityRange1 范围之间。

类型

整型

缺省值

白色

语法

```
Object.UnAckAlmBackColorRange1 [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckAlmBackColorRange2 属性

获取或设置未确认的报警的背景色。此颜色应用于控件中显示的特定记录，其状态为 UNACK_ALM，且优先级在 ColorPriorityRange1 到 ColorPriorityRange2 范围之间。

类型

整型

缺省值

白色

语法

```
Object.UnAckAlmBackColorRange2 [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckAlmBackColorRange3 属性

获取或设置未确认的报警的背景色。此颜色应用于控件中显示的特定记录，其状态为 UNACK_ALM，且优先级在 ColorPriorityRange2 到 ColorPriorityRange3 范围之间。

类型

整型

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

缺省值

白色

语法

```
Object.UnAckAlmBackColorRange3 [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckAlmBackColorRange4 属性

获取或设置未确认的报警的背景色。此颜色应用于控件中显示的特定记录，其状态为 UNACK_ALM，且优先级在 ColorPriorityRange3 到 999 范围之间。

类型

整型

缺省值

白色

语法

```
Object.UnAckAlmBackColorRange4 [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckAlmForeColor 属性

获取或设置未确认的报警的文本颜色。此颜色应用于控件中显示的状态为 UNACK_ALM 的所有记录。它会覆盖 UnAckAlmForeColorRange1 到 UnAckAlmForeColorRange4 属性值所作的任何设置。

类型

整型

缺省值

红色

语法

```
Object.UnAckAlmBackColor [= color]
```

值

color
确定文本颜色的值或常数。

UnAckAlmForeColorRange1 属性

获取或设置未确认的报警的前景色。此颜色应用于控件中显示的特定记录，其状态为 UNACK_ALM，且优先级在 1 到 ColorPriorityRange1 范围之间。

类型

整型

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

缺省值

红色

语法

```
Object.UnAckAlmForeColorRange1 [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckAlmForeColorRange2 属性

获取或设置未确认的报警的前景色。此颜色应用于控件中显示的特定记录，其状态为 UNACK_ALM，且优先级在 ColorPriorityRange1 到 ColorPriorityRange2 范围之间。

类型

整型

缺省值

红色

语法

```
Object.UnAckAlmForeColorRange2 [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckAlmForeColorRange3 属性

获取或设置未确认的报警的前景色。此颜色应用于控件中显示的特定记录，其状态为 UNACK_ALM，且优先级在 ColorPriorityRange2 到 ColorPriorityRange3 范围之间。

类型

整型

缺省值

红色

语法

```
Object.UnAckAlmForeColorRange3 [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckAlmForeColorRange4 属性

获取或设置未确认的报警的前景色。此颜色应用于控件中显示的特定记录，其状态为 UNACK_ALM，且优先级在 ColorPriorityRange3 到 999 范围之间。

类型

整型

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

缺省值

红色

语法

```
Object.UnAckAlmForeColorRange4 [= color]
```

值

color
确定特定对象颜色的值或常数。

UnAckOrAlarmDuration 属性

持续时间列显示“未确认持续时间”或“报警持续时间”。如果为 FALSE (0)，则显示“未确认持续时间”；如果为 TRUE (1)，则显示“报警持续时间”。

类型

整型

缺省值

False

语法

```
Object.UnAckOrAlarmDuration [= integer]
```

UserID 属性

返回或设置用于连接到 SQL Server 以检索数据的用户 ID。

类型

消息

语法

```
Object.UserID [= text]
```

值

text
为用户 ID 赋值的字符串表达式。

使用 Alarm DB View ActiveX 方法

使用 Alarm DB View ActiveX 方法可以：

- 控制数据库连接。
- 从报警数据库中检索记录。
- 检索报警的有关信息。
- 重置显示外观。
- 给报警记录排序。
- 显示上下文菜单。
- 访问过滤器收藏夹。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如需有关调用方法的详细信息，请参阅编写 ActiveX 控件脚本。

控制报警数据库连接

使用 Connect() 方法连接到报警数据库，使用 Disconnect() 方法断开连接。

Connect()

将控件连接到数据库；如果连接成功，则显示范围从 1 到 MaxRecords 之间的一组记录。

语法

```
Object.Connect()
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.Connect();
```

Disconnect()

断开控件与数据库的连接。

语法

```
Object.Disconnect()
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.Disconnect();
```

从数据库检索报警记录

使用以下方法选择查询、从数据库中检索记录以及刷新显示对象：

- SelectQuery()
- GetPrevious()
- GetNext()
- Refresh()

SelectQuery()

将当前显示对象设置为 .xml 文件中指定的查询名。

语法

```
Object.SelectQuery("QueryName");
```

参数

QueryName
过滤器收藏夹文件中定义的查询的名称。

示例

本例应用"HighPriority"查询定义的过滤器准则，此查询保存在当前与 AlmDbView1 控件关联的过滤器收藏夹文件中。

```
#AlmDbView1.SelectQuery("HighPriority");
```

GetPrevious()

从数据库中检索上一组记录（如果有）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
Object.GetPrevious();
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.GetPrevious();
```

GetNext()

从数据库中检索下一组记录（如果有）。

语法

```
Object.GetNext()
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.GetNext();
```

Refresh()

使用数据库中的数据来刷新控件；如果连接成功，则显示范围从 1 到 MaxRecords 之间一组记录。

语法

```
Object.Refresh
```

附注

通过调用 Alarm DB View 控件的 Refresh() 方法启动刷新操作之后，RowCount 与 TotalRowCount 属性的值更改为 -1，直至完成刷新（也就是，从数据库中检索到所有相关的记录之后）。完成刷新时，这两个属性都使用正确的当前行数进行更新。

Refresh() 方法采用异步工作方式 - 它立即将控制权返回给发出调用得脚本，然后在后台继续工作。这表示，在调用 Refresh() 方法之后查询 RowCount 与 TotalRowCount 的值时，很可能会返回 -1；原因在于，这是在尚未完成刷新的情况下查询它们的值。获取正确值的一种方法是：使用脚本确定属性值何时从 -1 更改为其它值，这样就可以知道现在有正确的值提供。

示例

控件名为 AlmDbView1。

```
#AlmDbView1.Refresh();
```

检索报警的有关信息

使用以下方法从数据库中检索与特定报警有关的记录：

- GetItem() 方法, GetItem() 方法
- GetSelectedItem() 方法

GetItem() 方法

将指定的行与列的数据作为字符串来返回。

语法

```
Object.GetItem(Integer, Message)
```

参数

整型

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

整型表达式，为控件中特定的行赋值。

消息
字符串表达式，为控件中的列名赋值。

示例

控件名为 AlmDbView1，tag 定义为消息标记。

```
tag = #AlmDbView1.GetItem(1, "Group");
```

GetSelectedItem() 方法

将所选行、给定列的数据作为字符串来返回。

语法

```
Object.GetSelectedItem(Message)
```

参数

Message
为控件中列名赋值的字符串表达式。

示例

控件名为 AlmDbView1，tag 定义为消息标记。

```
tag = #AlmDbView1.GetSelectedItem("State");
```

为报警记录排序

使用以下方法给报警记录排序及重置列宽的调整：

- SortOnCol() 方法
- ShowSort() 方法, ShowSort() 方法
- Reset() 方法

SortOnCol() 方法

在显示的报警记录上执行主排序。

语法

```
Object.SortOnCol(Message, Integer)
```

参数

Message
为控件中列名赋值的字符串表达式。

Integer
要使用的排序方向。0 = 升序，1 = 降序。

示例

控件名为 AlmDbView1。

```
tag = #AlmDbView1.SortOnCol("Name",1);
```

ShowSort() 方法

如果启用 SortMenu 属性，则显示第二位排序对话框。

语法

```
Object.ShowSort()
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

控件名为 AlmDbView1。

```
#AlmDbView1.ShowSort();
```

Reset() 方法

将所有的列还原为设计时保存的设置。

语法

```
Object.Reset()
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.Reset();
```

显示上下文菜单。

使用 ShowContext() 方法显示快捷菜单。

ShowContext() 方法

如果启用 RefreshMenu、ResetMenu 或 SortMenu 属性中的任何一个，则显示快捷菜单。

语法

```
Object.ShowContext()
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.ShowContext();
```

访问过滤器收藏夹。

使用 ShowFilter() 方法显示过滤器收藏夹对话框。

ShowFilter() 方法

显示过滤器收藏夹对话框。

语法

```
Object.ShowFilter()
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.ShowFilter();
```

显示其它信息

使用 AboutBox() 方法显示关于对话框。

AboutBox() 方法

显示关于对话框。

语法

```
Object.AboutBox()
```

示例

控件名为 AlarmPareto1。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
#AlarmPareto1.AboutBox();
```

处理与使用方法和属性有关的错误

使用 SilentMode 属性确定控件是否处于无提示模式。控件处于无提示模式时，不显示错误消息。要查看错误，请调用 GetLastError() 方法来返回错误消息。

GetLastError() 方法

如果 Alarm DB View 控件处于无提示模式，则返回上一条错误消息。

语法

```
Object.GetLastError()
```

示例

控件名为 AlmDbView1，Tagname 被定义为变量或字符串。

```
Tagname = #AlmDbView1.GetLastError();
```

使用 Alarm DB View ActiveX 事件触发脚本

您可以将 QuickScript 指定给 Alarm DB View 控件事件（如鼠标单击或双击）。该事件发生时，此 QuickScript 便会运行。

Alarm DB View 控件支持以下事件：

- 单击
- DoubleClick
- ShutDown
- StartUp

Click 事件有一个参数 ClicknRow，它可以确定运行时所选择的行。

DoubleClick 事件有一个参数 DoubleClicknRow，它可以确定运行时所双击的行。

Click 与 DoubleClick 事件都是零基的。发布 Click 与/或 DoubleClick 事件时，显示对象中的条形计数从 0 开始。

注意：Alarm DB View 控件从 StartUp 事件调用方法时会忽略用户界面方法，原因在于控件尚不可见。这些方法包括：ShowSort()、ShowContext()、GetSelectedItem()、GetNext()、GetPrevious() 及 AboutBox()。

如需有关编写 ActiveX 事件脚本的详细信息，请参阅编写 ActiveX 控件脚本。

分析标记间的报警分布

通过使用 Alarm Pareto ActiveX 控件，可以分析给定的生产系统中有哪些报警与事件最经常发生。您也可以按报警发生的时段来分析报警频率。

Alarm Pareto 控件的分析功能可以确定生产系统中最大的问题。Alarm Pareto 控件帮助确定应该将精力集中到哪些地方才能取得最显著的改善。

Alarm Pareto 控件显示一个代表报警活动的条形图。

如需有关 ActiveX 控件的详细信息，请参阅 ActiveX 控件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置 Alarm Pareto ActiveX 控件

配置 Alarm Pareto 控件时，可以指定：

- 与报警数据库的连接。
- Pareto 控件的外观，包括颜色与字体。
- 用户可以在运行时访问的功能。
- 图表中显示的报警及其显示方式。

配置与报警数据库的连接

您必须配置 Alarm Pareto 控件与报警数据库之间的连接。

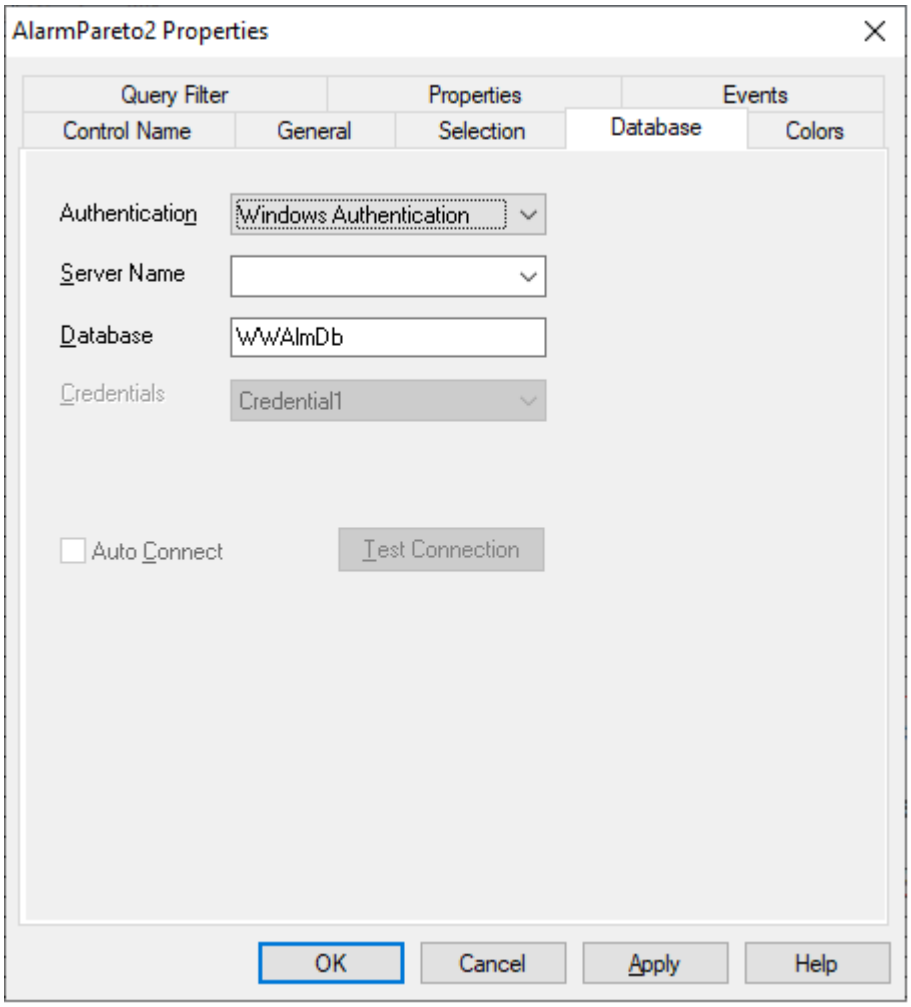
使用对报警数据库有适当的只读访问权限的帐户，而不要使用系统管理员帐户，这是一种很好的做法。

配置与报警数据库的连接

1. 使用鼠标右键单击 Alarm Pareto 控件，然后选择属性。

   此时出现 **AlarmPareto** 属性对话框。

2. 选择数据库选项卡。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 配置连接。执行以下操作：
   a. 在身份验证列表中，选择身份验证方法：**SQL Server 身份验证**或 **Windows 身份验证**（缺省）。

      注意：与 SQL 身份验证相比，Windows 身份验证可提供更好的应用程序安全性。因此，如果从 Windows 身份验证切换为 SQL 身份验证，则将显示一个对话框，建议您使用 Windows 身份验证。如果您选择忽略此警告并继续使用 SQL 身份验证，请选择确定。在 OCMC Log Viewer 中将记录一条类似的消息。

   b. 在服务器名框中，输入安装了报警数据库的计算机的节点名。
   c. 在数据库名框中，输入报警数据库的名称。
   d. 从凭据下拉列表中选择用于身份验证的凭据。

   注意：
   - 仅当选择“SQL Server 身份验证”类型时，才会启用凭据下拉列表。Windows 身份验证方法使用当前登录用户的凭证，并禁用用户名和密码字段。
   对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发》中的使用凭据管理器。

4. 如果希望 Alarm Pareto 控件在 WindowViewer 启动时立即自动连接到报警数据库，请选中自动连接复选框。

   如果未选中自动连接复选框，则必须通过明确调用 Connect() 方法将 Alarm Pareto 控件配置为连接到报警数据库。如需有关 Connect 方法的详细信息，请参阅 Connect() 方法。

5. 选择测试连接以验证与报警数据库的连接。此时出现一条消息，指出连接成功。
6. 选择应用。

配置 Alarm Pareto 控件的外观与颜色

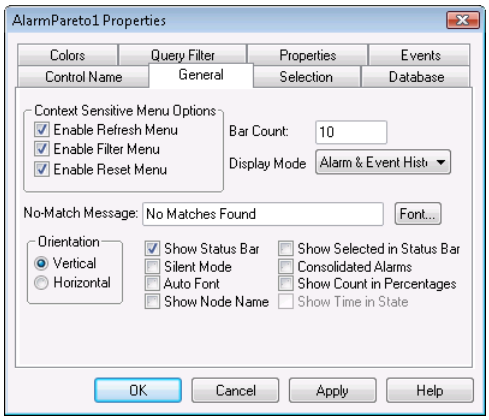
您可以配置 Alarm Pareto 控件的视觉外观。您可以：

- 包含状态栏。
- 设置 Pareto 条形图的方向。
- 包含条形图的描述。
- 选择 Alarm Pareto 图表的颜色。

设置 Alarm Pareto 控件的外观

1. 使用鼠标右键单击 Alarm Pareto 控件，然后选择属性。此时出现 **AlarmPareto 属性对话**框。
2. 选择常规选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



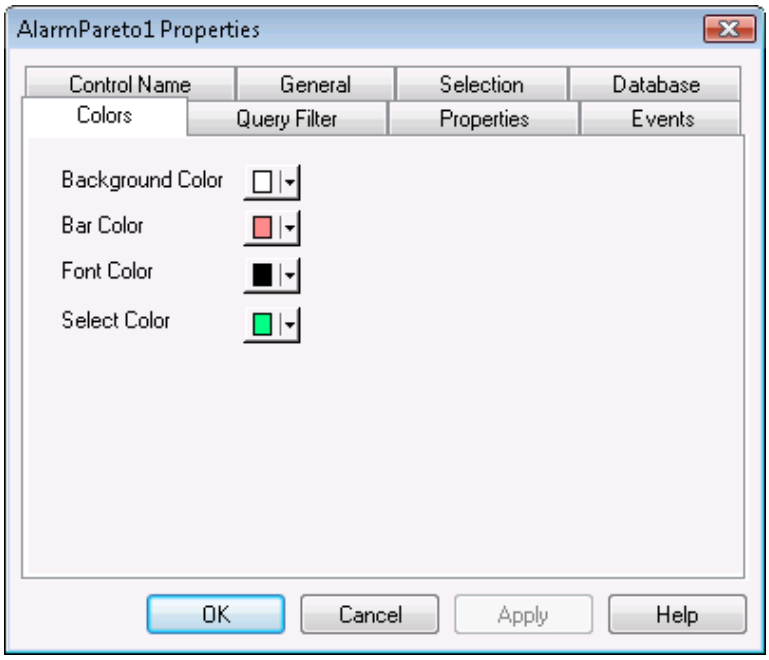
3. 配置常规选项。执行以下操作：

属性	描述
条形计数	设置要在 Alarm Pareto 控件中查看的条形数。
显示模式	此列表显示可用的视图选项。选项有“报警和事件历史”、“报警历史”以及“事件历史”。
无匹配消息	设置在 Alarm Pareto 控件中没有要处理的数据时显示的消息。
垂直	沿垂直方向显示条形。
水平	沿水平方向显示条形。
显示状态栏	启用状态栏。
无提示模式	Alarm Pareto 控件不显示运行时错误消息。如果未选择它，则报警显示对象显示错误消息。错误消息总是发送到 Logger 中。
自动调整字体	可用空间不足而无法在所选的条形上正确显示文本时，“自动调整字体”功能会隐藏文本，并且仅在选择该条形时才显示文本。
显示节点名	在条形图上显示节点名。
在状态栏中显示所选内容	在状态栏上显示所选条形的描述。
合并报警	将报警状态合并为两个状态。例如，如果某个模拟标记有三个报警状态： Hi、HiHi 及 Normal，则将 Hi 与 HiHi 报警状态 划分为一个状态。
以百分比形式显示计数	根据计数除以总数得到的百分比来显示条形。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	描述
显示特定状态的时间	根据每个标记处于报警状态的时间来显示 Alarm Pareto 控件。仅当显示模式设置为“报警历史”时，此选项才会启用。

4. 选择**应**用。
5. 选择**颜**色选项卡。



6. 选择每个颜色框以打**开调**色板。
7. 选择要给以下每个图表属性指定的颜色：

属性	描述
背景颜色	设置 Alarm Pareto 图的背景色
条形颜色	设置图表的条形颜色
字体颜色	设置图表中出现的文本的颜色。
选择颜色	设置所选条形的颜色

8. 选择**应**用。

配置字体显示

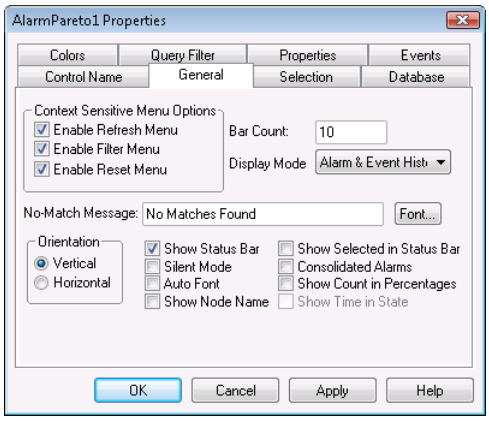
您可以给 Alarm Pareto 图中出现的文本指定字体属性。

配置字体属性

1. 使用鼠标右**键单击** Alarm Pareto 控件，然后**选择属性**。此时出现 **AlarmPareto 属性对话**框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 选择常规选项卡。



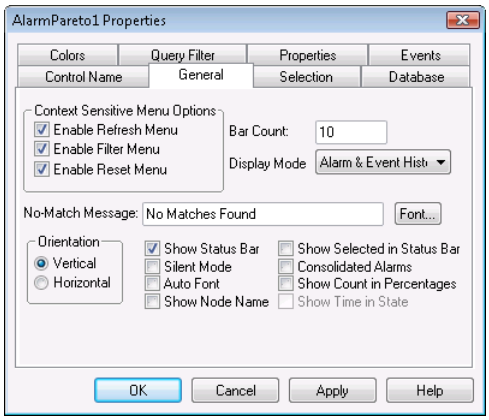
3. 选择字体。此时出现标准的 Windows 字体对话框。配置字体，然后单击确定。
4. 选择确定。

配置用户可以在运行时访问的功能

Alarm Pareto 图包含一个快捷菜单。操作员在运行时使用鼠标右键单击趋势时，会出现一个菜单，其中包含的选项可动态更新趋势中显示的数据。

配置运行时功能

1. 使用鼠标右键单击 Alarm Pareto 控件，然后选择属性。此时出现 **AlarmPareto** 属性对话框。
2. 选择常规选项卡。



3. 在上下文相关菜单选项区域中，配置菜单命令：
   a. 选中启用刷新菜单复选框，以允许运行时用户刷新 Alarm Pareto 趋势中显示的数据，并显示范围在 1 到 MaxRecords 属性所定义的数字之间的记录。
   b. 选中启用过滤器菜单复选框，以允许用户显示过滤器收藏夹对话框，用于选择包含 Alarm Pareto 趋势数据库查询值的文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

c. 选择启用复位菜单复选框，以允许用户将运行时的 Alarm Pareto 图恢复为从 WindowMaker 中指定的原始值。操作员在运行时所作的全部更改都会恢复成设计时的原始值。

4. 选择应用。

配置要分析的报警

您可以配置要使用 Alarm Pareto 图表来分析的报警。您可以指定：

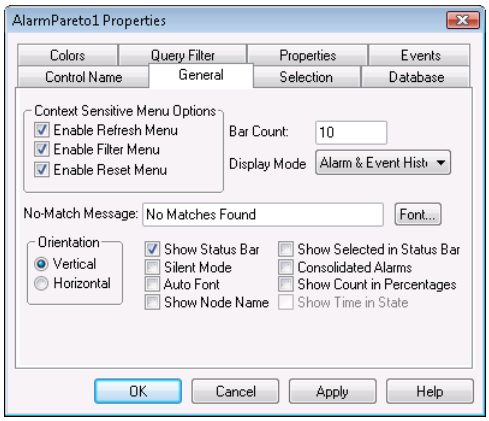
- 数据库数据的类型（报警或事件数据）
- 从中选择记录的时间段
- 过滤数据的准则

选择报警或事件数据

您可以配置在 Alarm Pareto 图中显示报警记录、事件记录，还是这两者都显示。

选择数据类型

1. 使用鼠标右键单击 **Alarm Pareto** 控件，然后单击属性。此时出现“AlarmPareto 属性”对话框。
2. 选择常规选项卡。



3. 在显示模式列表中，配置记录类型。执行以下任何操作。
   - 选择报警和事件历史，以同时显示报警与事件的历史数据库记录。
   - 选择报警历史以便仅显示历史报警记录。
   - 选择事件历史以便仅显示历史事件记录。
4. 选择应用。

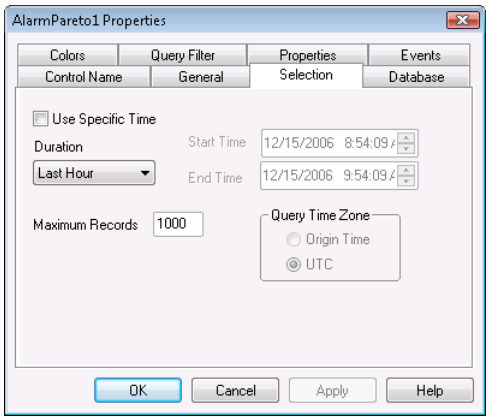
选择时段

您可以设置查询值来根据所选的时间选择记录。您也可以配置要查看的最大记录数、报警查询的开始与结束时间，以及查询时区。

选择数据的时段

1. 使用鼠标右键单击 **Alarm Pareto** 控件，然后单击属性。此时出现 **AlarmPareto** 属性对话框。
2. 选择选择选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 要使用**预**定义的**时间间**隔（**查询**数据**时总**是使用 UTC），**请选择持续时间**列表中的某个**间**隔。
4. 要使用特定的**开**始**时间**与**结**束**时间**，**请选择使用指定时间**，然后配置**详细资**料。
   a. 在**开始时间**框中，**输**入**检**索**报**警**记录**的**开**始**时间**。字符串必**须**采用 YYYY/MM/DD HH:MM:SS 格式。使用任何**时**区中自 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 之**间**的任何日期。
   b. 在**结束时间**框中，**输**入停止**检**索**报**警**记录**的**结**束**时间**。字符串必**须**采用 YYYY/MM/DD HH:MM:SS 格式。使用任何**时**区中自 1970 年 1 月 1 日午夜到 2038 年 1 月 18 日 19:14:07 之**间**的任何日期。
   c. 在**查询时**区区域中，**单击 UTC** 或**原始时间**。UTC **时间**是“格林尼治**标**准**时间**”，也称作“通用**协调时间**”或 Zulu。原始**时间**指操作**员**所在**时**区的当前**时间**。
5. 在**最大记录数**框中，**输**入在一个**实**例上可以从控件中**查**看的**记录**数。最大**记录**数的有效范**围**是从 0 到 1000000。
6. **选择应**用。

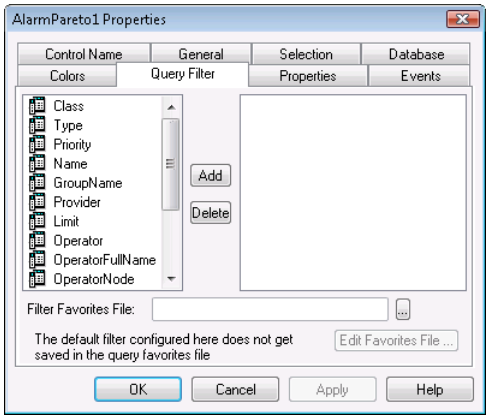
使用过滤器收藏夹创建自定义过滤器

您可以**选择**要在**查询结**果中包含哪些**记录**。例如，您可以按**记录**日期或**报**警状**态**来**选择过滤**器。您可以**选择**多个字段来限制或**扩**展**查询结**果。

如果没有定义自定义过滤器，则使用**查询**所有**记录**的缺省**过滤**器。

创建自定义过滤器

1. 使用鼠**标**右**键单击 Alarm Pareto** 控件，然后**单击属性**。此**时**出**现 AlarmPareto 属性对话**框。
2. **选择查询过滤器选项**卡。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 在左侧窗格中，选择过滤器字段，然后选择添加，以便将它们包含到右侧窗格显示的过滤器中。这些过滤器字段在下表中介绍：

字段名	查询过滤准则：
类	报警类。
类型	报警类型。
优先级	报警优先级。
名称	报警名。
GroupName	报警组名。
供应器	报警供应器。
限制	报警限。这些值是字母数字字符。 在“查询过滤器”中，这些值的比较是当作字符串比较来完成的。
操作员	操作员。
操作员全名	操作员的全名。
操作员节点	与报警关联的操作员的节点名。
操作员域	与报警关联的操作员的域名。
Comment	报警注释。
用户 1	用户自定义的报警数值 1。
用户 2	用户自定义的报警数值 2。
用户 3	用户自定义的报警字符串值。
持续时间	未确认与报警持续时间。“持续时间”列设置为零时，不会在查询中产生任何包含空值的记录。

4. 要从过滤器窗格中删除某个字段，请突出显示要删除的字段，然后单击删除。删除过滤器的操作无法撤消。出现消息时，选择是。
5. 配置每个过滤器字段的准则。如需详细信息，请参阅定义列过滤器准则, 定义列过滤器准则。
6. 配置运算符与过滤器的组合。如需详细信息，请参阅组合多个报警列, 组合多个报警列。
7. 配置过滤器收藏夹文件。
   a. 在过滤器收藏夹文件框中，输入网络路径与文件名，或选择省略号按钮以浏览文件。
   b. 要编辑过滤器收藏夹文件，请选择编辑收藏夹文件按钮。此时打开过滤器收藏夹窗口，可供您在收藏夹文件中添加、修改或删除过滤器。完成时选择确定，以保存更改并关闭窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

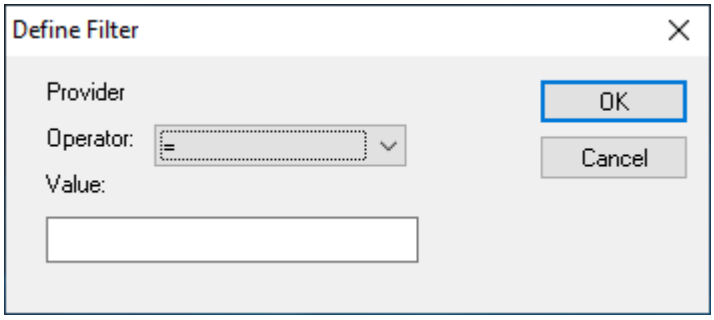
8. 选择应用。

定义列过滤器准则

对于查询中包含的每个列过滤器，都必须配置过滤器准则。例如，您可能只希望查看特定操作员的报警。

定义列过滤器

1. 使用鼠标右键单击字段，然后选择编辑过滤器。此时出现定义过滤器对话框。



2. 在运算符列表中，选择所需的运算符。
3. 在值框中，输入要匹配的过滤器准则。值框不接受所选的查询无法处理的值。给字母数字列名使用 **Like** 与 **Not Like** 过滤器运算符时，值框接受以下通配符：

字符	查找
%	包含零个或多个字符的任何字符串
_	任何单个字符
[ ]	指定的范围（如 [a-f]）或集合（如 [abcdef]）内的任何单个字符。
[^]	指定的范围（如 [^a-f]）或集合（如 [^abcdef]）之外的任何单个字符。

“值”框的以下限制适用于不同的字段：

字段	限制
所有字段	除“用户 1”、“用户 2”以及“优先级”之外，接受所有的字母数字字符。
优先级	接受从 1 到 999 的整数值。
用户 1、用户 2	仅接受负数、正数或小数。

4. 选择确定。

组合多个报警列

定义多个字段时，各个列使用布尔运算符进行合并。

- AND 运算符返回满足全部所选字段值的记录。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- OR 运算符返回满足任何所选字段值的记录。

要使用 AND/OR 运算符来设置过滤器选择准则，必须将相应的字段组合到一起。对于过滤器窗格中的某一项，只能创建一个过滤器表达式。如果需要多个表达式，则必须将该项目再次添加到过滤器窗格中。

缺省条件下，组合的字段会使用 AND 运算符。

AND 与 OR 运算符是父节点。每个父节点下的所选字段都是子节点。您无法将字段从父节点拖放到子节点上。

组合多个报警列

1. 使用鼠标右键单击字段，然后选择组合。
2. 将一个字段拖放到另一个字段上。

复制或移动查询过滤器

如果有多个 Alarm Pareto 控件的实例，并且希望为多个实例使用相同的过滤器，则可以将定义的过滤器从一个实例复制或剪切到另一个过滤器中。

复制过滤器

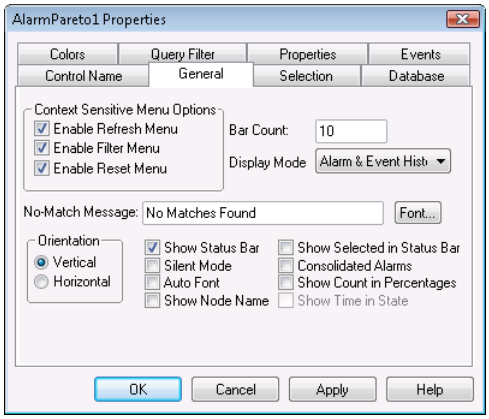
1. 在 Alarm Pareto 控件的第一个实例中定义过滤器。
2. 使用鼠标右键单击这些过滤器，然后选择复制。要移动这些过滤器，请选择剪切。
3. 关闭 Alarm Pareto 控件的第一个实例。
4. 打开 Alarm Pareto 控件的下一个实例，然后选择查询过滤器选项卡。
5. 将箭头放在右侧窗格中。使用鼠标右键单击所选的过滤器，然后选择粘贴。

配置分析结果的显示

您可以配置查询结果中报警的显示。

配置显示

1. 使用鼠标右键单击 **Alarm Pareto** 控件，然后单击属性。此时出现“AlarmPareto 属性”对话框。
2. 选择常规选项卡。



3. 选中合并报警复选框，以将多状态报警中的报警合并为单时间标签的记录。
4. 选中以百分比形式显示计数复选框，以便为图中显示的每个条形添加占总数的百分比。
5. 选择应用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在运行时使用 Alarm Pareto 控件

在运行时使用鼠标右键单击 Alarm Pareto 控件打开一个快捷菜单。下表列出该快捷菜单中出现的所有可能的选项。

菜单选项	描述
刷新	刷新显示对象。
过滤器	可用于编辑过滤器以更改 Pareto 控件接收的数据。只有在设置了过滤器收藏夹文件的情况下，此菜单项才可用。
复位	将图形复位成缺省查询。

理解状态栏上显示的信息

Alarm Pareto 控件的状态栏显示：

- 控件与报警数据库之间的数据库连接状态。
- 刷新图形中显示的数据的图形更新状态。

使用 Alarm Pareto ActiveX 属性

您可以使用脚本直接设置 Alarm Pareto 控件属性的值，或是将它指定给 InTouch 标记或 I/O 引用。如需有关设置属性的详细信息，请参阅编写 ActiveX 控件脚本。

下表列出了 Alarm Pareto 属性。

属性名	用途
身份验证	返回在“数据库”选项卡中所选的“身份验证类型”。
AuthenticationMode	设置用于确定“身份验证类型”的值（0 代表 SQL 身份验证，1 代表 Windows 身份验证）
AutoConnect	返回或设置一个值，用于确定控件处于运行时模式之后是否立即连接到数据库。
AutoFont	可用空间不足而无法在所选的条形上正确显示文本时，“自动调整字体”功能会隐藏文本，并且仅当选择该条形时才显示文本。
BackGndColor	设置 Alarm Pareto 图的背景色
BarColor	设置 Alarm Pareto 控件的条形颜色。
BarCount	设置 Alarm Pareto 控件上出现的条形数。
BarSelectColor	设置 Alarm Pareto 控件中所选条形的颜色。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性名	用途
已连接	确定 Alarm Pareto 控件是否连接到数据库。
ConsolidatedAlarms	将多个报警状态合并到两个状态。例如，如果某个模拟标记有三个报警状态：Hi、HiHi 及 Normal，则将 Hi 与 HiHi 报警状态划分为一个状态。
DatabaseName	设置 Alarm Pareto 控件要连接的数据库的名称。
DisplayMode	设置显示模式。显示模式选项包括“报警和事件历史”、“报警历史”以及“事件历史”。
Duration	返回或设置由控件用于设置“开始时间”与“结束时间”的持续时间。
EnableRefresh	启用或禁用可刷新 Alarm Pareto 控件的上下文菜单。
EnableReset	启用或禁用可重置 Alarm Pareto 控件的上下文菜单。
EnableSilentMode	启用或禁用“无提示模式”。 如果禁用“无提示模式”，则 Alarm Pareto 控件显示错误消息框。如果启用“无提示模式”，则错误消息框不会出现。 错误信息写入日志记录中。
EndTime	返回或设置结束日期与时间。
FilterMenu	启用或禁用某个上下文菜单项，供您编辑过滤器来更改 Pareto 控件接收的数据。仅当设置了“过滤器收藏夹文件”，此属性才会启用。
FilterFavoritesFile	按字符串格式指定过滤器收藏夹文件。
字体	设置控件中记录与标题的字体。
FontColor	设置 Alarm Pareto 控件中记录视图的字体颜色。
HorizontalChart	将图表显示为水平条形。 如果禁用 HorizontalChart，则图表显示为垂直条形。
MaxRecords	返回或设置一个值，指定要在给定的时间检索的最大记录数。
NoMatchMessage	设置在 Alarm Pareto 控件中没有要处理的数据时显示的消息。
QueryTimeZone	将时区设置为“UTC 时间”或“原始时间”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性名	用途
ServerName	返回当前服务器名。
ShowCountPercentage	如果选择此项，则每个条形的计数显示为占总数的百分比。如果未选择，则显示每个条形的实际计数。
ShowNodeName	将 Alarm Pareto 控件设置为除显示其它的信息之外，还在条形上显示节点名。
ShowSelectedInStatusBar	启用或禁用在状态栏上显示所选条形相关信息的功能。
ShowStatusBar	返回或设置一个值，确定是否显示状态栏。
ShowTimeinState	返回或设置一个值，确定 Alarm Pareto 控件是否根据标记处于报警状态的时间来显示条形。 如果禁用，则控件基于报警的发生次数来显示条形。
SpecificTime	返回或设置一个值，确定控件是使用“开始时间”与“结束时间”属性，还是根据“持续时间”属性的值来计算开始时间与结束时间。
StartTime	返回或设置开始日期与时间。
用户	返回或设置用作连接到 SQL Server 的控件的“用户”。

使用 Alarm Pareto ActiveX 方法

使用 Alarm Pareto 的 ActiveX 方法可以：

- 控制数据库连接。
- 从数据库检索记录。
- 检索特定 Pareto 条的有关信息。

如需有关调用方法的详细信息，请参阅编写 ActiveX 控件脚本。

控制数据库连接

使用 Connect() 方法连接到报警数据库。

Connect() 方法

连接到从 Alarm Pareto 控件属性的数据库选项卡中配置的数据库。

语法

```
Object.Connect()
```

示例

控件名为 AlarmPareto1。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
#AlarmPareto1.Connect();
```

从数据库检索记录

使用以下函数从数据库中检索记录：

- Refresh() 方法
- SelectQuery() 方法

Refresh() 方法

使用数据库中的数据来刷新控件；如果连接成功，则显示范围从 1 到 MaxRecords 属性定义的数字之间的一组记录。

语法

```
Object.Refresh()
```

示例

```
#AlarmPareto1.Refresh();
```

SelectQuery() 方法

选择配置为查询收藏夹文件的过滤器。

语法

```
Object.SelectQuery(Filter)
```

参数

Filter
查询过滤器的名称。

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.SelectQuery("MyFilter");
```

检索特定 Pareto 条的有关信息

使用以下函数检索特定 Pareto 条的有关信息：

- GetItemAlarmName() 方法
- GetItemAlarmType() 方法
- GetItemCount() 方法
- GetItemTotalTime() 方法
- GetItemEventType() 方法
- GetItemProviderName() 方法

GetItemAlarmName() 方法

获取特定条形的报警的名称。

语法

```
Object.GetItemAlarmName(BarIndex)
```

参数

BarIndex

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

条形的索引。

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.GetItemAlarmName(1);
```

GetItemAlarmType() 方法

获取特定条形的报警的类型。

语法

```
Object.GetItemAlarmType(BarIndex)
```

参数

BarIndex
条形的索引。

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.GetItemAlarmType(1);
```

GetItemCount() 方法

获取条形中的报警数量。

语法

```
Object.GetItemCount(BarIndex)
```

参数

BarIndex
条形的索引。

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.GetItemCount(1);
```

GetItemTotalTime() 方法

获取标记处于报警状态的总计时间（以秒为单位）。

语法

```
Object.GetItemTotalTime(BarIndex)
```

参数

BarIndex
条形的索引。

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.GetItemTotalTime(1);
```

GetItemEventType() 方法

获取特定条形的事件的类型。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
Object.GetItemEventType(BarIndex)
```

参数

BarIndex
条形的索引。

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.GetItemEventType(1);
```

GetItemProviderName() 方法

获取为特定的条形生成的报警的供应器名。

语法

```
Object.GetItemProviderName(BarIndex)
```

参数

BarIndex
条形的索引。

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.GetItemProviderName(1);
```

显示其它信息

使用 AboutBox() 方法显示关于对话框。

AboutBox() 方法

显示关于对话框。

语法

```
Object.AboutBox()
```

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.AboutBox();
```

使用方法与属性时的错误处理

Alarm Pareto 控件基于无提示模式选项处理运行时错误消息。如需有关详细信息，请参阅配置 Alarm Pareto 控件的外观与颜色。

如果选择无提示模式，则 Alarm Pareto 控件不显示运行时错误消息。如果未选择它，则报警显示对象显示错误消息。所有的 Alarm Pareto 错误消息都发送到 Logger 中。

使用 Alarm Pareto ActiveX 事件触发脚本

您可以将 QuickScript 指定给 Alarm Pareto 控件事件，如鼠标单击或双击。该事件发生时，此 QuickScript 便会运行。

Alarm Pareto 控件支持以下事件：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 单击
- DoubleClick
- ShutDown
- StartUp

Click 事件有一个参数 ClicknBarIndex，它可以确定在运行时所选择的条形的索引。

DoubleClick 事件有一个参数 DoubleClicknBarIndex，它可以确定在运行时所双击的条形的索引。

Click 与 DoubleClick 事件都是零基的。发布 Click 与/或 DoubleClick 事件时，显示对象中的条形计数从 0 开始。

注意：Alarm Pareto 控件从 StartUp 事件调用方法时会忽略用户界面方法，原因在于控件尚不可见。这些方法包括 AboutBox() 与 Refresh()。

如需有关编写 ActiveX 事件脚本的详细信息，请参阅编写 ActiveX 控件脚本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 11 脚本

您可以使用 InTouch 脚本语言 QuickScript 构建更强大可靠的应用程序。程序提供八种类型的脚本和许多内置的脚本函数。

脚本可以按运行时间，以及它们是否独立于其它正在进行的应用程序进程来进行分类。脚本通常可以按两种不同的方式运行：

- 基于事件的脚本在事件发生时运行一次。例如，某个基于事件的脚本可以在操作员按下某个键或某个标记值改变之后运行。
- 基于时间的脚本在满足某个条件时周期性地运行。例如，某个基于时间的脚本可以在某个窗口打开或某个按钮被按住期间运行。

您可以将多个基于事件和基于时间的脚本配置成使用相同的触发器来运行。例如，您可以将一个脚本配置成按下某个键时运行一次，将另一个脚本配置成在按住相同的键期间每隔 5 秒运行一次。

您可以在脚本语言中使用条件语句、循环以及局部变量，以便在应用程序中创建复杂的效果。

对于条件脚本，您可以让脚本同步或异步运行。

- 同步脚本运行时，所有的 InTouch 动画与标记值停止更新。在脚本停止之后，动画与标记值随后会恢复更新。
- 异步脚本运行时，所有的 InTouch 动画与标记值在脚本运行期间继续更新。

内置的脚本函数包括数学函数、三角函数、字符串函数等。使用这些函数可以帮助节省开发应用程序的时间。

InTouch 脚本可以包含“对象链接与嵌入”（Object Linking and Embedding，简称 OLE）对象与 ActiveX 控件。

基本脚本概念

在开始编写脚本之前，应该了解：

- 脚本是指示应用程序执行相关任务的一组指令。
- QuickScript 是 InTouch HMI 脚本语言。
- 函数是可以由另一个脚本调用的脚本。InTouch HMI 附带一组预定义的函数供您使用。
- QuickFunction 是在 QuickScript 中编写并在 QuickFunction 库中存储的可重复使用的函数。要创建 QuickFunction，只需创建一个 QuickScript 并给它命名即可。QuickFunction 可以由另一个脚本或是从动画链接表达式中调用。

脚本类型

在 InTouch 中，脚本按其运行原因进行分类。例如，如果希望在操作员按键盘上的某个键时执行脚本，可以创建一个“键脚本”。

在选择脚本类型之后，便可以进一步定义导致脚本执行的准则或条件。例如，您可能希望在释放键而不是按下键的时候执行脚本。

脚本类型如下：

- **应用程序**脚本在 WindowViewer 运行时连续执行，或是在 WindowViewer 启动或关闭时执行一次。
- **窗口**脚本在 InTouch 窗口打开期间定期执行，或是在打开或关闭 InTouch 窗口时执行一次。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 键脚本在按下或释放特定键或组合键时执行一次或定期执行。
- 条件脚本在满足或不满足特定的条件时执行一次或定期执行。
- 数据改变脚本在特定的标记或表达式的值发生改变时执行一次。
- 动作脚本在操作员选择 InTouch HMI 图形对象时执行一次或定期执行。动作脚本经常用于按钮。
- **ActiveX 事件**脚本在 ActiveX 事件（如选择 ActiveX 控件）发生时执行一次。

高级脚本概念

一些高级脚本功能可以实现那些基本 InTouch HMI 之外的复杂功能。

通过使用 OLE 对象与 ActiveX 控件<1>，</1>可以访问本地计算机系统函数<2>，</2>还能与其它程序<3>（</3>如<4>“</4>制造工程模块<5>”）</5>进行交互<6>。</6>

OLE 对象

在自定义的脚本中，可以调用 OLE 对象。OLE 对象允许访问本地计算机系统函数，或是与其它程序（如“制造工程模块”）进行交互。

例如，使用 OLE 可以：

- 产生随机数。
- 创建用户界面对话框。
- 打开 Windows 日期与时间属性面板。
- 读写注册表。
- 最小化窗口。

使用 ActiveX 控件编写脚本

InTouch HMI 的“向导”菜单中提供多个 ActiveX 控件。由于 InTouch HMI 基于 Windows 操作环境，因此它几乎可以使用任何 ActiveX 控件。

创建与编辑脚本

创建新脚本的步骤根据脚本类型的不同而各异。一般而言，需要打开脚本编辑器，选择条件类型，输入语句，然后保存脚本。

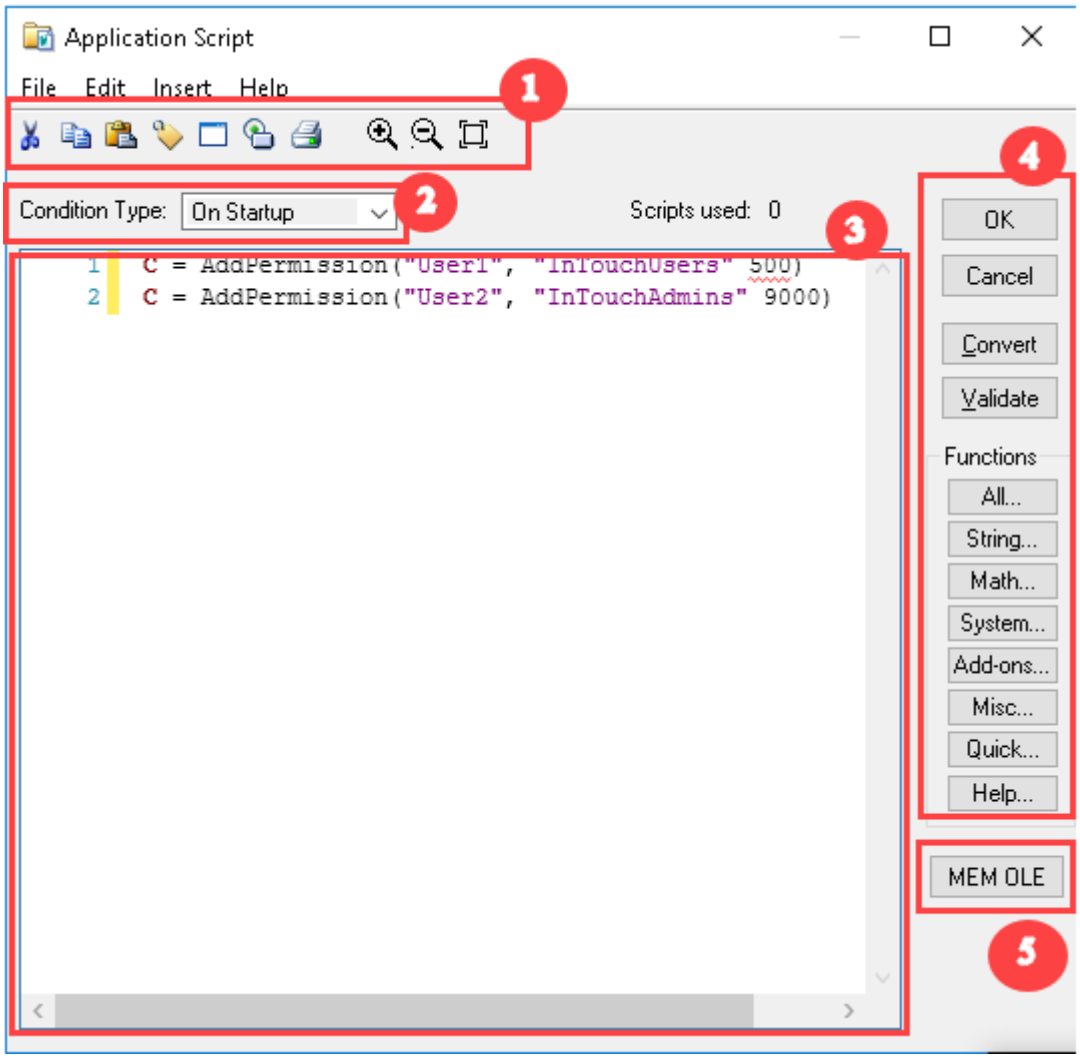
如需有关创建每种类型的脚本的详细信息，请参阅下面几节：

- 配置应用程序脚本。
- 配置窗口脚本。
- 配置键脚本。
- 配置条件脚本。
- 配置数据改变脚本。
- 配置动作脚本。
- 配置 ActiveX 事件脚本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用 InTouch 脚本编辑器

在 InTouch WindowMaker 内使用 InTouch HMI 脚本编辑器创建和编辑脚本。



区域	描述
1	工具栏
2	条件定义区域 **条件类型**框提供可供正在编写的脚本类型使用的执行条件。
3	脚本文本窗口
4	命令按钮
5	内置脚本函数按钮

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6	MEM OLE 按钮 仅当“制造工程模块”（Manufacturing Engineering Module，简称 MEM）随 InTouch HMI 程序一起安装时，右下角才会出现 **MEM OLE** 按钮。通过选择此按钮，可以使用 MEM 编写脚本。

<1>本例是应用程序脚本</1><2>。</2> <1>每种类型的脚本都有自己的脚本对话框</1><2>, </2><3>其中包含该脚本类型所特有的选项与选择</3><4>。</4>

编辑器的标题栏指出正在使用的脚本类型。如需有关脚本类型的信息，请参阅脚本类型。

在上下文相关的弹出窗口中，有一些文本、逻辑以及数学运算符按钮，选择它们即可在脚本中的光标位置插入该关键字、函数或符号。

InTouch 脚本元素的颜色指示器

InTouch 脚本编辑器使用不同的文本颜色标识不同的脚本元素。下表显示与脚本元素关联的文本颜色。

元素	颜色
关键字	蓝色 输入时语法突出显示。
注释（单行和多行）	绿色 输入时语法突出显示。
字符串	紫色 输入时语法突出显示。
函数名、数值常量、运算符、分号、dim 变量、别名变量等等	黑色 请参阅属性名和保留字的说明。
属性、InTouch 标记、引用字符串	褐紫红色，黑体
保留字	红色，非黑体
.NET 类型名称	青色，非黑体

InTouch 脚本编辑器自动完成功能

这些功能支持标记、点域、方法、编程构造和其它脚本元素的自动单词完成。脚本编辑器自动完成功能：“InTouch 脚本编辑器”自动完成功能融合了多个在编写 InTouch 脚本时使用的功能。这些功能支持标记、点域、方法、编程构造和其它脚本元素的自动单词完成。“脚本编辑器”自动完成功能：

- 在可选列表框中提供自动完成标记引用。
- 在自动完成列表框中提供方法参数帮助，包括涵盖定义、关键字、脚本元素、对象和属性名称以及编程构造的上下文特定建议。

这些功能用作方法参数和脚本语法以及增强输入方法的便利文档。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

按 **Ctrl+空格**可显示脚本中所选位置的所有可用自动完成选项和变量。您可以通过自动完成弹出窗口底部显示的图标来识别上下文。

图标	表示
	方法
	关键字
	运算符
	变量
	标记
	窗口
	ActiveX 实例

接受 InTouch 脚本编辑器自动完成建议

通过执行以下操作之一，在编辑器插入符号处从自动完成列表框插入项目（不附加换行符或 Tab 字符）：

- 双击项目
- 突出显示（选择）项目并按 **Enter** 键或 **Tab** 键。

键入空格、句点、逗号、左括号或右括号或者编程语言中使用的其它标点符号 (: ; [ ] = < > - + / *)，在自动完成列表框中突出显示的项目将在编辑器插入符号处插入，并附加字符。

禁用脚本自动完成

如果不想在编写脚本时查看自动完成建议，可以禁用脚本自动完成功能。

从 WindowMaker 配置屏幕禁用脚本自动完成

1. 在文件菜单上，选择配置，然后选择 **WindowMaker**。

   此时出现 WindowMaker 配置屏幕。

2. 选中禁用脚本自动完成复选框。

   自动完成建议将不会出现在脚本编辑器中。

从脚本窗口禁用脚本自动完成

1. 启动脚本窗口。
2. 在编辑菜单上，选择禁用脚本自动完成。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

自动完成建议将不会出现在脚本编辑器中。

InTouch 脚本中的多级撤消和恢复

您可以有选择地撤消对脚本的一系列更改。可以撤消的更改数量仅受可用内存的限制。

- 使用主菜单选项编辑，然后选择撤消或按 **Ctrl+Z** 撤消编辑。您也可以使用上下文菜单选项来撤消和恢复。
- 使用主菜单选项编辑，然后选择恢复或按 **Ctrl+Y** 恢复编辑。您也可以使用上下文菜单选项来撤消和恢复。

撤消的更改可以恢复。恢复操作可镜像撤消操作所做的更改。

单次撤消通常由输入或删除系列操作组成，与自动完成列表交互、用鼠标移动光标或选择脚本中的其它位置可中断此撤销操作。

如果关闭“脚本编辑器”或在“脚本编辑器”内切换到其它脚本，所有待处理的撤消和恢复操作都将丢失。

InTouch 脚本错误的可视化指示

InTouch 脚本文本中的错误会标记红色“波浪”下划线。

将鼠标光标悬停在错误上将以工具提示的方式显示错误消息。工具提示错误消息提供的信息与选择脚本验证按钮时显示的消息相同。

在某些情况下，会有多个错误被添加下划线。但有时可能并非如此，因为某些错误会阻止编译器继续通过错误。

当使用“\”字符串联两个字符串以形成路径时，脚本编辑器将在“\”下显示红色下划线。这是一个异常，可以忽略。路径在运行时会正确构造。

```
1  TagComment = InfoIntouchAppDir() + "\";
```

管理 InTouch 脚本编辑

“InTouch 脚本编辑器”提供了用于查找和管理编辑的工具和可视化参考。

脚本行号

“脚本编辑器”在左侧空白中显示行号。

- 当未缩放“脚本编辑器”时，将显示最多四位的行号。
- 使用右键单击上下文菜单转到选项可转到脚本中的特定行。
- 按 **Ctrl+L** 可删除当前文本行。
- 您可以选择文本行并将其拖到其它行。

更改条

脚本文本窗口左侧空白上的黄色更改条是进行中脚本更改的可视化参考，可指示添加、行插入以及编辑。

标记定义

您可以在“InTouch 脚本编辑器”中输入新标记名，然后按 **Ctrl+T** 或选择主菜单中的编辑来定义标记。将显示标记名字典，以便您完成标记名定义。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

查找与替换

“InTouch 脚本编辑器”提供可自定义的查找与替换功能。如需有关如何使用查找与替换功能的信息，请参阅在脚本内搜索。

打开脚本进行编辑

打开现有脚本的步骤根据脚本类型的不同而略有不同。

打开应用程序脚本

1. 在脚本窗格中，双击应用程序。

   此时出现“应用程序脚本”对话框。
2. 在条件类型列表中，选择要编辑的脚本类型。

打开窗口脚本

1. 在窗口窗格中，右键单击窗口名，然后选择窗口脚本。

   或者，打开脚本与之关联的窗口。使用鼠标右键单击窗口中的空白区域，然后选择窗口脚本。
2. 在条件类型列表中，选择导致脚本运行的条件。

打开 ActiveX 事件脚本

- 执行以下任意操作：
  - 在脚本窗格中，展开 ActiveX 事件，然后双击脚本名。
  - 双击脚本与之关联的 ActiveX 控件实例。选择事件选项卡，然后双击包含脚本名的单元。

打开动作脚本

1. 打开包含动作脚本与之关联的图形元素的窗口。动作脚本的典型用途是编写按钮上的动作的脚本。
2. 双击动作脚本与之关联的图形元素。
3. 在触动按钮区域中，选择动作。此时出现<1>“</1>脚本编辑器<2>”。</2>
4. 在条件类型列表中，单击导致脚本运行的动作。

打开键、条件或数据改变脚本

1. 在脚本窗格中，展开脚本类别，然后双击脚本名。
   - 此时出现<1>“</1>脚本编辑器<2>”。</2> 选择浏览按钮，然后选择脚本名。
2. 如果适用，在条件类型列表中，选择导致脚本运行的条件。

保存或放弃对脚本的更改

使用“脚本编辑器”期间或完成时，可以手动或自动保存脚本。也可以完全放弃脚本。

恢复选项在窗口与应用程序脚本中不可用。

注意：保存和放弃更改始终适用于一类脚本的所有条件类型，而不只是当前可见的条件类型。

保存更改并保持脚本的打开状态

- 在文件菜单上，选择保存。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

保存更改并关闭脚本

- 选择确定。

放弃更改并保持脚本的打开状态

- 选择恢复。

放弃更改并关闭脚本

- 选择取消。

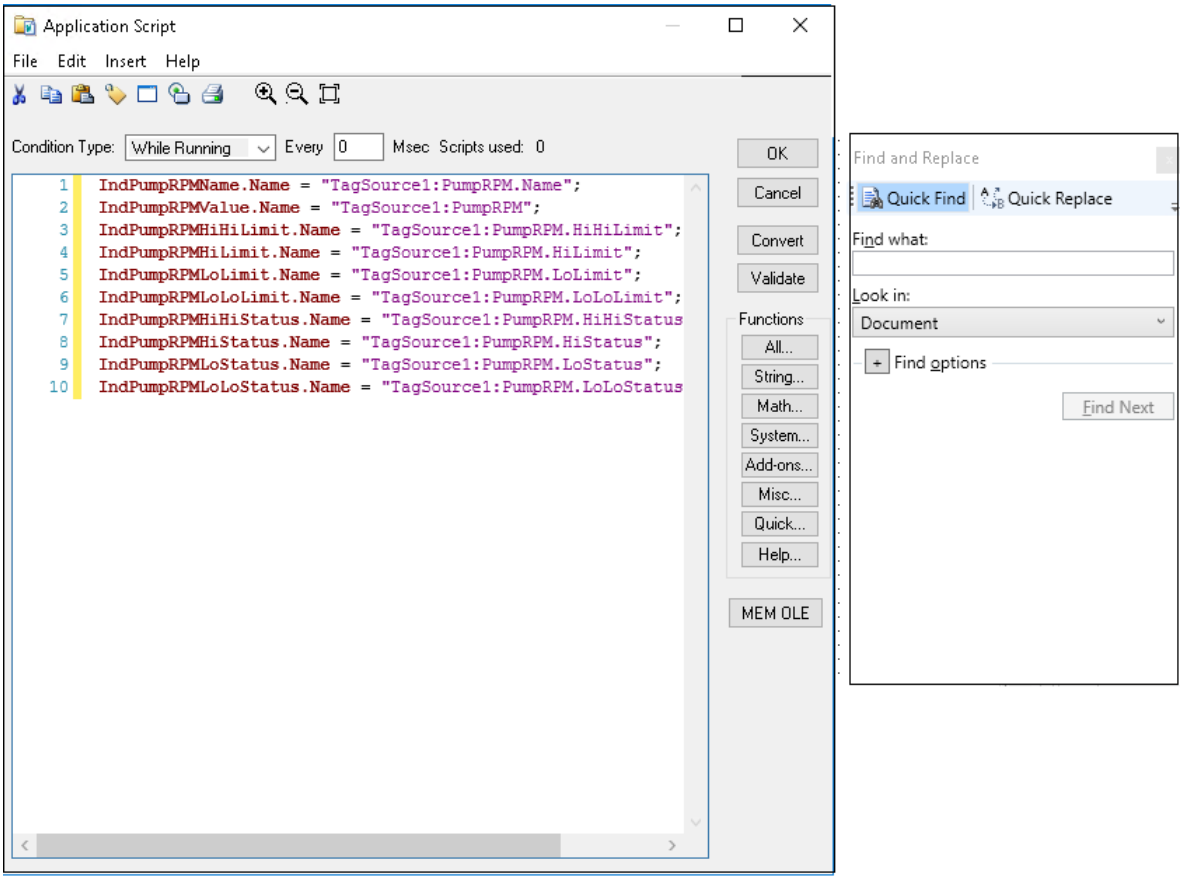
复制、剪切和粘贴文本

在<1>“</1>脚本编辑器<2>”</2>中<3>，</3>复制<4>、</4>剪切及粘贴文本与在任何其它 Windows 应用程序中的工作方式相同<5>。</5> 使用标准的键盘快捷键 Ctrl-C、Ctrl-X 以及 Ctrl-V，或是工具栏按钮。

还可以选择一行，然后将该行中的文本拖到脚本中的新位置。

在脚本内搜索

“InTouch 脚本编辑器”提供了在打开的脚本中根据搜索字符串和一组自定义搜索选项来查找和替换文本的功能。您可以通过“脚本编辑器”的菜单栏上的编辑选项来访问查找和替换。



还可以使用键盘快捷键 **Ctrl+F**（快速查找）和 **Ctrl+H**（快速替换）来显示查找与替换对话框。

- 在使用缺省搜索选项的简单搜索中，选择快速查找选项卡，然后在查找内容字段中输入词或短语。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

选择查找下一个开始搜索。琥珀色背景用于标识脚本中与搜索字符串匹配的词或短语。

- 在使用缺省选项的简单替换操作中，选择快速替换选项卡，然后在查找内容字段中输入词或短语，同时在替换为字段中输入替换的词或短语。

  选择查找下一个开始替换操作。脚本中用浅蓝色背景标识与在查找内容字段中输入的搜索字符串匹配的词或短语。在找到匹配的词或短语后，有三个替换选项。

  - 选择查找下一个将忽略当前的匹配词或短语，并继续在脚本中搜索下一个匹配。
  - 选择替换会将当前的匹配词或短语替换为在替换为字段中输入的字符串。
  - 选择全部替换会将所有匹配词或短语替换为在替换为字段中输入的字符串。

配置查找或替换

查找与替换对话框提供了一组选项来配置搜索选项

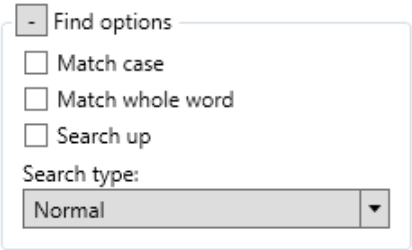
查找

查找字段包括用于搜索整个脚本的选项（文档）或仅搜索脚本选定部分的选项（所选内容）。文档是缺省选项。

当您只想在脚本的选定部分内进行搜索时，可先使用鼠标在脚本内选择搜索区域，然后执行搜索。蓝色背景标识可以搜索的选定脚本行。搜索结果仅显示选定脚本部分内的匹配项。

查找选项

您可以展开或折叠查找选项部分。还可以选择或清除以下选项来更精确地过滤搜索结果。



- 匹配大小写

  选择此选项时，搜索结果仅显示内容和大小写均匹配的查找内容字符串实例。例如，在选择匹配大小写的情况下搜索 Triangle4 将返回 Triangle4 而不是 triangle4。

- 全字匹配

  选择此选项时，搜索结果仅显示全字匹配的查找内容字符串实例。例如，搜索 LogicBit 将返回 LogicBit 而不是 LogicBits。

- 向上搜索

  选择此选项时，将从脚本内的当前位置向脚本顶部执行搜索。缺省情况下，从脚本内的当前位置向底部执行搜索。

搜索类型

搜索类型字段提供了用于根据搜索类型执行脚本搜索的选项。

- 正常

  这是缺省搜索类型，要求搜索字符串中的字符与脚本中的文本完全匹配。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 按正则表达式搜索

  正则表达式描述一个或多个要在搜索脚本时匹配的字符串。正则表达式包含一些普通字符，这些字符是将字符模式与所搜索的字符串进行匹配的模板。

- 按通配符搜索

  通配符使用星号 (*) 或问号 (?) 等键盘字符来代表在脚本内搜索时的一个或多个字符。

- 按首字母缩略词搜索

  首字母缩略词搜索先匹配单词开头的字符，然后匹配下划线后面的每个大写字母或字符。

- 按简写搜索

  简写搜索扩展了“按首字母缩略词搜索”选项，允许在搜索模式字符之间存在非空格字符。

所输入的搜索字符串必须符合所选搜索类型的语法和支持的字符。

按通配符搜索

通配符搜索使用单个键盘字符，在脚本中搜索字符串时，该键盘字符可以是多个文字字符，也可以是空字符串。

当不知道确切的字符或者不想输入整个搜索字符串时，通常会使用通配符来代替一个或多个字符。

通配符	功能
星号 (*)	搜索字符串中的星号匹配任何字符序列。使用星号可代替零个或多个字符。 **示例** • Logic* 找到 Logic1、LogicTest，但不会找到 ALogicTest • *Test* 找到 LogicTest1、PumpTestABC，但不会找到 LogicTst1
问号 (?)	搜索字符串中的问号匹配搜索字符串内一个位置处的任意字符。 **示例** • LogicTest? 找到 LogicTest1、LogicTestA，但不会找到 ALogicTest1 • LogicTest?2 找到 LogicTest12，但不会找到 LogicTest13

按正则表达式搜索

正则表达式使用字母数字字符和称为元字符的特殊字符来描述一个或多个要在搜索脚本时匹配的字符串。正则表达式充当一种字符模式，用来与被搜索的脚本文本进行比较。

正则表达式的结构很像算术表达式。使用各种元字符和运算符将简单的表达式组合起来，创建复杂的表达式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

正则表达式的组件可以是单个字符、成组的字符、字符范围，或在字符之间的选择。也可以是这些组件的任意组合。

脚本正则表达式

正则表达式	用途	示例
。	匹配任意单个字符（换行符除外）	s.e 匹配“step”中的“ste”和“transfer”中的“sfe”，但不匹配“across”中的“acro”。
*	*前面的表达式匹配零次或多次（匹配尽可能多的字符）	a*r 匹配“rack”中的“r”、“ark”中的“ar”和“aardvark”中的“aar”
.*	匹配任意字符零次或多次（通配符 *）	c.*e 匹配“racket”中的“cke”、“comment”中的“comme”和“code”中的“code”
+	+前面的表达式匹配一次或多次（匹配尽可能多的字符）	e.+e 匹配“feeder”中的“eede”，但不匹配“ee”。
.+	匹配任意字符一次或多次（通配符 ?）	e.+e 匹配“feeder”中的“eede”，但不匹配“ee”。
*?	*?前面的表达式匹配零次或多次（匹配尽可能少的字符）	e.*?e 匹配“feeder”中的“ee”，但不匹配“eede”。
+?	+?前面的表达式匹配一次或多次（匹配尽可能少的字符）	e.+?e 匹配“enterprise”中的“ente”和“erprise”，但不匹配整个词“enterprise”。
^	将匹配字符串定位到行首或字符串的开头	^car 仅匹配出现在行首的单词“car”。
\r?$	将匹配字符串定位到行尾	End\r?$ 仅匹配出现在行尾的单词“end”。
[abc]	匹配字符集中的任意单个字符	b[abc] 匹配“ba”、“bb”和“bc”。
[a-f]	匹配字符范围中的任意字符	be[n-t] 匹配“between”中的“bet”、“beneath”中的“ben”和“beside”中的“bes”，但不匹配“below”。
()	捕获括号内包含的表达式并隐式地为其编号	([a-z])X\1 与 "aXa" 和 "bXb" 匹配，但与 "aXb" 不匹配。". "\1" 是指第一个表达式组 "[a-z]"。
(?!abc)	使匹配无效	real (?!ity) 匹配“realty”和“really”中的“real”，但不匹配“reality”。它还会找到

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

		“realityreal”中的第二个“real”（而非第一个“real”）。
[^abc]	匹配不在给定字符集中的任何字符	be[^n-t] 匹配“before”中的“bef”、“behind”中的“beh”和“below”中的“bel”，但不匹配“beneath”。
|	匹配符号前面或符号后面的表达式。	(sponge|mud)bath 匹配“spongebath”和“mudbath”。
|\^	对反斜杠后面的字符进行转义	
{x},	指定前面的字符或组的出现次数	x(ab){2}x 匹配“xababx”，并且 x(ab){2,3}x 匹配“xababx”和“xabababx”，但不匹配 “xabababab x”。
\p{X}	匹配 Unicode 字符类中的文本, 其中“X”是 Unicode 数字。	\p{Lu} 匹配“Thomas Doe”中的“T”和“D”。
\b	匹配单词两侧的字符	\bin 匹配“inside”中的“in”，但不匹配“pinto”。
\r?\n	匹配换行符（即新行前面的回车符）。	End\r?\nBegin 仅当“End”是一行的最后一个字符串，且“Begin”是下一行的第一个字符串时，才匹配“End”和“Begin”。
\w	匹配任意字母数字字符	a\wd 匹配“add”和“a1d”，但不匹配“a d”。
(?[^\r\n])\s	匹配任意空格字符。	Public\sInterface 匹配短语“Public Interface”。
\d	匹配任意数字字符	\d 匹配“3456”中的“3”、“23”中的“2”和“1”中的“1”。
\uXXXX（其中 XXXX 指定 Unicode 字符值）	匹配 Unicode 字符	\u0065 匹配字符“e”。
\b(\w+| [\w-[0-9\]]\w*)\b	匹配标识符	匹配“Type1”，但不匹配“&type1”或“#define”。
((\".+?\")|('.+?'))	匹配引号中的字符串	匹配单引号或双引号内的任何字符串。
\b0[xX]([0-9a-fA-F])\b	匹配十六进制数	匹配“0xc67f”，但不匹配“0xc67fc67f”。
\b[0-9]\.\[0-9]+\b	匹配整数和小数	匹配“1.333”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

优先顺序

从左到右评估正则表达式，并遵守一定的优先顺序。

下表包含了正则表达式运算符的优先顺序，从最高到最低排列。

运算符	描述
\	转义
()、(?:)、(?=)、[]	圆括号和方括号
*、+、?、{*n*}、{*n*,}、{*n*,*m*}	量词
^、$、*任何元字符*	定位和顺序
|	交替

字符的优先级高于交替运算符，例如，允许“m|food”匹配“m”或“food”。

按首字母缩略词搜索

首字母缩略词搜索先匹配单词开头的字符，然后匹配下划线后面的每个大写字母或字符。

示例：搜索项“LB”会找到“LogicBits”的实例。

按简写搜索

简写搜索扩展了“按首字母缩略词搜索”选项，允许在搜索模式字符之间存在任意数量的非空格字符。

示例：搜索项“ta”会找到“Triangle”的实例。

插入代码元素

<1>通过从列表中进行选择</1><2>, </2><3>可以将各种代码元素自动插入脚本</3><4>。</4> <1>这可以节省时间</1><2>，</2><3>并减小输入错误的风险</3><4>。</4> 以下面两种方式之一访问代码元素：使用主菜单的插入选项，或使用自动完成弹出窗口并从列表中选择元素。

您可以从自动完成弹出窗口和“脚本编辑器”菜单项插入的代码元素包括：

- 函数
- 标记名和点域。对于点域，输入标记名后加句点可打开自动完成弹出窗口。
- 变量
- 窗口名
- ActiveX 实例
- 关键字
- 操作员

自动完成弹出窗口的底部包括一行图标，用于直接访问可用元素。

访问脚本函数的帮助

如需有关特定脚本函数的帮助<1>，</1>可以直接从<2>“</2>脚本编辑器<3>”</3>进行访问<4>。</4>

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

查看关于特定脚本函数的帮助

1. 在“脚本编辑器”的右下角，选择帮助。

   此时出现一个函数列表。

2. 选择列表中的函数名称可显示其帮助。

   此时出现相应的帮助主题。

验证脚本的语法是否正确

保存脚本时，“脚本编辑器”会自动检查语法是否正确。如果发现错误，则出现一条包含更多信息的消息。您必须纠正所有的语法错误，才可以保存脚本。您还可以在编辑脚本的过程中手动开始验证。

手动验证脚本语法

- 选择验证。

突出显示脚本错误

“InTouch 脚本编辑器”还会突出显示脚本错误。如需详细信息，请参阅 InTouch 脚本错误的可视化指示。

打印脚本

您可以从“脚本编辑器”中单独打印脚本，也可以使用 WindowMaker 中的打印功能来打印特定类型的所有脚本。

您可以从“脚本编辑器”中单独打印脚本，也可以使用 WindowMaker 中的打印功能来打印特定类型的所有脚本。

打印单个脚本

1. 在<1>“</1>脚本编辑器<2>”</2>中打开脚本<3>。</3>
2. 选择工具栏中的打印。此时脚本会通过 Windows 缺省打印机打印。

打印特定类型的所有脚本

1. 在快速访问工具栏上，选择打印。

   此时出现 **WindowMaker** 打印件对话框。

2. 要打印窗口脚本，请执行以下操作：
   a. 选择窗口。
   b. 选择要打印的窗口：

      全部打印应用程序中所有窗口的信息。

      所选的仅打印特定窗口的信息。此时会出现要打印的窗口对话框。选择应用程序中要打印的窗口，然后选择确定。

      批仅打印在 .csv 文件中指定的窗口的信息。

   c. 选择窗口脚本，以打印与窗口关联的脚本。
3. 要打印其它类型的脚本，请选择适当的复选框。要打印所有脚本，请选择全部脚本。
4. 选择下一步。此时出现选择输出目标对话框。
5. 执行以下操作之一：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 选择发送输出到打印机。
- 选择发送输出到文本文件。

6. 选择浏览以选择打印机或查找文件。
7. 选择打印。

打印所有脚本

1. 选择全部脚本以打印应用程序中使用的所有脚本。

   通过清除全部脚本复选框，可以仅打印所选类型的脚本。然后，对于希望打印的每种脚本类型，选择对应的复选框。
2. 选择下一步。此时出现选择输出目标对话框。
3. 选择相应选项以打印“标记名字典”的内容，或是将输出发送到文本或 .html 文件。
4. 选择打印。

删除脚本

删除脚本的步骤根据脚本类型的不同而各异。请参阅下面几节：

- 配置应用程序脚本。
- 配置窗口脚本。
- 配置键脚本。
- 配置条件脚本。
- 配置数据改变脚本。
- 配置动作脚本。
- 配置 ActiveX 事件脚本。

调整缩放选项以查看脚本

您可以调整缩放选项以查看脚本，如放大、缩小或缩放为正常大小。

放大

1. 打开“脚本编辑器”。
2. 在“脚本编辑器”工具栏上，选择放大。

   或者选择 编辑，然后选择 放大。

缩小

1. 打开“脚本编辑器”。
2. 在“脚本编辑器”工具栏上，选择缩小。

   或者，选择编辑，然后选择缩小。

要缩放为正常大小

1. 打开“脚本编辑器”。
2. 在“脚本编辑器”工具栏上，选择缩放为正常大小。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

或者，选择编辑，然后选择缩放为正常大小。

脚本触发器

所有的 InTouch HMI 脚本都由脚本触发器执行。每种类型的脚本都有一个或多个触发器用于启动它。

在“脚本编辑器”中，可以选择要用于执行脚本的脚本触发器。您可以根据执行脚本的时间与方式来选择脚本触发器。

您可以根据用户动作、内部状态与标记名值的更改来配置各种触发器。用户动作包括按键与选择图形元素。内部状态触发器可以包括启动 WindowViewer。

脚本由以下这些动作触发：

- 启动与关闭 WindowViewer。请参阅配置应用程序脚本。
- 打开与关闭窗口。请参阅配置窗口脚本。
- 按单键或组合键。请参阅配置键脚本。
- 满足特定条件，如标记名或表达式值。请参阅配置条件脚本。
- 更改标记名值或标记名字段值。请参阅配置数据改变脚本。
- 选择图形对象。请参阅配置动作脚本。
- 发生在 ActiveX 控件中的事件，如选择控件。请参阅配置 ActiveX 事件脚本。

此外，也可以暂停脚本执行。缺省条件下，启动 WindowViewer 时运行逻辑并执行脚本。在运行时，可以通过停止逻辑来暂停脚本执行。暂停之后可以恢复脚本执行。如需详细信息，请参阅在运行时暂停脚本执行。

脚本触发器的类型

在 InTouch HMI 中，脚本分为七种类型。每种类型的脚本都有一个或多个触发器可选择用于启动脚本。

- 应用程序脚本有三个触发器：启动时、关闭时及运行期间。每个触发器都可以执行不同的脚本。
- 窗口脚本有三个触发器：显示时、隐藏时及显示期间。每个触发器都可以执行不同的脚本。
- 键脚本有三个触发器：放开时、按下时，或是按下期间。每个触发器都可以执行不同的脚本。
- 条件脚本有四个触发器：为真时、为真期间、为假时以及为假期间。每个触发器都可以执行不同的脚本。
- 特定标记或表达式的值发生改变时，执行数据改变脚本。
- 动作脚本在操作员单击 InTouch HMI 图形对象时执行一次或定期执行。
- 发生特定的 ActiveX 事件（如选择 ActiveX 控件）时，ActiveX 事件脚本执行一次。

使用多个触发器

对于大多数脚本类型而言，都可以使用多个触发器，并可以将不同的脚本关联到每个触发器。

例如，您可以将一个应用程序脚本配置为在 WindowViewer 启动时执行某个脚本一次，在 WindowViewer 运行期间定期执行另一个脚本。

在条件类型列表中，选择触发器以查看触发器的现有脚本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

定期执行脚本

定期执行的脚本不是在触发之后立即执行，而是在指定的周期之后执行。

例如，如果将一个键脚本配置成在按下特定的键期间，每隔 5000 毫秒执行一次，则它会在按住该键 5 秒之后执行，此后每隔 5 秒执行一次。

配置应用程序脚本

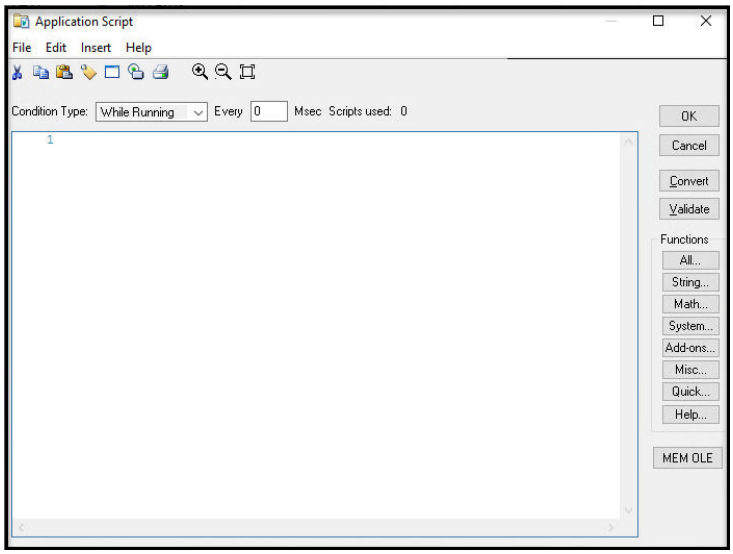
应用程序脚本链接到整个 InTouch HMI 应用程序。您可以使用应用程序脚本：

- 在 WindowViewer 启动时执行某个脚本一次。
- 在 WindowViewer 运行期间定期执行某个脚本。
- 在 WindowViewer 关闭时执行某个脚本一次。

配置应用程序脚本

1. 在脚本窗格中，右键单击应用程序，然后选择打开。

   此时出现应用程序脚本对话框。

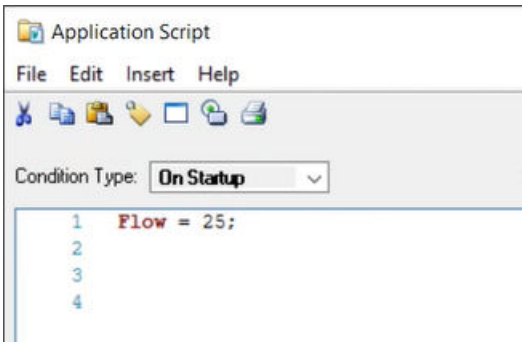


2. 在条件类型列表中，选择脚本执行的条件：
   - 选择启动时，以便将脚本配置成在 WindowViewer 启动时执行一次。

     示例：在 InTouch 应用程序中，如果名为 **Flow** 的标记的初始值为 50，并且要在应用程序启动时更改此值，则将条件类型选择为启动时，并使用以下脚本：

     ```
     Flow = 25;
     ```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



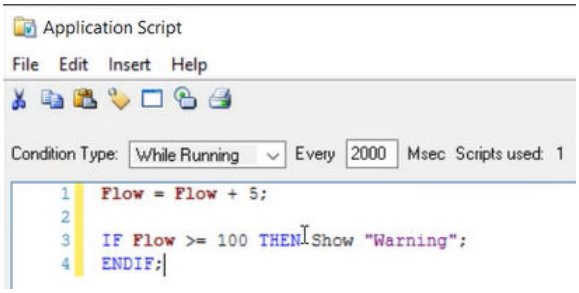
- **选择运行期间**，以便将脚本配置成在 WindowViewer 运行期间定期执行。

  示例：考虑一个带有名为 **Flow** 的标记的 InTouch 应用程序。应用程序运行期间，如果希望 **Flow** 的值每 2 秒增加 5，并且希望在标记值大于或等于 100 时显示名为**“警告”**的窗口，则将**条件类型设置为运行期间**，并使用以下脚本：

  ```
  Flow = Flow + 5;
  ```

  若 `Flow >= 100`，则显示“警告”

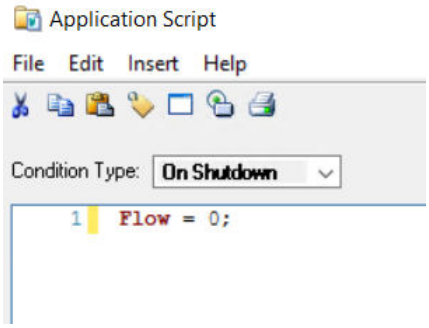
  ```
  ENDIF;
  ```



- **选择关闭时**，以便将脚本配置成在 WindowViewer 关闭时执行一次。

  示例：考虑一个带有名为 **Flow** 的标记的 InTouch 应用程序。关闭应用程序时，如果希望将 **Flow** 的值更改为 0，则将**条件类型设置为关闭时**，并使用以下脚本：

  ```
  Flow = 0;
  ```



3. 如果在上一步中选择了**运行期间**，请在**每**框中输入 1 到 360000 毫秒之间的一个时间间隔。该时间间隔指定脚本执行的频率。
4. 在窗口中输入脚本。
5. 选择**确定**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

删除应用程序脚本

1. 在脚本窗格中，右键单击应用程序，然后选择打开。

   此时出现应用程序脚本对话框。

2. 在条件类型列表中，选择要删除的脚本。

   此时脚本出现在应用程序脚本对话框的主要部分。

3. 在编辑菜单上，选择清除。

   此时该脚本会从该主要部分中清除，关联的脚本也被删除掉。

应用程序脚本的限制

在启动或关闭 WindowViewer 时执行的应用程序脚本在同其它对象进行交互方面存在一定的限制。

您无法使用<1>“</1>启动时<2>”</2>应用程序脚本<3>：</3>

- 引用 ActiveX 方法、属性或事件。
- 对控件与 I/O 标记名或远程引用进行读取或写入。
- 运行数据改变脚本与条件脚本。

您无法使用<1>“</1>关闭时<2>”</2>应用程序脚本<3>：</3>

- 对控件与 I/O 标记名或远程引用进行读取或写入。
- 启动其它应用程序。

配置窗口脚本

窗口脚本是链接到特定窗口的脚本。您可以使用 **GetWindowName** 脚本函数帮助运行时环境减少在加载窗口时需要编写的脚本量。您可以使用窗口脚本：

- 在打开 InTouch 窗口时执行某个脚本一次。
- 在打开 InTouch 窗口期间定期执行某个脚本。
- 在关闭 InTouch 窗口时执行某个脚本一次。

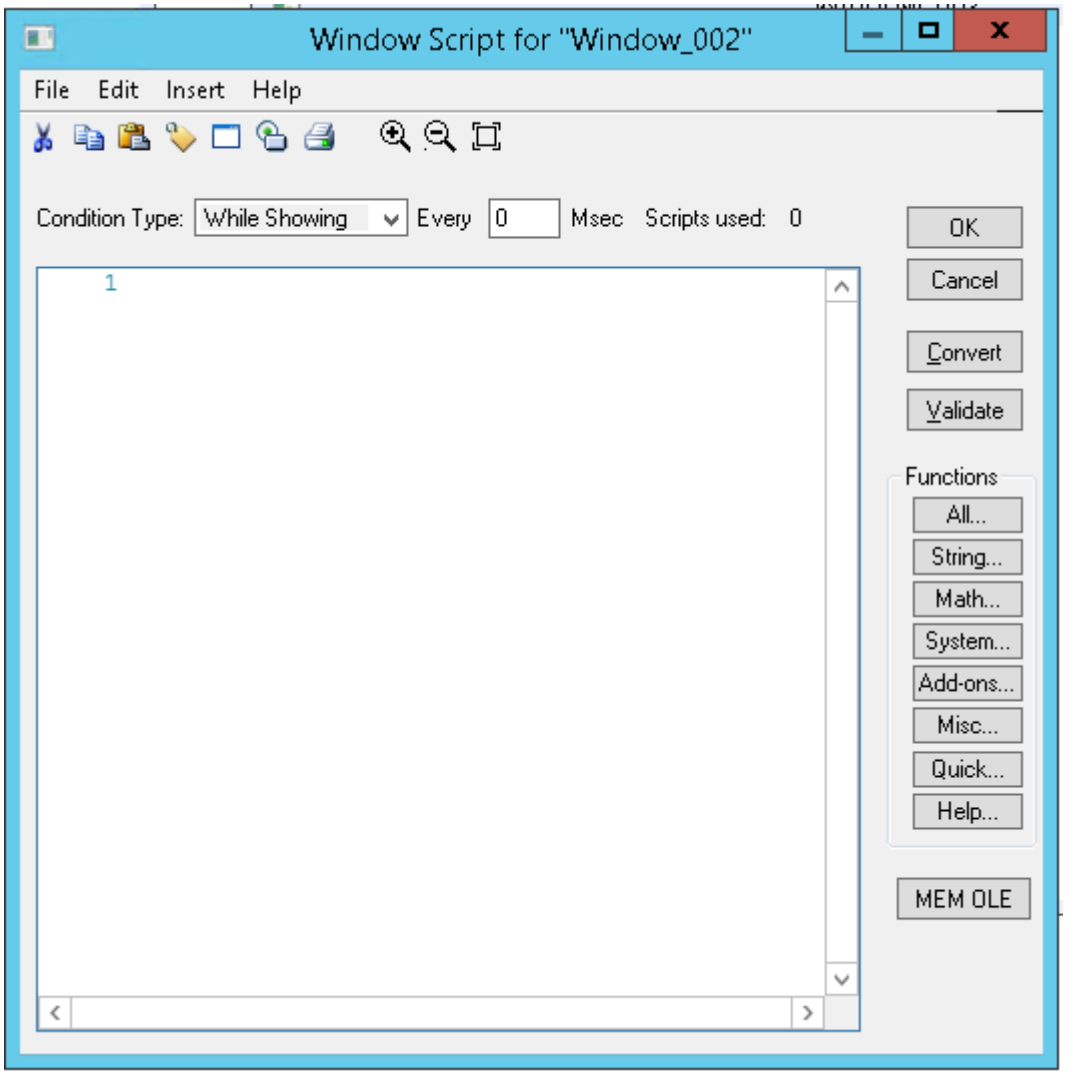
注意：打开 InTouch 窗口也称为“显示 InTouch 窗口”。关闭 InTouch 窗口也称为“隐藏 InTouch 窗口”。

配置窗口脚本

1. 在窗口窗格中，右键单击窗口，然后选择窗口脚本。

   此时出现窗口脚本 - 窗口名对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**条件类型**列表中，**执**行以下操作之一：
   - **选择显示时**，以便将脚本配置成在**关联**的窗口启**动时执**行一次。
   - **选择显示期间**，以便将脚本配置成在**关联**的窗口打**开**期**间**定期**执**行。
   - **选择隐藏时**，以便将脚本配置成**关联**的窗口**关闭时执**行一次。
3. 如果在上一**步**中**选择显示期间**，**请**在**每**框中**输**入介于 1 到 360000 毫秒之**间**的**时间间**隔。
4. 在窗口中**输**入脚本。
5. **选择确定**。

删除窗口脚本

1. 在**窗口**窗格中，右**键单击**窗口，然后**选择窗口脚本**。此**时**出**现窗口脚本 - 窗口名对话**框。
2. 在**条件类型**列表中，**选择**要**删**除的脚本的脚本触**发**器。此**时**脚本出**现**在**窗口脚本 - 窗口名对话**框的主要部分。
3. 在**编辑**菜**单**上，**选择清除**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要点：请勿使用**隐**藏**时**脚本**读**取或写入 I/O **标记**名。<1>I/O **值**更新不必在窗口**隐**藏前完成</1><2>。</2>要在窗口**关闭时读**取或写入 I/O **标记**名，**请**配置数据更改脚本并从**隐**藏**时**脚本将其激活。

配置**键**脚本

键脚本是**链**接到特定**单键**或**组**合**键**的按**键动**作的脚本。您可以使用**键**脚本：

- 在按**单键**或**组**合**键时执**行某个脚本一次。
- 在按住不放某个**单键**或**组**合**键**期**间**定期**执**行某个脚本。
- 在**释**放**单键**或**组**合**键时执**行某个脚本一次。

键脚本由启**动**脚本的**键**的名称来确定。例如：Ctrl+Q。

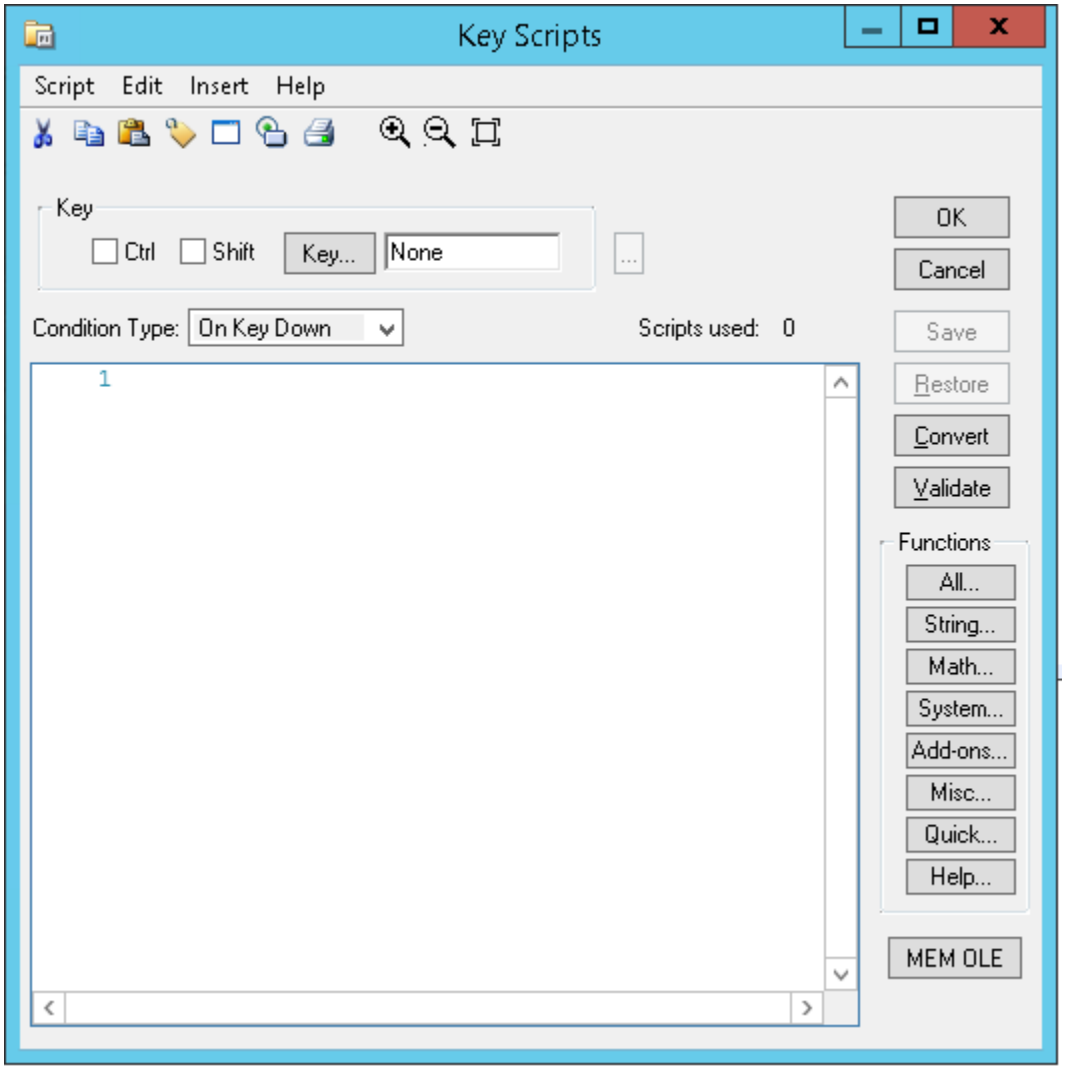
注意：如果已**经**将一个**动**作脚本配置成使用相同的**单键**或**组**合**键**来触**发**，**则**会忽略**键**脚本而**执**行**动**作脚本。

配置**键**脚本

1. 在**脚本**窗格中，**执**行以下操作之一：
   - 要配置新的**键**脚本，使用鼠**标**右**键单击键**，然后**选择新建**。

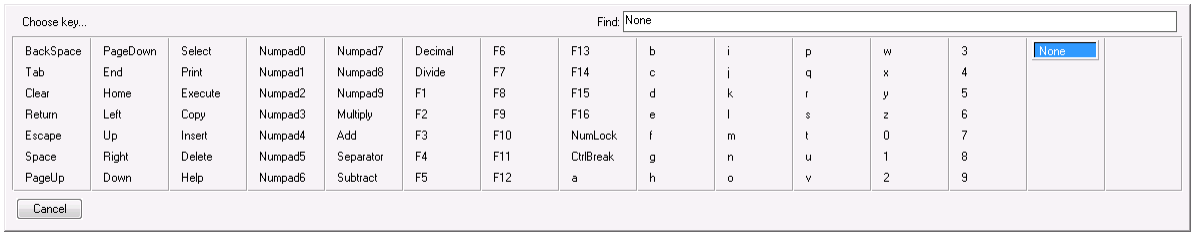
     此**时**出**现键脚本对话**框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有**权**利。



- 要配置现有的键脚本，展**开键**，使用鼠标右键单击脚本名，然后单击**编辑**。此时出现**编辑键脚本**对话框。

2. 选择**键**并从**选择键**对话框中选择一个键。



3. 选中 **Ctrl** 与/或 **Shift** 复选框，以便指定 Ctrl 与/或 Shift 键同所选键的组合。
4. 在**条件类型**列表中，执行以下操作之一：
   - **选择按下时**，以便将脚本配置为在按关联的单键或组合键时执行一次。
   - **选择按下期间**，以便将脚本配置为在按住关联的单键或组合键期间定期执行。
   - **选择放开时**，以便将脚本配置为在释放关联的单键或组合键时执行一次。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5. 如果在上一步中选择按下期间，请在每框中输入介于 1 到 360000 毫秒之间的时间间隔。
6. 在窗口中输入脚本。
7. 选择确定。

删除与某个键关联的所有键脚本

- 在脚本窗格中，展开键，右键单击键脚本名，然后选择删除。出现消息时，选择是。

删除与某个键关联的键脚本

1. 使用经典视图，在脚本窗格中，展开键，使用鼠标右键单击键脚本名，然后单击编辑。此时出现编辑键脚本对话框。
2. 在条件类型列表中，选择要删除的脚本的脚本触发器。该脚本会出现在编辑键脚本对话框的主要部分。
3. 在编辑菜单上，选择清除。此时该脚本会从该主要部分中清除，关联的脚本也被删除掉。

配置条件脚本

条件脚本根据何时满足特定的逻辑条件而触发。使用条件脚本：

- 在满足条件时执行脚本一次。
- 在不满足条件时执行脚本一次。
- 在满足特定的条件期间定期执行脚本。
- 在不满足特定的条件期间定期执行脚本。

条件脚本由启动脚本的条件语法来确定。例如：tag1>=13。

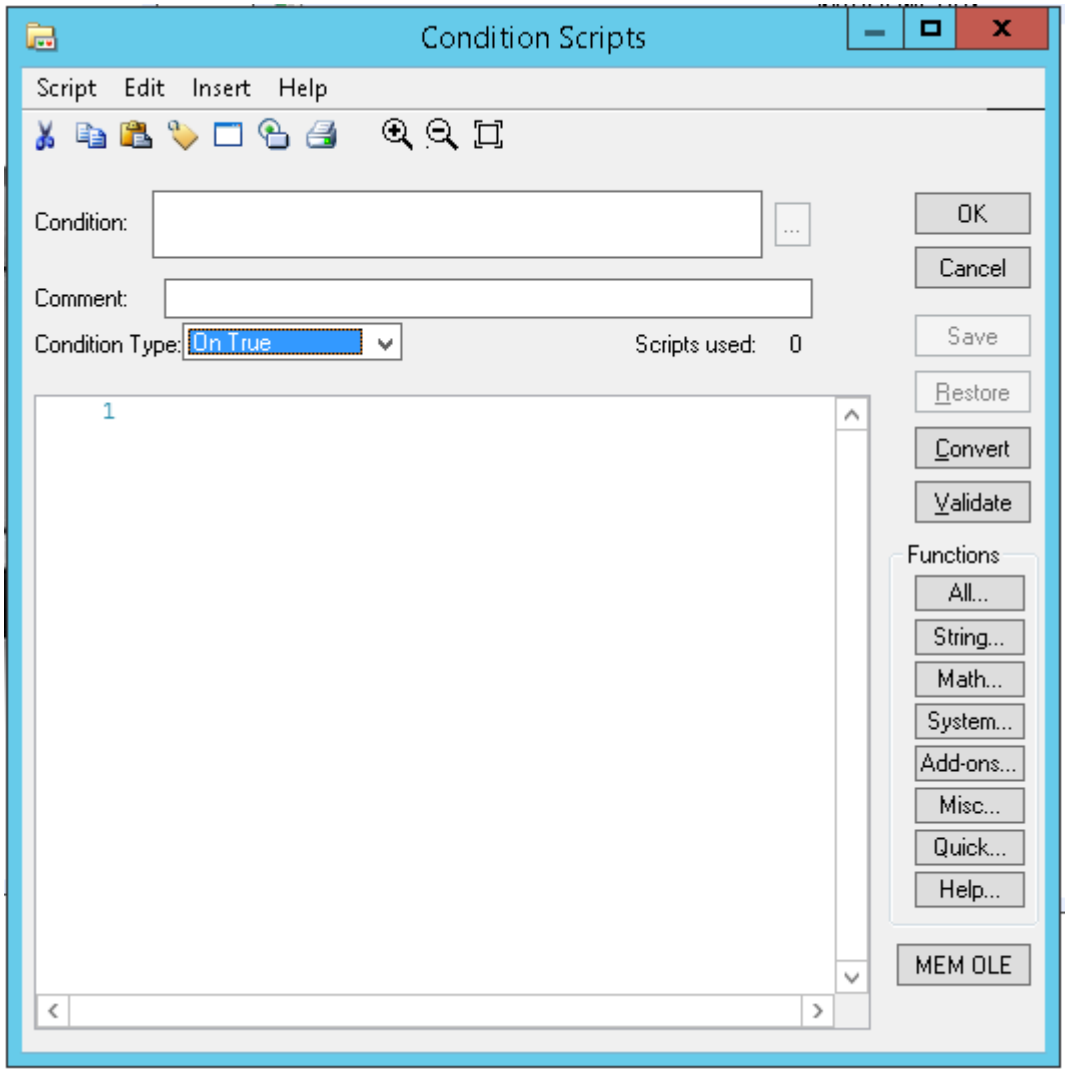
注意：指定了“为真时”条件类型的脚本仅当条件从假转变为真时执行。<1>指定了</1><2>“</2><3>为假时</3><4>”</4><5>条件类型的脚本仅当条件从</5><6>“</6><7>真</7><8>”</8><9>转变为</9><10>“</10><11>假</11><12>”</12><13>时执行</13><14>。</14>

配置条件脚本

1. 在脚本窗格中，右键单击条件，然后选择新建。

   此时出现条件脚本对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



- 要**编辑**现有的条件脚本，**请单击条件**旁边的加号，使用鼠标右**键单击**条件脚本名，然后**单击编辑**。此时出现**编辑条件脚本对话**框。

2. 在**条件**框中，**输**入要用作条件的表达式。

   表达式最**长**可以**输**入 1024 个字符。

3. 您可以在**注释**框中**输**入注**释**。
4. 在**条件类型**列表中，**执**行以下操作之一：
   - **选择为假时**，以便将脚本配置成在条件**变为**假**时执**行一次。
   - **选择为假期间**，以便将脚本配置成在条件**为**假期**间**定期**执**行。
   - **选择为真时**，以便将脚本配置成在条件**变为**真**时执**行一次。
   - **选择为真期间**，以便将脚本配置成在条件**为**真期**间**定期**执**行。
5. 如果在上一**步**中**选择为假期间**或**为真期间**，**请**在**每**框中**输**入介于 1 到 360000 毫秒之**间**的**时间间**隔。

注意：如果条件不再**为**真，条件 WindowViewer 定**时**器将自行停止。例如，如果不再按下鼠**标**按**键**，将不会触**发**“按下鼠**标**按**键**期**间**”事件，如果不再按下**键**，**键**脚本将停止。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 输入脚本，或修改窗口中现有的脚本。
7. 选择确定。

删除与某个条件关联的所有条件脚本

- 在脚本窗格中，展开条件，右键单击条件脚本名，然后选择删除。出现消息时，选择是。

删除与某个条件关联的单个条件脚本

1. 在脚本窗格中，展开条件，右键单击键脚本名，然后选择编辑。此时出现编辑条件脚本对话框。
2. 在条件类型列表中，选择要删除的脚本的脚本触发器。该脚本会出现在编辑条件脚本对话框的主要部分。
3. 在编辑菜单上，选择清除。此时该脚本会从该主要部分中清除，关联的脚本也被删除掉。

配置数据改变脚本

特定的标记名或点域的改变超过其定义的死区时，可以使用数据改变脚本执行某个脚本一次。

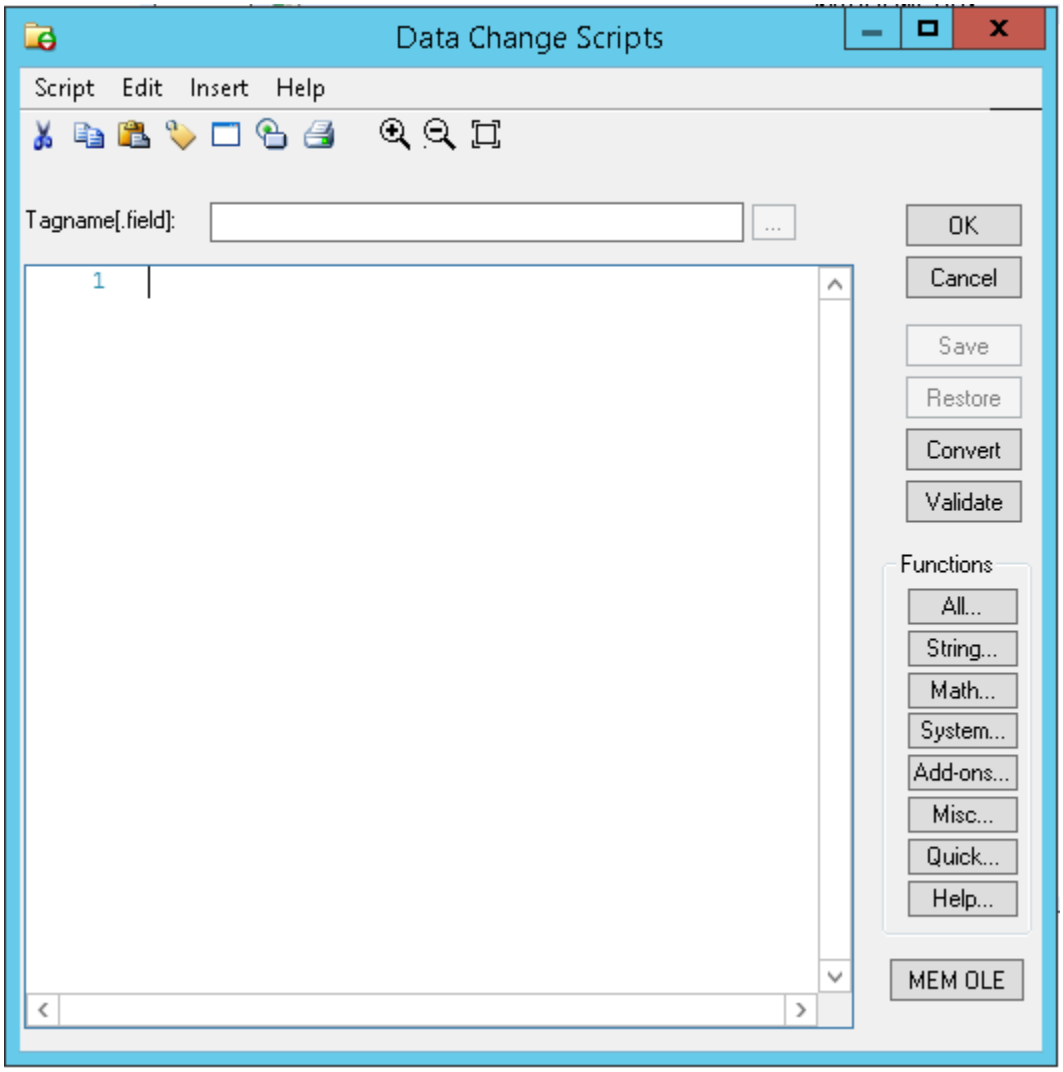
数据改变脚本由启动脚本的标记名或标记名字段来确定。例如：Tag1 或 Tag2.HiHiLimit。

配置数据改变脚本

1. 在脚本窗格中，右键单击数据更改，然后选择新建。

   此时出现数据改变脚本对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 要创建新脚本，在**标记名[点域]**框中，输入标记名或标记名字段。

   要编辑现有脚本，请选择**标记名[点域]** 框右侧的省略号按钮，然后从出现的列表中选择脚本。
3. 在窗口中输入脚本。
4. 选择确定。

删除数据改变脚本

- 在脚本窗格中，展开数据更改，右键单击数据更改脚本名，然后选择删除。出现消息时，选择是。

配置动作脚本

使用动作脚本将操作员动作关联到图形对象。您可以使用图形对象配置一个或多个以下事件：

- 释放鼠标左键、中键或右键。
- 按住鼠标左键、中键或右键。
- 释放鼠标左键、中键或右键。
- 双击鼠标左键、中键或右键。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

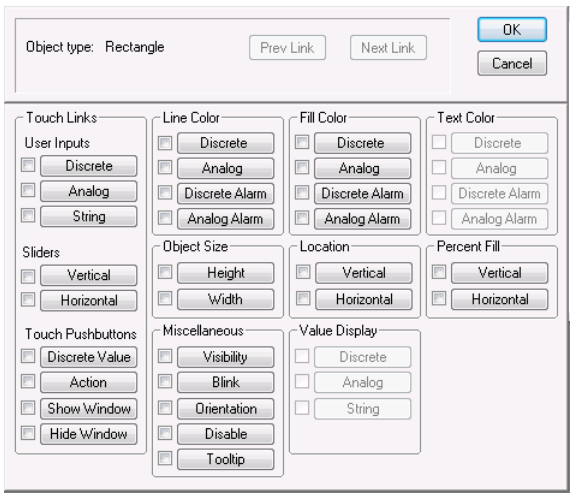
- 按单键或组合键。
- 按住单键或组合键。
- 释放单键或组合键。
- 将鼠标指针移到某个对象上。

动作脚本只能在对象自身的**动画链接选择**面板中配置。

要点：如果存在与动作脚本相同的单键或组合键来触发的键脚本，则执行动作脚本并忽略键脚本。

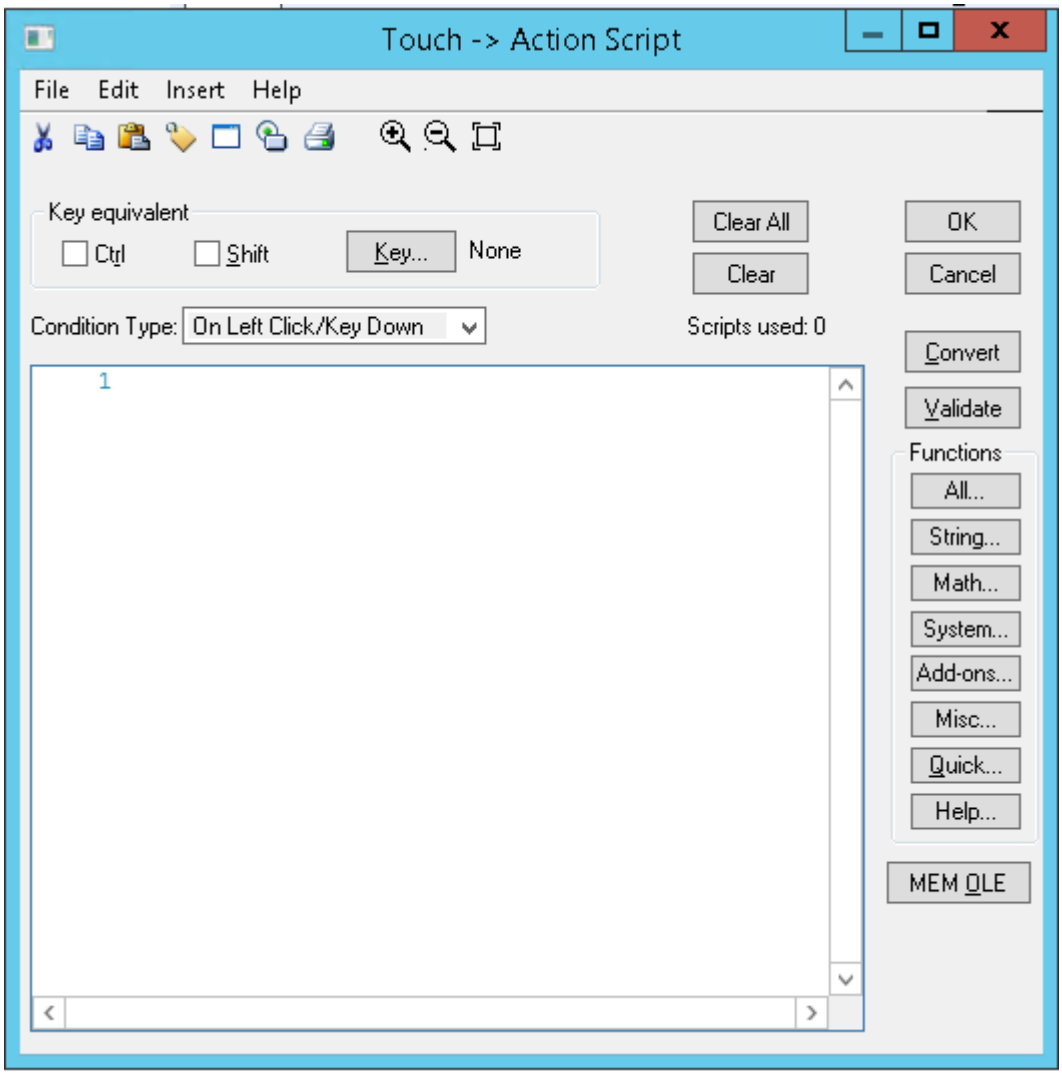
配置动作脚本

1. 双击图形对象。此时出现**动画链接选择**面板。



2. 选择操作。此时出现**触动 -> 动作脚本**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**条件类型**列表中，**执**行以下操作之一：

要将某个脚本配置成在此条件下**执**行	**选择**
按鼠标左键、特定的单键或组合键时执行一次	**鼠标左键/键按下时**
按住鼠标左键、特定的单键或组合键期间定期执行	**鼠标左键/键按下期间**
释放鼠标左键、特定的单键或组合键时执行一次	**鼠标左键/键放开时**
双击鼠标左键时执行一次	**双击鼠标左键时**
按鼠标右键时执行一次	**单击鼠标右键时**
按住鼠标右键期间定期执行	**鼠标右键按下期间**
释放鼠标右键时执行一次	**鼠标右键放开时**

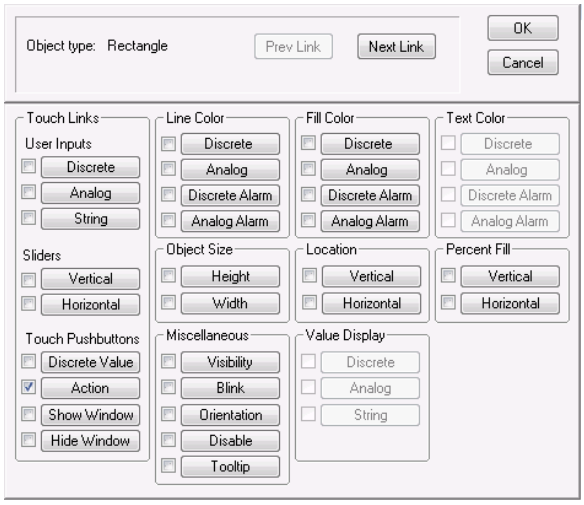
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要将某个脚本配置成在此条件下执行	选择
双击鼠标右键时执行一次	双击鼠标右键时
按鼠标中键时执行一次	鼠标中键单击时
按住鼠标中键期间定期执行	鼠标中键按下期间
释放鼠标中键时执行一次	鼠标中键放开时
双击鼠标中键时执行一次	鼠标中键双击时
将鼠标指针移到对象上时执行一次	鼠标悬停时

4. 如果选择鼠标左键/键按下时、鼠标左键/键按下期间，或鼠标左键/键放开时：
   a. 选择 键。此时出现选择键对话框。
   b. 选择一个键。
   c. 选择 Ctrl 与/或 Shift 复选框，以指定 Ctrl 与/或 Shift 键同所选键的组合。
5. 如果选择鼠标左键/键按下期间或鼠标右键按下期间，请在每框中输入 1 到 360000 毫秒之间的一个时间间隔。
6. 如果选择鼠标悬停时，请在之后框中，输入 1 到 360000 之间的毫秒数，以确定在鼠标移到对象上之后、脚本执行之前的时间。
7. 在窗口中输入脚本。
8. 选择确定。

删除与某个 InTouch 图形对象关联的所有动作脚本

1. 双击图形对象。此时出现对象属性面板。



2. 选择以清除动作复选框。此时动作脚本将不在运行时执行。如果选择动作按钮，则会打开编辑器，并显示为任何对象保存的上一个动作脚本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

删除单个动作脚本

1. 双击包含要删除的动作脚本的图形对象。此时出现对象属性面板。
2. 选择动作按钮。此时出现触动 -> 动作脚本对话框。
3. 在条件类型列表中，选择脚本触发器。
4. 在编辑菜单上，选择清除。此时该脚本会从该主要部分中清除，关联的脚本也被删除掉。

配置 ActiveX 事件脚本

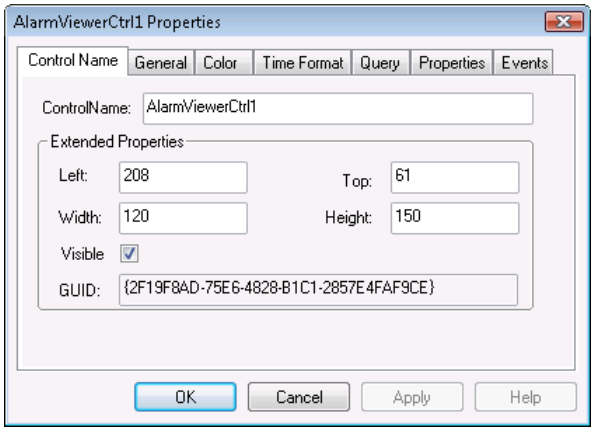
使用 ActiveX 事件脚本在发生 ActiveX 事件时运行某个脚本。根据具体的 ActiveX 控件，这些事件可以包括：

- 启动 ActiveX 控件：启动
- 关闭 ActiveX 控件：关闭
- 用户单击 ActiveX 控件：单击
- 用户双击 ActiveX 控件：双击

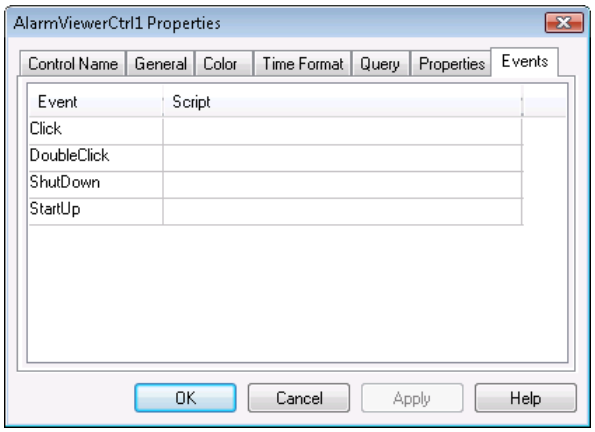
ActiveX 事件脚本通过名称来确定。缺省条件下，InTouch HMI 自动添加脚本与之关联的控件名与事件。例如：MyActiveXScript (AlarmViewerCtrl1::Click)。

配置新的 ActiveX 事件脚本

1. 双击要配置的 ActiveX 控件。此时出现该 ActiveX 控件的属性对话框。



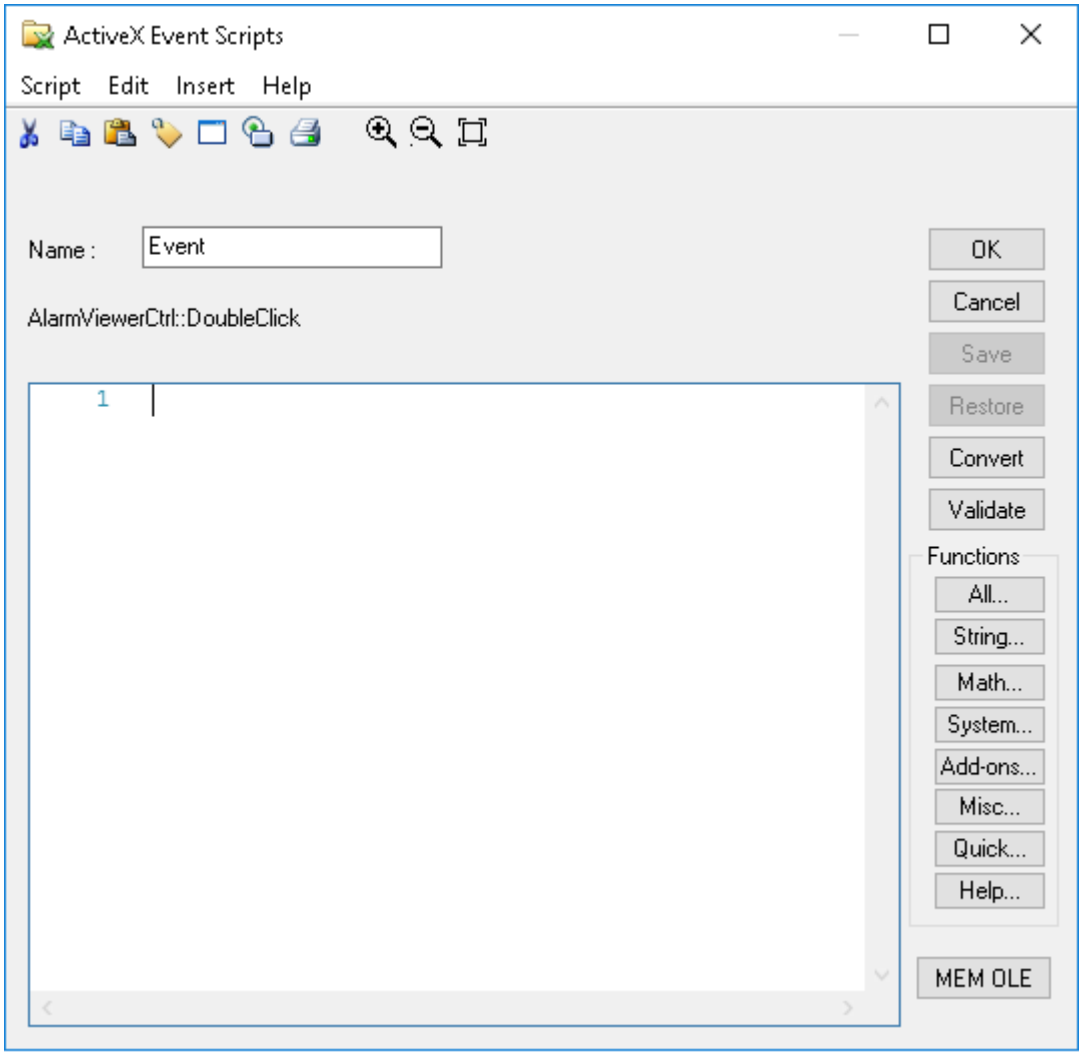
2. 选择事件选项卡。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 选择某个事件，如 Click（单击）、DoubleClick（双击）、ShutDown（关闭），或 StartUp（启动）。
4. 选择该事件的脚本单元。此时出现方括号。
5. 为事件脚本输入新的名称，然后选择确定。出现消息时，选择确定以创建一个新脚本。

   此时出现 **ActiveX** 事件脚本对话框。



1. 在名称框中，可以更改 ActiveX 事件脚本名。
2. 在窗口中输入脚本。
3. 选择确定。

编辑现有的 ActiveX 事件脚本

1. 在脚本窗格中，展开 **ActiveX** 事件，右键单击 ActiveX 脚本名，然后选择编辑。此时出现 **ActiveX** 事件脚本对话框。
2. 对脚本进行必要的更改，然后选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

删除现有的 ActiveX 事件脚本

1. 确保没有 ActiveX 控件正在使用要删除的 ActiveX 事件脚本。如果有 ActiveX 控件正在使用该脚本，请先执行以下操作：
   a. 从可能在使用 ActiveX 事件脚本的每个 ActiveX 控件的事件面板中，删除该 ActiveX 事件脚本引用。
   b. 关闭所有窗口并更新使用计数。
2. 在脚本窗格中，展开 **ActiveX 事件**，右键单击 ActiveX 脚本名，然后选择删除。出现消息时，选择是。此时该 ActiveX 事件脚本已删除。

在运行时暂停脚本执行

缺省条件下，启动 WindowViewer 时运行逻辑并执行同步脚本。在运行时，可以通过停止逻辑来暂停脚本执行。暂停之后可以恢复脚本执行。

在运行时从菜单中暂停脚本执行

- 在逻辑菜单上，选择停止逻辑。此时同步脚本停止运行。异步脚本继续运行，但不会启动新的异步脚本。

在运行时使用脚本暂停脚本执行

- 将值 0 写入离散系统标记 $LogicRunning。此时同步脚本停止运行。异步脚本继续运行，但不会启动新的异步脚本。

在运行时恢复脚本执行

- 在逻辑菜单上，选择开始逻辑。此时脚本恢复执行。

在运行时使用脚本恢复脚本执行

- 将值 1 写入离散系统标记 $LogicRunning。$LogicRunning 系统标记必须包含在暂停逻辑时所执行的异步脚本中。

$LogicRunning 系统标记

此系统标记监视与/或控制脚本的运行。

用法

$LogicRunning

附注

将值设为 1 使脚本开始运行。将值设为 0 使脚本停止运行。

此系统标记相当于在 WindowViewer 的逻辑菜单上选择开始逻辑或停止逻辑。

您无法停止当前正在运行的异步脚本。不过可以阻止运行新的脚本。

数据类型

离散（可读写）

脚本语言

使用 InTouch HMI 脚本语言编写脚本时可以使用这些概念、技术及语法规则。

- 基本语法规则。请参阅基本语法规则。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 调用预定义或自定义的函数。请参阅调用标准函数和调用自定义函数 (QuickFunction)。
- 使用赋值语句与各种运算符。请参阅赋值语句与运算符。
- 使用条件语句。请参阅使用条件程序分支结构。
- 使用循环。请参阅使用程序循环。
- 使用局部变量。请参阅使用局部变量。

如需有关脚本编辑器常规操作的详细信息，请参阅创建与编辑脚本。

如需有关各种类型脚本触发器的详细信息，请参阅脚本触发器。

如需标准脚本函数的参考，请参阅内置函数。

基本语法规则

基本语法规则涉及 InTouch HMI 脚本语言的以下这些概念：

- 子程序
- 语句
- 缩进
- 注释
- 标记引用
- 数据值
- 值表达式
- 语法验证

子程序

在相同的脚本中没有单独子程序的概念，如 Visual Basic 中的“子”过程。要使用多个子程序构建一个脚本，必须为每个子程序都创建一个自定义的 QuickFunction。请参阅自定义脚本函数。

语句

- 语句可以是赋值、函数调用或控制结构。
- 脚本中的每个语句都必须以英文分号 (;) 结尾。
- 只要每个语句都以英文分号结尾，同一行中便可以有多个语句。
- 通过使用换行符（按 Enter 键），可以将一条语句分布到多个行上。

缩进

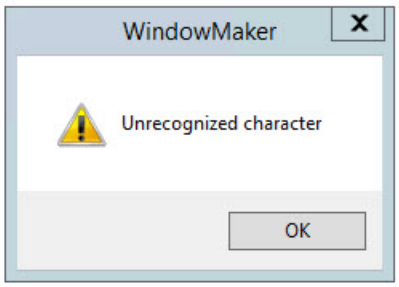
您可以按任何方式缩进脚本代码。缩进没有功能相关性。

注释

要将文本标记为注释，请使用大括号 { } 将它括起来。注释可以跨越多行。不支持嵌套注释。例如，在以下示例中，第一个 } 括号结束了注释，第二个 } 括号显示错误。

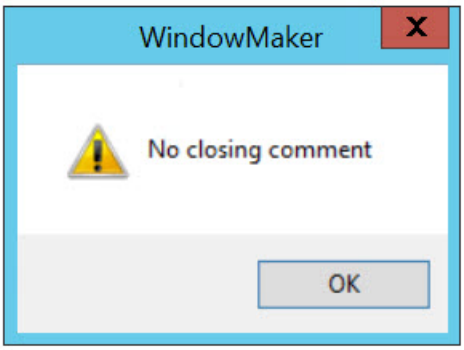
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
1  DIM msg AS Message;
2  {
3
4  Comments across multiple lines
5
6  {
7  Nested comments
8  }
9  }
10 msg = $DateString + ":" + $Operator;
```



如果不提供 } 括号，验证期间将显示错误。

```
1  DIM msg AS Message;
2  {
3
4  Comments across multiple lines
5
6
7  Nested comments
8
9  msg = $DateString + ":" + $Operator;
```



标记引用

有多种方式进行标记引用。

- 要引用本地<1>“</1>标记名字典<2>”</2>中定义的标记<3>，</3>只需简单地使用标记名即可<4>。</4>
- 要引用特定的点域，请使用常规引用格式 (`Tagname.Dotfield`)。
- 要引用远程节点上的数据项，请使用常规远程标记引用 (`AccessName:Item`)。
- 您还可以定义局部变量，它们的作用范围限制为当前脚本。请参阅使用局部变量。

数据值

- 您可以使用十进制或十六进制格式来指定整数值。例如，`255` 或 `0xFF`。
- 您可以使用十进制或科学计数法来指定浮点值。例如，`0.001` 或 `1E-3`。
- 要指定布尔值，请将数值 0 用作 FALSE、1 用作 TRUE。
- 要指定字符串值，请使用英文双引号将它括起来。例如：`"This is a string."`

值表达式

值表达式可以包含值、标记引用以及函数调用，所有这些都通过适当的运算符链接起来。请参阅赋值语句与运算符。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法验证

保存脚本时，“脚本编辑器”会自动检查语法是否正确。通过单击验证按钮，也可以手工开始这项验证。请参阅验证脚本的语法是否正确。

调用标准函数

标准函数是 InTouch HMI 中预定义好的。请参阅调用自定义函数 (QuickFunction)。

调用标准函数的语法

调用预定义脚本函数的语法取决于函数是否以及如何返回结果。

有些函数不返回任何结果；有些函数返回可选结果，可以赋值给标记或用在表达式中；有些函数返回的结果必须赋值给标记或用在表达式中。

要确定函数类型，可以看看函数描述。每个函数描述都有语法列表，显示该函数是否返回结果以及该结果是否为可选。

调用不返回结果的函数

- 仅在语句中使用函数名（与参数，如果有）。例如：
```
FunctionName(Parameters);
```

调用需要将结果赋值的函数

- 在脚本中任何可以使用值或相关数据类型的标记名的位置使用函数名（与参数，如果有）。例如，在赋值语句中：
```
ResultsTagname = FunctionName(Parameters);
```
  或者在嵌套的函数调用中，将它用作另一个标准函数的参数：
```
OtherStandardFunction(FunctionName(Parameters));
```

调用返回可选结果的函数

- 使用上述过程之一。

将参数传递给函数

标准预定义函数的参数通常通过*值*进行传递。这表示，只要表达式的求值结果属于参数所需的数据类型，便可以将任何有效的表达式作为参数进行传递。这类表达式可以包含值、标记引用及函数调用，所有这些都通过适当的运算符链接起来。如需有关表达式与运算符的详细信息，请参阅赋值语句与运算符。

脚本调用函数时，对表达式进行求值并将结果值传递给函数。

不过，有些函数需要将标记*引用*用作参数。例如：
```
RecipeSelectRecipe(Filename, RecipeName, Number);
```

在本例中，RecipeName 参数必须是标记引用（即，RecipeName 参数必须使用标记名）。即使该表达式通过求值可以得到有效的标记名，也不能使用传递字符串表达式的方式来代替。

注意：有些仅有一个参数的旧的预定义函数（例如，Ack() 函数）不遵循在括号中传递参数的标准语法。相反，参数使用空格同函数名进行分隔。如果对特定的函数有任何疑问，请查阅函数文档中的语法描述。

调用自定义函数 (QuickFunction)

调用自定义 QuickFunction 和调用预定义的标准函数略有不同：

- 关键字 CALL 必须加在 QuickFunction 名称之前。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- QuickFunction 返回的结果总是可选的；您可以使用它们，但并非必须如此。

调用不返回结果的 QuickFunction

- 在语句中使用函数名（以及参数，如果有），前面加上关键字 CALL。例如：
```
CALL QuickFunctionName(Parameters);
```

调用返回结果的 QuickFunction

- 执行以下任一操作：
  - 像不返回结果的那样调用 QuickFunction（请参阅上述过程）。
  - 在可以使用值或相关数据类型的标记名的脚本中的任何位置，使用函数名（以及参数，如果有），前面加上关键字 CALL。例如，在赋值语句中：
    ```
    ResultsTagname = CALL QuickFunctionName(Parameters);
    ```
    或者在嵌套的函数调用中，将它用作标准函数的参数：
    ```
    OtherStandardFunction(CALL FunctionName(Parameters));
    ```

注意：不能嵌套 QuickFunction 调用，让 QuickFunction 用作另一个 QuickFunction 的参数。例如，`Call QF1(Call QF2());` 是无效的语句。

将参数传递给 QuickFunction

QuickFunction 的参数总是按值传递。不能通过引用将参数传递给 QuickFunction。

只要表达式进行求值可以得到参数所需的数据类型，就可以将任何有效的表达式作为参数进行传递。这类表达式可以包含值、标记引用及函数调用，所有这些都通过适当的运算符链接起来。如需有关表达式与运算符的详细信息，请参阅赋值语句与运算符。脚本调用函数时，对表达式进行求值并将结果值传递给函数。

注意：不能嵌套 QuickFunction 调用，让 QuickFunction 用作另一个 QuickFunction 的参数。例如，`Call QF1(Call QF2());` 是无效的语句。

赋值语句与运算符

在脚本中，使用赋值语句将值写入标记。赋值语句的语法如下：
```
Tagname = ValueExpression;
```

执行此语句时，`ValueExpression` 写入由 `Tagname` 所引用的标记。`ValueExpression` 可以是数据类型与标记数据类型匹配的任何有效表达式。值表达式可以包含值、标记引用以及函数调用，所有这些都通过适当的运算符链接起来。

请参阅支持的运算符。

请参阅设置运算符的求值顺序。

请参阅有效表达式和无效表达式的示例。

支持的运算符

下表列出支持的所有运算符。如需有关使用特定运算符的详细信息，请参阅相关章节。

操作员	详细信息
+	加法或串联：+
-	减法：-

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

操作员	详细信息
*	乘法：*
/	除法：/
**	幂：**
MOD	模除：MOD
~	取补：~
SHL	左移位：SHL 与右移位：SHR
SHR	左移位：SHL 与右移位：SHR
和	位与：&
|	位或：|
^	位非：^
与	逻辑与：AND
或	逻辑或：OR
非	逻辑非：NOT
<	比较：<、>、<=、>=、==、<>
>	比较：<、>、<=、>=、==、<>
<=	比较：<、>、<=、>=、==、<>
>=	比较：<、>、<=、>=、==、<>
==	比较：<、>、<=、>=、==、<>
<>	比较：<、>、<=、>=、==、<>

注意：对于数值计算，总是选择运算数，使得计算结果仍在“实数”的数值范围内。否则结果将不正确。

加法或串联：+

将两个数值运算数相加，或串联两个字符串运算数。

有效运算数

加法<1>：</1>任何<2>“</2>整型<3>”</3>或<4>“</4>实型<5>”</5>值

串联<1>：</1>任何<2>“</2>消息<3>”</3>值

返回值的数据类型

加法<1>：“</1>整型<2>”</2>或<3>“</3>实型<4>”</4>

串联<1>：“</1>消息<2>”</2>

示例

```
MessageTag = "Setpoint value: " + Text(SetpointTag, "#.##");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

减法：-

用于两个运算数时，执行常规的数值减法。

有效运算数

任何“整型”或“实型”值

返回值的数据类型

整型或实型

示例

在此例中，如果 TemperatureSetpoint 为 70，则在脚本执行后，TemperatureSetpoint 为 65。

```
TemperatureSetpoint = TemperatureSetpoint - 5;
```

乘法：*

常规的数值乘法。

有效运算数

任何“整型”或“实型”值

返回值的数据类型

“整型”或“实型”

除法：/

常规的数值除法。如果试图在运行时除以 0，则 0 作为结果返回。

有效运算数

任何“整型”或“实型”值

返回值的数据类型

“整型”或“实型”

幂：**

将左侧运算数（基数）升为右侧运算数的幂（幂）。

有效运算数

“整型”或“实型”值。不能结合基数为 0、幂为负数，或基数为负、幂为小数的情况。在这些情况下，0 作为结果返回。

返回值的数据类型

“整型”或“实型”

示例

`8 ** (1/3)` 返回 2（8 的三次方根）

模除：MOD

返回两个整型值相除所得的余数。

有效运算数

任何“整型”值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

返回值的数据类型

整型

示例

`37 MOD 4` 返回 1

取补：~

返回整型值的补数。即，将每个 0 位转换为 1 位，反之亦然。

有效运算数

任何“整型”值。

返回值的数据类型

整型

左移位：SHL 与右移位：SHR

将整数值的二进制表达式向左或向右移动指定的位数。左侧的运算数是要移动的值，右侧的运算数是位数。从字中移出的位会丢失。移走后腾空的位设为 0。

有效运算数

任何“整型”值。

返回值的数据类型

整型

示例

初始标记值 5 时，反复执行 `IntTag = IntTag SHL 1;` 会得到以下结果：

迭代	二进制模式	标记值
初始值	`0[...]00000101`	5
执行 1 次	`0[...]00001010`	10
执行 2 次	`0[...]00010100`	20

位与：&

逐位比较两个整数的二进制表示法，然后按照下表返回结果：

第一个运算数中的位	第二个运算数中的位	结果中的位
0	0	0
0	1	0
1	0	0
1	1	1

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您可以使用这个运算符快速“屏蔽掉”（设为 0）位模式中特定的部分。例如，以下语句屏蔽掉 IntTag 标记的高 24 位：

```
IntTag = IntTag & 255;
```

如表中所示，如果运算数中的某个位是 0，则结果位总是 0。在 255 的二进制表示法中，只有低 8 位是 1，因此剩余的 24 个 0 位会使得结果中所有相应的位都设为 0。

有效运算数

任何“整型”值。

返回值的数据类型

整型

位或：|

逐位比较两个整数的二进制表示法，然后按照下表返回结果：

第一个运算数中的位	第二个运算数中的位	结果中的位
0	0	0
0	1	1
1	0	1
1	1	1

此运算也称为“同或”。

有效运算数

任何“整型”值。

返回值的数据类型

整型

位非：^

逐位比较两个整数的二进制表示法，然后按照下表返回结果：

第一个运算数中的位	第二个运算数中的位	结果中的位
0	0	0
0	1	1
1	0	1
1	1	0

此运算也称为“异或”。

有效运算数

任何“整型”值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

返回值的数据类型

整型

逻辑与：AND

如果两个离散运算数均为 TRUE（真），则返回 TRUE（真）；反之则返回 FALSE（假）。此运算符的真值表如下：

p	q	p AND q
F	F	F
F	T	F
T	F	F
T	T	T

有效运算数

任何<1>“</1>离散<2>”</2>值<3>。</3>

返回值的数据类型

离散

逻辑或：OR

如果离散运算数中至少有一个为 TRUE（真），则返回 TRUE（真）；反之则返回 FALSE（假）。此运算符的真值表如下：

p	q	p OR q
F	F	F
F	T	T
T	F	T
T	T	T

有效运算数

任何<1>“</1>离散<2>”</2>值<3>。</3>

返回值的数据类型

离散

逻辑非：NOT

如果离散运算数为 FALSE（假），则返回 TRUE（真），反之亦然。此运算符的真值表如下：

p	NOT p
F	T
T	F

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

有效运算数

任何<1>“</1>离散<2>”</2>值<3>。</3>

返回值的数据类型

离散

比较：<、>、<=、>=、==、<>

这些运算符将两个值进行比较，如果符合运算符指定的条件，则返回 TRUE（真）。运算数可以是任何数据类型。对于字符串运算数，以不区分大小写基于字母表的顺序进行比较，b 大于 a，c 大于 b，等等。对于离散运算数，将 TRUE 视作大于 FALSE。下表列出所有比较运算符及其条件：

运算	示例	条件
小于	a < b	a 小于 b
大于	a > b	a 大于 b
小于等于	a <= b	a 小于或等于 b
大于等于	a >= b	a 大于或等于 b
等于	a == b	a 等于 b
不等于	a <> b	a 不等于 b

有效运算数

任何数据类型值（两个值的数据类型必须相同）。

返回值的数据类型

离散型

设置运算符的求值顺序

在任何表达式中，都可以使用括号来强制运算符以某种顺序进行求值。这同任何数学表达式中的工作方式相同。如果不使用括号，则表达式根据运算符的缺省优先规则进行求值。最高优先级的运算最先执行，第二高优先级的运算随后执行，依此类推。

下表显示每个运算符的优先级。同一行上的运算符具有相同的优先级。

-、NOT、~	最高优先级
**	
*、/、MOD	
+、-	
SHL、SHR	
<、>、<=、>=	
==、<>	
和	
^	

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

|	
与	
或	
=	最低优先级

隐式数据类型转换

InTouch HMI 脚本语言在赋值语句中某些特定的数据类型之间提供隐式数值转换。不过，这可能导致异常的结果，因此应小心使用此功能。

下表显示将某种类型的值指定给不同类型的标记时会发生什么。

期望的数据类型	使用的数据类型	附注
离散	整型	值 0 诠释为 FALSE。任何其它值诠释为 TRUE。
离散	实型	值 0 诠释为 FALSE。任何其它值诠释为 TRUE。
整型	离散	FALSE 的值转换为 0。TRUE 的值转换为 1。
整型	实型	只使用小数点之前的值。所有小数位均丢弃。
实型	离散	FALSE 的值转换为 0。TRUE 的值转换为 1。
实型	整型	值保留下来而不进行任何更改。

如需有关使用脚本函数在其它数据类型之间进行转换的信息，请参阅转换数据类型。

有效表达式和无效表达式的示例

下表显示一些有效的表达式，以及表达式的结果和结果的数据类型。

表达式	结果的数据类型	结果
`37 MOD 4`	整型	1
`37 MOD 4 == 1`	离散	True
`NOT (37 MOD 4 == 1)`	离散	False
`InfoAppActive(InfoAppTitle("xyz")) == 1`	离散	如果 “xyz” 进程正在运行，则为 TRUE
`"Batch " + Text(IntTag, "000")`	消息	如果 `IntTag` 的值为 10，则为 “Batch 010”

下表显示一些无效的表达式及其无效的原因。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

表达式	问题
`NOT (37 MOD 4)`	NOT 要求使用离散运算数。
`NOT 37 MOD 4 == 1`	NOT 的优先级比其它运算符的更高，因此 InTouch HMI 会试图给整数值 37 应用 NOT，而不是比较运算的离散结果。
`"Batch " + IntTag`	使用 + 运算符串联字符串时，两个运算数都必须是字符串。

使用条件程序分支结构

基于满足某些特定的条件，可以动态控制脚本的执行路径。为此，InTouch HMI 支持 IF-THEN-ELSE 控制结构。

IF-THEN-ELSE 控制结构的基本语法如下：

语法

```
IF Condition THEN
   ... 语句与/或另一个 IF-THEN-ELSE 结构
[ELSE
   ... 语句与/或另一个 IF-THEN-ELSE 结构]
ENDIF;
```

使用 IF-THEN-ELSE 结构时请记住以下规则：

- IF-THEN-ELSE 结构可以嵌套，都在 THEN 部分与 ELSE 部分进行。
- 对于每个 IF 语句，必有一个 ENDIF 结束语句。在同一嵌套级别中，ENDIF 语句总是应用于前面最接近的 IF 语句。
- Condition 必须是有效的离散表达式。如果 Condition 为 TRUE，则执行 THEN 部分。如果 Condition 为 FALSE，则执行 ELSE 部分。
- ELSE 部分是可选的。
- 其它一些编程语言可以在 IF-THEN-ELSE 结构的相同层级上检查多个条件，并有一个常规的 ELSE 部分，如果全部条件赋 FALSE 值，则执行这个部分。（Visual Basic 中的 If-ElseIf-Else 结构就是这样的一个例子）。在 InTouch HMI 中不能这样。对于要检查的每个条件，都必须开始一个新的 IF-THEN-ELSE 结构。因此，要让单个代码段用作所有条件的 ELSE 代码，必须将它放在 IF-THEN-ELSE 结构的最后一个嵌套级别的 ELSE 部分。

简单条件结构

以下脚本显示一个简单的条件结构。如果 SuccessTag 为 TRUE，则打开 "Success" 窗口，否则打开 "Failure" 窗口。

```
IF SuccessTag == 1 THEN
   Show "Success";
ELSE
   Show "Failure";
ENDIF;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

嵌套条件结构

以下脚本显示如何检查多个条件并有一个常规 ELSE 部分，如果不满足任何条件则执行此部分代码。

```
IF ChoiceTag == 1 THEN
   Show "Procedure 1";
ELSE
   IF ChoiceTag == 2 THEN
      Show "Procedure 2";
   ELSE
      IF ChoiceTag == 3 THEN
         Show "Procedure 3";
      ELSE
         Show "Default Procedure";
      ENDIF;
   ENDIF;
ENDIF;
```

无效脚本示例（缺少 ENDIF）

如果熟悉 Visual Basic，可以尝试编写一个如下所示的简单 IF 语句：

```
IF OpenThisWindow == 1 THEN Show "This Window";
```

这在 InTouch HMI 中无效。对于每个 IF 语句，必有一个 ENDIF 结束语句。

Invalid scripting example (incorrect nesting)

如果熟悉一种语言（如 Visual Basic），可能希望编写如下所示的带多个条件与一个缺省条件的条件结构：

```
IF ChoiceTag == 1 THEN
   Show "Procedure 1";
ELSE IF ChoiceTag == 2 THEN
   Show "Procedure 2";
ELSE IF ChoiceTag == 3 THEN
   Show "Procedure 3";
ELSE
   Show "Default Procedure";
ENDIF;
```

这在 InTouch HMI 中无效。每个 IF 都开始一个新的嵌套级别，且必须有相应的 ENDIF 语句。如需本示例的正确版本，请参阅嵌套条件结构。

使用程序循环

循环可以反复执行一段代码。InTouch HMI 仅支持 FOR 循环。FOR 循环按所监视的每次循环迭代所产生的递增或递减的数值循环变量值来进行。循环一直执行到循环变量值达到固定的极限。

语法

```
FOR LoopTag = StartExpression TO EndExpression [STEP ChangeExpression]
... 语句或另一个 FOR 循环 ...
NEXT;
```

- StartExpression, EndExpression 与 ChangeExpression 共同定义迭代次数。
- StartExpression 设置循环范围的开始值。EndExpression 设置循环范围的结束值。
- STEP ChangeExpression 可选择设置每次循环迭代过程中循环标记所递增或递减的值；如果不指定此值，则使用缺省值 1。

执行 FOR 循环时，InTouch HMI：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1. 将 LoopTag 设置为 StartExpression 的值。
2. 测试 LoopTag 是否大于 EndExpression。如果是，InTouch HMI 退出循环。（如果 ChangeExpression 为负，InTouch HMI 测试 LoopTag 是否小于 EndExpression）。
3. 执行循环内的语句。
4. 按 ChangeExpression 的值（除非另外指定，否则设为 1）递增 LoopTag。
5. 重复步骤 2 到 4。

使用 FOR 循环时请记住以下规则：

- FOR 循环可以嵌套。最大嵌套级数取决于可用的内存与系统资源。
- 对于每个 FOR 语句，必有一个 ENDIF 结束语句。在同一嵌套级别中，NEXT 语句总是应用于前面最接近的 FOR 语句。
- LoopTag 必须是数值标记（或局部变量）。
- StartExpression、EndExpression 以及 ChangeExpression 必须是赋值为数值结果的有效表达式。
- 如果 ChangeExpression 为正，EndExpression 必须大于 StartExpression；如果 ChangeExpression 为负，StartExpression 必须大于 EndExpression。否则循环不会开始。
- 要退出循环，请使用 EXIT FOR 语句。如需详细信息，请参阅强制结束循环。
- 循环有时间限制。请参阅循环执行的时间限制。

警示：循环的执行会影响其它运行时进程。如需详细信息，请参阅循环对其它运行时进程的影响。

强制结束循环

您可以通过调用以下语句在任何时间退出循环：

```
EXIT FOR;
```

此语句使脚本继续执行紧接着循环 NEXT 语句之后的语句。

示例

下面的代码段使用循环将大量的虚拟记录插入数据库表。如果插入记录时发生错误，则放弃循环以防止产生更多错误。

```
FOR Counter = 1 TO 1000
   ResultCode = SQLInsert(ConnectionID, "BatchDetails", "BindList1");
   IF ResultCode <> 0 THEN
      LogMessage("Error creating records! Aborting...");
      EXIT FOR;
   ENDIF;
NEXT;
```

循环对其它运行时进程的影响

执行 FOR 循环时，WindowViewer 中的所有其它运行时进程都暂停。这包括以下范围：

- 屏幕更新（动画链接、值显示、趋势等）。这表示，到循环完成之前不会发生任何移动，因此不能给动画对象使用 FOR 循环。
- I/O 通讯。例如，如果修改 FOR 循环中 I/O 标记的值，则只有最终的迭代后面的值才会写入 I/O 设备。
- 其它脚本，包括异步 QuickFunction。

您可以通过将 FOR 循环放入异步 QuickFunction 来避免暂停其它运行时进程。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

循环执行的时间限制

为避免无限循环，这里设置了一个时间限制，FOR 循环必须在这个时间限制内完成执行。如果循环未在这个时间跨度之后完成，WindowViewer 会自动终止循环它，并将一条关于终止的消息写入 Log Viewer。

缺省的时间限制为 5 秒。通过向应用程序目录中的 intouch.ini 文件添加下面这行，可以对它进行自定义：

```
LoopTimeout=x
```

将 x 替换成以秒为单位的时间限制。

注意：时间限制仅在循环的 NEXT 语句中检查。因此，循环的第一次迭代总是会执行，即便它花费的时间比时间限制更长。

循环的示例

以下脚本通过一个简单的循环与一个间接标记，使用 0 值来重新初始化 100 个标记（Tag001 到 Tag100）。

```
DIM Counter AS INTEGER;
FOR Counter = 1 TO 100
   IndirectInteger.Name = "Tag" + Text(Counter, "000");
   IndirectInteger.Value = 0;
NEXT;
```

以下脚本通过两个嵌套的循环与一个间接标记，使用 0 来重新初始化 1000 个标记（Line01_Tag001 到 Line10_Tag100）。

```
DIM LineCounter AS INTEGER;
DIM TagCounter AS INTEGER;
FOR LineCounter = 1 TO 10
   FOR TagCounter = 1 TO 100
      IndirectInteger.Name = "Line" + Text(LineCounter, "00") + "_Tag" + Text(TagCounter,
      "000");
      IndirectInteger.Value = 0;
   NEXT;
NEXT;
```

使用局部变量

您可以在脚本中声明多个局部变量，以存储临时或中间结果。这可以提高性能并帮助维持低标记计数。您可以在脚本中像使用标记名那样使用局部变量。不过有些不同之处：

- 局部变量仅存在于声明它们的脚本的范围内。在脚本执行完毕时，它们的值会丢失。它们不能由应用程序中的任何其它脚本引用。
- 局部变量没有点域。
- 局部变量不计入标记计数。

可以在脚本中使用局部变量之前，必须先声明它；否则会将引用视作标记名。请参阅声明局部变量。

您可以声明与标记使用相同名称的局部变量。请参阅局部变量与标记之间的命名冲突。

声明局部变量

您可以在脚本中的任何位置声明局部变量，只要在第一次使用它们之前进行声明即可。要声明局部变量，请使用以下语句：

```
DIM LocVarName [AS DataType];
```

LocVarName 为局部变量的名称。名称必须符合标记名的命名惯例。如需详细信息，请参阅标记名惯例。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

DataType 是局部变量的数据类型。有效值是离散、整型、实型以及消息。如果不指定此选项，则缺省使用整型。

对于要声明的每个局部变量，必须使用一个单独的 DIM 语句。

您可以声明任何数量的局部变量。数量仅受可用内存的限制。

示例

要声明整型变量：

```
DIM MyLocalIntVar AS Integer;
```

要声明多个实型变量：

```
DIM MyLocalRealVar1 AS Real;
DIM MyLocalRealVar2 AS Real;
```

以下语句*无效*

```
DIM MyLocalRealVar1, MyLocalRealVar2 AS Real;
```

局部变量与标记之间的命名冲突

您可以使用与现有标记相同的名称来声明局部变量。不过，在脚本中引用该名称时，局部变量总是比标记优先。例如，假设有一个现有的“整型”标记 “iTag”，并运行以下脚本：

```
DIM iTag as Integer;
iTag = 20;
```

在这种情形中，赋值语句仅将一个值写入局部变量。同名标记的值保持不变。

自定义脚本函数

InTouch HMI QuickFunction 是其它环境中可能被称为宏、子程序或过程的脚本。

关于 QuickFunction

QuickFunction 是可以从其它脚本与动画链接中调用的脚本。QuickFunction 的主要优点是减少重复代码。

您可以将值传递给可以使用这些值并返回结果的 QuickFunction。

QuickFunction 可以异步运行。与其它脚本不同，它们可以在后台运行，而不会干扰主程序流程。异步运行的 QuickFunction 可用于耗时的操作，如 SQL 数据库调用。

备注：仔细设计 QuickFunction 及其参数，因为如果希望修改 QuickFunction 中的参数，必须先从使用 QuickFunction 的每个脚本中删除对该 QuickFunction 的所有调用。进行更改之后，必须再将 QuickFunction 调用添加回这些脚本中。请参阅配置 QuickFunction 中的备注。

QuickFunction 有三个基本部分：

- 名称
- 参数（可选）
- 带可选返回值的脚本主体

QuickFunction 通过在动画链接或其它脚本中使用 CALL 函数来执行。请参阅调用 QuickFunction。

配置 QuickFunction

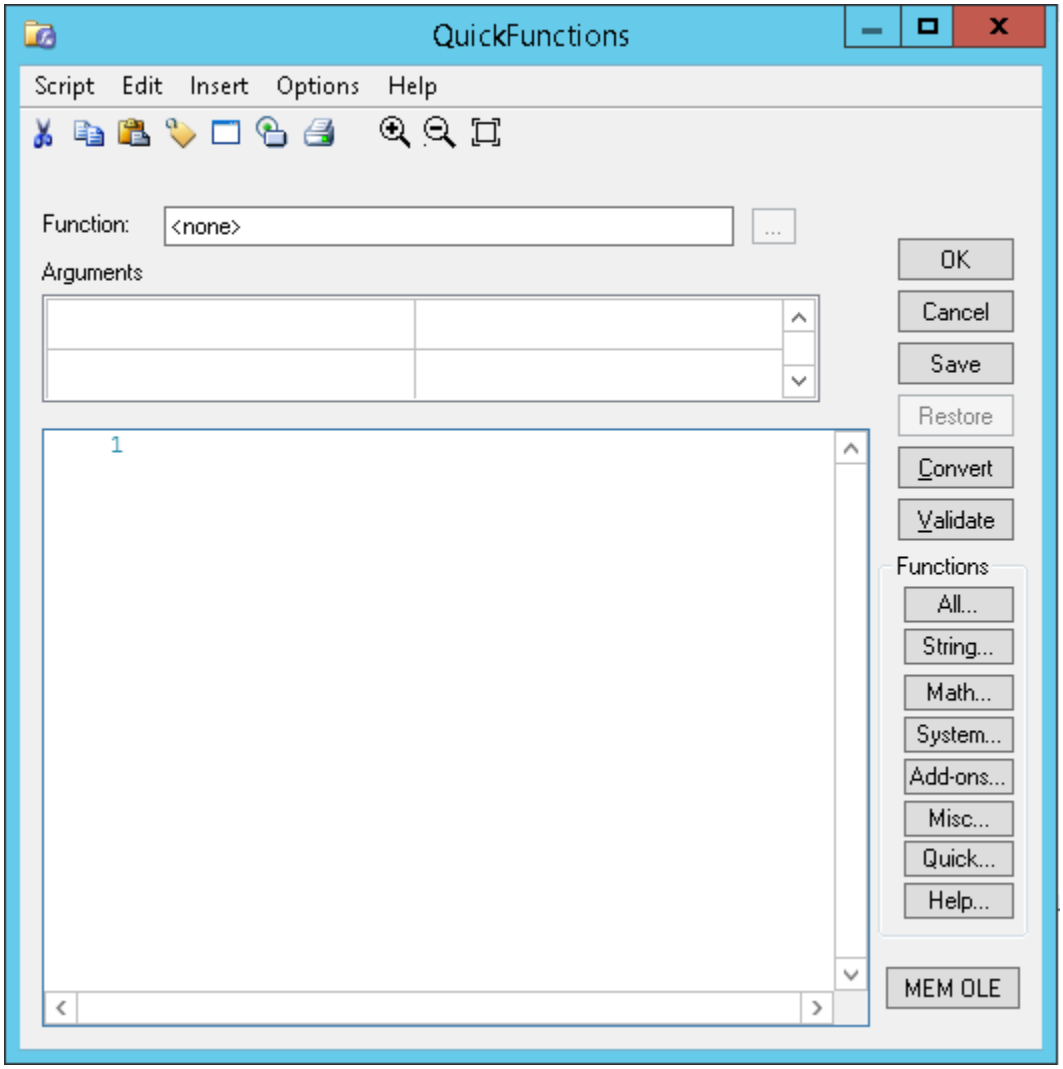
您可以创建、修改或删除 QuickFunction。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建 QuickFunction

1. 在脚本窗格中，右键单击 **QuickFunction**，然后选择新建。

   此时出现 **QuickFunctions** 对话框。



2. 在函数框中，输入 QuickFunction 的名称。
3. 在参数区域中，为每个参数在左侧输入名称、在右侧输入数据类型。

   这些参数是局部变量，它们仅存在于定义它们的 QuickFunction 中。每个 QuickFunction 最多可包含 16 个参数。参数名可以包含 31 个字符，但不能是空格。参数名必须以英文字母开头。参数名必须是唯一的。
4. 在窗口中输入脚本。
5. 要使 QuickFunction 返回结果，请将以下内容添加到脚本中：RETURN 值

   值可以是值、局部变量、全局标记名或计算表达式。脚本在 RETURN 命令处终止，在调用函数处继续。
6. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

修改 QuickFunction

1. 在脚本窗格中，展开 **QuickFunction**，右键单击要修改的 QuickFunction，然后选择打开。此时出现 **QuickFunctions** 对话框。
2. 对脚本主体进行修改，然后选择确定。

注意：如果在 InTouch 应用程序中存在对 QuickFunction 的调用，则不能修改参数列表。您必须先删除这些调用，关闭所有 InTouch 窗口，并更新使用计数。

删除 QuickFunction

1. 删除对 QuickFunction 的所有调用，关闭所有 InTouch 窗口，并更新使用计数。
2. 在脚本窗格中，展开 **QuickFunction**，右键单击要删除的 QuickFunction，然后选择删除。出现消息时，选择是。

调用 QuickFunction

您可以配置脚本与动画链接，以调用 QuickFunction 并处理或显示可能的返回值。

如果参数值没有更改，则不会调用 QuickFunction。您可以使用 $second 作为参数，以确保至少每秒执行一次 QuickFunction。

如需有关详细信息，请参阅调用自定义函数 (QuickFunction)。

创建异步 QuickFunction

您可以定义 QuickFunction 同主程序流程异步（即，并行）运行。

创建异步 QuickFunction

1. 在<1>“</1>脚本编辑器<2>”</2>中<3>，</3>创建 QuickFunction<4>。</4>
2. 在选项菜单上，选择异步。

异步 QuickFunction 的限制

您无法：

- 从异步 QuickFunction 中返回某个值。
- 同时运行相同 QuickFunction 的多个实例。
- 在 QuickFunction 开始执行异步之后将它们停止。

您不应：

- 同时运行三个以上不同的异步 QuickFunction。同时运行三个以上 QuickFunction 会大幅降低系统性能。
- 将异步函数用作动画链接表达式的一部分，如工具提示。

检查是否有任何异步 QuickFunction 正在运行

您可以使用 IsAnyAsyncFunctionBusy() 函数检查是否有任何异步 QuickFunction 正在运行。您可以使用此函数使调用异步 QuickFunction 的 QuickScript 等待所有其它异步 QuickFunction 完成处理。

IsAnyAsyncFunctionBusy() 函数

返回一个离散值，指出是否有任何异步 QuickFunction 正在运行。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
result = IsAnyAsyncFunctionBusy(timeout)
```

参数

result
离散值，指出是否有任何异步 QuickFunction 正在运行，含义如下：

- 0 = 没有异步 QuickFunction 正在运行。
- 1 = 有异步 QuickFunction 正在运行。

timeout
检查是否有任何异步 QuickFunction 正在运行之前要等待的秒数。整数值、整型标记名或整型表达式。

示例

假设要使用异步 QuickFunction 连接到多个 SQL 数据库，并且知道要 2 分钟才能建立这些连接。

首先，执行异步 QuickFunction 连接到 SQL 数据库。

接着，使用 QuickScript 中的 IsAnyAsyncFunctionBusy(120) 函数，让 SQL 有足够的时间在该 QuickFunction 完成之前建立这些连接。

如果在 2 分钟之后这些连接仍未建立，且异步 QuickFunction 仍忙于建立这些连接，则函数 IsAnyAsyncFunctionBusy() 返回值 1 (true)。

现在可以显示一条错误消息，告诉操作员 SQL 连接不成功。

以下脚本实现了这种情形的处理：

```
IF IsAnyAsyncFunctionBusy(120) == 1 THEN
   SHOW "SQL Connection Error Dialog";
ENDIF;
```

停止运行异步 QuickFunction

在异步 QuickFunction 开始之后无法将其停止，但可以通过停止脚本逻辑来停止启动其它的异步 QuickFunction。这会影响 InTouch 应用程序中的所有 QuickScript。

如需有关停止脚本执行的详细信息，请参阅在运行时暂停脚本执行。

内置函数

InTouch QuickScript 函数可以在满足指定条件的情况下执行命令与逻辑运算。您可以单独使用 QuickScript 函数并在满足特定条件时执行它们，也可以在动画显示链接中使用它们。预定义的函数按照功能组来组织管理。选择某个组之后，您可以选择预定义的函数以插入到脚本中。在选择预定义的数学函数组之后，您可以使用所出现的选择函数对话框。选择函数时，预定义的函数放入脚本中的当前光标位置。

要点：此章包括只能在 32 位版本的 Windows 操作系统中工作的旧有 InTouch QuickScript 函数。这些函数不应包括在任何设计为在 64 位版本的 Windows 中运行的 InTouch QuickScript 中。此章中的备注确定了这些旧有的仅支持 32 位系统的函数。

在动画显示链接中强制更新

如果在动画链接中使用 QuickScript，则仅当标记与这些动画链接关联时，它们才会更新。只要标记的值发生改变，此标记便充当触发器。使用 $Second 或 $Minute 系统标记来更新动画链接是个较好的选择。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在动画显示链接中进行强制更新

1. 在对象属性窗口中打开动画链接。
2. 将触发器标记（例如 $Second）添加到计算公式中。例如：
   - 如果动画链接是实型或整型，则可以使用 $Second/$Second 乘以表达式。
   - 如果动画链接是字符串，则可以将 StringMid($TimeString, 0, 0) 添加到表达式。
   - 如果动画链接是离散型，则可以将 ($second.00 - $second.00) 添加到表达式。

数学计算

InTouch HMI 支持基本数学函数，这些函数可以用在脚本与动画链接中，例如具有以下用途的函数：

- 舍入与截断数字。
- 计算正弦和余弦。
- 计算对数与指数。
- 计算平方根。

舍入、截断及确定符号

在脚本中，可以使用以下函数来舍入数字、截断数字以及确定数字的符号：

使用对象	作用
Abs()	计算某个值或表达式的绝对值。
Int()	计算某个值或表达式的整数部分。
Round()	舍入某个值或表达式。
Sgn()	确定某个值或表达式的符号（负、正、零）。
Trunc()	返回值或表达式中小数点之前的部分。

Abs() 函数

返回指定的数字的绝对值。您可以使用此函数将负数转换为正数。

语法

```
result = Abs (number)
```

参数

number
数字、模拟标记名或数值表达式。

示例

`Abs(14)` 返回 14。

`Abs(-7.5)` 返回 7.5。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Int() 函数

返回小于（或等于）指定数字的整数。

语法

```
result = Int (number)
```

参数

number
数字、模拟标记名或数值表达式。

示例

`Int(4.7)` 返回 4。

`Int(-4.7)` 返回 -5。

备注：对于负实数，此函数将返回小于指定数字的整数。例如，Int(-4.7) 不是 -4，而是 -5。要返回整数部分，请使用 Trunc() 函数。请参阅 Trunc() 函数。

Round() 函数

将数字舍入到指定的精度。结果是实数。

语法

```
result = Round (number, precision)
```

参数

number
数字、模拟标记名或数值表达式。

precision
数字所要舍入的精度。它可以是数字、模拟标记名或数值表达式。

示例

`Round(4.3, 1)` 返回 4.

`Round(4.3, 0.01)` 返回 4.30.

`Round(4.5, 1)` 返回 5.

`Round(-4.5, 1)` 返回 -4.

`Round(106, 5)` 返回 105.

`Round(43.7, 0.5)` 返回 43.5.

Sgn() 函数

返回某个数字的符号。使用它来确定数字、标记名或表达式是负、正还是零。

语法

```
result = Sgn (number)
```

参数

number
数字、模拟标记名或数值表达式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

`Sgn(425)` 返回 1。

`Sgn(0)` 返回 0。

`Sgn(-37.3)` 返回 -1。

Trunc() 函数

返回数字被截断后的值。截断后的值是小数点前面的部分。使用它得到实数的整数部分。

语法

```
result = Trunc (number)
```

参数

number
数字、模拟标记名或数值表达式。

示例

`Trunc(4.3)` 返回 4。

`Trunc(-4.3)` 返回 -4。

备注：您也可以使用此函数来处理数字的小数部分。要返回指定数字的小数部分，请按以下方式使用 Trunc() 函数：
result = number - trunc(number);

使用三角函数

在脚本中，可以使用以下函数来进行三角计算。

使用对象	作用
Sin()	计算某个角度的正弦。
ArcSin()	计算某个值或表达式的反正弦。
Cos()	计算某个角度的余弦。
ArcCos()	计算某个值或表达式的反余弦。
Tan()	计算某个角度的正切。
ArcTan()	计算某个值或表达式的反正切。

注意：InTouch HMI 中的 QuickScript 三角函数使用以度 (0 - 360) 表示的角度。要使用弧度取代之，必须在将参数传递给函数之前，或从函数检索结果之后，执行相应的计算。

Sin() 函数

返回某个数字的正弦。对于三角函数而言，该数字是以度表示的角度。

语法

```
result = Sin (number)
```

参数

number

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数字、模拟标记名或数值表达式。

示例

`Sin(90)` 返回 1。

`Sin(0)` 返回 0。

`Sin(30)` 返回 0.5。

`100 * Sin (6 * $second)` 返回振幅为 100、周期为 1 分钟的正弦波。

ArcSin() 函数

返回某个数字的反正弦。它是 Sin() 函数的反函数。使用 ArcSin() 函数计算介于 -90 和 90 度之间的角度，其正弦等于该数字。

语法

```
result = ArcSin (number)
```

参数

number
-1 到 1 范围内的一个数字、模拟标记名或数值表达式。

示例

`ArcSin(1)` 返回 90。

`ArcSin(0)` 返回 0。

`ArcSin(0.5)` 返回 30。

Cos() 函数

返回某个数字的余弦。对于三角函数而言，该数字是以度表示的角度。

语法

```
result = Cos (number)
```

参数

number
数字、模拟标记名或数值表达式。

示例

`Cos(90)` 返回 0。

`Cos(0)` 返回 1。

`Cos(60)` 返回 0.5。

`20 + 50 * Cos(6 * $second)` 产生振幅为 50、周期为一分钟围绕 20 振荡的正弦波。

ArcCos() 函数

返回数字的反余弦。它是 Cos() 函数的反函数。使用 ArcCos() 函数计算介于 0 到 180 度之间的角度，其余弦等于该数字。

语法

```
result = ArcCos (number)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

number
-1 到 1 范围内的一个数字、模拟标记名或数值表达式。

示例

`ArcCos(1)` 返回 0。

`ArcCos(-0.5)` 返回 120。

Tan() 函数

返回指定数字的正切。对于三角函数而言，该数字是以度表示的角度。

语法

```
result = Tan (number)
```

参数

number
数字、模拟标记名或数值表达式。

示例

`Tan(45)` 返回 1。

`Tan(0)` 返回 0。

ArcTan() 函数

返回某个数字的反正切。它是 Tan() 函数的反函数。使用 ArcTan() 函数计算其正切等于该数字的角度。

语法

```
result = ArcTan (number)
```

参数

number
数字、模拟标记名或数值表达式。

示例

`ArcTan(1)` 返回 45。

`ArcTan(0)` 返回 0。

返回 Pi 的值。

在脚本中，可以使用 Pi() 函数在数学计算中使用常量 Pi。Pi() 函数精确到小数点后 7 位。

语法

```
result = Pi ()
```

示例

`Pi()` 返回 3.1415927。

计算对数

在脚本中，可以使用以下函数来运行使用对数和指数函数的计算。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用对象	作用
Log()	计算某个值或表达式的自然对数。
Exp()	计算某个值或表达式的指数。
LogN()	计算以另一个值或表达式为底的值或表达式的对数。

Log() 函数

返回指定正数的自然对数。这是 Exp() 函数的反函数。

备注：0 与负数的自然对数未定义。如果将 0 或负数传递给 Log() 函数，则返回 -99.0000000。

语法

```
result = Log (number)
```

参数

number
正数、模拟标记名或数值表达式。

示例

`Log(100)` 返回 4.6051702。

`Log(1)` 返回 0。

Exp() 函数

返回指定数字的指数。这是 Log() 函数的反函数，它等于 e 的幂。

备注：如果将 -88.72 与 88.72 范围之外的值传递给 Exp() 函数，则返回 -99.0000000。

语法

```
result = Exp (number)
```

参数

number
-88.72 到 88.72 范围内的一个数字、模拟标记名或数值表达式。

示例

`Exp(1)` 返回 2.7182818。

`Exp(0)` 返回 1。

LogN() 函数

返回指定底的正数的对数。这是底的对数次幂的反函数。

示例

语法

```
result = LogN (number, base)
```

参数

number

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

正数、模拟标记名或数值表达式。

base
不等于 1 的正数、模拟标记名或表达式。

示例

`LogN(8,2)` 返回 3。

`LogN(num,btag)` 返回以 btag 为底的 num 的对数。

备注：如果将无效的参数传递给 LogN() 函数，则返回 -99.0000000。

计算平方根

在脚本中，可以使用 Sqrt() 函数计算指定的非负数的平方根。

注意：如果将负值传递给 Sqrt() 函数，则返回 -99.0000000。

语法

```
result = Sqrt (number)
```

参数

数字
非负的数字、模拟标记名或数值表达式。

示例

`Sqrt(36)` 返回 6。

`Sqrt(perftag)` 返回标记名 perftag 的值的平方根。

字符串运算

您可以在脚本与动画链接中使用许多基本字符串函数。您可以使用这些函数：

- 返回字符串的某些部分。
- 更改字符串大小写。
- 给字符串删除与添加空格。
- 处理字符串中的 ASCII 值。
- 在字符串中搜索并替换。
- 相互比较字符串。
- 返回有关字符串的其它信息，如长度。

返回字符串的某些部分

在脚本中，可以使用 StringLeft()、StringMid() 以及 StringRight() 函数返回字符串的某些部分。

StringLeft() 函数

从字符串的开头起返回指定数量的字符。

语法

```
result = StringLeft (string, length)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

string
文本、消息标记名或字符串表达式。

length
要返回的字符数。数字、模拟标记名或数值表达式。

示例

`StringLeft("Hello World",5)` 返回 “Hello”。

`StringLeft("Hello World",20)` 返回 “Hello World”。

`StringLeft("Hello World",0)` 返回 “Hello World”。

备注：如果将 0 作为 length 传递给 StringLeft() 函数，则返回整个字符串。

StringRight() 函数

从字符串的末尾起返回指定数量的字符。

语法

```
result = StringRight (string, length)
```

参数

string
文本、消息标记名或字符串表达式。

length
要返回的字符数。数字、模拟标记名或数值表达式。

示例

`StringRight("Hello World",5)` 返回 “World”。

`StringRight("Hello World",20)` 返回 “Hello World”。

`StringRight("Hello World",0)` 返回 “Hello World”。

备注：如果将 0 作为 length 传递给 StringRight() 函数，则返回整个字符串。

StringMid() 函数

返回字符串的一部分。您可以指定开始点与要返回的字符数。

语法

```
result = StringMid (string, startpos, length)
```

参数

string
文本、消息标记名或字符串表达式。

startpos
字符串中的开始位置。数字、模拟标记名或数值表达式。

length
要返回的字符数。数字、模拟标记名或数值表达式。

示例

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

`StringMid("Hello World",5,4)` 返回 "o Wo”。

`StringMid("Hello World",7,50)` 返回 “World”。

`StringMid("Hello World",4,0)` 返回 “lo World”。

备注：如果将 0 作为 length 传递给 StringMid() 函数，则返回开始位置之后的整个字符串。

更改字符串大小写

在脚本中，可以使用 StringLower() 与 StringUpper() 函数以小写或大写形式返回指定的字符串。通过将结果指定给指定的字符串，可以将大写转换为小写，反之亦然。

StringLower() 函数

返回字符串相应的小写形式。

语法

```
result = StringLower (string)
```

参数

string
文本、消息标记名或字符串表达式。

示例

`StringLower("TURBINE")` 返回 "turbine"。

`StringLower("The Value Is 22.2")` 返回 "the value is 22.2"。

`mtag = StringLower(mtag)` 将 mtag 的消息值转换为小写形式。

StringUpper() 函数

返回字符串相应的大写形式。

语法

```
result = StringUpper (string)
```

参数

string
文本、消息标记名或字符串表达式。

示例

`StringUpper(“abcd”)` 返回 “ABCD”。

`StringUpper("The Value Is 22.2")` 返回 “THE VALUE IS 22.2”。

`mtag = StringUpper(mtag)` 将 mtag 的消息值转换为大写形式。

从字符串中删除空格

在脚本中，可以使用 StringTrim() 函数从字符串中修剪掉前导与尾部空格（空白符）。您可以使用此函数在用户输入之后（举例而言）从字符串中删除多余的空格。

StringTrim() 函数

从字符串中修剪掉前导与尾部空格（空白）。您可以使用此函数在用户输入之后（举例而言）从字符串中删除多余的空格。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
result = StringTrim (string, trimtype)
```

参数

字符串
文本、消息标记名或字符串表达式。

trimtype
确定要删除的空格的值、模拟标记名或数值表达式：

- 1 = 前导空格。
- 2 = 尾部空格。
- 3 = 前导与尾部空格。

附注

此函数从字符串中删除所有的前导与尾部空白。空白是空格 (ASCII 0x20) 与 ASCII 0x09 到 0x0D 之间的控制字符。

示例

要删除包含动作脚本的消息标记 mtag 中的所有空格，使用以下脚本：

```
DIM i AS INTEGER;
DIM tmp AS MESSAGE;
mtag = StringTrim(mtag,3); {mtag 被截断}
FOR i = 1 TO StringLen(mtag) {在 mtag 的字符上运行变量 I}
  IF StringMid(mtag, i, 1)<>" " THEN {i-th c 字符不是空格} tmp = tmp + StringMid(mtag, i,
1); {
add that character to tmp}
  ENDIF;
NEXT;
mtag = tmp; {将 tmp 传递回 mtag}.
```

其它示例：

`StringTrim(" Joe ",1)` 返回 “Joe ”。

`StringTrim(" Joe ",2)` 返回 “ Joe”。

此脚本删除 mtag 值左侧与右侧的所有空格：

```
mtag = StringTrim(mtag,3)
```

使用空格设置字符串格式

在脚本中，可以使用 StringSpace() 函数将空格（空白）添加到字符串。

语法

```
result = StringSpace (number)
```

参数

数字
数字、数值标记名或数值表达式。

示例

`StringSpace(4)` 返回包含 4 个空格的字符串。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

"Pump"+StringSpace(1)+"Station" 返回 "Pump Station"。

在字符与 ASCII 码之间转换

在脚本中，通过使用 StringChar() 与 StringASCII() 函数，可以将字符串中的字符转换成 ASCII 码，也可以将 ASCII 码转换成字符。

这些函数不支持多字节字符集。仅支持范围在 0-255 的字符。

如果希望在字符串上执行某些数值计算（例如，对字符串进行编码），则使用 ASCII 码非常有用。

StringChar() 函数

返回指定的 ASCII 码对应的单个字符。

语法

```
result = StringChar (ASCIICode)
```

参数

ASCIICode
介于 0 到 255 之间的一个数字、数值标记名或数值表达式。

附注

对于将控制字符传递给外设（如打印机或调制解调器）或将英文双引号传递给 SQL 查询，此函数非常有用。

示例

```
StringChar(65) 返回 "A"。
```

此脚本返回用英文双引号括起的 “Hello World”：

```
StringChar(34)+"Hello World"+StringChar(34)
```

此脚本返回用回车换行符将两个词分开来的 “Hello World”：

```
"Hello"+StringChar(13)+StringChar(10)+"World"
```

StringASCII() 函数

返回字符串第一个字符的 ASCII 码。

语法

```
result = StringASCII (string)
```

参数

string
字符串、消息标记名或字符串表达式。

示例

```
StringASCII("A") 返回 65。
StringASCII("hello world") 返回 104。
```

在字符串中搜索并替换

对于使用单字节字符集的语言（如英语），可以使用脚本中的 StringInString() 与 StringReplace() 函数在消息标记上执行有限的搜索与替换功能。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用对象	作用
StringInString()	在另一个字符串中搜索特定的字符串，将位置作为结果返回。
StringReplace()	在指定的字符串中使用其它字符或字替换特定的字符或字，将新字符串作为结果返回。

StringInString() 函数

返回指定的字符串在另一个字符串中出现的第一个位置。

语法

```
result = StringInString (string, searchfor, startpos, casesens)
```

参数

string
这是要被搜索的字符串。字符串、消息标记名或字符串表达式。

searchfor
这是要搜索的字符串。字符串、消息标记名或字符串表达式。

startpos
这是字符串中搜索的起始位置。值、数值标记名或数值表达式。

casesens
确定搜索是否区分大小写。可以是 0 或 1、离散标记名或布尔表达式。

0 - 搜索不区分大小写（大写和小写形式视为相同）。

1 - 搜索区分大小写（大写和小写形式视为不同）。

附注

使用此函数来确定是否有特定的字符串包含在消息标记中。您可以指定搜索的起始位置以及是否考虑字母大小写。

示例

此脚本返回 5 — 由于 “MTX” 中的首字母 “M” 位于字符串中的第 5 个位置：

```
StringInString("DBO MTX-010","MTX",1,0)
```

此脚本返回 3 — 由于 “MTX” 中的首字母 “M” 位于字符串中的第 3 个位置：

```
StringInString("T-MTX 010 MTX","MTX",1,0)
```

此脚本返回 11 — 由于第 8 个位置之后的 “MTX” 中的首字母 “M” 位于字符串中的第 11 个位置：

```
StringInString("T-MTX 010 MTX","MTX",8,0)
```

此脚本返回 11 — 由于同 MTX 匹配且大小写一致的第一个字符串在第 11 个位置：

```
StringInString("t-mtx 030 MTX", "MTX",1,1)
```

此脚本返回 0 — 由于字符串中没有 “Mty”：

```
StringInString("t-mtx 030 MTY-Mtx","Mty",1,1)
```

StringReplace() 函数

在一个字符串中搜索另一个字符串，如果找到，则使用第三个字符串来替换它。您可以指定：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 区分大小写 - 这确定是否将大写字母与小写字母视为完全相同的字母。
- 要替换的次数 - 如果搜索字符串出现多次，这会非常有用。
- 全字匹配 - 如果搜索字符串是整个单词，则使用这项。

备注：此函数不支持双字节字符集。

语法

```
result = StringReplace (string, searchfor, replacewith, casesens, numtoreplace,
matchwholewords)
```

参数

string
要被搜索的字符串。字符串、消息标记名或字符串表达式。

searchfor
要搜索的字符串。字符串、消息标记名或字符串表达式。

replacewith
要用作替换的字符串。字符串、消息标记名或字符串表达式。

casesens
确定搜索是否区分大小写。可以是 0 或 1、离散标记名或布尔表达式。

0 - 搜索不区分大小写（大写和小写形式视为相同）

1 - 搜索区分大小写（大写和小写形式视为不同）

numtoreplace
要替换的量。将其设置为 -1 以替换找到的所有搜索字符串。整数值、整型标记名或整型表达式。

matchwholewords
确定是否仅限全字匹配。可以是 0 或 1、离散标记名或布尔表达式。

0 - 此函数查找字符串中任何位置的搜索字符串字符

1 - 仅先全字匹配

示例

此语句仅替换第一项，并返回 "MTY 030 MTX"。

```
StringReplace("MTX 030 MTX","MTX","MTY",0,1,0)
```

此语句替换所有项，并返回 "MTY 030 MTY"。

```
StringReplace("MTX 030 MTX","MTX","MTY",0,-1,0)
```

此语句替换大小写匹配的所有项，并返回 "MTY 030 mtx"。

```
StringReplace("MTX 030 mtx","MTX","MTY",1,-1,0)
```

此语句替换全字匹配的所有项，并返回 "MTY 030 QMTX"。

```
StringReplace("MTX 030 QMTX","MTX","MTY",0,-1,1)
```

返回关于字符串的信息

在脚本中，通过使用 StringLen() 与 StringTest() 函数可以返回指定字符串的长度，以及测试某个字符是否在特定的字符组中。

StringLen() 函数

返回指定字符串（包括不可见字符）的长度。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
result = StringLen (string)
```

参数

string
字符串、消息标记名或字符串表达式。

示例

```
StringLen("Twelve percent") 返回 14。
StringLen("12%") 返回 3。
StringLen("The end."+ StringChar(13)) 返回 9。
```

StringTest() 函数

测试字符串的第一个字符是否在特定的字符组中。

语法

```
result = StringTest (string, group)
```

参数

string
字符串、消息标记名或字符串表达式。

group
要测试字符的组数。1 到 11 范围内的值、整型标记名，或整型表达式。

1 - 字母数字字符（A-Z、a-z、0-9）

2 - 数值字符 (0-9)

3 - 字母字符（A-Z、a-z）

4 - 大写字符 (A-Z)

5 - 小写字符 (a-z)

6 - 标点字符 (ASCII 0x21 - 0x2F)，例如 !、@、#、$、%、^、&、* 等

7 - ASCII 字符 (ASCII 0x00 - 0x7F)

8 - 十六进制字符（0-9、A-F、a-f）

9 - 可打印字符 (ASCII 0x20 - 0x7E)

10 - 控制字符（ASCII 0x00 - 0x1F 与 0x7F）

11 - 空格字符（ASCII 0x09 - 0x0D 与 0x20）

示例

此字符串返回 1 — 由于 “A” 是字母数字字符：

```
StringTest("ACB123",1)
```

此字符串返回 0 — 由于 “A” 不是小写字符：

```
StringTest("ABC123",5)
```

比较字符串

在脚本中，通过使用 StringCompare()、StringCompareNoCase() 以及 StringCompareEncrypted() 函数，可以比较两个字符串。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用对象	作用
StringCompare()	比较时区分大小写。
StringCompareNoCase()	比较时不区分大小写。
StringCompareEncrypted()	比较加密的字符串和未加密的字符串。

StringCompare() 函数

相互比较两个字符串，并返回布尔结果（0 = 字符串相等）。每个字母的大小写也在考虑范围，因此举例而言，'A' 会视为不等于 'a'。

语法

```
result = StringCompare (string1, string2)
```

参数

string1
字符串、消息标记名或字符串表达式。

string2
字符串、消息标记名或字符串表达式。

示例

```
StringCompare ("Apple","Apple") 返回 0。
StringCompare ("Apple","apple") 返回 1。
```

此字符串比较两个消息标记，并返回离散结果（0 或 1）：

```
StringCompare (mtag1, mtag2)
```

StringCompareNoCase() 函数

相互比较两个字符串，并返回整型结果。每个字母的大小写不在考虑范围，因此举例而言，'A' 会视作等于 'a'。

整型结果返回：

- 0，如果两个字符串完全相同（忽略大小写）。
- 非零值（其它情况）。结果是不同字符的 ASCII 值之差（忽略大小写）。

备注：由于在 InTouch 脚本中，所有非零值都视作等于 TRUE，StringCompareNoCase() 函数的结果可以作为离散结果来使用。

语法

```
result = StringCompareNoCase (string1, string2)
```

参数

string1
字符串、消息标记名或字符串表达式。

string2
字符串、消息标记名或字符串表达式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

此字符串返回 0 — 由于这两个字符串被视作完全相同：

```
StringCompareNoCase("Apple","apple")
```

此字符串返回 -6 — 由于这两个字符串被视作不完全相同，第一个不同字符 “p” 的 ASCII 值减去相应字符 “v” 的 ASCII 值等于 -6：

```
StringCompareNoCase("Apple","Avocado")
```

StringCompareEncrypted() 函数

比较加密字符串与未加密的字符串，并返回布尔结果。您可以使用此函数来验证口令。如需有关口令加密的详细信息，请参阅动画对象。

语法

```
result = StringCompareEncrypted (plain, encrypted)
```

参数

plain
字符串、消息标记名或字符串表达式。

encrypted
加密的消息标记名。

示例

明文与密文完全相同时，此脚本返回 1；否则返回 0。Passwd 是消息标记，它包含加密的用户输入值。PlainTxt 是要与用户输入作比较的消息标记。

```
StringCompareEncrypted(PlainTxt, Passwd)
```

转换数据类型

在脚本中，通过使用转换 QuickScript 可以将标记名中包含的值转换为其它数据类型。这使您可以用数学函数来处理字符串数据，或是将这些值记录到 Log Viewer 以便进行调试。

- Text() 函数
- StringFromIntg() 函数
- StringFromReal() 函数
- StringToIntg() 函数
- StringToReal() 函数
- DText() 函数

Text() 函数

Text() 函数根据指定的格式将数字的值当作字符串来返回。您可能希望这样做，以通过某种方式设置某个值的格式，或是将结果同其它字符串值合并起来以便进一步处理。

语法

```
result = Text (number, format)
```

参数

number
数值、模拟标记名或数值表达式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

format
使用 “#”、“0”、“.” 或 “,”。
使用 “#” 代表数字、“.” 代表小数点、“0” 代表强制使用前导零、“,” 代表插入逗号。
如果在格式中使用零，则它后面必须跟零。小数点右边所有的位必须总是为零。例如，000.00 正确，而 #0#0.0# 则不正确。
此函数会根据需要对值进行舍入。字符串、消息标记名或字符串表达式。

示例

`Text(66,"#.00")` 返回 "66.00"。

`Text (1234,”#”)` 返回 “1234”。

`Text (123.4, "#,##0.0")` 返回 "123.4"。

`Text (12.3, "0,000.0")` 返回 "0,012.3"。

`Text(3.57,"#.#")` 返回 "3.6"。

如果模拟标记名 “pressure” 包含值 1690.2743，则此脚本返回字符串 “Reactor Pressure is 1690.3 mbar”。

```
"Reactor Pressure is "+Text(pressure,"#.#")+" mbar"
```

StringFromIntg() 函数

在脚本中，通过使用 StringFromIntg() 函数可以将整数值转换为字符串值。

此函数返回整数值的字符串值，并同时执行基数转换。举例来说，这可以用于同时显示文本与整数值，或是将整数值转换为十六进制数。

语法

```
result = StringFromIntg (number, base)
```

参数

number
整数值、整型标记名或整型表达式。

base
转换的基数。这用于将值转换到不同的基数，如二进制 (2)、十进制 (10) 或十六进制 (16)。整数值、整型标记名或整型表达式。

示例

`StringFromIntg(26,2)` 返回 “11010”（二进制）。

`StringFromIntg(26,8)` 返回 “32” — 由于
(base 8:26 = 3*8 + 2)

`StringFromIntg(26,10)` 返回 “26”（十进制）。

`StringFromIntg(26,16)` 返回 “1A”（十六进制）。

StringFromReal() 函数

在脚本中，通过使用 StringFromReal() 函数可以将实数值转换为字符串值。

您也可以指定：

- 将值舍入到指定的精度。
- 以指数形式传递值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

举例来说，这可以用于同时显示文本与实数值，或用于显示指数形式的实数。

语法

```
result = StringFromReal (number, precision, type)
```

参数

number
值、模拟标记名或数值表达式。

precision
指定要使用的小数位数。整数值、整型标记名或整型表达式。

类型
指定是否使用指数形式。字符串、消息标记名或字符串表达式。

“f” - 使用浮点记号。

“e” - 使用带小写 “e” 的指数形式。

“E” - 使用带大写 “E” 的指数形式。

示例

`StringFromReal(263.355, 2,”f”)` 返回 “263.36”。

`StringFromReal(263.355, 2,”e”)` 返回 “2.63e2”。

`StringFromReal(263.55, 3,”E”)` 返回 “2.636E2”。

`StringFromReal(0.5723, 2,”E”)` 返回 “5.72E-1”。

StringToIntg() 函数

在脚本中，通过使用 StringToIntg() 函数可以将字符串中包含的值转换为整数值。

您可以用它将字符串开头的值读取到整型标记，以便进行进一步的数学运算。

语法

```
result = StringToIntg (string)
```

参数

string
字符串、消息标记名或字符串表达式。

附注

此函数检查字符串的第一个字符。如果是数字，则它会试图将这个字符以及后续的字符读取到一个整数中，直至遇到非数值字符。此函数忽略字符串中的前导空格。

示例

`StringToIntg(“ABCD”)` 返回 0。

`StringToIntg(“13.4 mbar”)` 返回 13。

`StringToIntg("Pressure is 13.4")` 返回 0。

要从字符串 (mtag) 中提取不在开头位置的第一个整数并将它存储在整型标记 itag 中，请使用以下动作脚本：

```
DIM i AS INTEGER;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
DIM tmp AS INTEGER;
FOR i = 1 TO StringLen(mtag) {run variable i over the characters of mtag}
 tmp = StringASCII(StringMid(mtag, i, 1)) - 48; {detect ASCII value}
 IF (tmp>=0 AND tmp<10) THEN {if ASCII value represented "0" - "9"}
 itag = StringToIntg(StringMid(mtag, i, 0)); {set itag to value from that position and
exit loop}
 EXIT FOR;
 ENDIF;
NEXT;
```

StringToReal() 函数

在脚本中，通过使用 StringToReal() 函数可以将字符串中包含的值转换为实数值。

您可以用它将字符串开头包含的值读取到某个实型标记，以便进行进一步的数学运算。

备注：此函数也支持指数形式，并可以将 1e+6 这样的字符串表达式正确转换为 1000000。

语法

```
result = StringToReal (string)
```

参数

string
字符串、消息标记名或字符串表达式。

附注

此函数检查字符串的第一个字符。如果是数字，则它会将这个字符以及后续的字符读取到某个实数中，直至遇到非数值字符。此函数忽略字符串中的前导空格。

要从字符串（消息标记 mtag）中提取不在开头位置的第一个实数并将它存储在实型标记 rtag1 中，请使用以下脚本：

```
DIM i AS INTEGER;
DIM tmp AS INTEGER;
FOR i = 1 TO StringLen(mtag) {run variable i over the characters of mtag}
 tmp = StringASCII(StringMid(mtag, i, 1)) - 48; {detect ASCII value}
 IF (tmp>=0 AND tmp<10) THEN {if ASCII value represented "0" - "9"}
 rtag = StringToReal(StringMid(mtag, i, 0)); {set rtag to value from that position and exit loop}
 EXIT FOR;
 ENDIF;
NEXT;
```

示例

```
StringToReal("ABCD") 返回 0。
StringToReal("13.4 mbar") 返回 13.4。
StringToReal("Pressure is 13.4") 返回 0。
```

DText() 函数

在脚本中，通过使用 DText() 函数可以将布尔值转换为字符串值。您可以使用此函数来自定义消息显示动画链接。

此函数根据布尔值的值来返回不同的字符串值。

语法

```
result = Dtext (Boolean, stringtrue, stringfalse)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

Boolean
布尔值、离散标记名或布尔表达式。

stringtrue
Boolean 为真时要返回的字符串。字符串值、消息标记名或字符串表达式。

stringfalse
Boolean 为假时要返回的字符串。字符串值、消息标记名或字符串表达式。

示例

如果离散标记名 switch 为 TRUE，此脚本返回 “Running”；否则返回 “Stopped”。

```
DText(switch,"Running","Stopped")
```

此脚本根据离散标记 switch1 的值返回另一个离散标记 switch2 的“打开消息”与“关闭消息”。

```
DText(switch1,switch2.OnMsg,switch2.OffMsg)
```

在运行时处理 InTouch 窗口

在脚本中，可以控制 InTouch 窗口的行为与外观。您也可以使用 QuickScripts 编写脚本来打印单独的 InTouch 窗口或整个屏幕。

提供窗口名属性

您可以使用 **GetWindowName** 脚本函数帮助运行时环境通过当前的实施减少加载窗口时需要编写的脚本量。该函数可让您检索在其下调用函数的窗口名称。

GetWindowName() 函数

获取调用该函数的窗口的名称。

语法

该脚本函数的语法如下：

```
resultcode = GetWindowName(tagname);
```

Resultcode 可指出对脚本函数的调用是成功还是失败。Resultcode 的数据类型可能为离散型、整型或实型。Resultcode 可能为 1 或 0，具体取决于对脚本函数的调用是成功还是失败：

- 如果脚本函数是从窗口环境中调用的，则 Resultcode 为 1。
- 如果脚本函数是从非窗口环境中调用的，则 Resultcode 为 0。

备注：您可以选择是否要配置脚本函数的返回值。这与 InTouch 中的现有脚本函数类似。

参数

TagName
标记名是该函数的输出参数。它将用作检索窗口名的消息标记。

该函数可以采用以下的任意一个参数：

- InTouch 标记
- 点域
- 脚本中的局部变量
- 远程标记引用

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- Galaxy 引用

脚本浏览器加载的缺省文本如下：

GetWindowName(*标记名*);

备注：“标记名”是消息标记或属于消息类型的远程标记引用 (RTR)。

返回值

在以下情形中，**GetWindowName** 脚本函数会返回窗口名，返回值为 1：

- 窗口脚本（所有条件类型）
- 触动按钮动作脚本（所有条件类型）

在以下情形中，**GetWindowName** 脚本函数会返回空字符串，返回值为 0：

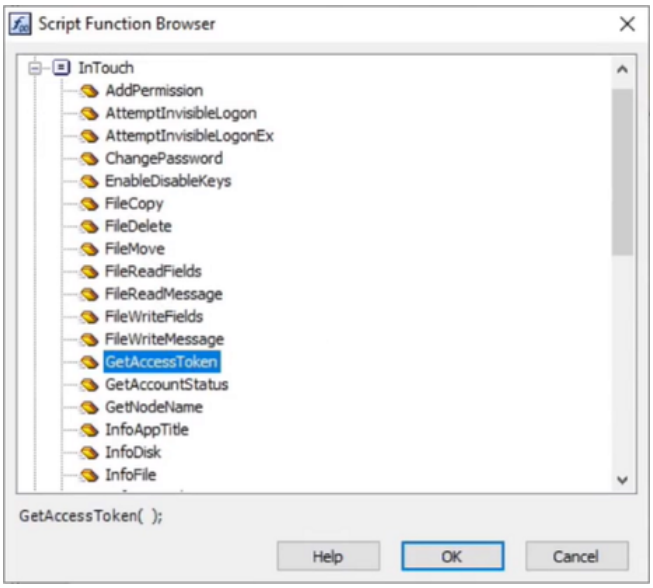
- 应用程序脚本
- 键脚本
- 条件脚本
- 数据改变脚本
- ActiveX 事件脚本
- 快速脚本函数（同步和异步）

公开最新的访问令牌

通过使用脚本 **GetAccessToken** 和 **GetSecureAccessToken**，CONNECT 访问令牌可以向 InTouch 公开，并且可由控件和小组件在运行时用于连接体验中的单一登录。

您可以使用 **GetAccessToken** 脚本函数返回最新的令牌值，并且可以使用 **GetSecureAccessToken** 脚本函数以安全或加密的方式返回最新的令牌值。由于访问令牌是有时间限制的，并且值会定期进行更改，因此此脚本函数有助于获取最新的访问令牌值。

在脚本函数浏览器中，您可以看到内置的脚本 **GetAccessToken** 和 **GetSecureAccessToken**。它将在记录器中记录访问令牌的值。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

GetAccessToken() 函数

GetAccessToken() 提供身份验证令牌，趋势控件、报警客户端控件和其它 InTouch 控件等控件将接受该令牌，以支持单一登录功能。

语法

该脚本函数的语法如下：

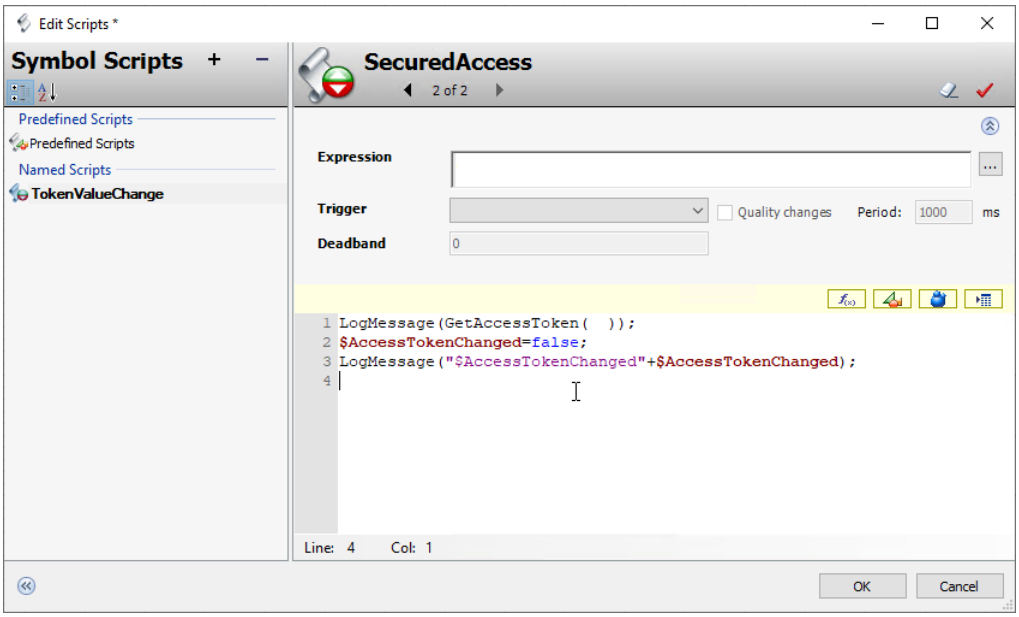
```
resultCode = GetAccessToken();
```

Resultcode 表示最新的访问令牌。

示例：

要使令牌值在每次过期时都得到更新，您可以按如下所示使用脚本：

1. 在“符号脚本”窗口中，选择添加脚本。
2. 为脚本提供一个名称。
3. 选择显示脚本函数浏览器图标。
4. 在脚本浏览器屏幕中，在 **InTouch** 下选择 **GetAccessToken** 以插入脚本函数，或者您可以手动输入。
5. 选择确定。



返回值

如果用户使用 CONNECT 进行身份验证，那么 GetAccessToken 脚本函数将返回带有令牌值的访问令牌。

如果用户未使用 CONNECT 进行身份验证，那么 GetAccessToken 脚本函数将返回空字符串。

GetSecureAccessToken() 函数

GetSecureAccessToken() 提供安全的身份验证令牌，趋势控件、报警客户端控件和其它 InTouch 控件等控件将接受该令牌，以支持单一登录功能。

语法

该脚本函数的语法如下：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

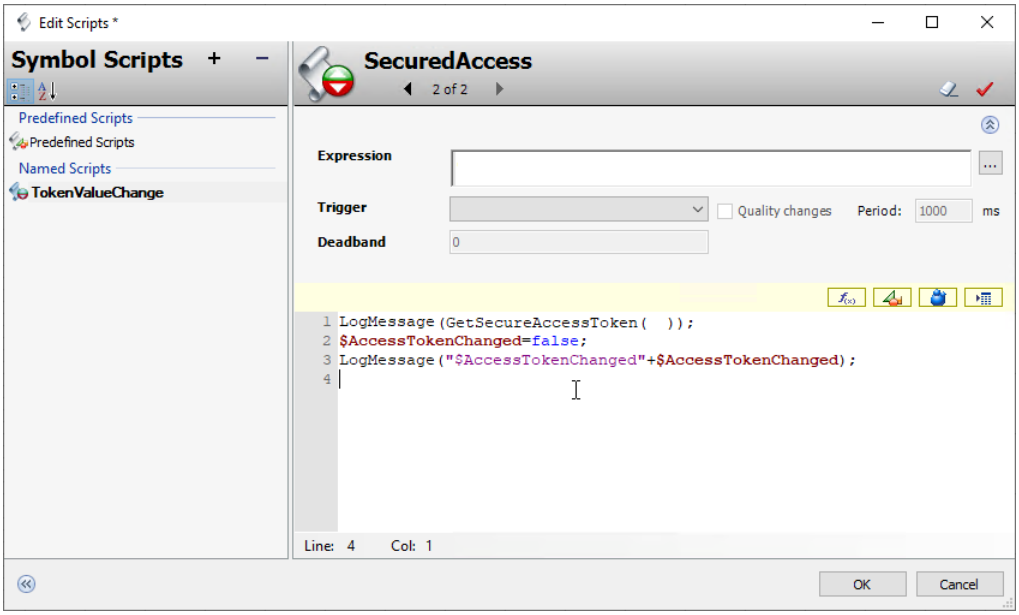
```
resultCode = GetSecureAccessToken();
```

Resultcode 表示最新的访问令牌。

示例：

要使安全令牌值在每次过期时都得到更新，您可以按如下所示使用脚本：

1. 在“符号脚本”窗口中，选择添加脚本。
2. 为脚本提供一个名称。
3. 选择显示脚本函数浏览器图标。
4. 在脚本浏览器屏幕中，在 **InTouch** 下选择 **GetSecureAccessToken** 以插入脚本函数，或者您可以手动输入。
5. 选择确定。



返回值

如果用户使用 CONNECT 进行身份验证，那么 GetSecureAccessToken 脚本函数将返回带有令牌值的访问令牌。

如果用户未使用 CONNECT 进行身份验证，那么 GetSecureAccessToken 脚本函数将返回空字符串。

检索连接状态

您可以使用 **GetTokenConnectionStatus** 脚本函数在 AVEVA Operations Control 连接体验中检索连接状态。

GetTokenConnectionStatus() function

在 AVEVA Operations Control 连接体验中检索 AVEVA Identity Manager 和 CONNECT 的连接状态。

语法

该脚本函数的语法如下：

```
resultcode = GetTokenConnectionStatus();
```

Resultcode 表示连接状态的令牌。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

返回值

- 当连接状态是“完全连接”时，resultcode 为 0。也就是说，应用程序同时连接到 AVEVA Identity Manager 和 CONNECT。
- 当连接状态是“部分连接”时，resultcode 为 1。也就是说，应用程序连接到 AVEVA Identity Manager，并断开与 CONNECT 的连接。
- 当连接状态是“完全断开连接”时，resultcode 为 2。也就是说，应用程序同时断开与 AVEVA Identity Manager 和 CONNECT 的连接。

示例：

int AccessTokenStatus=GetTokenConnectionStatus()

显示打开的窗口的列表

在脚本中，通过使用 OpenWindowsList() 函数可以显示一个对话框，该对话框包含当前打开的 InTouch 窗口的列表。

OpenWindowList() 函数

显示一个对话框，包含当前打开的 InTouch 窗口的列表。

您无法在动画链接中使用此函数。

语法

```
[result = ]OpenWindowsList();
```

示例

此脚本打开打开窗口列表对话框，并显示当前打开的所有 InTouch 窗口。

```
OpenWindowsList()
```

备注：启用使用内存窗口缓存 WindowViewer 选项时，使用 OpenWindowList() 函数创建的列表中可能会显示已关闭的窗口。

检查窗口是打开、关闭或是否存在

在脚本中，通过使用 WindowState() 函数可以检查 InTouch 窗口是打开、关闭或是否不存在。

WindowState() 函数

检查某个 InTouch 窗口是打开、关闭或是否不存在。

语法

```
result = WindowState (windowname)
```

参数

windowname
窗口的名称。字符串值、消息标记名或字符串表达式。

返回值

整数值，含义如下：

0 - InTouch 窗口存在，且当前处于关闭状态

1 - InTouch 窗口存在，且当前处于打开状态

2 - InTouch 窗口不存在

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

如果 InTouch 窗口 Main 存在但并未打开，则此脚本返回 0。

```
WindowState("Main")
```

打开 InTouch 窗口

在脚本中，通过使用以下 QuickScript 函数之一，可以打开 InTouch 窗口：

使用对象	作用
显示	在其位置设置所定义的位置上打开 InTouch 窗口。
ShowAt()	在指定的位置上打开 InTouch 窗口。打开的窗口在该位置处居中显示。此函数也可用于移动打开的窗口。
ShowHome	打开在 WindowViewer 属性对话框起始窗口选项卡中所指定的 InTouch 窗口，并关闭任何其它窗口。
ShowTopLeftAt()	在指定的位置上打开 InTouch 窗口。打开的窗口左上角对齐该位置。此函数也可用于移动打开的窗口。

Show() 函数

在缺省位置上打开 InTouch 窗口。

语法

```
Show windowname
```

参数

windowname
要打开的窗口的名称。字符串值、消息标记名或字符串表达式。

示例

此脚本打开窗口 Main。

```
Show "Main";
```

此脚本打开使用 wname 消息标记中所存储的名称的窗口。

```
Show wname;
```

ShowAt() 函数

在指定的位置上打开 InTouch 窗口。它也可以将打开的 InTouch 窗口移到指定的位置。该位置是窗口的中点。

备注：如果窗口的一条边在屏幕显示范围之外，则该窗口不会居中。

语法

```
ShowAt (windowname, xpos, ypos)
```

参数

windowname

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要打开或移动的窗口的名称。

xpos
窗口中心要移到的水平位置，以像素计。值、模拟标记名或数值表达式。

ypos
窗口中心要移到的垂直位置，以像素计。值、模拟标记名或数值表达式。

示例

此脚本打开 Main 窗口，使它在 (x:450, y:130) 位置居中。

```
ShowAt("Main",450,130);
```

此脚本打开 UserDialog 窗口，并使它在调用此函数（例如，按钮）的对象的中心位置上进行居中。

```
ShowAt("UserDialog",$ObjHor,$ObjVer);
```

ShowHome() 函数

打开在 WindowViewer 属性对话框起始窗口选项卡中所指定的 InTouch 窗口，并关闭任何其它窗口。

语法

```
ShowHome;
```

ShowTopLeftAt() 函数

在指定的位置上打开 InTouch 窗口。也可用于移动打开的窗口。

语法

```
ShowTopLeftAt (windowname, xpos, ypos)
```

参数

要打开或移动的窗口的名称。

xpos
窗口左侧边缘要移到的水平位置，以像素计。值、模拟标记名或数值表达式。

ypos
窗口顶部边缘要移到的垂直位置，以像素计。值、模拟标记名或数值表达式。

示例

此脚本打开 Main 窗口，并将它的左上角放到 (x:450，y:130) 位置。

```
ShowTopLeftAt("Main",450,130);
```

移动窗口与调整大小

在脚本中，通过使用 WWMoveWindow() 函数可以移动打开的 InTouch 窗口或调整其大小。指定的窗口打开时，暂时应用新的位置与新的大小。

WWMoveWindow() 函数

将打开的 InTouch 窗口移到指定的位置，并将它调整到指定的大小。指定的窗口打开时，暂时应用新的位置与新的大小。

语法

```
WWMoveWindow (windowname, xpos, ypos, xsize, ysize)
```

参数

windowname
要打开或移动的窗口的名称。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

xpos
窗口左侧边缘要移到的水平位置，以像素计。值、模拟标记名或数值表达式。

ypos
窗口顶部边缘要移到的垂直位置，以像素计。值、模拟标记名或数值表达式。

xsize
指定的窗口的水平大小，以像素计。值、模拟标记名或数值表达式。

ysize
指定的窗口的垂直大小，以像素计。值、模拟标记名或数值表达式。

隐藏 InTouch 窗口

在脚本中，通过使用以下函数之一，可以隐藏 InTouch 窗口。

使用对象	作用
隐藏	隐藏指定的窗口。
HideSelf	隐藏当前活动窗口。

Hide() 函数

隐藏（关闭）InTouch 窗口。

语法

```
Hide windowname;
```

参数

windowname
要隐藏的窗口的名称。字符串值、消息标记名或字符串表达式。

示例

此脚本隐藏 UserConfirmation 窗口。

```
Hide "UserConfirmation";
```

HideSelf() 函数

隐藏（关闭）当前活动的 InTouch 窗口。

备注：此函数仅能用在动作 QuickScript 中。

语法

```
HideSelf;
```

示例

```
HideSelf;
```

更改窗口的颜色

在脚本中，通过使用 ChangeWindowColor() 函数可以更改打开的 InTouch 窗口的颜色。

ChangeWindowColor() 函数

更改打开的 InTouch 窗口的颜色并返回结果码。

语法

```
Result = ChangeWindowColor (windowname, rValue, gValue, bValue)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

windowname
要更改颜色的窗口的名称。字符串值、消息标记名或字符串表达式。

rValue
红色的浓度。0 到 255 范围内的整数值、整型标记名或整型表达式。

gValue
绿色的浓度。0 到 255 范围内的整数值、整型标记名或整型表达式。

bValue
蓝色的浓度。0 到 255 范围内的整数值、整型标记名或整型表达式。

返回值

值的含义如下：

0 - 失败，窗口未定义或 RGB 值超出范围。

1 - 成功。

2 - 失败。窗口存在，但并未打开。

在运行时打印窗口

在脚本中, 通过使用 PrintWindow() 或 PrintScreen() 函数可以打印单独的 InTouch 窗口或整个 WindowViewer 屏幕。您也可以使用 SetWindowPrinter() 函数设置希望使用的打印机。

SetWindowPrinter() 函数

在运行时，可以使用 SetWindowPrinter() 函数设置希望使用的打印机。

备注：使用此函数设置的打印机也是 PrintHT() 函数所使用的打印机。

语法

```
SetWindowPrinter (printername)
```

参数

printername
打印机的名称或者显示为网络共享，或者显示为其属性窗口中所显示的打印机名。字符串值、消息标记名或字符串表达式。

示例

在本例中，PRTSRV1 是节点名，PRT22SW1 是赋予打印机的共享名。

```
SetWindowPrinter("\\PRTSRV1\PRT22SW1");
```

本例中，Epson LX-300 是打印机的名称，如打印机的“属性”窗口中所显示的那样。

```
SetWindowPrinter("Epson LX-300");
```

在本例中，MyPrinter 是消息标记，它包含所安装的 Windows 打印机的名称，或是指向共享网络打印机的路径。

```
SetWindowPrinter(MyPrinter);
```

关于打印的建议

以下列表包含打印时应注意的一些事项。对于打印单个窗口或 WindowViewer 屏幕，这些事项都适用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 在打印之前，打开要打印的窗口。否则<1>“</1>窗口<2>”</2>与<3>“</3>ActiveX 控件<4>”</4>可能不会正确打印<5>。</5>
- 您无法打印到当前正在打印报警的打印机。
- 打印时避免在窗口上叠放窗口与对象。
- 尽可能使用 True Type 字体。缺省 InTouch 字体 (System) 不是 True Type 字体。
- 为了快速打印，请考虑使用白色背景、减少对象，并使用文本代替图形。
- 在将窗口发送给打印机队列之前，WindowViewer 会等待一定的时间。在此期间，WindowViewer 会在后台更新该窗口的任何 I/O 值。要更改此等待时间，请打开 intouch.ini 文件，然后更改或添加下面这行（以毫秒计）：`PrintWindowWait=10000`

PrintWindow() 函数

在脚本中，通过使用 PrintWindow() 函数可以打印 InTouch 窗口。

备注：含有 PrintWindow() 函数的脚本无法在 InTouch 窗口内打印以下工业图形控件：ListBox、DateTimePicker、CalenderControl、EditBox、CheckBox、RadioButtonGroup、ComboBox、AlarmClient 或 TrendClient。

语法

```
[result = ] PrintWindow (windowname, leftmargin, topmargin, width, height, options);
```

参数

windowname
要打印的窗口的名称。字符串值、消息标记名或字符串表达式。

leftmargin
左侧边距偏移量（以英寸计）。数值、模拟标记名或数值表达式。

topmargin
顶部边距偏移量（以英寸计）。数值、模拟标记名或数值表达式。

width
打印输出宽度（以英寸计）。将此值设置为 0 可得到最大的纵横比。数值、模拟标记名或数值表达式。

height
打印输出高度（以英寸计）。将此值设置为 0 可得到最大的纵横比。数值、模拟标记名或数值表达式。

options
离散值 0 或 1，仅当 width 与 height 都为 0 时使用。布尔值、离散标记名或布尔表达式。设置为：

1 - 窗口按最大纵横比，也就是窗口大小的整数倍打印。

0 - 窗口按适合页面的最大纵横比打印。

备注：如果窗口包含位图，则将 options 设置为 1 可以防止拉伸位图。

返回值

0 - 打印作业未能成功进入队列，或窗口不存在

1 - 打印作业成功进入队列

PrintScreen() 函数

您可以使用 PrintScreen() 函数编写脚本来打印整个 WindowViewer 屏幕。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
PrintScreen (ScreenOption, PrintOption)
```

参数

ScreenOption
确定 WindowViewer 屏幕要打印的大小。整数值、整型标记名或整型表达式。

1 - 打印客户端区域，无菜单（缺省值）

2 - 打印整个窗口区域，包括菜单

PrintOption
确定如何拉伸打印的图像，使之适合打印输出。

- 1 - 最佳适配：
  拉伸图像，使其在不改变长宽比的前提下水平或垂直适配打印页面（缺省值）。
- 2 - 垂直适配：
  拉伸图像，使其在不改变长宽比的前提下垂直适配打印页面。可能会水平修剪图像。
- 3 - 水平适配：
  拉伸图像，使其在不改变长宽比的前提下水平适配打印页面。可能会垂直修剪图像。
- 4 - 拉伸至页面：
  拉伸图像，使其同时水平、垂直铺满打印页面。纵横比可能会更改，但不会截断图像。
- 无效选项，包含 0，缺省为“最佳适合”。

注意：超出 WindowViewer 屏幕区域的弹出窗口将被裁剪。

示例

此脚本将当前的整个 WindowViewer 屏幕区域（不含菜单）的打印输出发送到打印队列。打印输出包含经过拉伸以填满打印输出尺寸的屏幕区域。

```
PrintScreen(1,4);
```

在上述示例中，函数传入的第一个参数为 *ScreenOption* (1)。表示打印不含菜单栏的客户端区域；

第二个参数为 *PrintOption* (4)。表示将指定区域拉伸至铺满打印页面尺寸。

PrintHT() 函数

在脚本中，通过创建按钮并将它链接到执行 PrintHT QuickScript 函数的动作 QuickScript， 可以打印历史趋势。

希望打印整个窗口而不只是趋势图表时，请使用 PrintWindow() 函数代替 PrintHT() 函数。

备注：使用打印选项或 PrintHT() 函数打印历史趋势时，不会打印 x 轴与 y 轴的值。要打印 x 轴与 y 轴的值，请使用 PrintWindow() 或 PrintScreen()。

语法

```
PrintHT(HistTrendTagname);
```

参数

HistTrendTagname
要打印的历史趋势的历史趋势标记名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

启动标记查看器

标记查看器是一款运行时应用程序，允许您观察和监视标记并修改标记的值。如需有关标记查看器及其用法的信息，请参阅《*AVEVA™ InTouch HMI 应用程序运行时*》。

LaunchTagViewer() function

只有在 WindowViewer 正在运行，且已在 WindowMaker 中启用标记查看器后，您才可以启动标记查看器。

如需有关启用标记查看器的信息，请参阅《*AVEVA™ InTouch HMI 为 InTouch HMI 组件创建标准*》中的配置常规 WindowViewer 属性。

语法

```
LaunchTagViewer()
```

附注

LaunchTagViewer() 函数可以通过应用程序脚本（OnStartup 和 OnShutdown）以外的任意类型脚本执行。

如果尚未在 WindowMaker 中启用标记查看器，那么您将无法通过调用函数启动标记查看器且记录器中会出现一条警告消息。

要启动标记查看器，您必须拥有足够的安全权限。

处理日期与时间信息

在脚本中，通过系统标记与 QuickScript 函数，可以在计算中使用系统时间与日期设置。InTouch 脚本也支持涉及到多个时区与“夏令时”的计算。

检索数值日期与时间信息

在脚本中，通过使用多个数值系统标记与一个脚本函数，可以检索有关系统日期与时间的信息。这些标记与脚本函数可以用在其它数学运算中。系统标记与脚本函数包括：

使用对象	作用
$Year	返回当前年份。
$Month	返回当前月份。
$Day	返回月份中的当前日期值。
$Hour	返回一日中的当前小时值。
$Minute	返回当前时刻的分钟值。
$Second	返回当前这秒的值。
$Msec	返回当前这毫秒的值。
$Time	返回自午夜以来的时间（以毫秒为单位，本地时区）。
$Date	返回自 1970 年 1 月 1 日以来的整天数（本地时区）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用对象	作用
$DateTime	返回自 1970 年 1 月 1 日以来的天数（包括非整天，本地时区）。
DateTimeGMT()	返回自 1970 年 1 月 1 日以来的天数（包括非整天，通用协调时间 (UTC)）。

$Year 系统标记

返回当前年份。

语法

```
$Year
```

数据类型

整型（只读）

示例

此脚本将字符串 “Welcome to xxxx” 指定给字符串 Welcome， 其中 xxxx 是当前年份。

```
Welcome = "Welcome to " + StringFromIntg($Year,10)
```

$Month 系统标记

返回当前月份。

语法

```
$Month
```

数据类型

整型（只读）

示例

如果当前月份是 10， 此脚本将字符串 “October” 指定给字符串 MonthName。

```
IF $Month==10 THEN
   MonthName="October";
ENDIF;
```

$Day 系统标记

返回本月的当前日期值。

语法

```
$Day
```

数据类型

整型（只读）

示例

如果当前日期是 2 月 29 日， 则此脚本会将字符串 "It is a leap year!" 指定给字符串 Msg2User。

```
IF $Day==29 AND $Month==2 THEN
   Msg2Usr="It is a leap year!";
ENDIF;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

$Hour 系统标记

返回一日中的当前小时值。

语法

```
$Hour
```

数据类型

整型（只读）

示例

此脚本检查是否是下午 8 点并且备份尚未运行（用离散标记 BackupAlreadyRun 表示），如果是，则调用 RunBackup() QuickFunction 脚本并将 BackupAlreadyRun 标帜设为 TRUE。

```
IF $Hour==20 AND BackupAlreadyRun==0 THEN
   CALL RunBackup();
   BackupAlreadyRun=1;
ENDIF;
```

$Minute 系统标记

返回当前时刻的分钟值。

语法

```
$Minute
```

数据类型

整型（只读）

示例

此脚本检查是否是下午 4:50，如果是，则显示 Shift End 窗口。

```
IF $Minute==50 AND $Hour==16 THEN
   Show "Shift End";
ENDIF;
```

$Second 系统标记

返回当前的秒值。

语法

```
$Second
```

数据类型

整型（只读）

示例

此脚本生成一个振幅为 100、周期为 1 分钟的正弦波函数。

```
100*Sin(6*$Second)
```

此脚本生成一系列每秒钟交替更改的 0 与 1。

```
$second.00
```

$Msec 系统标记

返回当前的毫秒值。

备注：缺省条件下，InTouch 每 1000 毫秒更新一次所有的标记。因为如此，$Msec 系统标记似乎并未更改。如果增加 WindowViewer 属性中的更新速率，则可以看见 $Msec 标记在更新。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
$Msec
```

数据类型

整型（只读）

$Time 系统标记

返回当地时间自午夜以来的毫秒数。

语法

```
$Time
```

数据类型

整型（只读）

示例

此脚本返回自午夜以来的秒数。

```
$Time/1000
```

$Date 系统标记

返回自 1970 年 1 月 1 日以来的整天数（本地时间）。

语法

```
$Date
```

数据类型

整型（只读）

示例

此脚本返回当前时间。

```
StringFromTime(($Date*86400)+($Time/1000),3);
```

$DateTime 系统标记

返回自 1970 年 1 月 1 日以来的天数（包括非整天，本地时间）。

语法

```
$DateTime
```

数据类型

实型（只读）

示例

此脚本返回当前时间。

```
StringFromTime($DateTime*86400,3);
```

DateTimeGMT() 函数

返回自 1970 年 1 月 1 日以来的天数（包括非整天，通用协调时间 (UTC)）。

备注：此函数无法用在动画显示链接中。

语法

```
result = DateTimeGMT();
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

返回值

自 1970 年 1 月 1 日以来的天数（UTC 时间）。实数值。

示例

此脚本返回 UTC 格式的当前日期/时间。

```
StringFromTime(DateTimeGMT() * 86400.0, 3);
```

检索字符串日期与时间信息

在脚本中，可以检索字符串形式的日期与时间信息。对于在屏幕上显示日期与时间，或是在需要对整个日期/时间字符串进行计算时，这非常有用。

您可以使用以下系统标记与脚本函数。

使用对象	作用
$DateString	返回短格式的系统日期。
$TimeString	返回系统时间。
UTCDateTime	返回 UTC 日期与/或时间以及本地计算机的时区。

$DateString 系统标记

返回本地操作系统“区域设置”中所定义的短格式的系统日期。

语法

```
$DateString
```

数据类型

字符串（只读）

示例

根据操作系统“区域设置”中的短日期格式设置，此脚本可能返回 4/28/2006。

```
$DateString
```

$TimeString 系统标记

返回本地操作系统“区域设置”中所定义的系统时间。

语法

```
$TimeString
```

数据类型

字符串（只读）

示例

根据操作系统“区域设置”中的时间格式设置，此脚本可能返回 02:40:37 PM。

```
$TimeString
```

UTCDateTime() 函数

返回 UTC 时间、UTC 日期与时间，或本地时区。

语法

```
result = UTCDateTime (format)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

format
确定返回的内容。字符串值、消息标记名，或带以下可能值的字符串表达式：

UTC_SHORT - 函数返回 UTC 时间

UTC_LONG - 函数返回 UTC 日期与时间

UTC_LOCAL - 函数返回本地操作系统时区设置中所设置的时区的名称

任何其它值返回缺省格式的 UTC 日期与时间 (ddd mm dd hh:mm:ss yyyy)。

示例

在太平洋时区 2003 年 1 月 6 号星期一上午 09:24，UTCDateTime() 函数返回以下内容。

此脚本返回 17:24:05
```
UTCDateTime("UTC_SHORT")
```

此脚本返回 01/06/2003 17:24:05
```
UTCDateTime("UTC_LONG")
```

此脚本返回 Pacific Standard Time -8:0:1
```
UTCDateTime("UTC_LOCAL")
```

此脚本返回 Mon Jan 06 17:24:05 2003。
```
UTCDateTime("Invalid")
```

将日期与时间信息转换为字符串

在脚本中，可以将日期与时间信息转换为字符串，以便于解释及满足显示要求。您可以使用以下函数。

使用对象	作用
StringFromTime()	将 UTC 时间标签转换为本地时间并作为时间字符串返回。
wwStringFromTime()	将本地时间标签转换为 UTC 时间并将它作为时间字符串返回。
StringFromTimeLocal()	将时间标签转换为时间字符串。

StringFromTime() 函数

将 UTC 时间格式的时间标签转换为本地时间并返回作为字符串的结果。此函数支持“夏令时”。

备注：此函数等同于 StringFromGMTTimeToLocal() 函数。

语法

```
result = StringFromTime (timestamp, format)
```

参数

timestamp
自 1970 年 1 月 1 日午夜以来经过的秒数（UTC 时区）。整数值、整型标记名或整型表达式。

format
确定如何显示字符串结果。1 到 5 范围内的整数值、整型标记名或整型表达式，含义如下：

1 - 根据本地操作系统“区域设置”中所设置的格式显示日期

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2 - 根据本地操作系统“区域设置”中所设置的格式显示时间

3 - 按 24 个字符的字符串格式显示日期与时间 (ddd mmm dd hh:mm:ss yyyy)

4 - 按短格式显示星期几

5 - 按长格式显示星期几

示例

本例假设本地节点上的时区是“太平洋标准时间”(PST, UTC-0800)。传递给函数的 UTC 时间是 1970 年 1 月 2 日星期五上午 12 点。由于 PST 比 UTC 晚 8 小时，所以函数会返回以下结果。

此脚本返回 "1/1/70"
```
StringFromTime(86400,1)
```

此脚本返回 "04:00:00 PM"
```
StringFromTime(86400,2)
```

此脚本返回 "Thu Jan 01 16:00:00 1970"
```
StringFromTime(86400,3)
```

此脚本返回 "Thu"
```
StringFromTime(86400,4)
```

此脚本返回 "Thursday"
```
StringFromTime(86400,5)
```

wwStringFromTime() 函数

将本地时间格式的时间标签转换为 UTC 时间并返回作为字符串的结果。此函数支持“夏令时”。

语法

```
result = wwStringFromTime (timestamp, format)
```

参数

timestamp
自本地时区 1970 年 1 月 1 日午夜以来经过的秒数。整数值、整型标记名或整型表达式。

format
确定如何显示字符串结果。1 到 5 范围内的整数值、整型标记名或整型表达式，含义如下：

1 - 根据本地操作系统“区域设置”中所设置的格式显示日期

2 - 根据本地操作系统“区域设置”中所设置的格式显示时间

3 - 按 24 个字符的字符串格式显示日期与时间 (ddd mmm dd hh:mm:ss yyyy)

4 - 按短格式显示星期几

5 - 按长格式显示星期几

示例

本例假设本地节点上的时区是“太平洋标准时间”(PST, UTC-0800)。传递给函数的本地时间是 1970 年 1 月 1 日星期四下午 04:00:00。因为 PST 比 UTC 晚 8 小时，函数返回以下结果。

此脚本返回 “1/2/70”
```
wwStringFromTime(57600,1)
```

此脚本返回 “12:00:00 AM”
```
wwStringFromTime(57600,2)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此脚本返回 “Fri Jan 02 00:00:00 1970”

```
wwStringFromTime(57600,3)
```

此脚本返回 “Fri”

```
wwStringFromTime(57600,4)
```

此脚本返回 “Friday”

```
wwStringFromTime(57600,5)
```

StringFromTimeLocal() 函数

将时间标签转换为时间并返回作为字符串的结果。

语法

```
result = StringFromTimeLocal (timestamp, format)
```

参数

timestamp
自 1970 年 1 月 1 日午夜以来经过的秒数。整数值、整型标记名或整型表达式。

format
确定如何显示字符串结果。1 到 5 范围内的整数值、整型标记名或整型表达式，含义如下：

1 - 根据本地操作系统“区域设置”中所设置的格式显示日期

2 - 根据本地操作系统“区域设置”中所设置的格式显示时间

3 - 按 24 个字符的字符串格式显示日期与时间 (ddd mmm dd hh:mm:ss yyyy)

4 - 按短格式显示星期几

5 - 按长格式显示星期几

示例

此脚本返回 “1/2/70”

```
StringFromTimeLocal(86400,1)
```

此脚本返回 “12:00:00 AM”

```
StringFromTimeLocal(86400,2)
```

此脚本返回 “Fri Jan 02 00:00:00 1970”

```
StringFromTimeLocal(86400,3)
```

此脚本返回 “Fri”

```
StringFromTimeLocal(86400,4)
```

此脚本返回 “Friday”

```
StringFromTimeLocal(86400,5)
```

检查夏令时状态

在脚本中，通过使用 wwIsDaylightSaving() 函数可以检查是否在使用夏令时。

wwIsDaylightSaving() 函数

返回当前是否在使用夏令时。

语法

```
result = wwIsDaylightSaving()
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

返回值

含义如下的布尔值：

0 - “夏令时”未在使用。

1 - “夏令时”正在使用。

与其它应用程序交互

在脚本中，通过使用各种 QuickScript 可以与其它 Windows 应用程序进行交互。例如，可以：

- 启动应用程序<1>，</1>如<2>“</2>记事本<3>”。</3>
- 选择应用程序标题名。
- 检查某个应用程序是否正在运行。
- 激活正在运行的应用程序。
- 模拟键击。
- 关闭、最小化或最大化应用程序窗口。
- 执行命令并与支持 DDE 的应用程序交换数据。

启动 Windows 应用程序

在脚本中，可以使用 StartApp 命令启动 Windows 应用程序。

语法

```
StartApp appname;
```

参数

appname
要启动的应用程序的路径与文件名。字符串值、消息标记名或字符串表达式。

注意：您需要知道应用程序的路径与文件名。如果应用程序所在的目录是 Windows PATH 环境变量的一部分，则只要传递文件名（不需要路径）。

示例

此脚本启动 Microsoft 的<1>“</1>计算器<2>”。</2>

```
StartApp "calc"
```

检索正在运行的应用程序的应用程序标题

在脚本中，通过使用 InfoAppTitle() 函数，可以找到指定的正在运行的应用程序的应用程序标题或 Windows 任务列表名。举例来说，可以通过 InTouch 脚本来获取此信息，然后利用此信息来检查指定的应用程序当前是否正在运行，或用来激活它。

注意：此函数不会为 Internet Explorer 返回值。作为变通，请使用 Google Chrome 浏览器。

InfoAppTitle() 函数

返回指定的正在运行的应用程序的应用程序标题或 Windows 的任务列表名。

语法

```
result = InfoAppTitle (appname)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

appname
不带 .exe 扩展名的应用程序名。字符串值、消息标记名或字符串表达式。

示例

此脚本返回“计算器”
```
InfoAppTitle("calc")
```

此脚本返回 “Microsoft Excel”
```
InfoAppTitle("excel")
```

检查应用程序是否正在运行

在脚本中，通过使用 InfoAppActive() 函数可以检查指定的应用程序是否已在运行。您需要先知道应用程序标题或 Windows 任务列表名，然后才可以检查特定的应用程序是否正在运行。

InfoAppActive() 函数

返回应用程序的运行状态。

语法
```
result = InfoAppActive (apptitle)
```

参数

apptitle
希望查询其运行状态的应用程序的应用程序标题或 Windows 任务列表名。字符串值、消息标记名或字符串表达式。

返回值

含义如下的布尔值：

0 - 应用程序未在运行

1 - 应用程序正在运行

示例

此脚本查询“记事本”应用程序；如果已在运行，则激活它。否则该脚本启动新的“记事本”实例。这就避免了多次启动“记事本”。
```
IF InfoAppActive(InfoAppTitle("Notepad"))==1
THEN
   ActivateApp InfoAppTitle( "Notepad" );
ELSE
   StartApp "Notepad";
ENDIF;
```

激活正在运行的 Windows 应用程序

在脚本中，通过使用 ActivateApp() 函数，可以激活正在运行的 Windows 应用程序。这会将指定的应用程序带到前台并将焦点切换到它上面。

激活正在运行的 Windows 应用程序之前，需要执行以下操作：

- 查找应用程序标题或 Windows 任务列表名。请参阅检索正在运行的应用程序的应用程序标题。
- 确保 Windows 应用程序正在运行。请参阅检查应用程序是否正在运行。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ActivateApp 函数

激活已在运行的 Windows 应用程序。

语法

```
ActivateApp apptitle;
```

参数

apptitle
希望激活的正在运行的应用程序的应用程序标题或 Windows 任务列表名。

示例

此脚本检查命令提示符窗口是否已打开；如果是，则激活它。否则启动命令提示符窗口。

```
IF InfoAppActive( InfoAppTitle("cmd")) == 1 THEN
   ActivateApp InfoAppTitle("cmd");
ELSE
   StartApp "cmd";
ENDIF;
```

将模拟键击发送到应用程序

在脚本中，可以模拟键盘上的按键序列。例如，您可以使用它来：

- 在打开的应用程序中自动输入数据
- 控制任何应用程序（包括 InTouch HMI）。

SendKeys 函数

模拟键击序列。

重要：SendKeys() 函数不适用于 64 位版本的 Windows 操作系统。

语法

```
SendKeys sequence;
```

参数

sequence
要模拟的键击序列。字符串值、消息标记名或字符串表达式。
除键盘上的常规字符（如字母数字字符）之外，也可以使用代码来指定控制键：

```
   {BACKSPACE} - 模拟 Backspace 键
   {BREAK} - 模拟 Break 键
   {CAPSLOCK} - 模拟 Caps Lock 键
   {DELETE} - 模拟 Delete 键（或 {DEL}）
   {DOWN} - 模拟下箭头键
   {END} - 模拟 End 键
   {ENTER} - 模拟 Enter 键（或 ~）
   {ESCAPE} - 模拟 ESC 键（或 {ESC}）
   {F1} ..{F12} - 模拟 F1 ..F12 keys
   {HOME} - 模拟 Home 键
   {INSERT} - 模拟 Insert 键
   {LEFT} - 模拟左箭头键
   {NUMLOCK} - 模拟 Num Lock 键
   {PGDN} - 模拟 Page Down 键
   {PGUP} - 模拟 Page Up 键
   {PRTSC} - 模拟 Print Screen 键
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
{RIGHT} - 模拟右箭头键
{TAB} - 模拟 Tab 键
{UP} - 模拟上箭头键
+ - 模拟 Shift 键
同要和 Shift 键构成组合键并用括号括起来的键一起使用。
^ - 模拟 Ctrl 键
同要和 Ctrl 键构成组合键并用括号括起来的键一起使用。
% - 模拟 Alt 键
同要和 Alt 键构成组合键并用括号括起来的键一起使用。
```

附注

使用 StartApp 与/或 ActivateApp() 命令，在将模拟的键击发送到另一个应用程序之前将它激活。

示例

此脚本模拟按 B 键。

```
SendKeys "b";
```

此脚本模拟按 Ctrl 与 P 组合键，这可用于在另一个应用程序中启动“打印”对话框。

```
SendKeys "^(p)";
```

此脚本模拟按 F1（可以打开帮助功能），按 Tab 键（可以将光标放入搜索字段），输入 HAL，然后按 Enter 键（可以启动搜索）。

```
SendKeys "{F1}{TAB}HAL{ENTER}";
```

此脚本模拟按 Ctrl、Shift、1 组合键，这与切换到 WindowMaker 效果相同。这项强大的组合可以用于开发可自我修改的（动态的） InTouch HMI 应用程序。

```
SendKeys "^(+(1))";
```

关闭、最小化或最大化应用程序窗口

在脚本中，通过使用 WWControl() 命令，可以关闭、最小化或最大化其它 Windows 应用程序。

在关闭、最小化或最大化 Windows 应用程序之前，需要执行以下操作：

- 查找应用程序标题或 Windows 任务列表名。请参阅检索正在运行的应用程序的应用程序标题。
- 确保该 Windows 应用程序正在运行。请参阅检查应用程序是否正在运行。

WWControl() 函数

还原、最小化、最大化或关闭 Windows 应用程序。

语法

```
WWControl (apptitle, control);
```

参数

apptitle
要还原、最小化、最大化或关闭的运行中应用程序的应用程序标题或 Windows 任务列表名。字符串值、消息标记名或字符串表达式。

control
确定要对指定的 Windows 应用程序执行的操作。使用以下值的字符串值、消息标记名或字符串表达式：

Restore - 激活并显示应用程序窗口

Minimize - 激活并最小化应用程序窗口

Maximize - 激活并最大化应用程序窗口

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Close - 关闭应用程序

示例

如果计算器应用程序已在运行，此脚本可以将它还原。

```
WWControl ("Calculator","Restore");
```

此脚本关闭 WindowViewer。

```
WWControl (InfoAppTitle("View"),"Close");
```

使用 DDE 执行命令与交换数据

您可以编写一个脚本，同支持 DDE 的应用程序进行交互。

使用对象	作用
WWExecute()	发送与执行命令。
WWRequest()	从 DDE 项目读取数据。
WWPoke()	向 DDE 项目写入数据。

WWExecute() 函数

向应用程序发送命令、执行它并返回状态结果。您可以使用它来让 Excel 运行某个宏。

重要：WWExecute() 函数不适用于 64 位版本的 Windows 操作系统。

语法

```
Result = WWExecute (appname, topic, command)
```

参数

appname
命令所发送到的应用程序的名称。字符串值、消息标记名或字符串表达式。

topic
命令所发送到的应用程序内的主题的名称。字符串值、消息标记名或字符串表达式。

command
要发送的命令。字符串值、消息标记名或字符串表达式。

返回值

值 -1、0 或 1 的含义如下：
-1 - 命令未成功执行。可能的原因是应用程序未在运行、主题不存在或命令有错误。
0 - 应用程序正忙，导致命令未成功执行。
1 - 命令成功执行。

示例

此脚本通过向 Excel 发送命令 [Run("Macro1”,0)] 指示 Microsoft Excel 来执行 Macro1 这个宏。

```
Macro="Macro1";
Command="[Run(" + StringChar(34) + Macro + StringChar(34) + ",0)]";
WWExecute("excel","system",Command);
```

WWRequest() 函数

从应用程序的项目中读取数据。例如，您可以使用它来读取 Microsoft Excel 电子表格单元格的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

重要：WWRequest() 函数不适用于 64 位版本的 Windows 操作系统。

语法

```
Result = WWRequest(appname, topic, item, messagetag)
```

参数

appname
应用程序的名称。字符串值、消息标记名或字符串表达式。

topic
应用程序中主题的名称。字符串值、消息标记名或字符串表达式。

item
属于该主题与应用程序的项目的名称。字符串值、消息标记名或字符串表达式。

messagetag
检索项目值的消息标记名。通过使用 StringToIntg() 或 StringToReal() 函数，可以将消息标记名值转换成整数或实数值。

返回值

值 -1、0 或 1 的含义如下：
-1 - 数据未成功读取。可能的原因是应用程序未在运行，或是主题或项目不存在。
0 - 应用程序正忙，导致数据未成功读取。
1 - 数据读取成功。

示例

此脚本将 Microsoft Excel 工作薄 Book1.xls 工作表 Sheet1 第 1 行第 1 列中的值读取到消息标记名 MTag，并将值放入实型标记名 CellValue 中。

```
Result = WWRequest("excel","[Book1.xls]sheet1", "r1c1",Mtag);
CellValue=StringToReal(MTag);
```

如果使用非英文操作系统，则在将 *MTag* 转换为不同的数据类型之前，可能需要使用 StringReplace() 函数来更改其内容。例如，对于使用英文逗号作为小数点的操作系统，在将 *MTag* 转换为实型数据类型之前，可能需要使用小数点替换其中的所有逗号。

WWPoke() 函数

向应用程序的项目写入数据。例如，您可以使用它将值写入 Microsoft Excel 中的电子表格单元格。

重要事项：WWPoke() 函数不适用于 64 位版本的 Windows 操作系统。

语法

```
result = WWPoke (appname, topic, item, string)
```

参数

appname
应用程序的名称。字符串值、消息标记名或字符串表达式。

topic
应用程序中主题的名称。字符串值、消息标记名或字符串表达式。

item
属于该主题与应用程序的项目名。字符串值、消息标记名或字符串表达式。

string

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要写入的值。字符串值、消息标记名或字符串表达式。通过使用 StringFromIntg()、StringFromReal() 或 Text() 函数，可以将整数或实型标记名的值转换为消息标记名。

返回值

值 -1、0 或 1 的含义如下：
-1 - 数据未成功写入。可能的原因是应用程序未在运行，或是主题或项目不存在。
0 - 应用程序正忙，导致数据未成功写入。
1 - 数据写入成功。

附注

请勿使用 WWPoke() 或 WWRequest() 函数在不同节点或会话上的 InTouch 应用程序之间读取和写入数据。要在 InTouch 应用程序之间读取和写入数据，请使用“访问名”来替代。请参阅设置访问名。

示例

此脚本将实型标记名 CellValue 的值放入消息标记名 Mtag 中，并将该值写入 Microsoft Excel 工作薄 Book1.xls 中工作表 Sheet1 的第 1 行第 1 列这个电子表格单元格。

```
MTag = Text(CellValue,"0");
Result = WWPoke("excel","[Book1.xls]sheet1", "r1c1",Mtag);
```

处理文件

您可以使用各种文件管理与访问操作来编写脚本。

使用对象	作用
FileCopy()	复制文件。
FileDelete()	删除文件。
FileMove()	移动文件。
FileReadFields()、FileWriteFields()	读取/写入 csv 数据。
FileReadMessage()、FileWriteMessage()	读取/写入文本数据。

管理文件

在脚本中，可以复制、删除或移动文件。

FileCopy() 函数

将源文件复制到目标文件，并返回状态结果。此函数可能要花费较长时间并分为多个阶段来执行：

1. 调用 FileCopy() 函数并返回中间结果，指出文件复制初始化操作是成功还是失败。
2. FileCopy() 函数在后台执行复制过程，InTouch 脚本在复制文件的过程中继续执行。您可以使用一个整型标记来监视文件复制进度。
3. FileCopy() 函数返回文件复制结果，指出文件复制过程是成功还是失败。

如果目标文件夹不可用（例如，在网络中的另一台计算机上），则该函数最多等待 10 秒便会超时，然后向 Logger 发送一条消息。

注意：请勿在异步 QuickFunction 中使用 FileCopy() 函数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
result = FileCopy (sourcefile, destfile, progresstag)
```

参数

sourcefile
要复制的文件的完整路径与文件名。字符串值、消息标记名或字符串表达式。您可以在此参数中使用通配符（* 与 ?）只复制与指定的准则匹配的文件。路径名也可以是 UNC 路径名。

destfile
目标文件的完整路径与文件名（或仅为路径名）。字符串值、消息标记名或字符串表达式。路径名也可以是 UNC 路径。

progresstag
括在英文双引号中的整型标记的名称，该标记将包含指示文件复制进度的值。字符串值、消息标记名（如包含 "IntTag.Name" 值的消息标记）或字符串表达式。这些值的含义如下：
0 - FileCopy() 过程仍在进行。
1 - FileCopy() 过程成功完成。
-1 - FileCopy() 过程已完成，但发生了错误。

返回值

值 -1、0 或 1 的含义如下：
1 - FileCopy() 函数调用成功。
0 - 调用 FileCopy() 函数时发生错误，原因是另一个 FileCopy() 过程已经在进行。
-1 - 调用 FileCopy() 函数时发生错误，原因是源文件不存在或目标文件为只读。

示例

此脚本将 c:\MyData\output.log 文件复制到 d:\archive 目录，并将该文件重命名为 output.txt。文件复制的进度写入整型标记 Monitor。

```
Status=FileCopy("c:\MyData\output.log","d:\archive\output.txt","Monitor");
```

此脚本将 c:\ root 目录中以 .txt 结尾的所有文件复制到目标目录 c:\Backup。

```
Status=FileCopy("c:\*.txt", "c:\Backup", "Monitor");
```

此脚本将完整的路径与文件名包含在消息标记 LogFile 中的文件复制到目标目录 c:\results\，并将它重命名为 logxxx.txt，其中 xxx 是时间标签。

```
Status=FileCopy(LogFile, "c:\results\log" + $DateString + $TimeString + ".txt", "Monitor");
```

FileDelete() 函数

删除单独的文件。

语法

```
result = FileDelete (filename)
```

参数

filename
要删除的文件的路径名与文件名。字符串值、消息标记名或字符串表达式。支持 UNC 路径名。

附注

请勿在 FileDelete() 函数中使用通配符（* 与 ?），也不要在异步 QuickFunction 中使用 FileDelete() 函数。

FileDelete() 函数不删除目录。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

返回值

指出文件删除操作是成功还是失败的值：
1 - 文件成功删除
0 - 文件删除失败。 可能的原因是试图删除只读或不存在的文件。

示例

此脚本删除文件 c:\Data.txt， 如果找到该文件并成功删除则返回 1。

```
Status=FileDelete("c:\Data.txt");
```

FileMove() 函数

将源文件移到目标文件并返回状态结果。 它也可以用于重命名文件。 此函数可能要花费较长时间并分为多个阶段来执行：

1. 调用 FileMove() 函数并返回中间结果， 指出文件移动初始化操作是成功还是失败。
2. FileMove() 函数在后台执行移动过程， InTouch 脚本在文件移动的过程中继续执行。 您可以使用整型标记来监视文件移动进度。
3. FileMove() 函数返回文件移动结果， 指出文件移动过程是成功还是失败。

请勿在异步 QuickFunctions 中使用 FileMove() 函数。

语法

```
result = FileMove (sourcefile, destfile, progresstag)
```

参数

sourcefile
要移动的文件的完整路径与文件名。 字符串值、消息标记名或字符串表达式。 您可以在此参数中使用通配符（* 与 ?）只移动与指定的准则匹配的文件。 路径名也可以是 UNC 路径名。

destfile
目标文件的完整路径与文件名（或仅为路径名）。 字符串值、消息标记名或字符串表达式。 路径名也可以是 UNC 路径。

progresstag
括在英文双引号中的整型标记的名称， 该标记将包含指出文件移动进度的值。 字符串值、消息标记名（例如， 包含 "IntTag” 值的消息标记名）或字符串表达式。 这些值的含义如下：

0 - FileMove() 过程仍在进行

1 - FileMove() 过程成功完成

-1 - FileMove() 过程已完成， 但发生了错误

返回值

值 -1、0 或 1 的含义如下：
1 - FileMove() 函数调用成功
0 - 调用 FileMove() 函数时发生错误， 原因是另一个 FileMove() 过程已经在进行
-1 - 调用 FileMove() 函数时发生错误。 可能的错误是试图移动不存在的文件。

示例

此脚本将 c:\MyData\output.log 文件移到 d:\archive 目录, 并将该文件重命名为 output.txt。 文件移动的进度写入整型标记 Monitor。

```
Status=FileMove("c:\MyData\output.log","d:\archive\output.txt","Monitor");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此脚本将 c:\ root 目录中以 .txt 结尾的所有文件移到目标目录 c:\Backup。

```
Status=FileMove("c:\*.txt", "c:\Backup", "Monitor");
```

此脚本将完整的路径与文件名包含在消息标记 LogFile 中的文件移到目标目录 c:\results\，并将它重命名为 logxxx.txt，其中 xxx 是时间标签。

```
Status=FileMove(LogFile, "c:\results\log" + $DateString + $TimeString + ".txt", "Monitor");
```

读取和写入 CSV 数据

通过使用 FileReadFields() 与 FileWriteFields() 函数，可以编写脚本将 csv（逗号分隔变量）文件中包含的数据读取到一系列的标记名中，也可以将一系列标记名中的数据写入 csv 文件。

FileReadFields() 与 FileWriteFields() 函数仅支持逗号作为分隔符。

FileReadFields() 函数

将 csv 文件中包含的值读取到一系列的标记名中。您可以使用此函数来加载一组标记名值。

逗号是支持的唯一分隔符。

此函数仅能用于同步调用。

语法

```
[result = ] FileReadFields (filename, offset, starttag, numberoffields)
```

参数

filename
要从中读取数据的 csv 文件的名称。字符串值、消息标记名或字符串表达式。

offset
文件中要开始读取的位置（以字节计）。整数值、整型标记名或整型表达式。

starttag
接收第一个读取数据项的第一个标记名的名称。标记名必须用英文双引号括起且以数字结尾，如 "MyTag1"。字符串值、消息标记名（例如，包含 "MyTag1" 值的消息标记名）或字符串表达式。

numberoffields
要从 csv 文件中读取的数据项的数量。整数值、整型标记名或整型表达式。第一个数据项读取到 starttag 参数定义的标记名，后续的数据项读取到编号为 starttag 参数后缀的增量的标记名（MyTag1、MyTag2、MyTag3...）。

返回值

可选的新文件读取数据之后的偏移量（以字节计）。这可以用于读取下一组数据。

示例

如果 csv 文件 c:\set.csv 包含以下数据：Flour、27.23、14、1，且定义了以下标记：RecipeTag1:message、RecipeTag2:real、Recipe3:integer、RecipeTag4:discrete，则此脚本将 "Flour" 值读取到 RecipeTag1、将 27.23 读取到 RecipeTag2、将 14 读取到 RecipeTag3、将 1 读取到 RecipeTag4，并返回新文件偏移量。

```
FileReadFields("c:\set.csv",0,"RecipeTag1",4);
```

FileWriteFields() 函数

将一系列标记名中包含的值写入 csv 文件。您可以使用此函数保存一组标记名值。

逗号是支持的唯一分隔符。

语法

```
[result = ] FileWriteFields (filename, offset, starttag, numberoffields)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

filename
要写入数据的 csv 文件的名称。如果先前不存在，则创建新文件。字符串值、消息标记名或字符串表达式。

offset
文件中要开始写入的位置（以字节计）。使用 -1 可写入文件末尾（附加）。整数值、整型标记名或整型表达式。

starttag
包含要写入的第一个数据项的第一个标记名的名称。标记名必须用英文双引号括起且以数字结尾，如 "MyTag1"。字符串值、消息标记名（例如，包含 "MyTag1" 值的消息标记名）或字符串表达式。

numberoffields
要写入 csv 文件的数据项的数量。整数值、整型标记名或整型表达式。第一个数据项从 starttag 参数定义的标记名写入文件，后续的数据项从后缀为 starttag 参数的增量的标记名（MyTag1、MyTag2、MyTag3...）写入。

返回值

可选的新文件写入数据之后的偏移量（以字节计）。这可以用于写入下一组数据。

示例

一系列 InTouch 标记定义如下：

标记名	数据类型	值
RecipeTag1	消息	Flour
RecipeTag2	实型	27.23
RecipeTag3	整型	14
RecipeTag4	离散	1

此脚本将 RecipeTag1 到 RecipeTag4 中包含的值写入 csv 文件 c:\set.csv。

```
FileWriteFields("c:\set.csv",0,"RecipeTag1",4);
```

使文件 c:\set.csv 包含以下数据：

```
Flour,27.23,14,1
```

读取和写入文本数据

通过使用 FileReadMessage() 与 FileWriteMessage() 函数，可以编写脚本从文件中读取文本数据，以及将文本数据写入文件。您可以读取/写入指定数量的字节，也可以是整行的文本（用换行符分割）。

FileReadMessage() 函数

从文件中读取指定字节数（或一行）的字符串数据。

语法

```
[result = ] FileReadMessage (filename, offset, messagetag, charstoread)
```

参数

filename
要从中读取数据的文件的名称。字符串值、消息标记名或字符串表达式。

offset

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

文件中要开始读取的位置（以字节计）。整数值、整型标记名或整型表达式。

messagetag
从文件接收第一行或一定字节数的消息标记名。在工业图形编辑器脚本编辑器内使用函数时，用双引号将标记名括起来。

charstoread
从文件读取的字节数。将它设置为 0 可以一直读取到下一个换行符 (LF)。整数值、整型标记名或整型表达式。

返回值

包含读取之后的新字节位置。您可以用它来从文件中继续进行读取。

示例

此脚本将 c:\Data\File.txt 文件中的第一行数据读取到消息标记名 MsgTag。

```
FileReadMessage ("c:\Data\File.txt",0,MsgTag, 0);
FileReadMessage ("c:\Data\File.txt",0,"InTouch:MsgTag", 0);
```

FileWriteMessage() 函数

将指定字节数（或一行）的字符串数据写入文件。

语法

```
[result = ] FileWriteMessage (filename, offset, messagetag, linefeed)
```

参数

filename
要写入数据的文件的名称。字符串值、消息标记名或字符串表达式。

offset
文件中要开始写入的位置（以字节计）。将它设置为 -1 可以将数据写入文件的末尾（附加）。整数值、整型标记名或整型表达式。

messagetag
包含要写入文件的数据的消息标记名。

linefeed
指定在将数据写入文件之后是否写入换行符 (LF)。设置为 1 可写入换行符；否则，设置为 0。文本布尔值、离散标记名或布尔表达式。

返回值

包含写入之后的新字节位置。您可以用它来继续写入文件。

示例

此脚本将消息标记名 MsgTag 的值写入 c:\Data\File.txt 文件的末尾。

```
FileWriteMessage("c:\Data\File.txt",-1,MsgTag,1);
```

检索系统相关信息

在脚本中，可以使用以下 QuickFunction 来检索系统相关信息。

使用对象	作用
GetNodeName()	检索计算机的节点名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用对象	作用
InfoDisk()	检索磁盘空间信息。
InfoFile()	检索文件的有关信息。

检索计算机的节点名

在脚本中，通过使用 GetNodeName() 函数可以检索计算机的节点名。例如，在处理访问名时，这可以用于使 InTouch 应用程序保持动态。

GetNodeName() 函数

返回计算机的节点名。

语法

```
GetNodeName (messagetag, nodenum);
```

参数

messagetag
将包含节点名的消息标记名。在工业图形编辑器脚本编辑器内使用函数时，用双引号将标记名括起来。

nodenum
要从节点名检索的字符数。0 到 131 范围内的整数值、整型标记名或整型表达式。

示例

此脚本检索节点名，并将它指定给 NodeName 消息标记名。

```
GetNodeName(NodeName,131);
GetNodeName("InTouch:NodeName",131);
```

检索磁盘空间信息

在脚本中，通过使用 InfoDisk() 函数可以检索磁盘空间信息。您可以检索：

- 磁盘驱动器的总计大小（以字节或千字节计）。
- 磁盘驱动器的可用空间（以字节或千字节计）。

通过指定触发器标记，也可以确定在什么时间、按什么频率来更新信息（在动画链接中）。

InfoDisk() 函数

返回本地或网络磁盘驱动器的总计空间或可用空间。

语法

```
result = InfoDisk (drive, infotype, trigger);
```

参数

drive
要检索其信息的盘符。仅使用字符串的第一个字符。字符串值、消息标记名或字符串表达式。

infotype
指定信息类型。使用以下可能值的整数值、整型标记名或整型表达式：

1 - 函数返回磁盘驱动器的总计大小（以字节计）

2 - 函数返回磁盘驱动器的可用空间（以字节计）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3 - 函数返回磁盘驱动器的总计大小（以千字节计）

4 - 函数返回磁盘驱动器的可用空间（以千字节计）

trigger
用作重新计算磁盘信息的触发器的标记名（或表达式）。如果触发器值发生改变，则重新计算磁盘信息。离散或模拟标记名，或是离散或模拟表达式。

附注

触发器标记仅当 InfoDisk() 函数用在动画显示链接中时才有意义。如果此函数用在脚本中，则可以指定任何数值、模拟型标记名或数值表达式。

示例

在动画显示链接中使用此脚本显示磁盘驱动器 C 的可用空间，并且每分钟更新一次信息。

```
InfoDisk("C", 4, $Minute)
```

检索文件或目录的有关信息

在脚本中，通过使用 InfoFile() 函数，可以检索特定文件或目录的有关信息。通过使用不同的参数，可以查找：

- 文件是否存在。
- 指定的文件名实际上是不是一个目录。
- 文件大小（以字节计）。
- 文件或目录的时间标签。
- 与通配符搜索匹配的文件数。

InfoFile() 函数

返回文件或目录的各种有关信息。

语法

```
result = InfoFile (filename, infotype, trigger)
```

参数

filename
要检索相关信息的文件或目录的完整文件名或目录名。字符串值、消息标记名或字符串表达式。也可以包含通配符，如 "*" 与 "?”。

infotype
您希望检索的关于指定文件或目录的信息类型。使用以下可能值且含义如下的整数值、整型标记名或整型表达式：

1 - 存在。如果文件存在，InfoFile() 函数返回 1；如果文件是目录，返回 2；如果文件或目录不存在，返回 0。

2 - 大小。InfoFile() 函数返回文件大小（以字节计）。

3 - 创建时间标签。InfoFile() 函数返回时间标签（使用自 1970 年 1 月 1 日午夜以来经过的秒数表示）。使用 StringFromTimeLocal() 函数将此值转换为消息时间标签。

4 - 通配符搜索匹配。InfoFile() 函数返回同指定的通配符搜索相匹配的文件数。

trigger

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用作重新计算文件信息的触发器的标记名（或表达式）。如果触发器值发生改变，则重新计算文件信息。离散或模拟标记名，或是离散或模拟表达式。

附注

触发器标记仅当 InfoFile() 函数用在动画显示链接中时才有意义。如果此函数用在脚本中，则可以指定任何数值、模拟型标记名或数值表达式。

示例

此脚本在 c:\data\log.txt 文件存在时返回 1。

```
InfoFile("c:\data\log.txt",1,$minute)
```

如果 c:\data\log.txt 文件的大小为 14223 字节，此脚本返回 14223。

```
InfoFile("c:\data\log.txt",2,$minute)
```

如果 c:\data\log.txt 文件是在 2006 年 1 月 26 日上午 11:14:26 创建的，此脚本返回 1138245266。

```
InfoFile("c:\data\log.txt",3,$minute)
```

如果 c:\data\ 目录中以 txt 结尾的文件有 14 个，此脚本返回 14。

```
InfoFile("c:\data\*.txt",4,$minute)
```

检索 InTouch 相关信息

在脚本中，可以使用以下函数来检索 InTouch 相关信息。

使用对象	作用
InfoInTouchAppDir()	获取正在开发的 InTouch 应用程序所在目录的有关信息。
InTouchVersion()	获取 InTouch 版本信息。

检索 InTouch 应用程序目录的名称

在脚本中，通过使用 InfoInTouchAppDir() 函数，可以检索 InTouch 应用程序所在目录的名称。对于查找要随您的 InTouch 应用程序一起交付的任何外部文件，此函数很有用。

InfoInTouchAppDir() 函数

返回当前 InTouch 应用程序目录。

语法

```
result = InfoInTouchAppDir();
```

返回值

包含当前运行的 InTouch 应用程序所在目录的消息标记名。

附注

由于 131 个字符的限制，传递给消息标记名或在动画链接中显示时，应用程序目录名可能被截断。

示例

此脚本可能返回 c:\documents and settings\user1\my documents\my intouch applications\packaging。

```
InfoInTouchAppDir()
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

检索 InTouch 版本

在脚本中，通过使用 Version() 函数可以检索当前正在运行的 InTouch 应用程序的版本号。

InTouchVersion() 函数

返回完整的 InTouch 版本号或其一部分。

语法

```
result = InTouchVersion (infotype);
```

参数

infotype
指定如何返回版本信息。含义如下的整数值、整型标记名或整型表达式：

0- 函数返回整个版本号

1- 函数仅返回主版本号

2- 函数仅返回次版本号

3- 函数仅返回补丁级别

4- 函数仅返回内部版本级别

示例

功能	可能的结果
InTouchVersion(0)	10.5.1626.0521.0045.0012
InTouchVersion(1)	10
InTouchVersion(2)	5
InTouchVersion(3)	0
InTouchVersion(4)	1626

安全性相关脚本

您可以使用各种 QuickScript 函数与系统标记在 InTouch 中添加与管理安全性。如需有关安全性函数的详细信息，请参阅《*AVEVA™ InTouch HMI 管理*》中的保护 InTouch 安全。

登录和注销

您可以使用以下函数与系统标记进行登录与注销。

使用对象	作用
AttemptInvisibleLogon()	通过在参数中提供身份验证数据将用户登录进来。
LogonCurrentUser()	将当前已登录到 Windows 的用户（如果身份验证模式为 "OS"）登录进来。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用对象	作用
PostLogonDialog()	显示登录对话框。
Logoff()	注销当前用户。
$PasswordEntered	设置口令。
$OperatorEntered	设置有效的用户名。
$OperatorDomainEntered	设置有效的用户域名（如果身份验证模式为 "OS"）。

如需有关安全性函数的详细信息，请参阅《*AVEVA™ InTouch HMI 管理*》中的保护 InTouch 安全。

更改和设置口令

您可以使用以下函数与系统标记来更改口令：

使用对象	作用
ChangePassword()	为当前登录的用户调用改变口令对话框。
$ChangePassword	为当前登录的用户调用改变口令对话框。

如需有关安全性函数的详细信息，请参阅《*AVEVA™ InTouch HMI 管理*》中的保护 InTouch 安全。

指定与配置用户

您可以使用以下系统标记指定与配置用户。

使用对象	作用
$ConfigureUsers	调用配置用户对话框。

如需有关安全性函数的详细信息，请参阅《*AVEVA™ InTouch HMI 管理*》中的保护 InTouch 安全。

管理安全性及其它信息

您可以使用以下系统标记与函数来管理安全性。

使用对象	作用
$AccessLevel	检索当前登录的用户的访问级别。
AddPermission()	将访问级别指定给特定的用户组（本地/域）。
GetAccountStatus()	检索帐户信息（口令过期、锁定、禁用标帜）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用对象	作用
$InactivityTimeout	显示在用户自动注销之前经过的时间。
$InactivityWarning	显示超时警告的时间。
InvisibleVerifyCredentials()	检索操作系统用户的 InTouch 访问级别信息。
IsAssignedRole()	查看当前登录的用户是否具有特定用户角色。
QueryGroupMembership()	查看当前登录的用户是不是特定用户角色的成员。

如需有关安全性函数的详细信息，请参阅《*AVEVA™ InTouch HMI 管理*》中的保护 InTouch 安全。

其它脚本

InTouch 脚本支持声音输出，这样就可以通过声音进行人机交互。InTouch 脚本也支持获取与设置“向导”属性。

从 InTouch 应用程序播放声音文件

在脚本中，可以将事件与条件同特定的声音关联起来。例如，可以将警告对话框或临界条件与警告声音关联起来。

PlaySound() 函数

播放波形文件的声音或 Windows 缺省声音。

语法

```
Playsound (soundname, flag)
```

参数

soundname
声音或波形文件的名称。字符串值、消息标记名或字符串表达式。如果给声音定义名称，它必须在 Win.ini 文件中的 [Sounds] 部分进行定义，例如 MC="c:\test.wav”

flag
指定如何播放声音。文字整数值、整型标记名或整型表达式，含义如下：
0 - 同步播放声音一次（脚本等到声音播放完毕才继续执行）。
1 - 异步播放声音一次（脚本不必等待声音播放完毕再继续执行）。
9 - 连续播放声音（直到再次调用 PlaySound() 函数为止）。

示例

此脚本播放 c:\welcome.wav 文件的声音一次，并暂停脚本执行，直到声音播放完毕。

```
PlaySound("c:\welcome.wav",0);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此脚本连续播放声音警告。在 win.ini 文件 [Sounds] 部分，需要将声音名称 Alert 与声音文件关联起来，例如：

```
Alert=c:\alert.wav.
PlaySound("Alert",9);
```

获取与设置向导属性

有些向导<1>（</1>如<2>“</2>分布式报警对象<3>”</3>与 Windows 控件<4>）</4>包含可设置或读取的属性<5>。</5> 这些属性可以是文本框中的值或是复选框的选取状态。

在脚本中，通过以下函数可以访问这些属性。

使用对象	作用
SetPropertyD(), GetPropertyD()	设置或读取离散属性。
SetPropertyI(), GetPropertyI()	设置或读取整型属性。
SetPropertyM(), GetPropertyM()	设置或读取消息属性。

如需所支持的属性的列表，请参阅向导描述。

下面是如何设置与读取这些属性的一般方法。

GetPropertyD() 函数

读取向导中的离散属性并返回成功码。

语法

```
result = GetPropertyD (controlname.property, dtag)
```

参数

controlname
支持属性的向导的名称。字符串值、消息标记名或字符串表达式。

property
向导中要读取的离散属性。与 controlname 合起来，可以是字符串值、消息标记名或字符串表达式。

dtag
将接收离散属性值的离散标记名。

返回值

整型错误码。如需有关错误码的详细信息，请参阅理解 Windows 控件错误消息。

示例

通过复选框向导 Checkbox1 与离散标记名 dtag，可以使用以下脚本函数来检查复选框的可见性：

```
result=GetPropertyD("Checkbox1.visible",dtag);
```

如果复选框为可见，此脚本将 dtag 设为 1；否则它将 dtag 设为 0。

SetPropertyD() 函数

设置向导中的离散属性并返回成功码。

语法

```
result = SetPropertyD(controlname.property, Boolean)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

controlname
支持属性的向导的名称。字符串值、消息标记名或字符串表达式。

property
向导中要设置的离散属性。与 controlname 合起来，可以是字符串值、消息标记名或字符串表达式。

布尔型
要传递给向导属性的布尔值。布尔值、离散标记名或布尔表达式。

返回值

整型错误码。如需有关错误码的详细信息，请参阅理解 Windows 控件错误消息。

示例

通过复选框向导 Checkbox1 与离散标记名 dtag，可以使用以下脚本函数来控制复选框的可见性：

```
result=SetPropertyD("Checkbox1.visible",dtag);
```

如果将 dtag 设为 0 并调用上面的脚本函数，复选框向导会消失。

GetPropertyI() 函数

读取向导中的某个整数并返回成功码。

语法

```
result = GetPropertyI (controlname.property, itag)
```

参数

controlname
支持属性的向导的名称。字符串值、消息标记名或字符串表达式。

property
向导中要读取的整型属性。与 controlname 合起来，可以是字符串值、消息标记名或字符串表达式。

itag
将接收整型属性值的整型标记名。

返回值

整型错误码。如需有关错误码的详细信息，请参阅理解 Windows 控件错误消息。

示例

通过单选按钮向导 Radiobutton1 与整型标记名 itag，可以使用以下脚本函数来检查单选按钮组中当前所选的项目：

```
result=GetPropertyI("Radiobutton1.value",itag);
```

此脚本会在第一个（第二个、第三个、...）单选按钮被选中时将 itag 设置为 1（2、3、...）。

SetPropertyI() 函数

设置向导中的整型属性并返回成功码。

语法

```
result = SetPropertyI (controlname.property, integer)
```

参数

controlname

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

支持属性的向导的名称。字符串值、消息标记名或字符串表达式。

property
向导中要设置的整型属性。与 controlname 合起来，可以是字符串值、消息标记名或字符串表达式。

integer
要传递给向导属性的整数值。整数值、整型标记名或整型表达式。

返回值

整型错误码。如需有关错误码的详细信息，请参阅理解 Windows 控件错误消息。

示例

通过单选按钮向导 Radiobutton1，可以使用以下脚本函数来设置第二个单选按钮：

```
result=SetPropertyI("Radiobutton1.value",2);
```

GetPropertyM() 函数

读取向导中的消息属性并返回成功码。

语法

```
result = GetPropertyM (controlname.property, mtag)
```

参数

controlname
支持属性的向导的名称。字符串值、消息标记名或字符串表达式。

property
向导中要读取的消息属性。与 controlname 合起来，可以是字符串值、消息标记名或字符串表达式。

mtag
将接收消息属性值的消息标记名。

返回值

整型错误码。如需有关错误码的详细信息，请参阅理解 Windows 控件错误消息。

示例

通过复选框向导 Checkbox1 与消息标记名 mtag，可以使用以下脚本函数来检查复选框的标题：

```
result=GetPropertyM("Checkbox1.caption",mtag);
```

此脚本将 mtag 设为复选框的标题。

SetPropertyM() 函数

设置向导中的消息属性并返回成功码。

语法

```
result = SetPropertyM (controlname.property, message)
```

参数

controlname
支持属性的向导的名称。字符串值、消息标记名或字符串表达式。

property
向导中要设置的消息属性。与 controlname 合起来，可以是字符串值、消息标记名或字符串表达式。

消息

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要传递给向导属性的消息值。字符串值、消息标记名或字符串表达式。

返回值

整型错误码。如需有关错误码的详细信息，请参阅理解 Windows 控件错误消息。

示例

通过复选框向导 Checkbox1，可以使用以下脚本函数来动态设置复选框向导的标题：

```
result=SetPropertyM("Checkbox1.caption","Start Engine 1");
```

此脚本将 Checkbox1 复选框的标题设为“Start Engine 1”。

使用 OLE 对象编写脚本

您可以使用 OLE 对象来扩展 InTouch HMI 应用程序的功能。使用 OLE 对象可以：

- 为操作员界面创建弹出式对话框。
- 访问操作系统功能<1>，</1>如<2>“</2>控制面板<3>”。</3>
- 使“制造执行模块”中的数据可以在 InTouch HMI 中处理。请参阅“制造执行模块”文档。

创建、验证及释放 OLE 对象

您可以创建与验证 OLE 对象以便在 InTouch 脚本中使用。使用 OLE 对象之后，可以释放它以释放内存。

使用以下函数来创建、验证及释放 OLE 对象。

- OLE_CreateObject() 函数
- OLE_IsObjectValid() 函数
- OLE_ReleaseObject() 函数

OLE_CreateObject() 函数

必须先创建 OLE 对象，然后才能在脚本中引用它。创建时会得到引用该 OLE 对象的指针。

在脚本中，可以通过使用 OLE_CreateObject() 函数来创建 OLE 对象并指定指针。

语法

```
OLE_CreateObject(%pointer, classname);
```

参数

%pointer
为 OLE 对象所选择的指针的名称。它可以包含字母数字字符（A-Z、0-9）与下划线。它不区分大小写。

classname
OLE 类的名称。类名区分大小写。字符串值、消息标记名或字符串表达式。

附注

如果使用相同的对象名创建另一个对象，则该对象更新为引用新的 OLE 类。它会从旧的 OLE 类中释放掉。

示例

此脚本创建一个 OLE 对象 %WShell，该对象引用 Wscript.Shell 类。

```
OLE_CreateObject(%WShell, "Wscript.Shell");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

OLE_IsObjectValid() 函数

在脚本中，可以使用 OLE_IsObjectValid() 函数来验证 OLE 对象是否有效。对于使用 OLE 对象，这并不是一个必需的步骤，但为了确保使用 OLE 对象时不会遇到问题，建议执行该步骤。

语法

```
result = OLE_IsObjectValid(%pointer)
```

参数

%pointer
引用要测试的 OLE 对象的指针。

result
“布尔”值，具体如下：

0 - 指针引用的 OLE 对象无效。

1 - 指针引用的 OLE 对象有效。

示例

此脚本基于 Wscript.Shell 类创建一个 OLE 对象，并创建引用该对象的指针 %WS。isvalid 是离散标记，如果成功创建 OLE 对象则为 TRUE。否则为 FALSE。

```
OLE_CreateObject(%WS, "Wscript.Shell");
isvalid = OLE_IsObjectValid(%WS);
```

OLE_ReleaseObject() 函数

在脚本中使用 OLE 对象之后，可以释放它并删除其指针，以释放系统资源。释放 OLE 对象之后，不能使用其指针来访问关联的 OLE 类的属性和方法。

语法

```
OLE_ReleaseObject(%pointer);
```

参数

%pointer
引用 OLE 对象的指针的名称。它可以包含字母数字字符（A-Z、0-9）与下划线。它不区分大小写。

示例

此脚本释放与指针 %WShell 关联的 OLE 对象，并删除指针 %WShell。

```
OLE_ReleaseObject(%WShell);
```

使用 OLE 对象的属性与方法

在脚本中，可以使用指针对 OLE 属性中的值进行读写。您也可以使用指针调用 OLE 方法。属性与方法是否可用取决于 OLE 对象。

访问 OLE 对象的属性

在脚本中，可以像访问大多数编程语言那样访问 OLE 对象的属性。属性通常使用点 "." 运算符来确定。

注意：在脚本中使用 OLE 对象属性时，确保其引用不超过 98 个字符（包括前导字符 "%”）。尽可能保持 OLE 指针名简短一些。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

读取 OLE 对象属性

在脚本中，可以通过将属性赋值给标记来读取 OLE 对象属性。在动画显示链接中，无法将直接引用用于 OLE 对象属性。

语法

```
tagname = %pointer.property;
```

参数

%pointer
引用 OLE 对象的指针。在读取属性之前，必须使用 OLE_CreateObject() 函数创建，或赋给另一个指针。

property
要读取的属性的名称。

tagname
要写入值的标记。

示例

此脚本基于 System.Random OLE 类创建 OLE 对象，创建引用它的指针 %SR，并将 Math.Random OLE 对象的 .NextDouble 属性的值指定给实型标记名 randtag。

在运行时，实型标记名 Randtag 接收到 0 到 1 之间的随机双精度浮点值。

```
OLE_CreateObject(%SR,"System.Random");
randtag = %SR.NextDouble;
```

写入 OLE 对象属性

在脚本中，可以通过将值赋给属性来将它写入 OLE 对象属性。

语法

```
%pointer.property = value;
```

参数

%pointer
引用 OLE 对象的指针。在写入属性之前，必须使用 OLE_CreateObject() 函数创建，或赋给另一个指针。

property
要写入的属性的名称。

值
要写入属性的值。它可以数值、标记名或表达式。不支持直接从动画输入链接写入 OLE 属性。

调用 OLE 对象的方法

在脚本中，可以调用 OLE 对象方法。

语法

```
%pointer.method(parameters);
```

参数

%pointer
引用 OLE 对象的指针。在调用方法之前，必须使用 OLE_CreateObject() 函数创建，或赋给另一个指针。

方法
属于 OLE 对象一部分的方法的名称。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数
传递给该方法的参数的列表。这些参数必须使用逗号隔开。值、标记名或表达式。

示例

此脚本基于 OLE 类 Shell.Application 创建一个 OLE 对象，创建引用该 OLE 对象的指针 %sa，并调用方法 .MinimizeAll()。此方法最小化桌面上的所有窗口。

```
OLE_CreateObject(%SA,"Shell.Application");
%SA.MinimizeAll();
```

注意：OLE InTouch HMI 脚本中不允许使用可选参数。所有参数都必须指定。

将多个指针指定给相同的 OLE 对象

在脚本中通过使用等号，可以将多个指针指定给相同的 OLE 对象。

语法

```
%newpointer = %pointer
```

参数

%pointer
已引用某个已创建的 OLE 对象的指针的名称。

%newpointer
应该引用相同 OLE 对象的新指针的名称。它可以包含字母数字字符（A-Z、0-9）与下划线。它不区分大小写。

示例

此脚本基于 Wscript.Shell 类创建一个 OLE 对象，并创建引用该对象的指针 %WS。指针 %WS2 设置为 %WS 时指向相同的 OLE 对象。它可以用于读取或写入相同 OLE 对象的属性及调用其方法。

```
OLE_CreateObject(%WS,"Wscript.Shell");
%WS2=%WS;
```

注意：您可以使用与指针相关的消息标记名。如果将消息标记名赋给指针，则它可以获取一个 ID 值。您可以使用它来创建更多指向相同 OLE 对象的指针。

排解 OLE 错误

在脚本中，可以使用 OLE 函数来排解 OLE 错误。

功能	描述
OLE_GetLastObjectError() 函数	获取上一个 OLE 错误的错误号。
OLE_GetLastObjectErrorMessage() 函数	获取有关上一个 OLE 错误的信息。
OLE_ResetObjectError() 函数	复位上一个错误。
OLE_ShowMessageOnObjectError() 函数	显示或隐藏 OLE 错误消息对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

功能	描述
OLE_IncrementOnObjectError() 函数	统计含 InTouch HMI 标记名的 OLE 错误的数量。

OLE_GetLastObjectError() 函数

此函数返回上一个 OLE 错误的错误号。

语法

```
errnum = OLE_GetLastObjectError();
```

参数

errnum
上一个 OLE 错误的编号。

OLE_GetLastObjectErrorMessage() 函数

此函数返回上一个 OLE 错误的错误消息。

语法

```
errmsg = OLE_GetLastObjectErrorMessage();
```

参数

errmsg
上一个 OLE 错误的错误消息。

OLE_ResetObjectError() 函数

在脚本中，使用 OLE_ResetObjectError() 函数复位上一个 OLE 错误，使上一个 OLE 错误号设置为零，且上一个 OLE 错误消息设置为空白。

这可以用于确定一批 OLE 函数中的任何错误。

语法

```
OLE_ResetObjectError()
```

OLE_ShowMessageOnObjectError() 函数

缺省条件下，发生 OLE 错误时，显示一个错误消息对话框。

在脚本中，通过使用函数 OLE_ShowMessageOnObjectError()，可以指定是否显示错误消息对话框。

语法

```
OLE_ShowMessageOnObjectError(Boolean)
```

参数

Boolean
用于确定是否显示 OLE 错误消息对话框的值。布尔值、离散标记名或布尔表达式，含义如下：

0 - 发生 OLE 错误时不显示 OLE 错误消息对话框

1 - 发生 OLE 错误时显示 OLE 错误消息对话框

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

此脚本抑制所有的 OLE 错误消息对话框。发生 OLE 错误时，不显示任何错误消息对话框。

```
OLE_ShowMessageOnObjectError(0);
```

OLE_IncrementOnObjectError() 函数

在脚本中，可以使用 OLE_IncrementOnObjectError() 函数将一个整型标记名指定为 OLE 错误数量的计数器。

语法

```
OLE_IncrementOnObjectError(integertag)
```

参数

integertag
充当计数器的标记名。

附注

如果显示 OLE 错误消息对话框，则计数器标记名仅在 OLE 错误消息对话框关闭之后递增。

示例

此脚本将整型标记名 errorcount 指定为错误计数器，隐藏错误消息对话框，并尝试基于无效的 OLE 类名来创建一个 OLE 对象。这会导致错误，且标记名值 errorcount 递增到 1。

```
errorcount = 0;
OLE_IncrementOnObjectError(errorcount);
OLE_ShowMessageOnObjectError(0);
OLE_CreateObject(%WS,"InVaLiD.cLaSs.nAmE");
```

使用 OLE 的好处

使用 OLE 对象可以给应用程序添加强大的功能；通过以下脚本，您将对此有所认识。

产生随机数

在脚本中，使用以下命令产生 0 到 255 之间的一个随机数：

```
OLE_CreateObject(%SR,"System.Random");
randtag = (%SR.NextDouble)*255;
```

创建用户界面对话框

在脚本中，使用以下命令产生用户界面对话框：

```
dim DlgBody as message;
dim DlgTitle as message;
dim Style as integer;
dim Result as integer;
<1>DlgBody = "Do you want to open the valve </1>’<2>MR-3-FF</2>’<3>?";</3>
DlgTitle = "Confirm Opening Valve MR-3-FF";
Style = 48;
OLE_CreateObject(%WS,"Wscript.Shell");
result = %WS.Popup(DlgBody,1,DlgTitle,Style);
```

本例创建以下用户界面对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Style 标记名确定有哪些图标与按钮出现在对话框上。使用以下值：

图标	样式	值
（无图标）	无图标	0
	错误图标	16
	问号图标	32
	警告图标	48
	信息图标	64

要使用特定的按钮，请将以下值之一添加到 Style 值：

值	样式
0	仅确定按钮
1	确定与取消按钮
2	放弃、重试及忽略按钮
3	是、否及取消按钮
4	是与否按钮
5	重试与取消按钮
6	取消、重试及继续按钮

Result 标记名包含用户选择的按钮编号。这可用作 InTouch 脚本中的条件分支。可能的结果码如下：

结果值	含义
1	按了确定按钮

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

结果值	含义
2	按了取消按钮
3	按了放弃按钮
4	按了重试按钮
5	按了忽略按钮
6	按了是按钮
7	按了否按钮
10	按了重试按钮
11	按了继续按钮

打开 Windows 日期与时间属性面板

在脚本中<1>，</1>使用以下命令打开 Windows 的<2>“</2>日期/时间属性<3>”</3>面板<4>：</4>

```
OLE_CreateObject(%WP,"Shell.Application");
%WP.SetTime();
```

通过调用不同的方法并将它们传递给所引用的 OLE 对象，可以执行类似的任务：

此方法	打开面板
TrayProperties()	工具栏属性
FileRun()	文件运行对话框
FindFiles()	查找文件对话框
FindComputer()	查找计算机对话框
ShutdownWindows()	关闭 Windows 面板

读写注册表。

在脚本中，可以通过以下方法使用 OLE 对 Windows 注册表进行读写：

- 基于 Windows 类 Wscript.Shell 创建 OLE 对象。
- 使用 OLE 对象的 RegRead() 与 RegWrite() 方法。

例如，这些命令直接从注册表键读取安装的 InTouch HMI 的版本，然后将值存储到 rkey 消息标记名：

```
OLE_CreateObject(%WS,"Wscript.Shell");
rkey = %WS.RegRead("HKLM\SOFTWARE\Wonderware\InTouch\Installation\Version");
```

这些命令将值 1 写入某个注册表键，该键确定对于当前登录的用户，文件扩展名是否隐藏：

```
OLE_CreateObject(%WS,"Wscript.Shell");
%WS.RegWrite("HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced\HideFileExt"
,1,"REG_DWORD");
```

最小化窗口

在脚本中，可以使用以下命令最小化桌面上的所有窗口：

```
OLE_CreateObject(%WA,"Shell.Application");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
%WA.MinimizeAll();
```

通过调用以下方法，可以执行一些类似的任务：

此方法	排列窗口
TileHorizontally()	横向平铺所有窗口
TileVertically()	纵向平铺所有窗口
CascadeWindows()	层叠所有窗口
UndoMinimizeALL()	还原所有窗口

编写 ActiveX 控件脚本

您可以使用 ActiveX 控件读取和写入标记名与 I/O 引用。在脚本中，可以引用 ActiveX 控件。

您也可以创建在 ActiveX 控件发生事件时执行的脚本。这些脚本可以复用，也可以导入到其它应用程序中。

调用 ActiveX 控件方法

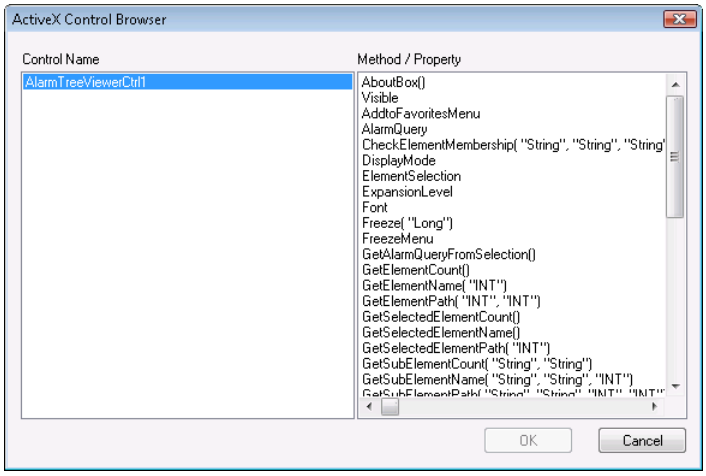
在脚本中，可以调用 ActiveX 控件的方法以执行 ActiveX 控件所支持的操作。ActiveX 方法可以从任何类型的 InTouch QuickScript 或 ActiveX 事件脚本中调用。

注意：要在发生 ActiveX 事件时调用 ActiveX 方法，需要执行一些必要的操作。请参阅配置 ActiveX 事件脚本。

调用 ActiveX 控件方法

1. 在脚本对话框中的插入菜单上，单击 **ActiveX**。

   此时出现 **ActiveX 控件浏览器**对话框。



2. 从左侧窗格中选择 ActiveX 控件的名称。此时右侧窗格包含该 ActiveX 控件所支持的属性与方法的名称。
3. 从右侧窗格中单击要使用的方法的名称，然后选择确定。此时方法名与缺省参数粘贴到脚本窗口中的光标位置。
4. 根据规格配置括号内的方法参数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从 InTouch HMI 访问 ActiveX 控件属性

在脚本中，可以读取和写入 ActiveX 控件属性，以便在 ActiveX 控件与 InTouch 标记名及显示链接之间交换数据。

配置 ActiveX 控件属性以读取和写入数据

在脚本中，可以对 ActiveX 控件进行数据的读取和写入。您使用与特定的 ActiveX 控件关联的 ActiveX 控件属性。

实现方法有两种：

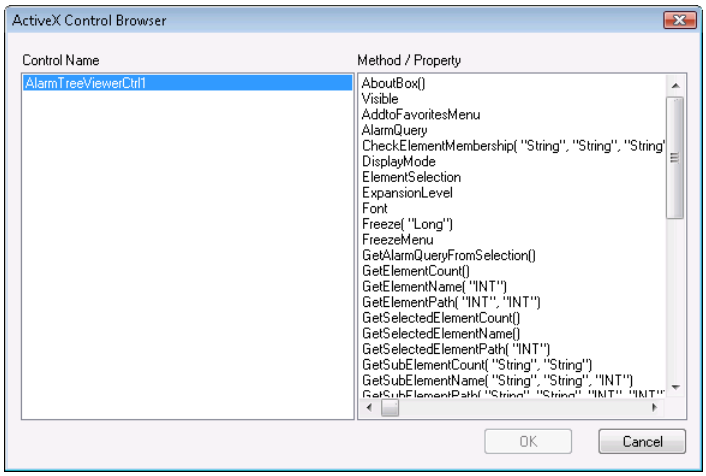
- 在 InTouch HMI QuickScript 或 ActiveX 事件脚本中使用 ActiveX 控件属性。属性值在每次执行脚本时读取或写入。
- 将 ActiveX 控件属性直接链接到 InTouch HMI 标记或 I/O 引用。属性值在每个更新间隔进行读取或写入。

配置脚本以读取和写入 ActiveX 控件属性

在脚本中，可以配置 ActiveX 控件属性以便将值写入 InTouch HMI 标记名或其它表达式，或是从中读取值。

对 ActiveX 控件属性进行数据的读取或写入

1. 打开脚本窗口，指向插入，然后单击 **ActiveX**。此时出现 **ActiveX 控件浏览器**对话框。



2. 从左侧窗格中选择 ActiveX 控件的名称。此时右侧窗格包含所选 ActiveX 控件的属性与方法的名称。
3. 从右侧窗格中选择要使用的属性的名称。此时属性名插入到脚本窗口中的光标位置。
4. 将属性名指定给标记，或根据您自己的规格来使用。
5. 选择确定。

示例

以下脚本将 ActiveX 控件实例 AlarmViewerCtrl1 的 ToPriority 属性读取到整型标记名 topri 中。

```
topri = #AlarmViewerCtrl1.ToPriority;
```

以下脚本将 MS Comic 这个值写入 AlarmViewerCtrl1 这个 ActiveX 控件的 Font 属性。本例动态更改 AlarmViewer ActiveX 控件的显示字体。

```
#AlarmViewerCtrl1.Font = "MS Comic";
```

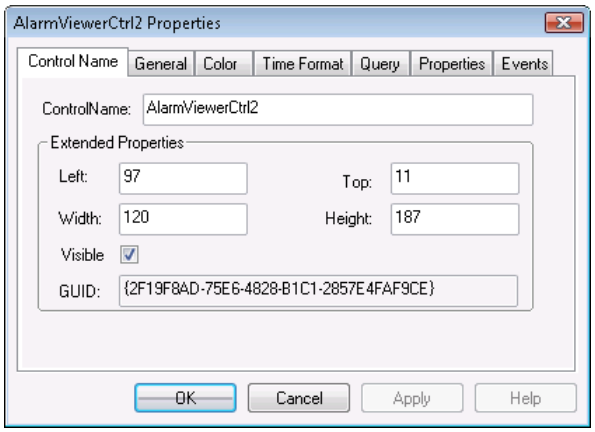
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将 ActiveX 控件属性链接到标记或 I/O 引用

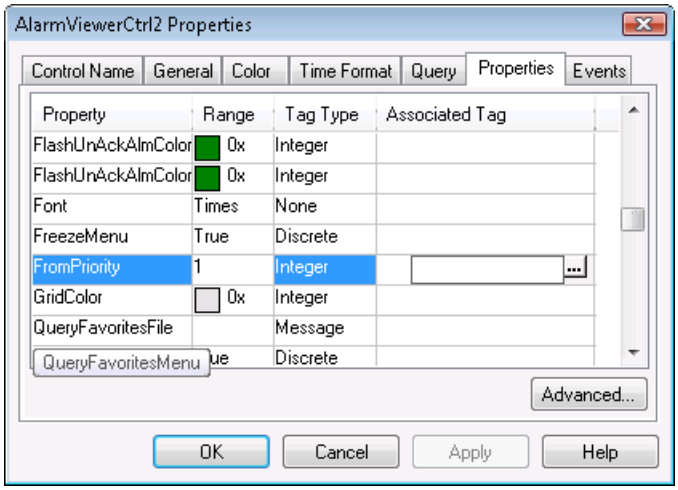
您可以将 ActiveX 控件属性链接到 InTouch HMI 标记或 I/O 引用。

将 ActiveX 控件属性链接到标记或 I/O 引用

1. 双击 ActiveX 控件。此时出现该 ActiveX 控件的属性对话框。



2. 选择属性选项卡并滚动到右侧。
3. 选择列表中的属性。



4. 指定标记或 I/O 引用。执行以下任一操作：
   - 将标记或 I/O 引用直接输入到**关联标记**列。
   - 选择方括号之间的**关联标记**列中的省略号按钮。此时出现**选择标记**对话框。选择一个标记，然后选择确定。
5. 选择确定。

创建与复用 ActiveX 事件脚本

ActiveX 控件可以支持事件，例如可用于关联特定操作的控件单击事件。这些操作存储在 ActiveX 事件脚本中。

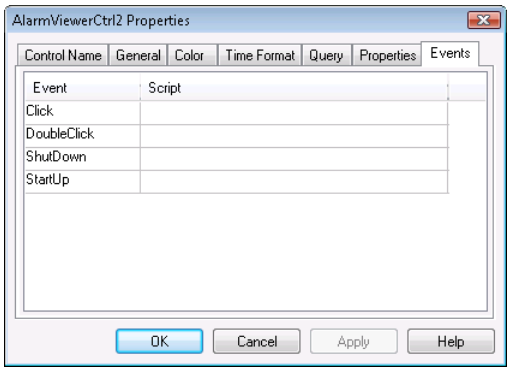
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建 ActiveX 事件脚本

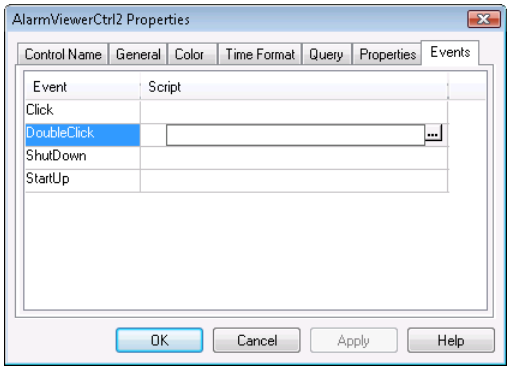
您可以创建或复用在每次发生特定的 ActiveX 控件事件（如选择 ActiveX 控件）时执行的事件脚本。

创建 ActiveX 事件脚本

1. 双击 ActiveX 控件。此时出现属性对话框。
2. 选择事件选项卡。



3. 选择要关联的事件。此时括号与省略号出现在脚本列中。



4. 在相应行的脚本列中，选择括号之间的区域。
5. 输入新名称，然后选择确定。出现消息时，选择确定。此时出现 ActiveX 事件脚本对话框。
6. 根据您自己的规格创建脚本。

复用 ActiveX 事件脚本

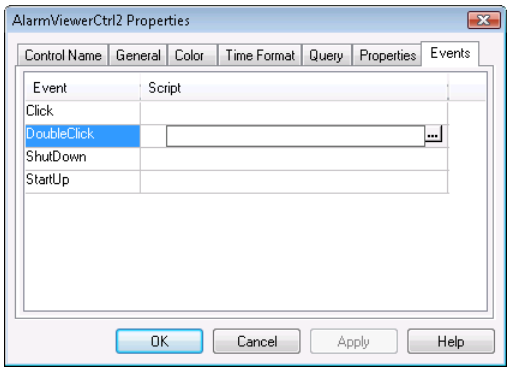
如果 ActiveX 事件脚本由相同的父 ActiveX 控件与事件创建，便可以复用它们。

例如，如果应用程序中有多个 AlarmViewer ActiveX 控件，则它们可以共享 DoubleClick 事件的事件脚本。

复用 ActiveX 事件脚本

1. 双击 ActiveX 控件。此时出现属性对话框。
2. 选择事件选项卡。
3. 选择要关联的事件。此时括号与省略号出现在脚本列中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在相应行的脚本列中，选择省略号按钮。此时出现**选择 ActiveX 脚本对话**框。
5. **单击** ActiveX 脚本，然后**选择确定**。
6. 再次**选择确定**。

创建自引用的 ActiveX 事件脚本

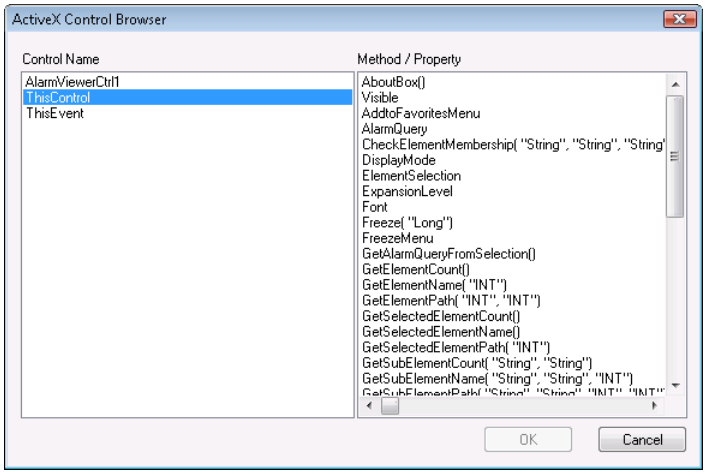
如果使用 ActiveX 事件脚本，则可以将它们配置为引用自身，而不是引用绝对 ActiveX 控件名。这在创建将要复用的 ActiveX 事件脚本时很有用处。ActiveX 事件脚本可以：

- 引用产生事件的特定 ActiveX 控件 (ThisControl)。
- 引用调用脚本的特定事件 (ThisEvent)。

通过引用特定事件，可以让 ActiveX 控件将其它参数传递给 ActiveX 控件脚本。

创建自引用的 ActiveX 事件脚本

1. 为特定的 ActiveX 事件创建 ActiveX 事件脚本。请参阅创建 ActiveX 事件脚本。
2. 在 **ActiveX 事件脚本对话**框中，**选择插入**，然后**选择 ActiveX**。此时出现 **ActiveX 控件浏览器对话**框。



3. 在左侧窗格中，执行以下操作之一：
   - **选择 ThisControl**，以查看可用于连接此控件（以及在其中复用此脚本的任何其它控件）的属性与方法。
   - **选择 ThisEvent**，以查看可用于连接自引用事件的 ActiveX 控件的属性与方法。
4. 在右侧窗格中，突出显示属性或方法之一，然后**选择确定**。此时所选的属性或方法粘贴到脚本窗口中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5. 配置脚本。
6. 选择确定。

例如，此语句将 ClicknRow 事件参数的值写入 ClickedRow 标记：

```
ClickedRow = ThisEvent.ClicknRow;
```

导入 ActiveX 事件脚本

您可以从其它 InTouch HMI 应用程序中导入 ActiveX 事件脚本，以便在当前正在开发的应用程序中复用它们。

从其它应用程序导入 ActiveX 事件脚本

1. 打开 WindowMaker。
2. 在文件菜单上，选择导入，选择可视化，然后选择窗口与脚本。

   此时出现打开文件夹对话框。
3. 浏览到包含要导入的 ActiveX 事件脚本的 InTouch HMI 应用程序。
4. 选择确定。
1. 选择 **ActiveX 事件脚本复选**框，然后选择导入。此时所有的 ActiveX 事件脚本都导入到当前的 InTouch HMI 应用程序中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 12 ActiveX 控件

ActiveX 控件是一些独立的软件组件，它们可以给 InTouch 应用程序带来额外的功能。ActiveX 控件包含可以在应用程序运行时修改的属性、方法及事件，以更改控件的行为。

您可以在 InTouch 应用程序中使用多种类型的 ActiveX 控件。

- InTouch HMI 包含用于报警的 ActiveX 控件。
- 其它产品（如 ActiveFactory）则提供用于操纵和分析数据的控件。
- 您可以使用第三方的 ActiveX 控件。
- 您可以使用 Visual Basic 或 C 语言来开发自己的 ActiveX 控件。

您可以在 InTouch 应用程序中使用任意数量的 ActiveX 控件。您可以：

- 选择 ActiveX 控件并将它粘贴到任何应用程序窗口。
- 调整控件大小（如果控件支持调整大小）。
- 剪切、复制、粘贴、删除 ActiveX 控件，以及为其创建副本。
- 对齐 ActiveX 控件：左侧、右侧、顶部、底部，以及中心点。
- 添加 ActiveX 控件
- 在创建单元时合并 ActiveX 控件与其它对象。
- 使用特定 ActiveX 控件属性所支持的属性、方法和事件。

InTouch 应用程序不支持以下类型的 ActiveX 控件：

- 无窗口控件
- 单框架位置或区域框
- 容器
- 数据控件
- 调度对象

InTouch 应用程序仅支持这些数据类型：离散型（布尔）、整型（32 位数字）、实型（32 位 IEEE 浮点数表示法），以及消息（最长 131 个字符的字符串）。不支持的数据类型包括变量、指针、数组、结构，以及参数化的属性。

ActiveX 控件不能同其它的 InTouch 对象（如窗口控件或图形对象）重叠。一个窗口上的 ActiveX 控件太多会导致系统性能降低。

使用 ActiveX 控件

您可以选择一个 ActiveX 控件，将它粘贴到窗口中，然后再双击它。此时出现它的配置对话框。

配置 ActiveX 控件时，可以给它指定一个唯一的控件名。随后在脚本中可以引用此控件名。

在 InTouch 应用程序中使用 ActiveX 控件

1. 安装 ActiveX 控件。请参阅安装与删除 ActiveX 控件。
2. 选择 ActiveX 控件并将它粘贴到 InTouch 窗口中。

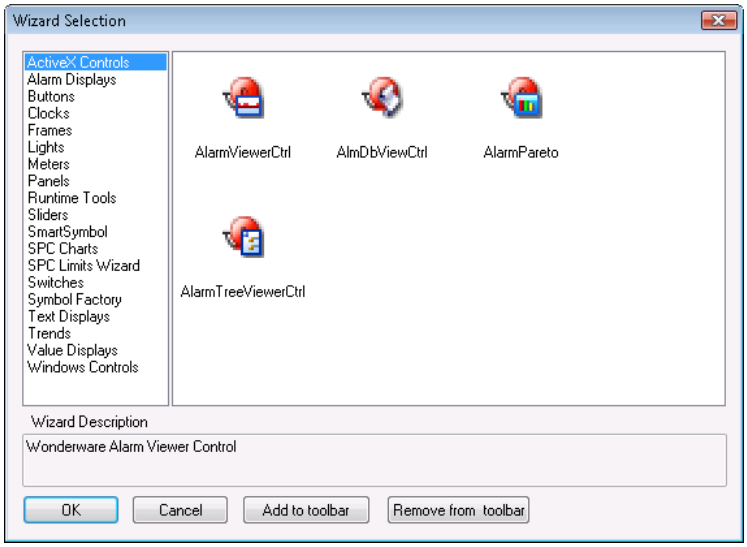
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 配置 ActiveX 控件属性，并将它们绑定到标记。
4. 将 ActiveX 事件关联到 ActiveX 事件脚本。
5. 在 ActiveX 事件脚本或其它 InTouch QuickScript 中，调用 ActiveX 方法并设置 ActiveX 控件属性。

在窗口中放置 ActiveX 控件

1. 在绘图菜单上的插入组中，选择向导。

   此时出现向导选择对话框。



2. 在向导列表中，选择 ActiveX 控件类别。此时所有可用的 ActiveX 控件都出现在显示区域。
3. 双击要使用的 ActiveX 控件。此时对话框关闭，光标变为弯头符号。
4. 选择要粘贴 ActiveX 控件的位置。

将 ActiveX 控件添加到工具栏

1. 在绘图菜单上的插入组中，选择向导。

   此时出现向导选择对话框。

1. 选择要添加的 ActiveX 控件。
2. 选择添加到工具栏。

从工具栏中删除 ActiveX 控件

1. 在绘图菜单上的插入组中，选择向导。

   此时出现向导选择对话框。

2. 选择从工具栏中删除。此时出现从工具栏删除向导对话框。
3. 选择要删除的 ActiveX 控件。
4. 选择确定。

配置 ActiveX 控件

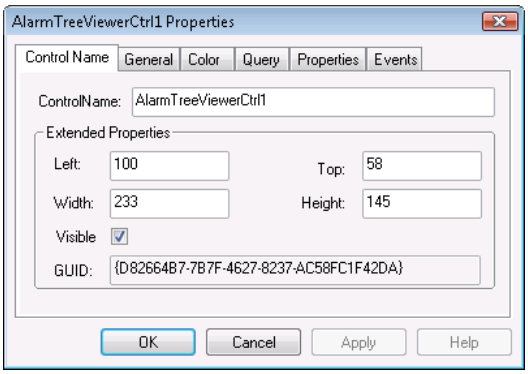
您可以为特定的 ActiveX 控件配置的属性取决于该 ActiveX 控件自身。所有属性都有一个缺省值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在配置 ActiveX 控件的属性之前，必须将它粘贴到 InTouch 窗口中。

- 在将 ActiveX 控件粘贴到窗口时，会生成缺省的控件名，如 Calendar1。您可以在脚本中引用此控件名。ActiveX 控件必须在打开的应用程序窗口中运行，运行的脚本才能引用它。
- 您可以将 ActiveX 控件属性指定给 InTouch 标记。您必须将每种属性类型指定给一个相应的 InTouch 标记类型。

ActiveX 控件有三个标准选项卡：控件名、属性以及事件。



使用事件选项卡将脚本指定给可用的控件事件，如用户双击鼠标时。

任何其它选项卡及其配置都是控件独有的，并取决于控件的属性。例如，有些控件可能要求您配置颜色与字体，而其它的则可能不具备这些属性。

为 ActiveX 控件命名

在以下情况中，会创建一个新的控件实例，并给它指定一个独特的名称：

- 选择复制。
- 选择剪切或复制，然后选择粘贴。
- 选择窗口另存为。
- 选择撤消，然后选择恢复。
- 导入包含控件的窗口。

ActiveX 控件的名称必须唯一。如果试图给控件使用一个已经存在的名称，则会显示一条错误消息。

为 ActiveX 控件命名

1. 将 ActiveX 控件粘贴到开发窗口中。
2. 双击控件。此时出现该控件的属性对话框。
3. 选择控件名选项卡，然后在控件名框中给该 ActiveX 控件输入名称。

ActiveX 控件上的标准操作

像对任何其它 InTouch 对象那样，您也可以在 ActiveX 控件上执行各种标准的操作。如需有关详细信息，请参阅所有对象的特殊操作。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

安装与删除 ActiveX 控件

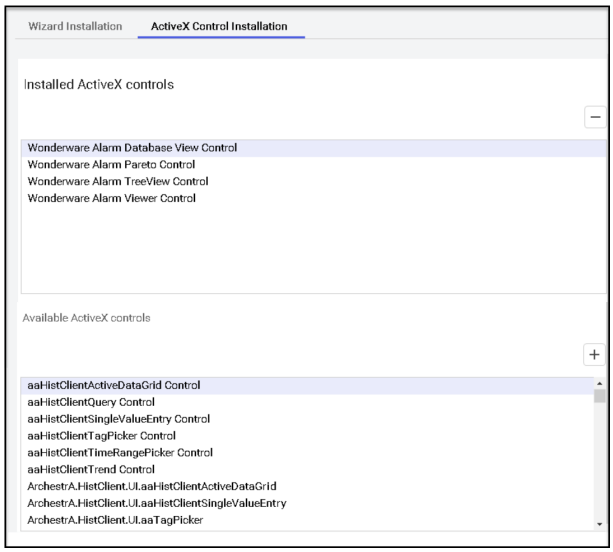
即便已经在计算机上安装了某个 ActiveX 控件，还是必须将此控件安装到 WindowMaker 上以使 InTouch HMI 了解到这个情况。

从 WindowMaker 中移除控件时，它并没有从计算机上彻底删除。它只是从内存中移除掉，不再起作用。

安装 ActiveX 控件

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择向导/ActiveX。

   此时出现向导/ActiveX 屏幕。
3. 选择 ActiveX 控件安装选项卡。此时出现 ActiveX 控件安装属性页。



4. 在安装的 ActiveX 控件列表中，选择要安装在可用的 ActiveX 控件列表内的控件，然后选择 + 图标。

   提示：要选择多个控件，请使用 SHIFT 键或 CTRL 键。
5. 选择保存。

删除 ActiveX 控件

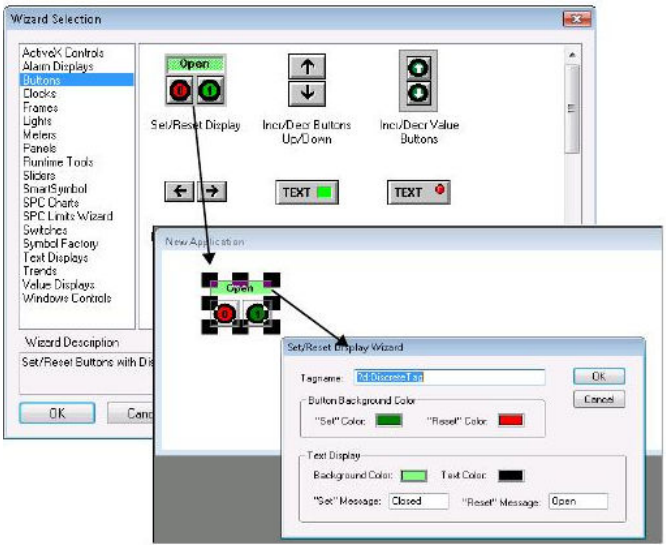
1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择向导/ActiveX。此时出现向导/ActiveX 屏幕。
3. 选择 ActiveX 控件安装选项卡。此时出现 ActiveX 控件安装属性页。
4. 在安装的 ActiveX 控件列表中，选择要从应用程序中删除的控件，然后选择 - 图标。

   提示：要选择多个控件，请使用 SHIFT 键或 CTRL 键。
5. 选择是以删除该控件。此时该控件移到可用的 ActiveX 控件列表中。
6. 选择保存。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 13 向导

向导是一组预先设计、构建及编程的对象，您只需要选择这些对象、将它们放置到应用程序中并进行相应的配置即可。



使用向导

通过使用向导，不必花费时间绘制去对象的各个单独组件、输入对象的值范围或是给对象设置动画效果。

- 您可以从“向导选择”对话框中选择向导。
- 您可以通过在配置对话框中输入标记与 QuickScript 来配置向导。
- 将所选的向导粘贴到窗口中，然后双击它时，会出现配置对话框。

例如，对于游标向导，配置项目包括要使用的标记名、游标的最小与最大范围标签以及填充颜色，等等。提供所需的配置信息之后，该向导便已经可以使用。

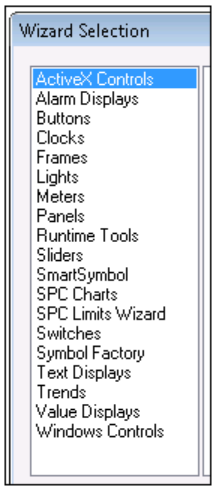
您也可以自行开发复杂的向导来提供“幕后”操作。这些操作可能包括创建完整的显示窗口、创建或转换数据库、导入 AutoCAD 图形，以及配置其它应用程序。

在创建自己的向导之前，应该先研究一下一些工业图形，这些图形不仅提供向导功能，而且还不必编程。

向导类型

向导的类别显示在向导选择对话框中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



“趋势对象”与“Windows 控件向导”有一些独特的参数。如需有关详细信息，请参阅趋势对象和 Windows 控件向导。

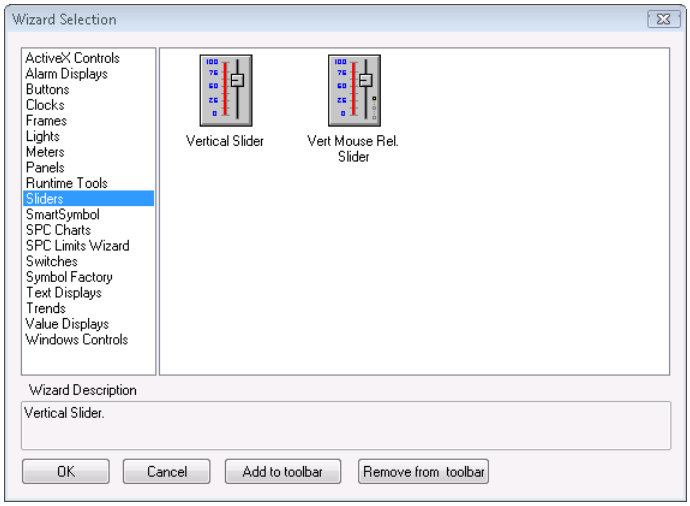
将向导添加到工具栏

您可以将常用的向导添加到“向导/ActiveX 工具栏”以便快速访问。

将向导添加到向导工具栏

1. 在**绘图**菜单上的**插入组**中，**选择向导**。

   此时出现**向导选择**对话框。



2. 在左侧窗格中，**选择**向导类别，如**游标**。
3. 在右侧窗格中，**选择**向导，然后**选择添加到工具栏**。此时向导按钮出现在工具栏中。

从工具栏中删除向导

1. 在**绘图**菜单上的**插入组**中，**选择向导**。

   此时出现**向导选择**对话框。

2. 选择从工具栏中删除。此时出现从工具栏删除向导对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 选择要删除的向导，然后选择确定。

粘贴向导实例

您可以将向导实例放入窗口中。

将向导放入窗口

1. 在**绘图**菜单上的**插入组**中，选择**向导**。

   此时出现**向导选择**对话框。
2. 在左侧窗格中，选择向导类别。
3. 在右侧窗格中，选择向导。
4. 选择确定。

   此时对话框关闭，光标变成弯头符号。
5. 选择要放置该向导的位置。

配置向导

在应用程序中放置向导之后，双击该向导以配置其属性。此时出现一个根据所选的向导而定制的属性对话框。

如需有关每种特定类型的向导的详细信息，请参阅向导的“帮助”（如果有）。

对向导执行标准操作

像处理其它对象那样，您可以使用相同的方法来剪切、复制、粘贴、删除向导，以及给向导创建副本，结果也与其它额对象相似。

安装或删除向导

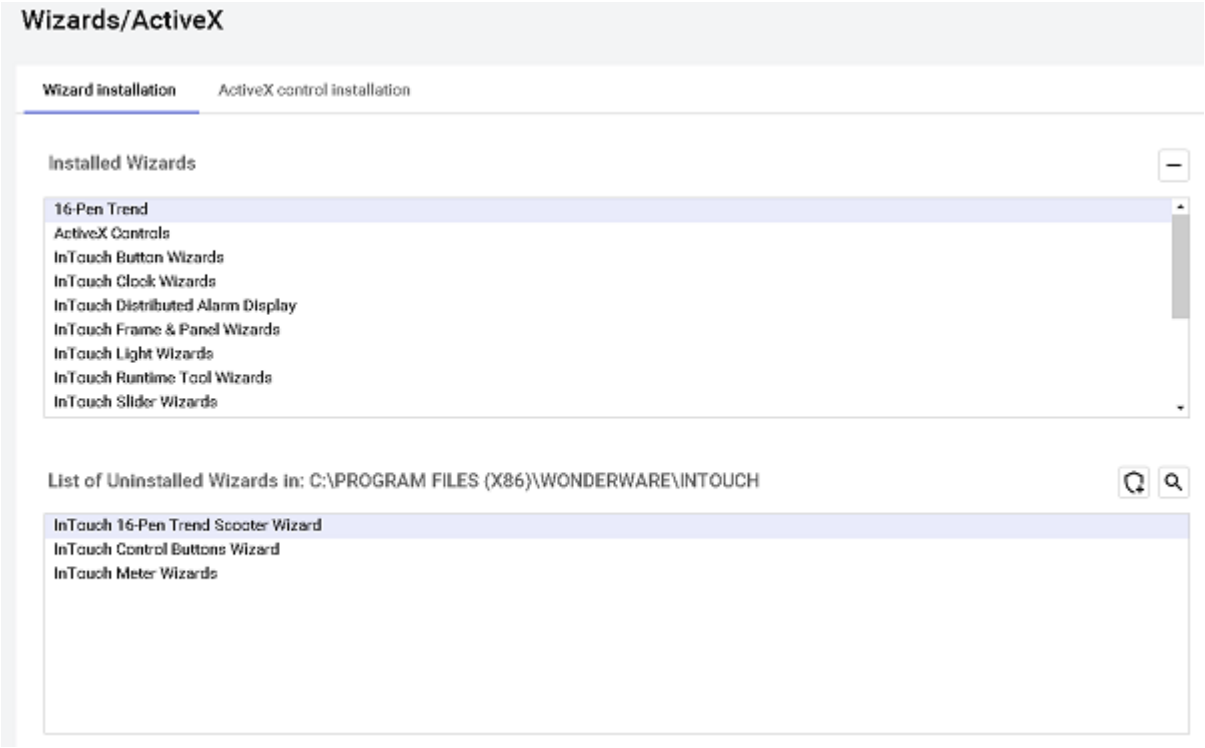
您必须给 WindowMaker 安装向导，以便在应用程序中使用它。从 WindowMaker 中删除向导时，它并没有从计算机上彻底删除。

安装向导

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择**向导/ActiveX**。

   此时出现**向导/ActiveX** 配置屏幕。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 从已卸载的向导列表中选择向导，然后选择 + 图标。
4. 选择保存。

删除向导

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择向导/ActiveX。

   此时出现向导/ActiveX 配置屏幕。
3. 在已安装的向导列表中，选择要删除的向导，然后选择 - 图标。

   提示：通过在按住 SHIFT 或 CTRL 键的同时进行单击，可以选择多个向导。

   该向导出现在已卸载的向导列表中。
4. 选择保存。

从其它目录导入向导

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后单击向导/ActiveX。此时出现向导/ActiveX 屏幕。
3. 单击搜索图标。此时出现浏览文件夹对话框。
4. 浏览到包含要安装的向导的目录，然后选择确定。此时再次出现向导安装对话框，且导入的向导出现在已卸载的向导列表中。

趋势对象

趋势对象是绘制标记值随时间变化的一些向导。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

趋势对象主要有三种类型：

- “实时”趋势，最多使用四个标记实时绘制图表。
- “历史”趋势，最多使用八个标记绘制过去一段时间的图表。
- “16 笔趋势”，最多使用十六个标记绘制实时与历史数据图表。

对于可以在窗口中放置的趋势对象（实时或历史），没有数量上的限制。

使用以下项目配置趋势对象：

- 时间跨度
- 值范围
- 网格分辨率
- 时间标签与值标签的位置
- 笔与颜色

在使用趋势向导之前，必须为每个标记启用记录功能以便进行跟踪，并且也要在 InTouch 应用程序中启用记录功能。

启用标记的记录功能

1. 从标记名字典中选择标记，然后选择记录数据。
2. 如果之前没有这样做，则需要在 InTouch 应用程序中启用记录功能。
   a. 打开 WindowMaker。
   b. 在文件菜单上，指向配置，然后选择历史记录。
      此时出现历史记录屏幕。
   a. 选中启用历史记录复选框或启用存储到 Historian 复选框，然后选择确定。

如需有关配置和使用趋势对象的详细信息，请参阅绘制标记数据的趋势。

Windows 控件向导

“Windows 控件向导”是一组用户界面对象，如下拉列表、组合框、单选框或复选框。

“Windows 控件向导”可以向用户显示一组预定义的选项。例如，您可能会创建一个下拉列表，列出过程、配方或操作员 ID。您也可以启用和禁用指定的控件。您甚至可以加载与修改下拉列表的内容。

通过 QuickScript 函数而不是动画链接表达式，可以访问“Windows 控件向导”的运行时属性。

“Windows 控件向导”具有一些类似于 InTouch 标记点域的属性。它们既可以是可读写的，也可以是只读的。有些属性可以在开发时访问，有些则在运行时访问。它们使用 ControlName.x 的形式进行标识，其中 x 是属性。

例如，如果 Windows 控件的 .Visible 属性等于 0，则该控件在窗口中不可见。同 InTouch 标记类似，.Value 是“Windows 控件向导”的缺省属性。

注意：如果使用工业图形，则可以使用一组基于 .NET 的更强大灵活的 Windows 控件。

创建与配置 Windows 控件

创建与配置“Windows 控件向导”时，请注意以下事项：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- “Windows 控件向导”不能相互重叠。
- 运行时期间，如果“Windows 控件向导”与窗口中的其它对象重叠，那么“Windows 控件向导”将始终位于最上层。
- 每个“Windows 控件向导”都有唯一的控件名 — 这并不添加到标记计数中。
- 指定给列表框或组合框的标记初始值并不初始化列表框或组合框的值。您必须在脚本中使用 SetPropertyX QuickScript 函数，以指定需要与缺省值不同的初始值。

提示：您可以像操作其它向导一样，将“Windows 控件向导”粘贴到窗口中。要取得最佳显示效果，请为 Windows 控件选择灰色背景。如果背景颜色无法设置为灰色，请在“Windows 控件向导”背后放置一个灰色的“面板向导”。

对于每个“Windows 控件向导”，必须指定一个字母数字控件名，其中首字符是字母。允许使用下划线，但不允许使用其它特殊字符。例如，允许使用“Checkbox_1”，但不允许使用 “Checkbox#1”。

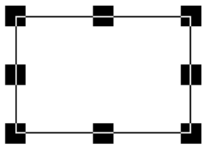
您可以使用 InTouch QuickScript 配置“Windows 控件向导”。

创建文本框

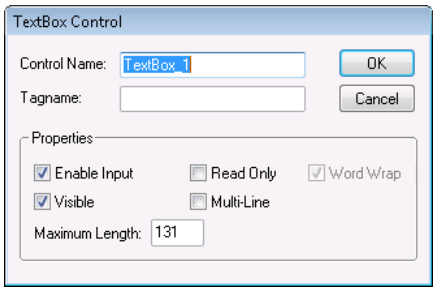
您可以在应用程序中使用文本框，以供操作员输入文本字符串。

创建与配置文本框

1. 创建与放置文本框。执行以下操作：
   a. 在向导选择对话框中，选择 **Windows 控件**向导。此时出现各个控件向导图标。
   b. 双击文本框图标。此时再次出现应用程序窗口，光标变成左弯头符号。
   c. 在应用程序窗口中选择要放置该向导的位置。此时出现拐角处带浓黑色手柄的文本框向导。



   a. 根据应用程序拖动该向导并调整其大小。
2. 双击向导。此时出现文本框控件对话框。



3. 配置该对话框。执行以下操作：
   a. 在控件名框中输入控件名，如 *TextBox_1*。
   b. 在标记名框中输入内存消息标记名，如 *New_Value*。
   c. 在属性区域中，选择允许输入与可见。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 选择确定。此时文本框控件对话框关闭。

创建列表框

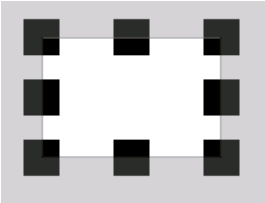
您可以创建应用程序，使用户可以从选项列表中选择项目。

将列表框添加到应用程序时，可以将控件放置在屏幕上，设置属性以配置该列表，然后编写可能需要的任何脚本。

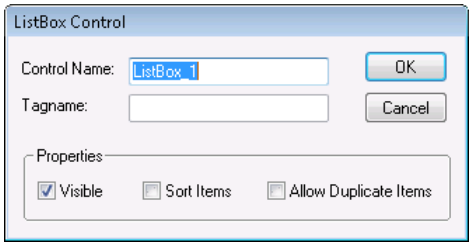
在运行时，列表框可以从文件加载项目，或直接从键盘输入。如需有关详细信息，请参阅编写 Windows 控件脚本。

创建列表框

1. 创建并放置列表框控件。执行以下操作：
   a. 在向导选择对话框中，选择 **Windows** 控件。
   b. 双击列表框图标。此时再次出现应用程序窗口，光标变成左弯头符号。
   c. 在应用程序窗口中选择要放置该控件的位置。此时出现列表框控件。



2. 双击控件。此时出现列表框控件对话框。



3. 配置控件。执行以下操作：
   a. 在控件名框中输入控件名，如 *ListBox_1*。
   b. 在标记名框中输入内存消息标记名，如 *LB1_Value*。
   c. 在属性区域，配置控件的显示方式与功能。
4. 选择确定。

创建组合框

您可以创建应用程序使用组合框让用户从选项列表中选择项目。组合框是将文本框与列表框综合到一起的一种 Windows 控件。

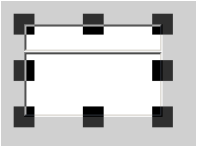
将组合框添加到应用程序时，可以将控件放置在屏幕上，设置属性以配置组合框，然后编写可能需要的任何脚本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

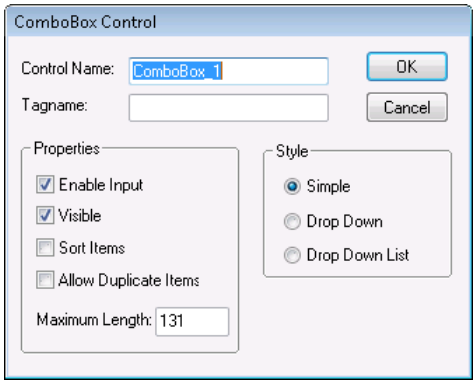
在运行时，组合框可以从文件加载项目，也可以直接从键盘输入。如需有关详细信息，请参阅编写 Windows 控件脚本。

创建组合框

1. 创建并放置组合框控件。执行以下操作：
   a. 在向导选择对话框中，选择 Windows 控件。
   b. 双击组合框图标。此时再次出现应用程序窗口，光标变成左弯头符号。
   c. 在应用程序窗口中选择要放置该控件的位置。此时出现组合框控件。



2. 双击控件。此时出现组合框控件对话框。



3. 配置控件。执行以下操作：
   a. 在控件名框中输入控件名，如 *ComboBox_1*。
   b. 在标记名框中输入内存消息标记名，如 *CB1_Value*。

      其中，CB1_Value 可以是内存标记、IO 标记或远程标记，这些标记可以相应地连接到 PLC。如需有关如何配置 I/O 标记以连接到 PLC 的信息，请参阅配置 I/O 标记属性。

   c. 在属性区域，配置控件的显示方式与功能。
   d. 在样式区域中，选择组合框的类型。
4. 选择确定。

创建复选框

复选框控件允许操作员选择选项。

配置控制图

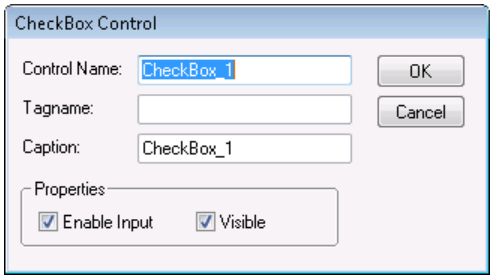
1. 创建复选框。执行以下操作：
   a. 在向导选择对话框中，选择 Windows 控件向导。此时出现各个控件向导图标。
   b. 双击复选框图标。此时再次出现应用程序窗口，光标变为“左”弯头符号。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

c. 在应用程序窗口中选择要放置该向导的位置。此时出现复选框向导。

a. 拖动并调整向导大小。

2. 双击向导。此时出现**复选**框控件对话框。



3. 配置向导。执行以下操作：
   a. 输入控件名。
   b. 输入离散标记名，或双击空标记名框以显示**选择标记**对话框，然后选择标记。
   c. 输入显示在按钮表面的标题。
4. 选择确定。

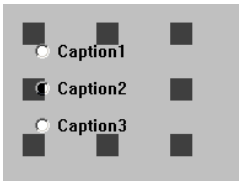
创建单选按钮组

用户必须从多个选项中选择其一的时候，使用单选按钮。用户选择一个选项时，会取消之前所选的选项。您可以为用户创建多个选项作为单选按钮组。每个单选按钮具有一个题注，并向脚本提供唯一值。

您可以仅将整型标记指定给单选按钮控件。

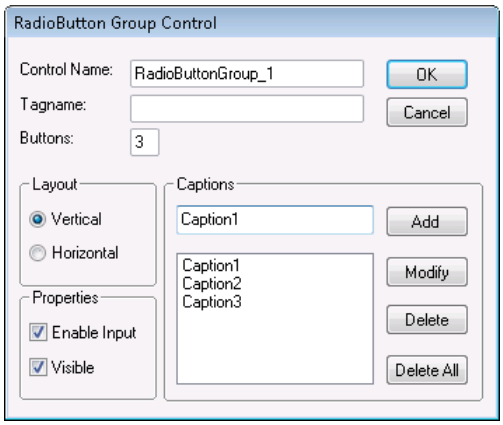
创建单选按钮组

1. 创建“单选按钮”控件向导。执行以下操作：
   a. 在向导选择对话框中，选择 **Windows** 控件向导。此时出现各个控件向导图标。
   b. 双击“单选按钮组”图标。此时再次出现应用程序窗口，光标变成左弯头符号。
   c. 在应用程序窗口中选择要放置该向导的位置。此时出现“单选按钮组”控件向导，显示三个单选按钮。



   a. 根据应用程序拖动该向导并调整其大小。
2. 双击向导。此时出现**单选按钮组**控件对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 配置向导。执行以下操作：
   a. 输入控件名。
   b. 输入要连接到此控件的整型标记名。
   c. 输入要显示的按钮数。
   d. 输入每个按钮的题注。
   e. 设置布局与属性。
4. 选择确定。

编写 Windows 控件脚本

您可以在脚本中使用 QuickScript 函数以：

- 获取或设置控件的值。
- 启用、禁用或隐藏控件。
- 使用组合框、列表框、文本框以及复选框中的项目。

运行时属性可以是可读写的或只读的，具体取决于属性本身。

使用 GetPropertyX() 与 SetPropertyX() 函数来控制或检索这些属性。

获取或设置控件的值

.Value 属性是所有 InTouch“Windows 控件向导”的缺省属性。

对此属性所作的更改与 InTouch 标记名以及“Windows 控件向导”同步。

.Value 点域

所有 InTouch Windows 控件向导的缺省属性。对此属性所作的更改与 InTouch 标记名以及“Windows 控件向导”同步。

类别

Windows 控件

用法

M、I、D 分别是 GetProperty 与 SetProperty 函数的内存 (Memory)、整型 (Integer)、离散 (Discrete) 形式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
[ErrorNumber=]GetPropertyM("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyM("ControlName[.Value]", Value);
[ErrorNumber=]GetPropertyI("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyI("ControlName[.Value]", Value);
[ErrorNumber=]GetPropertyD("ControlName[.Value]", Tagname);
[ErrorNumber=]SetPropertyD("ControlName[.Value]", Value);
```

参数

ControlName
Windows 控件的名称，例如 ChkBox_4。

标记名
要写入属性值的标记名。

[.Value]
此属性是可选的。如果未指定，函数总是假设使用 .Value 属性。

值
处理函数时，要写入的实际值，或用于存放要写入的属性值的有效 InTouch 标记名（与要写入的属性的类型相同）。

附注

指定给列表框或组合框的标记名初始值无法用于初始化列表框或组合框的值。

此点域在开发与运行时为可读写。如果通过将标记名关联到列表框或组合框来访问 .Value 点域，则此点域是只读的。如果将 .Value 点域指定给复选框、单选按钮、文本框，则它是可读写的。开发时指定的值用作运行时的缺省值。

数据类型

对于文本框、列表框、组合框，为消息（可读写）。

对于单选按钮，为整型（可读写）。

对于复选框，为离散（可读写）。

适用范围

文本框、列表框、组合框、复选框、单选按钮。

示例

以下语句将单选按钮对象 "RadioButton_1" 的 .Value 点域的值设置为 4：

```
SetPropertyI( "RadioButton_1.Value", 4 );
```

另请参阅

GetPropertyM()、SetPropertyM()、GetPropertyI()、SetPropertyI()、GetPropertyD()、SetPropertyD()

启用或禁用用户输入控件

使用 .Enabled 属性确定控件对象是否可以响应操作员生成的事件。

.Enabled 点域

确定控件对象是否可以响应用户生成的事件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

Windows 控件

用法

```
[ErrorNumber=] GetPropertyD("ControlName.Enabled",
Tagname);
[ErrorNumber=] SetPropertyD("ControlName.Enabled",
Discrete);
```

参数

ControlName
Windows 控件的名称。例如，ChkBox_4。

标记名
离散型标记名，用于存放所请求的属性。

离散
离散值或离散型标记名，处理函数时用于存放要写入的值。对于离散值：

0 = 禁用控件。

1 = 启用控件。

附注

此属性在开发与运行时都是可读写的。

数据类型

离散型（可读写）

适用范围

文本框、列表框、组合框、复选框、单选按钮。

示例

以下语句禁用列表框对象 "ListBox_1"。

```
SetPropertyD("ListBox_1.Enabled", 0);
```

另请参阅

GetPropertyD()、SetPropertyD()

动态隐藏 Windows 控件

使用 .Visible 属性来确定窗口中的 Windows 控件是否可见。

.Visible 点域

确定 Windows 控件在窗口中是否可见。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyD("ControlName.Visible", Tagname);
[ErrorNumber=]SetPropertyD("ControlName.Visible", Number);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ControlName
Windows 控件的名称。例如，ListBox_1。

标记名
标记名（与返回值的类型相同），处理函数时用于存放属性值。

编号
离散值或离散型标记名，处理函数时用于存放要写入的值。对于离散值：

0 = 控件不可见。

1 = 控件可见。

附注

此属性在开发与运行时都是可读写的。

数据类型

离散型（可读写）

有效值

适用范围

文本框、列表框、组合框、复选框、单选按钮。

示例

以下语句隐藏文本框 "TextBox_1"。

```
SetPropertyD("TextBox_1.Visible",0);
```

另请参阅

GetPropertyD()、SetPropertyD()

对组合框中的项目进行添加与删除

使用以下脚本函数对组合框与列表中的项目进行添加与删除

脚本函数	效果
wcAddItem()	将某个项目添加到列表框或组合框的列表末尾。如果启用了排序功能，则在添加项目之后给列表排序。
wcInsertItem()	将项目添加到列表框或组合框的列表的指定位置。
wcDeleteItem()	删除列表框或组合框的列表中指定的位置上的项目。
wcDeleteSelection()	从列表或组合框中删除当前所选的项目。
wcClear()	从列表或组合框中删除所有项目。

wcAddItem() 函数

将某个项目添加到列表框或组合框的列表末尾。如果启用了排序功能，则在添加项目之后给列表排序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

Windows 控件

语法

```
[ErrorNumber=]wcAddItem("ControlName", "MessageTag");
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

MessageTag
要显示的消息字符串。实际的字符串或消息标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框与组合框。

示例

在打开包含列表框的窗口（使用显示时窗口 QuickScript）时，以下语句将消息字符串的内容添加到列表框中。

```
wcAddItem("ListBox_1", "Chocolate");
wcAddItem("ListBox_1", "Vanilla");
wcAddItem("ListBox_1", "Strawberry");
```

另请参阅

wcInsertItem()

wcInsertItem() 函数

将指定的字符串插入列表框或组合框列表中指定的位置上。与 wcAddItem() 函数不同，wcInsertItem() 函数并不给列表排序，即便它是作为排序的列表框或组合框来创建的。

类别

Windows 控件

语法

```
[ErrorNumber=]wcInsertItem("ControlName", ItemPosition, "Message");
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

ItemPosition
与要添加的项目的位置对应的数字。如果此参数为 -1，则将字符串添加到列表的末尾。任何数字或整型标记名。

消息
包含要插入到 ItemPosition 所指出的位置的字符串。实际的字符串或消息标记名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句将“Blueberry”这个新项目插入到列表框中的第四个位置上（从上至下）。

```
wcInsertItem("ListBox_1", 4, "Blueberry");
```

另请参阅

wcAddItem()

wcDeleteItem() 函数

从列表框或组合框列表中指定的位置上删除某个项目。

类别

Windows 控件

语法

```
[ErrorNumber=]wcDeleteItem("ControlName", ItemPosition);
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

ItemPosition
与项目的位置对应的数字。任何数字或整型标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句删除列表中的第三个项目：

```
wcDeleteItem("ListBox_1", 3);
```

wcDeleteSelection() 函数

从列表中删除当前所选的项目。

类别

Windows 控件

语法

```
[ErrorNumber =]wcDeleteSelection("ControlName");
```

参数

ControlName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句删除列表框中当前所选的项目：

```
wcDeleteSelection("ListBox_1");
```

wcClear() 函数

从列表框或组合框中删除所有的项目。

类别

Windows 控件

语法

```
[ErrorNumber=]wcClear("ControlName");
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句清除列表框中所有的项目：

```
wcClear("ListBox_1");
```

从文件中加载列表项或将列表项保存到文件

使用以下脚本函数从文件中将一些项目加载到组合框或列表中，或是将组合框或列表中的项目保存到文件。

脚本函数	效果
wcLoadList()	将文件中的一些新项目加载到列表框或组合框的内容中。
wcSaveList()	将列表框或组合框的内容保存到文件。

wcLoadList() 函数

将文件中的一些新项目加载到列表框或组合框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

Windows 控件

语法

```
[ErrorNumber=]wcLoadList("ControlName", "Filename");
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

文件名
包含文件的名称。如果没有将完整的路径名作为消息参数的一部分来提供，则函数会在应用程序目录中查找消息文件。实际的字符串或消息标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

如果使用外部文件来填充列表与组合框，则它们必须采用特定的格式并包含特定的信息。格式：

ControlType, ListCount

ListItem, ItemIndex

ListItem, ItemIndex

: :

: :

ListItem, ItemIndex

ControlType 为 COMBOBOX 或 LISTBOX。

例如，您希望将某个列表文件加载到组合框，它包含三个可供选择的项目，并且这些项目没有指定的项目数据。文件的格式是：

COMBOBOX, 3

Chocolate, 0

Vanilla, 0

Strawberry, 0

COMBOBOX 是控件类型。列表计数为 3：Chocolate、Vanilla、Strawberry。然后 Chocolate 被列为第一个项目或位置 1。Vanilla 是位置 2、Strawberry 是位置 3。每个项目的数据值都是 0。

如需有关项目数据的详细信息，请参阅 wcSetItemData() 函数。

适用范围

列表框与组合框。

示例

以下语句将适当格式的列表（位于 c:\wclist.txt 中）加载到组合框中：

```
wcLoadList("Combobox_1", "c:\wclist.txt");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

wcAddItem()、wcSaveList()

wcSaveList() 函数

将列表框或组合框中的项目保存到文件。

类别

Windows 控件

语法

```
[ErrorNumber=]wcSaveList("ControlName","Filename");
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

文件名
包含文件的名称。如果该文件不存在，则创建它。实际的字符串或消息标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句将列表框中的当前项目保存到文件（c:\newlist.txt）中：

```
wcSaveList("ListBox_1", "c:\newlist.txt");
```

另请参阅

wcLoadList()、wcSetItemData()

在组合框或列表中查找项目

使用 wcFindItem() 函数在列表框或组合框中搜索指定的项目。如果找到项目，此函数将相应的位置作为第四个参数返回给某个整型标记名。

wcFindItem() 函数

确定列表框或组合框中第一个同所提供的消息字符串匹配的项目所对应的位置。

类别

Windows 控件

语法

```
[ErrorNumber=]wcFindItem ("ControlName", "MessageTag", CaseSens, Tagname);
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

MessageTag

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要比较的消息字符串。实际的字符串或消息标记名。

CaseSens
确定字符串比较的类型。它可以是离散值或标记名。以下值有效：

0 = 不区分大小写。

1 = 区分大小写。

Tagname
返回匹配项位置的整型标记。如果未找到匹配项，则返回 -1。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框、组合框

示例

假设 ListBox_1 这个列表框包含“ItemA”、“ItemB”、“ItemC”，则此函数将以下这些值返回给 Result：

```
wcFindItem("ListBox_1", "ItemB", 0, Result);
```

返回 2

```
wcFindItem("ListBox_1", "Itemb", 1, Result);
```

返回 -1

```
wcFindItem("ListBox_1", "itemc", 0, Result);
```

返回 3

```
wcFindItem("ListBox_1", "XYZ", 0, Result);
```

返回 -1

在组合框或列表中使用项目索引

使用以下点域处理列表框或组合框中的项目索引。

点域	效果
.TopIndex	列表框中最顶端项目的整型索引。
.NewIndex	通过 wcAddItem() 或 wcInsertItem() 函数添加到列表框或组合框的最后一个项目的整型索引（标记名）。
.ListIndex	列表中当前所选项目的索引（标记名或数字）。

.TopIndex 点域

设置或读取列表框中最顶端项目的整型索引。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyI("ControlName.TopIndex", Tagname);
[ErrorNumber=]SetPropertyI("ControlName.TopIndex", Number);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ControlName
Windows 控件的名称。例如，ListBox_1。

Tagname
整型标记名，处理函数时用于存放属性值。

Number
定义列表框中最顶端项目的索引号。可以是整数值或整型标记名，也可以是提供整数值的表达式。

附注

此属性仅在运行时可用。

数据类型

整型（可读写）

适用范围

列表框。

示例

以下语句将列表框对象“ListBox_1”的 TopIndex 值设置为 14：

```
SetPropertyI("ListBox_1.TopIndex",14);
```

另请参阅

GetPropertyI()、SetPropertyI()、.ListIndex、.NewIndex

.NewIndex 点域

返回通过 wcAddItem() 或 wcInsertItem() 添加到列表框或组合框的最后一个项目的整型索引（标记名）。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyI("ControlName.NewIndex", Tagname);
```

参数

ControlName
Windows 控件的名称。例如，ListBox_4。

Tagname
标记名，包含添加到列表或组合框的最后一个项目的整型索引。对于空白列表，返回值为 -1。

附注

此属性仅在运行时可用。

数据类型

整型（只读）

适用范围

列表框与组合框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

以下语句检索“ListBox_1”列表框中最近添加的项目的索引，并将该值写入内存整型标记名 NewItemIndex。

```
GetPropertyI("ListBox_1.NewIndex", NewItemIndex);
```

另请参阅

GetPropertyI()、wcAddItem()、wcInsertItem()、.ListIndex、.TopIndex

.ListIndex 点域

设置或读取列表中当前所选项目的索引（Tagname 或 Number）。

使用列表框时，索引 -1 指出当前未选择任何项目。

使用组合框时，索引 -1 指出用户已经将新文本输入到控件的文本输入区域中。

语法

```
[ErrorNumber=]GetPropertyI("ControlName.ListIndex", Tagname);
[ErrorNumber=]SetPropertyI("ControlName.ListIndex", Number);
```

参数

ControlName
Windows 控件的名称。例如，ListBox_4。

Tagname
要写入当前所选项目的索引的标记名。

Number
定义列表中某个特定项目的索引号。

附注

索引号定义列表中的某个特定项目。使用 .ListIndex 点域来设置或确定列表或组合框中当前所选项目的索引。

此属性仅在运行时可用。

数据类型

整型（读或写）

适用范围

列表框与组合框。

示例

此语句检索“ListBox_1”列表框中当前所选的项目，并将该值写入内存整型标记名 MyListBoxIndex 中。

```
GetPropertyI( "ListBox_1.ListIndex", MyListBoxIndex );
```

另请参阅

GetPropertyI()、SetPropertyI()、.NewIndex、.TopIndex

统计列表框或组合框项目数

.ListCount 点域包含列表框或组合框中项目的数量。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.ListCount 点域

读取列表框或组合框中项目的数量。

类别

Windows 控件

语法

```
[ErrorNumber=]GetPropertyI("ControlName.ListCount", Tagname);
```

参数

ControlName
Windows 控件的名称。

Tagname
有效的标记名，包含列表中项目的整型计数。

附注

此属性仅在运行时可用。

数据类型

整型（只读）

适用范围

列表框与组合框。

示例

以下语句检索 *ListBox_1* 列表框中的项目数量，并将该值写入内存整型标记 *MyListBoxCount* 中。

```
GetPropertyI("ListBox_1.ListCount", MyListBoxCount);
```

另请参阅

GetPropertyI()、.ListIndex

获取或设置列表项的值

使用 wcGetItemData() 函数来查找与项目索引所确定的列表项关联的整数值。

使用 wcSetItemData() 函数将整数值指定给项目索引指定的列表中的项目。这会将一个数字指定给字符串。

wcGetItemData() 函数

读取与 ItemIndex 参数所确定的列表项关联的整数值。

类别

Windows 控件

语法

```
[ErrorNumber=]wcGetItemData("ControlName", ItemIndex, Tagname);
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ItemIndex
与项目的位置对应的数字。任何数字或整型标记名。

Tagname
实型或整型标记名的实际名称。从函数返回时，wcGetItemData() 函数将该项目对应的数值赋予此标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框与组合框。

示例

在动作脚本运行时，以下语句检索与列表框中的第五个项目关联的数值，并将它返回给 ItemValue 整型标记名：

```
wcGetItemData("ListBox_1", 5, ItemValue);
```

如果给列表中的第五个项目指定了整数值 4500，则 ItemValue 标记名包含 4500。

另请参阅

wcSetItemData()

wcSetItemData() 函数

将项目的整数值（Number 参数）指定给列表中 ItemIndex 参数所指定的项目。此函数允许将数字指定给字符串。

类别

Windows 控件

语法

```
[ErrorNumber=]wcSetItemData("ControlName", ItemIndex, Number);
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

ItemIndex
指定要编辑的列表项的整数值。任何数字或整型标记名。

Number
代表项目数据的整数值。任何数字或整型标记名。

附注

您可以使用“记事本”之类的程序创建包含各个项目的完整列表，然后使用一个函数调用来加载它们。根据 wcSaveList() 函数的要求设置列表格式。

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

使用 wcGetItemData() 函数返回与列表项关联的值（项目数据）。此标记名参数包含返回的数值。此参数可以是直接写入实际设备的 I/O 整型标记名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

某配方包含三种成分：面粉、糖、盐。面粉的质量是 4500 克、糖是 1500 克、盐是 325 克。通过使用由所选配方（tagname, RecipeName）触发的数据改变脚本，将值指定给每个列表框项目。

```
wcSetItemData("ListBox_1", 1, 4500); {set 1st item in the list (flour)=4500}
wcSetItemData("ListBox_1", 2, 1500); {set 2nd item in the list (sugar)=1500}
wcSetItemData("ListBox_1", 3, 325);   {set 3rd item in the list (salt)=325}
```

另请参阅

wcLoadList()、wcSaveList()、wcGetItemData()

获取列表项的名称

使用 wcGetItem() 函数返回与列表框或组合框中相应的项目索引关联的项目字符串。

wcGetItem() 函数

返回一个字符串，包含与列表框或组合框中的 ItemIndex 所对应的项目的内容。

类别

Windows 控件

语法

```
[ErrorNumber=]wcGetItem("ControlName", ItemIndex, Tagname);
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

ItemIndex
与项目的位置对应的数字。任何数字或整型标记名。

Tagname
消息标记名。从函数返回时，wcGetItem 函数将该项目所对应的数据放入此标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框与组合框。

示例

以下语句会在动作脚本运行时，将组合框中第十个项目的字符串值返回到 ListSelection 消息标记：

```
wcGetItem("Combobox_1", 10, ListSelection);
```

如果列表中的第十个项目是“Vanilla”，则 ListSelection 标记包含字符串“Vanilla”。

加载文本框的内容

使用 wcLoadText() 函数从文件加载文本框的内容。使用 wcSaveText() 函数将文本框的内容保存到文本文件中。

注意：如果给标记名定义了最大长度，则只能从文本框内容中将该数量的字符给标记。如果未给文本框指定标记，其内容最多可以包含 65535 个字符。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

wcLoadText() 函数

使用文件的内容替换文本框的内容。

类别

Windows 控件

语法

```
[ErrorNumber=]wcLoadText("ControlName", "Filename");
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

文件名
包含文件的名称。如果未将完整的路径名作为消息参数的一部分来提供，此函数会在应用程序目录中查找文件。实际的字符串或消息标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

文本框。

示例

在打开包含文本框的窗口（“显示时”窗口脚本）时，以下语句将一个文本文件 (c:\InTouch.32\readme.txt.) 加载到文本框中：

```
wcLoadText("Textbox_1", "c:\InTouch.32\readme.txt");
```

wcSaveText() 函数

将文本框中包含的文本保存到指定的文件中。如果文件不存在，则创建它。如果存在，则必须是可读写的。

类别

Windows 控件

语法

```
[ErrorNumber=]wcSaveText("ControlName", "Filename");
```

参数

ControlName
Windows 控件对象的名称。例如，ListBox_1。实际的字符串或消息标记名。

文件名
包含目标文件的名称。如果不提供完整的路径名，则文件保存到应用程序目录中。如果该文件存在，则它会被覆盖。如果该文件不存在，则创建它。随后可以使用 wcLoadText() 函数将产生的文件加载到文本框对象中。实际的字符串或消息标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

适用范围

文本框。

示例

在动作脚本运行时，以下语句将文本框中所输入的当前信息保存到文件 (c:\InTouch.32\newtext.txt) 中：

```
wcSaveText("Textbox_1", "c:\InTouch.32\newtext.txt");
```

另请参阅

wcLoadText()

检查文本框是否为只读

使用 .ReadOnly 点域来确定文本框的内容是只读还是可读写的。

.ReadOnly 点域

确定文本框的内容是只读还是可读写的。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyD("ControlName.ReadOnly", Tagname);
```

参数

ControlName
Windows 控件的名称。例如，Textbox_1。

Tagname
离散标记名，处理函数时用于存放属性值。

0 = 文本框的内容为可读写

1 = 文本框的内容为只读

附注

此属性在开发与运行时都可用。

数据类型

离散（只读）

适用范围

文本框。

示例

以下语句检索“TextBox_1”文本框的只读状态：

```
GetPropertyD("TextBox_1.ReadOnly",A_Tagname);
```

另请参阅

GetPropertyD(), SetPropertyD()

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

获取或设置复选框的标签

.Caption 点域定义复选框的消息文本。

.Caption 点域

确定要使用复选框显示的消息。

类别

Windows 控件

用法

```
[ErrorNumber=]GetPropertyM ("ControlName.Caption", Tagname);
[ErrorNumber=]SetPropertyM ("ControlName.Caption", "Message");
```

参数

ControlName
Windows 控件的名称。例如，ChkBox_4。

标记名
消息标记名，用于存放所请求的属性。

消息
用英文引号括起的消息字符串。

附注

此属性在开发与运行时都是可读写的。

数据类型

消息（可读写）

适用范围

复选框。

示例

此语句将复选框对象“CheckBox_1”的题注设置为“Blue Paint Option”。

```
SetPropertyM("CheckBox_1.Caption","Blue Paint Option");
```

另请参阅

GetPropertyM()、SetPropertyM()

理解 Windows 控件错误消息

只要给出错误号，wcErrorMessage() 便会返回描述该错误的字符串消息。它适用于列表框、文本框、组合框、单选钮及复选框。

Window 控件功能会根据 QuickScript 函数的处理结果返回一些值。返回值用于错误诊断。您可以将这些值指定给整型标记名。例如：

```
ErrorNumber = wcGetItem("ControlName", Number, Tagname);
```

在此脚本中，`ErrorNumber` 是包含所返回的错误值的整型标记。函数的返回值可以传递给 wcErrorMessage()。wcErrorMessage() 将返回该错误的字符串描述。例如：

```
ErrorMsg = wcErrorMessage(ErrorNumber);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在此脚本中，ErrorMsg 是一个消息型标记，它包含所返回的错误的文本。下表列出错误值及其定义。

错误消息	定义
0	成功
-1	一般错误
-2	可用内存不足
-3	属性为只读
-4	指定的项目已经存在
-5	未知对象名
-6	未知属性名
-x	未知错误。

wcErrorMessage() 函数

返回描述错误的消息字符串。

类别

Windows 控件

语法

```
ErrorMessage=wcErrorMessage(ErrorNumber);
```

参数

ErrorMessage
消息标记名。

ErrorNumber
所有 Windows 控件函数返回的数字。任何数字或整型标记名。

附注

如需已返回错误号的列表，请参阅理解 Windows 控件错误消息。

适用范围

列表框、文本框、组合框、复选框、单选钮。

示例

如果在加载列表时发生错误，则在 ErrorDescription 消息标记名中显示该错误的文本描述。在本例中，通过给 ErrorDescription 标记名指定一个“值显示”动画链接来显示错误消息。

在“显示时”窗口的 QuickScript 中：

```
ErrorNumber=wcLoadList("ListBox_1","c:\recipe.txt");
ErrorDescription=wcErrorMessage(errornumber);
```

您可以将此函数与所有 Windows 控件函数结合使用，以显示错误消息：

```
ErrorNumber=wcAddItem("ListBox_1","AM_4A4356");
ErrorMsg=wcErrorMessage(ErrorNumber);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

小组件

小组件是可以扩展网页或网站功能的小型 Web 小组件。定制的网站也可以通过使用开源代码或框架来包含小组件，从而提供全部或部分特定功能。小组件是插入网站的独立代码块，不会更改网站的任何功能。小组件最常用于提供与其它平台和数据源集成的屏幕用户界面元素。小组件可以在网站的任何网页上运行，具有一致的放置位置和用户界面。例如，社交媒体、天气、RSS 或播客小组件。

缺省情况下，图形工具箱中的“小组件”文件夹下提供以下小组件：

- 轮播
- Web 浏览器
- QR 码扫描仪
- Map_App
- DataGrid
- Teamwork

轮播小组件

轮播小组件可以像旋转木马一样循环显示元素（文本图像或幻灯片），无需任何输入。该小组件可用于在工厂车间内的大型监视器上显示仪表板、警报或报警信息。

注意：WindowViewer 和 Web 客户端中不支持在轮播小组件中显示客户端控件（报警客户端控件和趋势客户端控件）。

属性

除了标准图形属性，还可以在**小组件属性**下配置特定于小组件的属性。

名称	描述	缺省值
Autoplay	如果 Autoplay 属性设置为 True，轮播小组件将在加载时自动启动。如果设置为 False，用户必须选择下一项才能启动轮播。	True
BackgroundColor	为小组件设置背景颜色。在 RGB、HTML 代码 (#FF0000) 或有效的 HTML 颜色名称中指定颜色值。	白色
GraphicNames	轮播小组件将在运行时显示的图形列表，用逗号分隔。	空
间隔	项目自动循环之间的延迟时间（以毫秒为单位）。	5000
键盘	如果 Keyboard 属性设置为 True，轮播小组件将响应键盘输入。	True
Loop	如果 Loop 属性设置为 True，轮播小组件将连续循环显示图形，否则将在一次循环后停止。	True

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

名称	描述	缺省值
Pause（暂停）	如果 Pause 属性设置为 True，轮播小组件在检测到鼠标悬停或触摸按下事件时将暂停循环显示图形。当鼠标移走后，图形将恢复循环显示。	True

轮播小组件基于 Bootstrap 4.0 轮播组件，如需有关 bootstrap 的详细信息，请访问：https://getbootstrap.com/docs/4.0/components/carousel/

Web 浏览器小组件

使用 Web 浏览器小组件，用户可以在 WindowViewer 和 Web 客户端中显示网站。

如果 Web 客户端正在以 HTTPS 运行，则只能加载 HTTPS URL 页面。如果 Web 客户端正在以 HTTP 运行，则可以同时加载 HTTP 和 HTTPS。如果网站的策略阻止跨域访问，此小组件将不工作。URL 不需要位于双引号内，但必须是有效的 URL。

属性

URL：网站的地址。

限制

- 如果未指定任何协议，缺省条件下将使用 https 协议。
- 如果将 Web 客户端配置为使用 HTTPS 协议，则只会加载 HTTPS URL 页面。如果使用 HTTP URL，Web 浏览器小组件将显示一条消息：已通过 HTTPS 加载地址为“`https://localhost/intouchweb`”的页面，但是请求的框架“`http://**********`”不安全。此请求已被阻止：必须通过 HTTPS 提供内容。
- 如果网站策略阻止跨域（跨源）访问，Web 浏览器小组件将不起作用。将提供一个链接以在单独的选项卡中打开网页。

QR 码扫描仪

QRCode_Scanner 小组件连接到相机以扫描 QR 码，并返回所生成的字符串。

属性

属性名	描述	缺省值
QRCode	扫描的 QR 码所生成的字符串。缺省值为空。	空
AutoStart	如果设置为 true，相机将会自动启动。	True
AutoStop	如果设置为 true，相机将在扫描 QR 码之后停止。	True
开始	如果设置为 true，相机将会启动。	False
停止	如果设置为 true，相机将会停止。	False
BackgroundColor	设置小组件的背景颜色。在 RGB、HTML 代码 (#FF0000) 或有效的 HTML 颜色名称中指定颜色值。	黑色

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

限制

- 设备必须带有相机。
- 建议使用物理机器上的 QR 码，而不是虚拟机上的 QR 码。
- 当远程使用 Web 客户端时，请使用安全的 URL (https://) 访问 Web 客户端。

用法

您可以配置一个脚本来读取 QR 码，并根据扫描的值显示图形。

在运行时，“QR 码扫描仪”小组件将随浮动工具栏一起出现，其中包含以下按钮 - AutoStart、AutoStop 和 StartStop。

在加载小组件后，如果将 AutoStart 设置为 True，相机将会自动启动。要使相机保持打开状态，**请选择 AutoStop**。

要手动启动相机，**请选择 StartStop** 并扫描 QR 码。

扫描 QR 码之后，相机将保持打开状态，从而允许用户扫描其它 QR 码。要停止相机，**请选择 StartStop**。

浮动工具栏将显示从相机所扫描的 QR 码衍生而来的 QRCode。

用户可以根据返回的 QRCode 编写动作脚本。

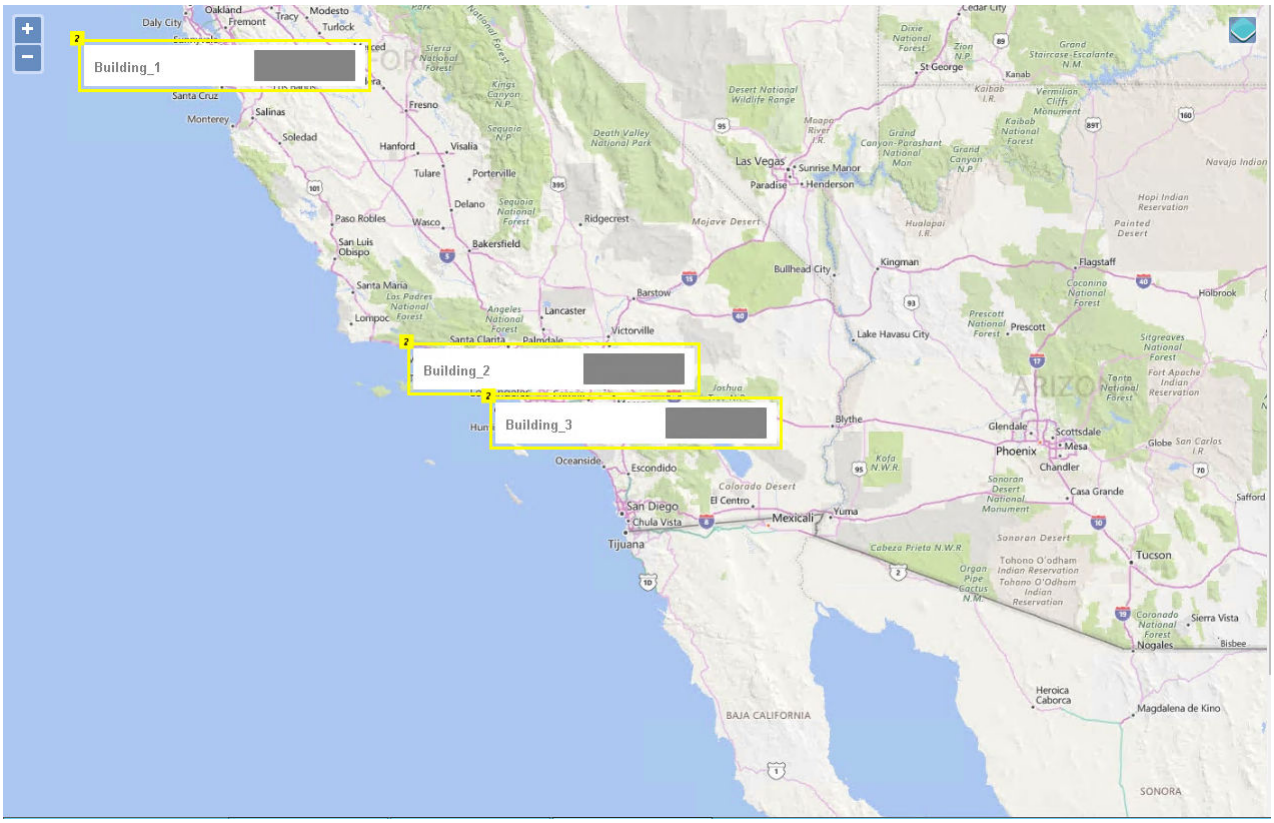
Map_App 小组件

Map_App 小组件显示地图，其中包含运行中应用程序中的符号。在运行时，地图提供控件和触摸支持，以允许用户平移到地图的不同区域，并放大或缩小以显示更多或更少的地图细节。放置在地图中的图形通常表示位于地图所示区域内的业务资产。这些图形可以包含报警，以显示每个业务位置的当前流程状态。

以下示例地图显示的报警指示符指示大楼位置以及每个大楼当前的报警状态。在运行时，用户可以单击地图左上角的缩放按钮，以放大或缩小视图。根据您向缩放层分配公司资产数据的方式，将显示不同的地图数据。选择地图右上角的图标会列出用户可以选择的地图，以在当前地图视图内显示不同的数据。

当 ViewApp 在触屏设备上运行时，MapApp 支持触屏手势。用户可以使用标准触屏手势平移、缩放或选择地图中显示的项目。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



配置 MapApp：概述

要查看 WindowViewer 中的地图，可以执行以下步骤：

请注意，您可以根据自己的需求跳过其中一些步骤。

1. 完成一套准备任务：
   - 如果您想使用 UDT 在 MapApp 中显示，请执行此步骤。在新的 InTouch 应用程序中，创建 UDT、实例和成员。添加并配置位置属性，如“纬度”和“经度”。有关更多信息，请参阅 MapApp 主题。
   - 使用您希望在 WindowViewer 中显示其数据的受支持地图供应器设置帐户，并使用地图供应器定义您希望能够使用的每个地图。
   - 如果要使用需要进行本地身份验证的本地地图服务器，请执行此步骤。创建用户凭据以控制对特定地图供应器所示数据的访问。有关创建凭据的信息，请参阅使用凭据管理器。
   - 使用工业图形编辑器创建一组将出现在地图中的图形。
2. 为 MapApp 配置全局设置：
   - 定义地图源。源地图定义包括地图的名称、地图供应器、地图是基本地图还是叠加地图，以及其他配置信息，具体信息因地供应器而异。您必须定义将在 WindowViewer 使用的每个源地图。当您将 MapApp 配置为用于特定窗口时，您需要选择要包含到该窗口中的源地图。

     创建一组已命名的缩放层。每个层介于最小缩放百分比和最大缩放百分比之间。
   - 将表示您公司资产的图形放到每个缩放层上的相应地理位置。在运行时，每个图形仅在在其缩放层范围内设置地图缩放时显示。如果需要，您可以将任何图形分配给多个缩放层以及缩放层上的多个位置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

有关更多信息和说明，请参阅地图设置。

3. 在“工业图形编辑器”中创建图形。
   - 使用嵌入工业图形添加 **Map_App** 小组件。
   - 编辑 Map_App 小组件的小组件属性。有关更多信息，请参阅在在图形中配置 MapApp。
4. 在 WindowMaker 中创建窗口，并通过拖放方式嵌入所创建的图形。
5. 切换到 WindowViewer。

注意：“MapApp”是指 InTouch MapApp，可用于在 InTouch WindowViewer 中包括地图。将 MapApp 放置到窗口中以用于 WindowViewer 时，它会创建一个“MapControl”。配置 MapApp 是一个全局过程，尽管您可以随时添加或更改信息，但仅需要执行一次。每次将 MapApp 添加到窗口中时，它会创建一个新的 MapControl，您必须对其进行配置。

地图设置

MapApp 的全局属性包含两类信息：

- 源地图定义。您必须定义您想要从地图供应器获取的以用于 WindowViewer 的每个地图。您可以从任意或所有支持的地图供应器定义一个或多个地图。每个源地图要么是基本地图，要么是叠加地图。（一些供应器仅支持基本地图，其他供应器则支持基本地图和叠加地图。）在运行时，基本地图显示为不透明的背景，一个或多个叠加地图可以显示在该地图之上，并且您为每个用户设定了一定的透明度。
- 缩放层及其包含的图形：在**缩放层选项**卡下创建缩放层；每个层定义一个缩放范围（例如，在 10% 到 20% 之间）。然后，在**位置选项**卡下，将代表您企业资产的图形（或任何其他所需图形）添加到每个缩放层的特定地理位置。您可以输入位置，也可以根据指定属性的值定位位置。仅当地图缩放级别位于该缩放层的范围内时，才会显示分配给特定缩放层的图形。缩放层及其包含的图形与源地图无关，因此在运行时使用不同源地图的 WindowViewer 中可以显示相同的图形。

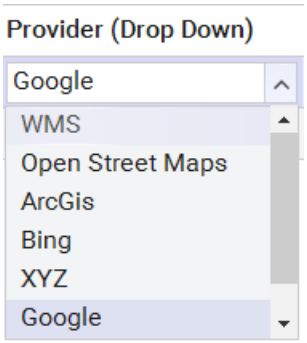
您可以先定义这些全局属性，然后将 MapControls 添加到 Windows 以用于 WindowViewer，也可以选择使用提供的缺省选项。但是，您可以随时返回并更改或添加全局属性。有关添加和配置 MapControl 的更多信息，请参阅在图形中配置 MapApp。

注意：如果更改了 MapApp 全局属性，需要重新打开使用 MapApp 的 WindowViewer，以使它们反映这些更改。

配置 MapApp 全局属性

1. 打开 WindowMaker 以转至工业图形 > 小组件，并双击 **Map_App** 以显示地图设置对话框。
2. 定义源地图：
   a. 选择 源选项卡。
   b. 在第一个空白名称字段中，输入名称并将其赋予该源地图。
   c. 在供应器字段中，选择地图供应器。

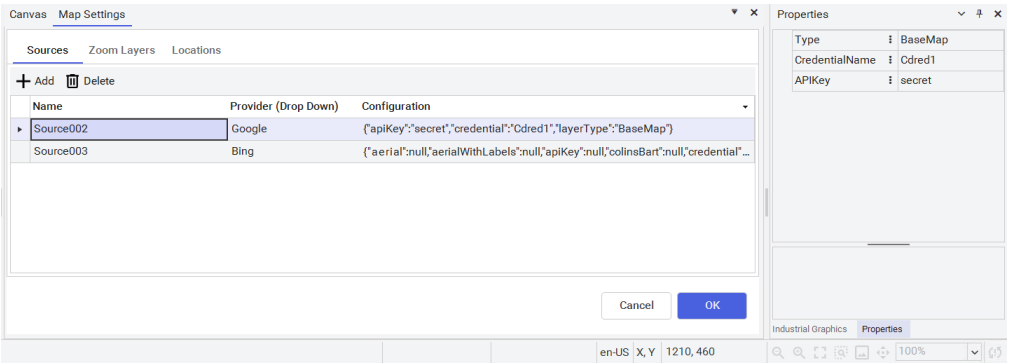
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



属性区域会更新，以包含适用于此供应器的字段。另外，配置 字段会显示此供应器的参数。您不能直接在配置字段中进行更改；该字段反映您在属性字段中输入的信息。

d. 在工业图形窗格的底部，选择属性选项卡以打开选定地图供应器的属性。在属性区域为选定的源地图赋值。有关每个地图供应器属性的更多信息，请参阅地图供应器属性。

配置字段中列出的地图供应器属性会更新，以显示从属性字段中选择的属性 。



e. 如果地图供应器需要凭据，请在 **CredentialName** 字段中选择相应的凭据。

注意：只有在使用需要身份验证的本地地图供应器时，才需要执行此步骤。如果地图供应器不需要用户凭据，可跳过此步骤。

f. 重复执行步骤 b-e 以定义所有源地图。

注意：如果要使用同一供应器的不同源地图，必须单独定义每个源地图。

3. 创建缩放层。缩放层允许您根据给定的缩放级别确定在地图中显示哪些图形。有关详细信息，请参阅缩放层。

缩放级别为 0% 表示无缩放，即整个地球的视图。100% 缩放级别是支持的最大缩放比例，显示街道级别的地图视图。

a. 选择缩放层选项卡。

b. 在第一个空白名称字段中，输入名称并将其赋予该缩放层。

c. 输入最小缩放和最大缩放百分比。

d. 重复执行步骤 b-c，以根据需要创建多个缩放层。

e. 选择一个缩放层并选择设置缺省值，以将其设置缺省缩放层。在下一步向缩放层添加图形时，这些图形将分配给此层，除非您选择其他缩放层。

4. 将图形添加到缩放层的所需位置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

a. 选择位置选项卡。

b. 在工业图形选项卡上，导航到所需图形并将其拖动到位置网格。

注意：此操作仅在已有 UDT 实例可用于与图形关联时有效。

c. 在层中，为图形选择缩放层。缩放层决定了图形在运行时何显示。有关更多信息，请参阅缩放层

d. 重复执行步骤 b-c，直到所有需要的图形添加完毕。您可以将同一图形添加到不同的缩放层以及同一缩放层上的多个位置。位置 选项卡会列出 UDT 实例，这些实例也可以与图形相关联。

地图供应器属性

下表列出了 MapApp 支持的每个地图供应器所需的属性。

如果您使用本地地图服务器存储来自地图供应器的地图，您为每个地图源输入的 URL 应当指向本地服务器上的地图，而不是地图供应器的 Web 站点。

要点：您可能需要为帐户设置一些地图供应器，才能从服务器中访问数据。这些地图供应器要求您在配置地图源属性时使用 API 密钥或特定帐户。

ArcGIS

注意：要使用 ArcGIS 地图并包含 x、y（纬度、经度）及可选 z（海拔）坐标，请使用 XYZ 地图供应器。下面介绍了 XYZ 供应器。

类型	基本地图和叠加地图。
源层	ArcGIS 地图可以包含多个层。您可以选择在此源地图中使用的 ArcGIS 地图层。 有四个源层 操作可供选择。选择操作，然后在右侧字段中列出层： • 显示：仅使用列出的层。 • 隐藏：使用列表之外的任意层。 • 包括：使用缺省层和列出的层。 • 排除：使用缺省层，但不包括列出的层。 要列出层，请使用 ArcGIS 地图中定义的层编号。用逗号分隔多个层。例如， 1,5 指定层 1 和 5。
CredentialName	CredentialName 属性用于指定表示 Windows 用户登录的凭据。当地图服务器需要 Windows 身份验证时，使用此功能。CredentialName 属性是一个下拉控件，列出了凭据管理器中配置的用户名和密码或域、用户名和密码类型的命名凭据 。
源 URL	ArcGIS REST 服务的 URL 的您想要使用的已定义的 ArcGIS 地图。其格式类似于： https://<服务器名称>.arcgisonline.com/ArcGIS/rest/services/Specialty/<地图名称>/MapServer
不透明度	仅适用于叠加地图，表示叠加区域的不透明度。使用数字 0 到 1 表示，0 表示 100% 透明，1 表示 100% 不透明。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

百芜

类型	仅基本地图。
源 URL	要用作基本层的百度地图的 URL。
CredentialName	CredentialName 属性用于指定表示 Windows 用户登录的凭据。当地图服务器需要 Windows 身份验证时，使用此功能。CredentialName 属性是一个下拉控件，列出了凭据管理器中配置的用户名和密码或域、用户名和密码类型的命名凭据 。

必应

类型	仅基本地图。
API 密钥	用于向提供数据的必应地图服务器验证 MapApp 的已注册密钥。 有关获取 API 密钥的信息，请参阅 Microsoft 网站。URL 为 ： https://www.microsoft.com/maps/create-a-bing-maps-key.aspx
CredentialName	CredentialName 属性用于指定表示 Windows 用户登录的凭据。当地图服务器需要 Windows 身份验证时，使用此功能。CredentialName 属性是一个下拉控件，列出了凭据管理器中配置的用户名和密码或域、用户名和密码类型的命名凭据 。
基本源	必应地图支持的基本地图层类型。 • 路：地理区域内的道路。 • 卫星：正射投影航空和卫星图像。 • 混合：具有覆盖道路和标签的正射投影航空和卫星图像。 您可以选择一个或多个基本地图层类型。运行时，用户可以选择要通过 MapApp 显示的基本源。

谷歌

类型	仅基本地图。
CredentialName	CredentialName 属性用于指定表示 Windows 用户登录的凭据。当地图服务器需要 Windows 身份验证时，使用此功能。CredentialName 属性是一个下拉控件，列出了凭据管理器中配置的用户名和密码或域、用户名和密码类型的命名凭据 。
API 密钥	用于向提供数据的谷歌地图服务器验证 MapApp 的已注册密钥。 有关获取 API 密钥的更多信息，请参阅谷歌地图平台网站。URL 为 ： https://developers.google.com/maps/documentation/javascript/get-api-key

打开街道地图

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	描述
类型	仅基本地图。
源 URL	提供基本地图瓦片的地图服务器的 URL。在编写时，对于整个世界的街道地图，您可以使用： https://www.openstreetmap.org 虽然 Open Street Maps 免费提供地图瓦片，但 OSM 服务器存在使用限制，网址：https://openstreetmaps.org。如果使用 OSM 基本地图瓦片服务器，请确保您的使用符合 OSM 许可证要求。
CredentialName	CredentialName 属性用于指定表示 Windows 用户登录的凭据。当地图服务器需要 Windows 身份验证时，使用此功能。CredentialName 属性是一个下拉控件，列出了凭据管理器中配置的用户名和密码或域、用户名和密码类型的命名凭据。

WMS

WMS（Web 地图服务）不是特定的地图供应器，而是一个受多个地图供应器（包括 U. S. Geological Survey、ArcGIS 等）支持的接口标准。

类型	基本地图和叠加地图。
源 URL	您希望使用的地图的 URL。不同的地图源使用不同的 URL 格式，您可以从地图供应器获取格式。
源层	在从 WMS 服务器检索查询时，必须指定一个层，因为地图供应器服务器支持多个层。 如果您不确定用于特定 WMS 服务器的层，可在浏览器中输入以下 URL 查询可用层： <Source URL> ?request=GetCapabilities&service=WMS 此查询会返回一个 XML 文件，其中列出了服务器的所有功能。在“功能/层”下可以找到层列表。每个层都有一个名为“name”的节点，这是应在此属性中指定的层名称。
自定义数据	这作为 WMS 服务器的自定义数据发送。
不透明度	仅适用于叠加地图，表示叠加区域的不透明度。使用数字 0 到 1 表示，0 表示 100% 透明，1 表示 100% 不透明。
CredentialName	CredentialName 属性用于指定表示 Windows 用户登录的凭据。当地图服务器需要 Windows 身份验证时，使用此功能。CredentialName 属性是一个下拉控件，列出了凭据管理器中配置的用户名和密码或域、用户名和密码类型的命名凭据。

XYZ

XYZ 是 ArcGIS 地图供应器的功能，用于向地图中添加包含地理位置（x,y 坐标的形式）的表格数据。如果表中还包含 z 坐标（如海拔值），您可以将表格数据作为 3D 内容添加到您的地球或场景。要添加 x,y 坐标的

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

表，该表必须包含两个字段：一个字段表示 x 坐标，另一个字段表示 y 坐标。字段中的值可以表示任意坐标系统和单位，如纬度和经度或米。支持 3D 几何的 z 坐标字段为可选字段。

类型	基本地图和叠加地图。
源 URL	XYZ 服务器的 URL，采用类似于以下格式： https://<服务器名称>/ArcGIS/rest/services/<地图名称>/MapServer/tile/{z}/{y}/{x}
不透明度	仅适用于叠加地图，表示叠加区域的不透明度。使用数字 0 到 1 表示，0 表示 100% 透明，1 表示 100% 不透明。
CredentialName	CredentialName 属性用于指定表示 Windows 用户登录的凭据。当地图服务器需要 Windows 身份验证时，使用此功能。CredentialName 属性是一个下拉控件，列出了凭据管理器中配置的用户名和密码或域、用户名和密码类型的命名凭据 。

有关几个属性的更多信息

地图源 API 密钥

API 密钥是一个唯一标识符，出于使用和构建的目的，对与地图服务器关联的请求进行身份验证。为了避免将敏感数据（如 API 密钥）保存到应用程序文件夹，可以选择在凭据管理器配置的命名凭证。地图源配置中的 API 密钥属性支持来自命名凭据的密钥。键入 API 密钥时，编辑框将列出“密钥”类型的命名凭据。您可以选择其中一个命名凭据作为 API 密钥。您还可以在编辑框中手动指定 API 密钥。如果您选择了使用命名凭据，在运行时，地图应用程序将从安全位置读取 API 密钥信息。与地图源建立连接时，使用 API 密钥。

注意：仅在使用独立 InTouch 应用程序时，才支持在 API 密钥编辑框中显示命名凭据。

地图源 CredentialName

CredentialName 属性用于指定表示 Windows 用户登录信息的凭据。当地图服务器需要 Windows 身份验证时，使用此功能。**CredentialName** 属性是一个下拉控件，列出了“用户名和密码”或“域、用户名和密码”类型的命名凭据 。对于托管 InTouch 应用程序，列出的凭据将是在 Repository 储备库中配置的凭据。对于独立 InTouch 应用程序， 列出的凭据将是在应用程序管理器中配置的凭据。在运行时，WindowViewer 和地图应用程序将检索凭据，并使用该凭据对本地地图服务器进行身份验证。

设置本地地图服务器身份验证：

1. 安装本地地图服务器（例如：GeoServer。有关更多信息，请参阅设置地图服务器。
2. 创建一个用户，使其用户名和密码与地图服务器中的用户帐户信息相匹配，并授予访问本地服务器中的地图所需的角色。使用此帐户在 InTouch 应用程序管理器中执行相应的身份验证。
3. 在 WindowMaker 中打开该应用程序查看所需的配置。
4. 在 WindowMaker 中，打开 MapApp 源配置页面，并工业图形窗格的底部选择属性选项卡，以打开选定地图供应器的属性。

   将显示相应的属性。
5. 指定 **SourceURL** 以指向本地地图服务器，并选择 **SourceLayer**。
6. 更新 **CredentialName** 属性。

   随后将列出您可以在凭据管理器中创建的凭据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

7. 如果不使用需要身份验证的本地地图服务器，您也可以不作任何选择。
8. 打开 WindowViewer，可以看到 MapApp 已连接到本地地图服务器，而且该服务器已使用配置的命名凭据进行身份验证。

设置地图服务器

以下示例过程介绍了如何使用开源 Java Runtime Environment (JRE) 和名为 GeoServer 的开源地图服务器安装和配置地图服务器。此过程仅举例说明配置基于 Web 的地图服务器的典型工作流程。您可以使用这些产品或其他开源或商业产品作为本地地图服务器，以保存 MapApp 中显示的地图数据。

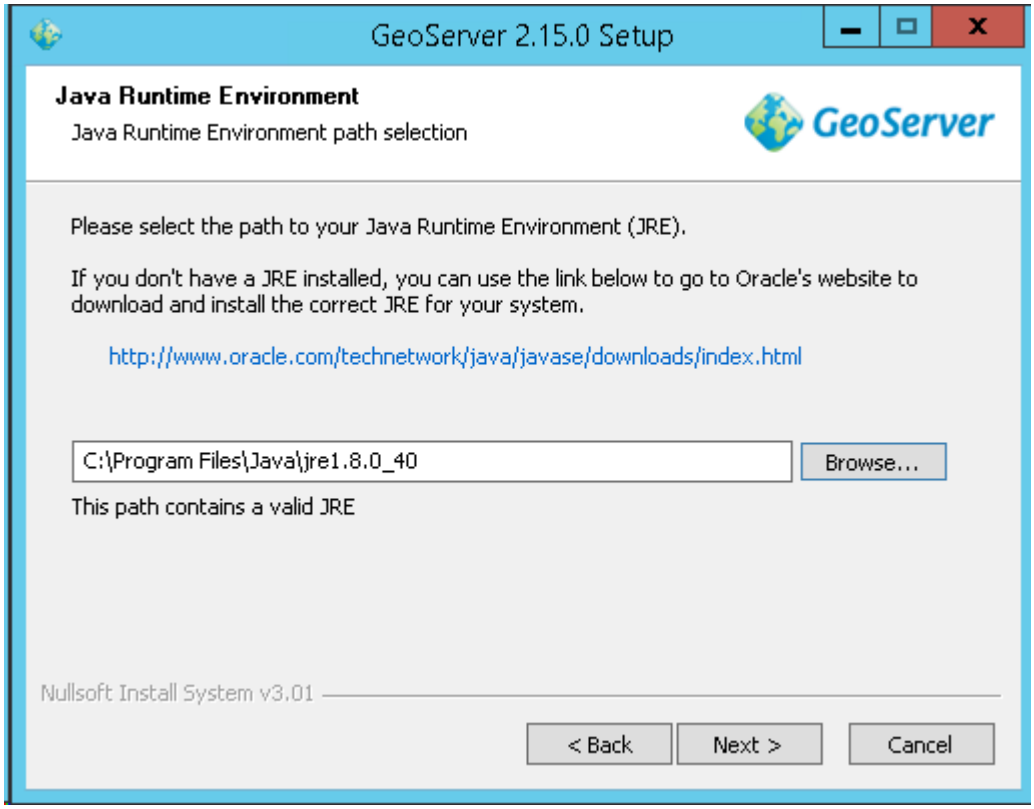
设置本地地图服务器并保存到所需的源地图后，必须配置 MapApp 地图源，以指向本地服务器上的地图位置，而不是指向原始供应器的网站。有关定义地图源的信息，请参阅地图供应器属性。

注意：为了使用户在从另一个域连接时能够查看地图数据，您必须在地图服务器上启用跨域资源共享 (CORS)。CORS 允许一台服务器上的 JavaScript 访问另一台服务器上的资源。**此操作存在安全风险**，因此大多数服务器会禁用 CORS，以减少恶意脚本访问服务器资源的可能性。在确定是否启用 CORS 之前，请务必考虑启用 CORS 的所有安全含义。

配置地图服务器

这只是过程示例。您可以根据自己的需求使用任一可用的服务器产品。

1. 安装 64 位版本的 Java Runtime Environment (JRE)。
2. 安装 GeoServer 并找到 JRE 的文件夹路径。



3. 接受缺省数据目录。

```
C:\Program Files (x86)\GeoServer 2.15.0\data_dir
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 设置 GeoServer 管理员的用户名和密码。

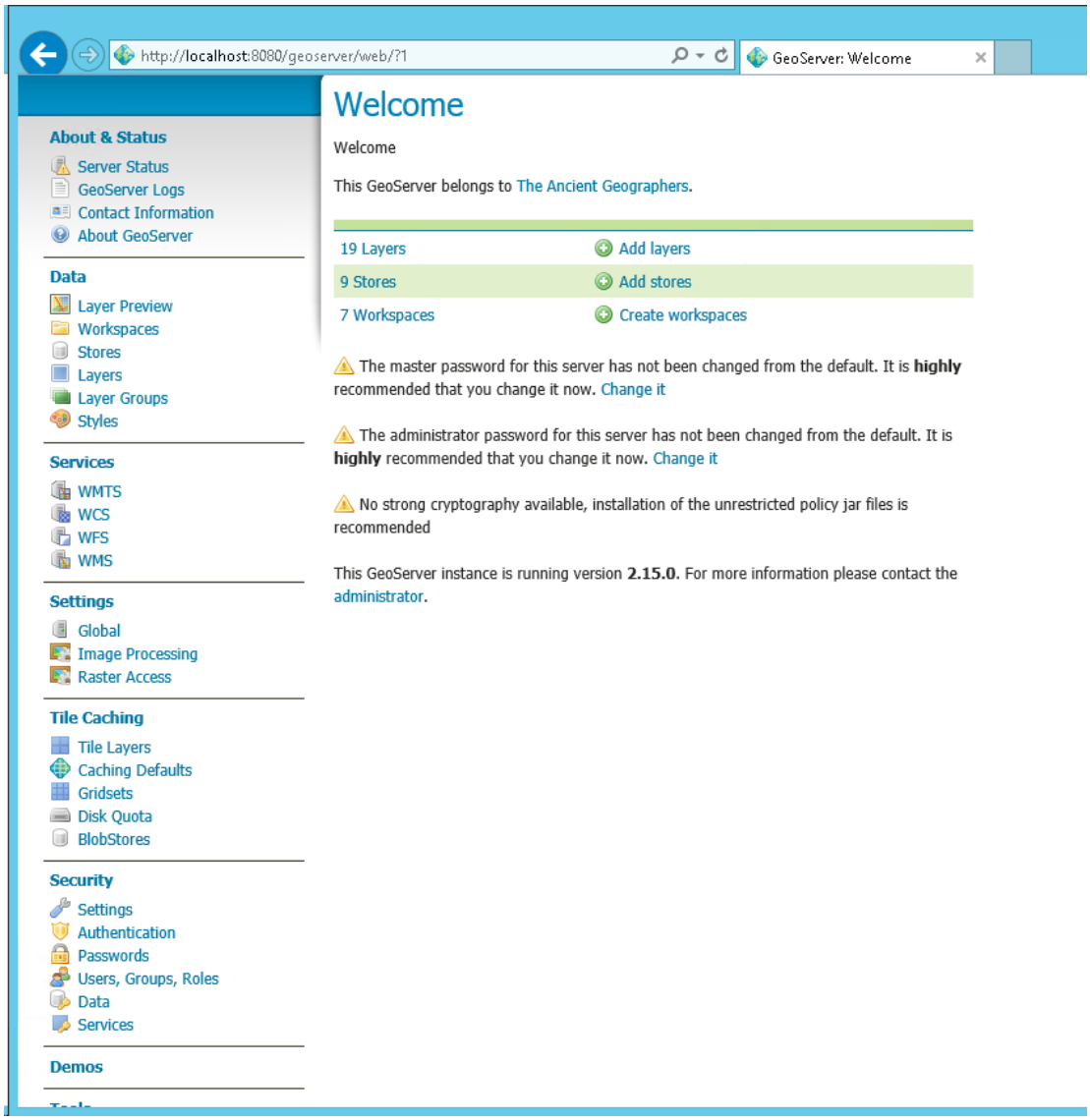
   缺省的管理员用户名和密码是 `admin` 和 `geoserver`。

5. 设置 GeoServer Web 服务器端口，缺省为 8080。
6. 对于安装类型，**选择手动运行选项**。
7. **选择安装开**始安装地图服务器。
8. **选择完成结**束安装过程。
9. 启动 GeoServer。

   您可以在 Windows 的“**开**始”菜单中找到**启动 GeoServer**，也可以从 GeoServer 文件夹（例如 C:\Program Files (x86)\GeoServer 2.15.0\bin）中运行 startup.bat。

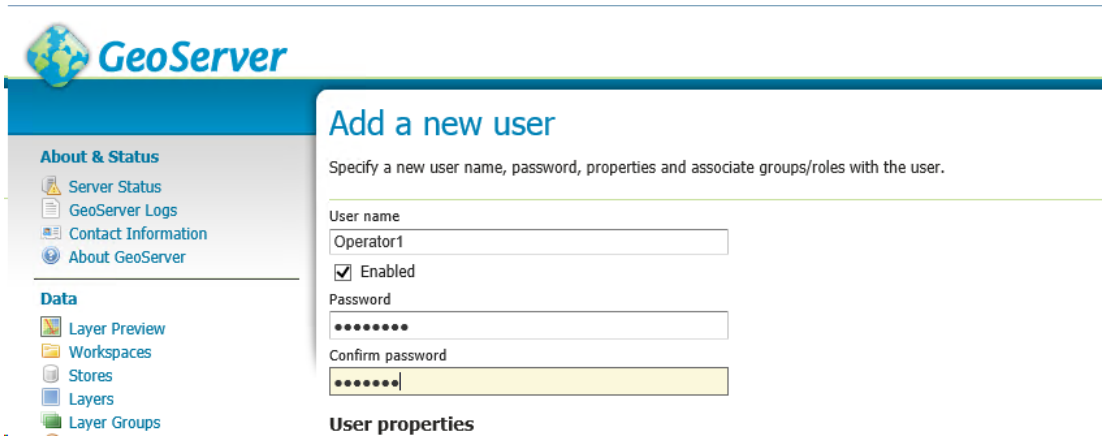
10. 打开 Web 浏览器，然后使用管理员凭据登录 GeoServer 管理员页面。

    `http://localhost:8080/geoserver/web`

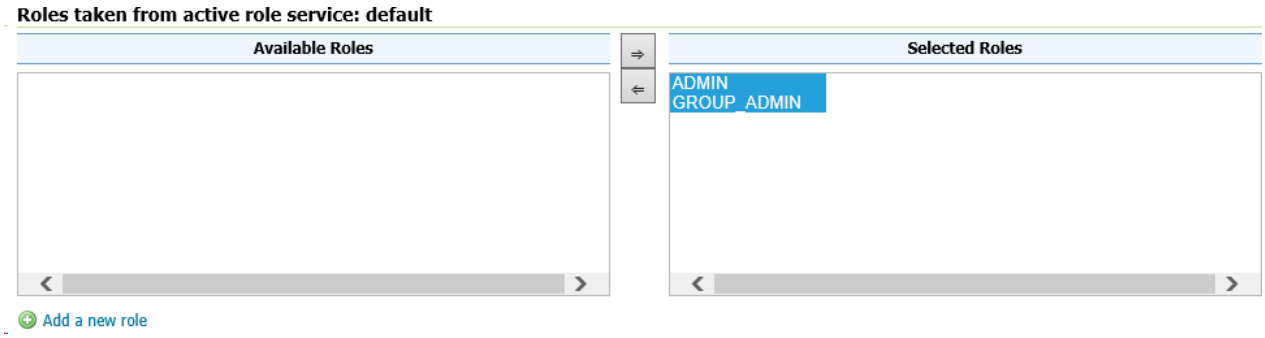


© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

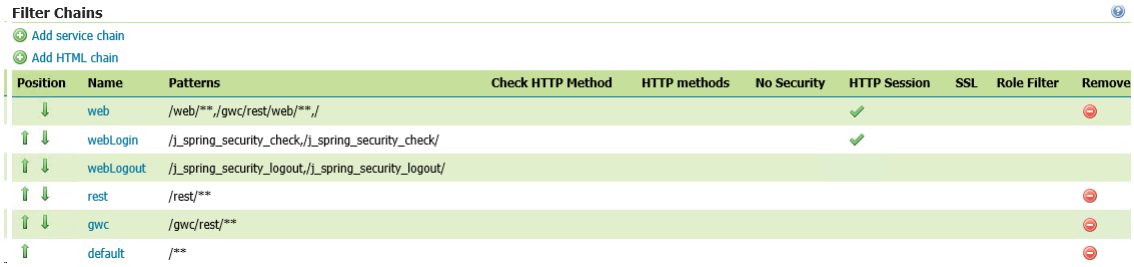
11. 从管理员页面左列中的安全性区域选择用户、组和角色。
12. 从用户、组和角色页面选择添加新用户 。
13. 添加作为凭证组成员的新用户并保存更改。



14. 将分配给凭据用户的可用角色移到所选角色字段。



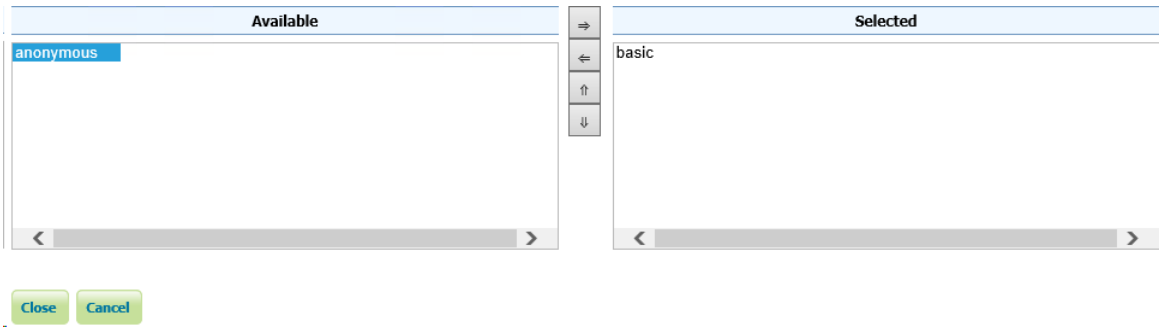
15. 从 管理员网页的“安全性”区域选择身份验证。
16. 从身份验证网页的过滤器链区域选择缺省值。



过滤器链网页随即打开。

17. 将 anonymous 移动到可用列表。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



缩放层

缩放层允许您控制当地图缩放到特定百分比后可在地图上显示的图形。配置缩放层分为两步：

- 创建一个或多个缩放层：每个层定义最小和最大缩放百分比。运行时，如果缩放百分比在这些限制内，则分配给此缩放层的图形将出现在地图中。您可以定义任意多个缩放层，您分配给层的缩放范围可重叠。例如，如果您创建一个范为围 10% 到 30% 的缩放层，并创建另一个范围为 25% 到 50% 的缩放层，当地图缩放到 25% 到 30% 之间时，将包括分配给其中一个缩放层的图形。
- 为每个缩放层分配图形：您可以向位置选项卡下的每个缩放层分配一个或多个图形。当您分配图形时，请选择缩放层，然后通过指定位置的纬度和经度将其分配到该位置。

可能需要尝试在缩放层之间转换多次，才能获得运行时的最佳外观。应将地图放大到多大，才能显示各个位置及这些位置中的资产（而非较大的省/市/自治态或区域符号）？您可以随时修改现有的缩放层或创建新缩放层。覆盖缩放层边界也有所帮助，这样在达到某个百分比时，用户就能同时看到较小的区域和较大的单个位置等情况。

注意： 如果更改缩放层定义或图形，则需要重新打开 WindoeMaker。

位置资产

您可以从位置网格上的“工业图形”窗格拖放图形，以将其放在地图层的指定坐标位置上。位置网格的每一行都会显示运行时将在地图上出现的图形的已分配配置值。

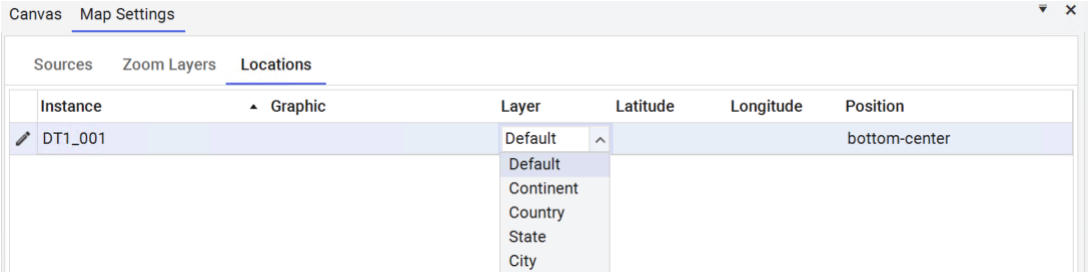
将图形放置到地图上

1. 从地图设置上选择位置选项卡。

   所有可用的 UDT 都将在“实例”下显示。
2. 从“工业图形”窗格找到要放置到地图的图形。
3. 拖动图形并将其放置在位置网格区域。
4. 在列表中选择该图形。

   暗背景用于标识所选的行。
5. 选择图形的层列，以列出定义的地图缩放层。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



每个图形最初都指定给缺省缩放层。

6. 选择要放置图形的缩放层。在运行时，只有地图缩放至该缩放层对应范围内的某个百分比时，图形才会显示。
7. 保存更改。

在图形中配置 MapApp

您可以从“工业图形编辑器”配置 Map_App 小组件的特定属性。

属性	描述
ConfigName	Map_App 小组件全局配置文件的名称。
InitialLatitude	初始中心点地图位置的纬度（以十进制度数表示）。有效纬度值是 +/- 0-90。
InitialLongitude	初始中心点地图位置的经度（以十进制度数表示）。有效经度值是 +/- 0-180。
InitialZoom	地图在运行时最初显示的缩放级别百分比。
MinZoom	运行时可以缩小地图的最小缩放百分比（0-100%）。
MaxZoom	运行时可以放大地图的最大缩放百分比（0-100%）。
MaxBoundsSouth	南部地图边界的纬度，以十进制度数（+/- 0-90）表示，以将屏幕视区中点的垂直平移运动限制到地图底部边界。
MaxBoundsWest	西部地图边界的经度，以十进制度数（+/- 0-180）表示，以将屏幕视区中点的水平平移运动限制到地图左侧边界。
MaxBoundsNorth	北部地图边界的纬度，以十进制度数（+/- 0-90）表示，以将屏幕视区中点的垂直平移运动限制到地图顶部边界。
MaxBoundsEast	东部地图边界的经度，以十进制度数（+/- 0-180）表示，以将屏幕视区中点的水平平移运动限制到地图右侧边界。
Asset	从所显示地图中选择的资产的名称。
CurrentLatitude	地图上显示的所选项目的当前纬度。
CurrentLongitude	地图上显示的所选项目的当前经度。
CurrentZoom	所显示地图的当前缩放级别。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

FollowCurrentAsset	将此属性设置为 true 以使 MapApp 小组件跟随当前所选资产（上下文），并自动平移和缩放地图以显示资产的位置和相关符号（资产应添加到地图应用程序编辑器页面的位置选项卡中，并且资产级别应为 “-1”）。 • 打开地图时，将所选资产置于地图的中心。您可以使用这些资产在 ViewApp 中导航。例如，可以显示所有状态，并为每个状态显示一个标记。然后，通过从地图选择一个状态，您可以将 ViewApp 的焦点设置为显示该状态详细信息的独立窗格。 • 地图将缩放到所选资产的缩放层再加上 1%。 • 如果未选择任何资产或所选资产不在地图上，则地图将显示初始缩放层和地图中心点。 如果已配置 **Asset** 属性，那么将 **FollowCurrentAsset** 属性设置为 false 可使地图能够跟随配置为使用资产位置和关联符号加载地图的资产。
源	在应用程序的地图设置中配置的地图数据源。(All) 是缺省值，其中包括为 Map 应用程序指定的所有地图数据源。 注意：必须将 All 须放在括号内 (All) 作为源属性值。 如果您想要将 Map 应用程序限制为仅显示某些地图源的数据，请使用逗号分隔的字符串来指定多个源。 OSM,Bing,TemperatureOverlay
ZoomLayers	为地图应用程序配置的地图缩放层。(All) 是缺省值，其中包括为地图应用程序指定的所有缩放层。 注意：All 必须放在括号内 (All) 作为 **ZoomLayers** 属性值。 如果您想要将 Map 应用程序限制为仅显示某些缩放层的数据，请使用逗号分隔的字符串按名称指定缩放层。 country,state,city

示例

1. 创建新的 InTouch 应用程序并启动 WindowMaker。
2. 创建 UDT 并重命名为 $Tank。
3. 打开 Tank UDT 并添加属性：
   - Flow：整型
   - Temp：实型
4. 添加位置属性 **Latitude** 和 **Longitude**。
5. 创建第一个 $Tank 实例并将其重命名为 Tank1。
6. 创建第二个实例并将其重命名为 Tank2。
7. 在 Tank1 下创建成员，按如下所示设置成员（属性）值：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- Tank1.Flow = 11
- Tank1.Temp = 54.5
- Tank1.Latitude = 33
- Tank1.Longitude = -114

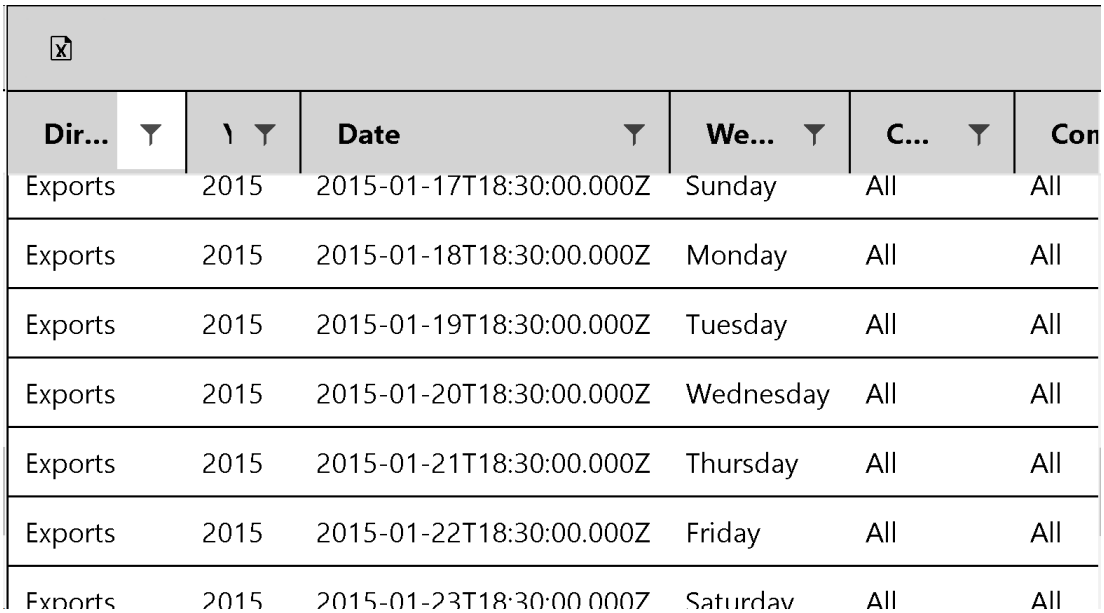
8. 在 Tank2 下创建成员并按如下所示设置成员（属性）值：
   - Tank2.Flow = 21
   - Tank2.Temp = 65.5
   - Tank2.Latitude = 36.746
   - Tank2.Longitude = -119.773
9. 创建新图形 Tank1Sym，在工业图形编辑器中打开。
10. 绘制一个圆角矩形。
11. 为模拟元素添加两个文本基元和值显示动画，如下所示：
    - Tank1\Flow
    - Tank1\Temp
12. 为圆角矩形内的 Tank1 添加标签。
13. 保存并关闭“工业图形编辑器”。
14. 对 Tank2Sym 重复执行步骤 11-15
    - Tank2\Flow
    - Tank2\Temp
15. 在工业图形窗格中，打开小组件文件夹并双击 **Map_App** 打开 Map_App 设置。
16. 在位置 选项卡中，键入或拖放每个资产的图形。将 Tank1Sym 作为 Tank1 资产的图形。将 Tank2Sym 作为 Tank2 资产的图形。
17. 对于 Tank2，从下拉列表中将层设置为状态。
18. 单击确定。
19. 创建新图形，命名为 MyMap，然后在工业图形编辑器中打开。
20. 使用鼠标右键单击画布，然后单击嵌入工业图形。
21. 在小组件文件夹中，选择 Map_App 小组件并将其嵌入画布，然后调整大小，使其适应页面。
22. 编辑 Map_App 小组件的小组件属性。
    - ConfigName = 缺省值
    - CurrentZoom = 5
    - InitialLatitude = 33
    - InitialLongitude = -11
    - InitialZoom = 30
23. 保存并关闭“工业图形编辑器”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

24. 导航到文件 > 新建并创建 Frame_window。
25. 将 MyMap 图形嵌入到步骤 19 中创建的框架窗口。
26. 关闭并保存窗口。
27. 快速切换到 WindowViewer，然后打开包含 MyMap 图形的窗口。
28. 单击 - 按钮进行缩小。

DataGrid 小组件

DataGrid 小组件在运行时会在 InTouch 窗口内展示表格数据。用户可对小组件内数据进行筛选。该小组件支持一系列视觉属性，可自定义小组件外观与表格数据展示样式。

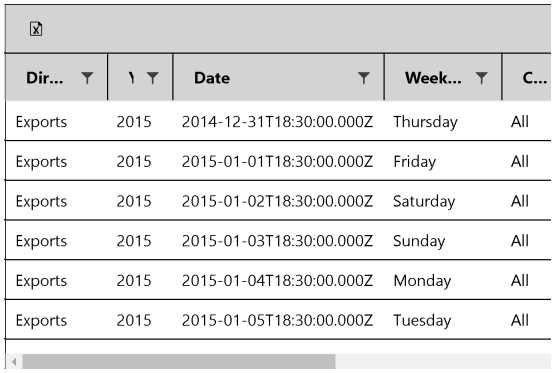


Dir...	\	Date	We...	C...	Con
Exports	2015	2015-01-17T18:30:00.000Z	Sunday	All	All
Exports	2015	2015-01-18T18:30:00.000Z	Monday	All	All
Exports	2015	2015-01-19T18:30:00.000Z	Tuesday	All	All
Exports	2015	2015-01-20T18:30:00.000Z	Wednesday	All	All
Exports	2015	2015-01-21T18:30:00.000Z	Thursday	All	All
Exports	2015	2015-01-22T18:30:00.000Z	Friday	All	All
Exports	2015	2015-01-23T18:30:00.000Z	Saturday	All	All

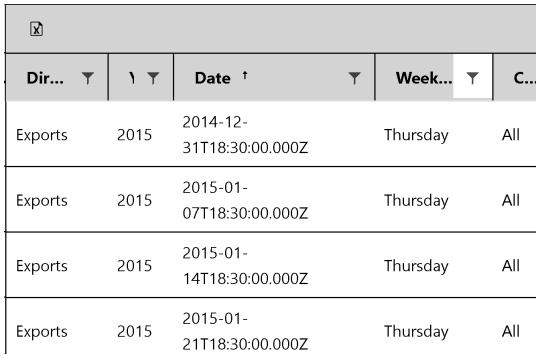
运行时筛选数据

网格的每一列数据均可筛选。应用筛选后，Data Grid 小组件仅显示列项匹配筛选条件的行数据。

下例展示了在“周”列应用筛选器（仅显示周四的数据）前后，表格数据的对比。



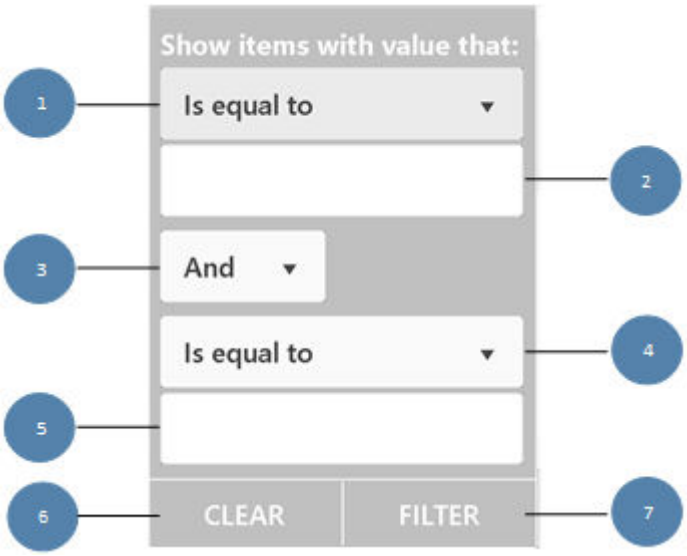
Dir...	\	Date	Week...	C...
Exports	2015	2014-12-31T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-01T18:30:00.000Z	Friday	All
Exports	2015	2015-01-02T18:30:00.000Z	Saturday	All
Exports	2015	2015-01-03T18:30:00.000Z	Sunday	All
Exports	2015	2015-01-04T18:30:00.000Z	Monday	All
Exports	2015	2015-01-05T18:30:00.000Z	Tuesday	All



Dir...	\	Date	Week...	C...
Exports	2015	2014-12-31T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-07T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-14T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-21T18:30:00.000Z	Thursday	All

选择各列标题中的筛选图标，将显示一个下拉窗格，用于指定筛选条件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



1	筛选运算符下拉列表。所选运算符将与配对的数据输入字段 (2) 中输入的值进行比较。如果列数据符合筛选条件，网格将显示该行的数据。不符合条件的数据将被移除，且不会显示在网格中。 Data Grid 小组件支持以下筛选运算符： • Is equal to（等于） • Is not equal to（不等于） • Starts with（开头为） • Contains（包含） • Does not contain（不包含） • Ends with（结尾为） • Is null（为空值） • Is not null（不为空值） • Is empty（为空） • Is not empty（不为空） • Has no value（没有值） • Has value（有值）
2	用于输入筛选值的输入字段，该值将由在 (1) 中选定的筛选运算符进行评估。
3	用于通过布尔运算符 (AND, OR) 评估筛选语句 1 和 2 的筛选逻辑。如果筛选语句 2 为空，则不应用任何筛选逻辑。
4	与 1 相同。如果配对的输入字段 (5) 为空，则不评估筛选运算符。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5	与第 2 点相同。可留空。
6	选择清除以移除筛选条件并显示网格中的所有数据。
7	选择筛选以应用筛选条件。

属性

下表列出了 DataGrid 小组件的属性。所有属性都公开为动态运行时属性，在这些属性中，它们可以绑定到对 ViewApp 命名空间的引用，如下所示：

- 属性
- 对象属性
- MyContent 属性
- 复合表达式

Data Grid 小组件属性	描述
类型	获取或设置源数据字符串的格式。 • csv • json
AllowExport	允许将数据导出到文件。缺省值为 true。
数据来源	获取或设置 DataGrid 小组件显示的 JSON 或 CSV 数据字符串。
ColumnDelimiter	获取或设置用于分隔所选网格行中字段值的分隔符。 • csv 字符串的逗号 • json 字符串的垂直管道 (|)
SelectedRow	获取或设置所选行字段值的字符串。
DateFieldFormat	获取或设置日期的日期格式。支持所有日期格式字符串。
BackgroundColor	小组件的背景颜色。
ExportFileName	如果 AllowExport = True，则指定导出文件的名称。
FilterMode	获取或设置网格的筛选模式；可选值为 default 或 checklist。
前景色	获取或设置小组件的文本颜色。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

HeaderBackColor	显示列名的表格标题的背景颜色。
HeaderForeColor	标题列名的文本颜色。
SelectionBackColor	网格中所选行的背景颜色。
SelectionForeColor	网格内所选行的文本颜色。
RowHoverBackColor	鼠标悬停时网格内该行的背景颜色。
RowHoverForeColor	鼠标悬停时网格内该行的文字颜色。
Font Family	小组件用于显示文本的字体系列名称。
字体大小	获取或设置 Data Grid 小组件中显示的文本的字体大小（以像素为单位）。
PageSize	获取或设置一次要显示的记录数。

配置 Data Grid 小组件

将 Data Grid 小组件准备好在 InTouch HMI 运行时中运行，包括几个配置工作流。

- 为 Data Grid 小组件属性赋值
- 准备要在 Data Grid 小组件中显示的源数据。

本章的以下各节描述了每个配置工作流。

配置 Data Grid 小组件属性

将小组件放置在图形中后，Data Grid 小组件属性将在工业图形编辑器中显示。在图形中选中该小组件后，其属性将显示在属性选项卡上，并可为其赋值。

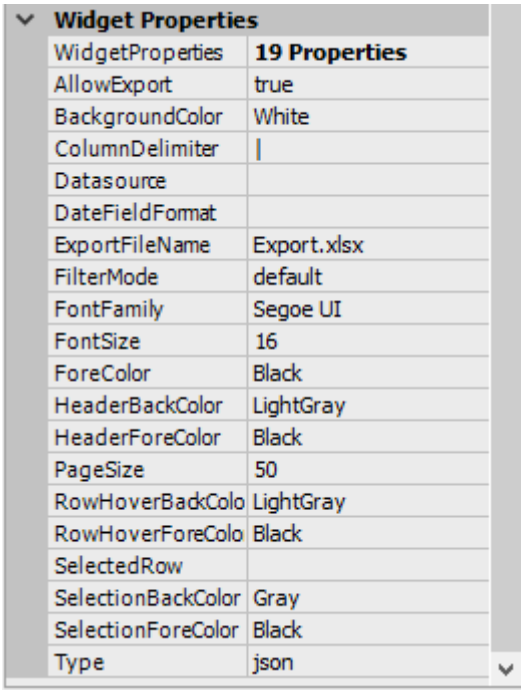
配置 Data Grid 小组件属性

1. 打开 WindowMaker 并通过右键单击工业图形窗格并选择新建图形来创建工业图形。
2. 双击新建的图形，在工业图形编辑器中打开。
3. 在工具栏中选择嵌入工业图形图标 。
4. 在弹出的窗口中，于图形工具箱窗格中选择小组件文件夹，然后选择 **Data Grid** 小组件缩略图并单击确定。
5. 在画布上选择您希望放置 Data Grid 的任意位置。
6. 在画布上选中 Data Grid 小组件的缩略图，并选择右侧的属性选项卡。

   属性选项卡显示 Data Grid 小组件的属性。下图显示了分配给该小组件的缺省值。

   有关 Data Grid 小组件属性的更多信息，请参阅 Data Grid 属性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



Widget Properties	
WidgetProperties	**19 Properties**
AllowExport	true
BackgroundColor	White
ColumnDelimiter	|
Datasource	
DateFieldFormat	
ExportFileName	Export.xlsx
FilterMode	default
FontFamily	Segoe UI
FontSize	16
ForeColor	Black
HeaderBackColor	LightGray
HeaderForeColor	Black
PageSize	50
RowHoverBackColo	LightGray
RowHoverForeColo	Black
SelectedRow	
SelectionBackColor	Gray
SelectionForeColor	Black
Type	json

7. 为属性赋值并保存更改。

Data Grid 属性

下表列出了 Data Grid 小组件的属性。所有属性都公开为动态运行时属性，在这些属性中，它们可以绑定到对 ViewApp 命名空间的引用，如下所示：

- 属性
- 对象属性
- MyContent 属性
- 复合表达式

Data Grid 小组件属性	描述
类型	获取或设置源数据字符串的格式。 • csv • json
数据来源	获取或设置 DataGrid 小组件显示的 JSON 或 CSV 数据字符串。
ColumnDelimiter	获取或设置用于分隔所选网格行中字段值的分隔符。 • csv 字符串的逗号 • json 字符串的垂直管道 (|)

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Data Grid 小组件属性	描述
SelectedRow	获取或设置所选行字段值的字符串。
DateFieldFormat	获取或设置日期的日期格式。支持所有日期格式字符串。
BackgroundColor	小组件的背景颜色。
前景色	小组件的文本颜色。
HeaderBackColor	显示列名的表格标题的背景颜色。
HeaderForeColor	标题列名的文本颜色。
SelectionBackColor	网格中所选行的背景颜色。
SelectionForeColor	网格内所选行的文本颜色。
RowHoverBackColor	鼠标悬停在网格内某行上的该行的背景颜色。
RowHoverForeColor	鼠标悬停在网格内某行时的该行文字颜色。
Font Family	Data Grid 小组件用于显示文本的字体系列名称。
字体大小	Data Grid 小组件中显示的文本的字体大小（以像素为单位）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Data Grid 小组件属性	描述
Pagesize	小组件显示的网格行数。

为 Data Grid 小组件准备源数据

Data Grid 小组件显示的源数据可以是 CSV 或 JSON 字符串格式。

对于格式为 CSV 字符串的源数据，第一行被视为标题（列名）。后续行均为数据。通常对于 CSV 格式文件，行内数据元素之间使用逗号作为分隔符。

```
Direction,Year,Date,Weekday,Country,Commodity,Transport_Mode,Measure,Value,Cumulative
Exports,2015,01/01/2015,Thursday,All,All,All,$,104000000,104000000
Exports,2015,02/01/2015,Friday,All,All,All,$,96000000,200000000
Exports,2015,03/01/2015,Saturday,All,All,All,$,61000000,262000000
Exports,2015,04/01/2015,Sunday,All,All,All,$,74000000,336000000
```

对于格式为 JSON 字符串的源数据，系统会将其解析为字典对象数组。数组中的第一个项被视为 Data Grid 小组件的标题（列名）。

在解析数据的过程中，字段的数据类型也会被确定。确定的数据类型可以是以下任何一种：

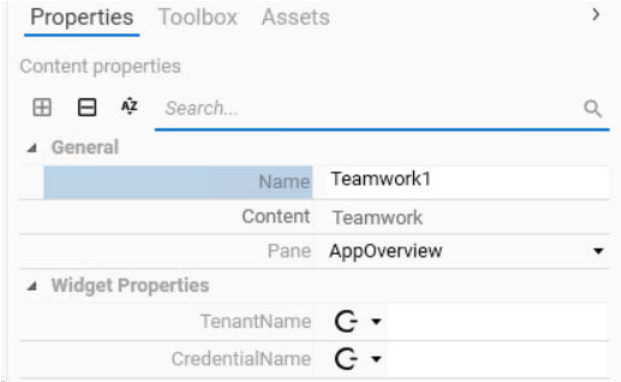
- 字符串型
- 编号
- 日期

  解析器仅在值符合 ISO 日期格式（例如 2021-08-26T07:44:36.079Z）或小组件的 **DateFieldFormat** 属性指定的格式时，才会将字符串识别为日期。

Teamwork 小组件

Teamwork 小组件包含两个属性：

- **TenantName**（必填）需设置为您分配的租户名称。
- **CredentialName**（可选）需设置为用户凭据名称或留空。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	描述	数据类型	缺省值
TenantName	运行时用于显示特定 Teamwork 页面的租户名称。每个租户的名称都是唯一的。 请咨询您的管理员以获取租户名称。	字符串常量或引用	<Empty string>
CredentialName （可选）	目前不支持 **CredentialName** 属性。要自动登录 Teamwork，用户应使用 SSO 方法。	字符串常量或引用	<Empty string>

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 14 使用 XML 文件导入窗口

您需要创建命令文件，然后使用该命令文件运行 WindowMaker 以从 XML 文件导入窗口。除了导入窗口外，命令文件还可用于：

- 创建新的应用程序。
- 删除窗口。
- 重命名窗口。
- 将打印信息发送到文件或打印机。
- 将交叉引用发送到文件。

正在进行 XML 导入时，不支持对引用的符号执行以下操作：

- 删除符号
- 重命名符号
- 调换符号名

但是，可以在 XML 导入完成后执行这些操作。

不支持以下系统级别的操作：

- 启动控制台会话
- 将焦点切换到其它应用程序

您可以从命令行运行 InTouch DBDump 和 DBLoad 实用程序。但是，对于托管的 InTouch 应用程序，请从 System Platform IDE 扩展运行 DBDump 和 DBLoad 实用程序。DBDump 会创建一个导出文件，其中包含 InTouch 应用程序中的标记信息。DBLoad 会将标记信息导入到 InTouch 应用程序中。如需有关 DBDump 和 DBLoad 实用程序的详细信息，请参阅《AVEVA™ InTouch HMI 应用程序维护》。

预备 XML 文件

以下操作程序描述了如何创建 XML 文件并将窗口定义导入到 InTouch 应用程序中。

1. 创建 XML 文件。此文件没有定义要使用的工具或基本 XML 语法。
2. 检查 XML 文件是否符合 XML 格式标准。您可以使用 XML 验证工具来验证 XML 文件的语法。要查看 XML 文件，请在 Internet Explorer 中将其打开。
3. 预备应用程序中的标记。您可以使用 InTouch DBDump 和 DBLoad 实用程序来提取、修改和重新加载标记。如需有关 DBDump、DBLoad 和 CSV 文件的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序维护*》。
4. 预备 WindowMaker 命令以执行要完成的函数。
5. 从 InTouch 应用程序、WindowMaker 和 WindowViewer 退出。
6. 从命令行提示符运行 WindowMaker 命令文件，或从某个程序或 IDE 扩展运行命令文件。
7. 检查是否有错误。如果定义了错误日志文件，请检查该文件并检查 Logger。如果从某个程序运行了命令文件，请检查返回代码。

以下样本文件可帮助您使用 XML 导入功能来创建和导入窗口定义：

- 用于将窗口定义加载到 InTouch 应用程序的样本 XML 文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 样本命令文件。
- 两个架构文件。导入分析器不使用这些架构文件。架构文件比 XML 导入分析器的限制多。

注意：安装完成后，架构文件位于 InTouch 安装文件夹中。

预备命令文件

WindowMaker 命令文件是一种文本文件。系统会从该文件中读取命令，然后对 InTouch 应用程序数据集执行操作。

该命令文件使用单字节 ANSI 字符集。您不能使用多字节字符集 (MBCS) 或 Unicode 字符。

文本文件语法的细节如下：

- 使用 8 位 ANSI 字符集，但字符数不能超过 127 个。大多数控制字符会被去除。
- 以 # 字符开头的行被视为注释。
- 空白行会被忽略。
- 行以回车换行符 (CRLF) 序列终止。
- 空格可以出现在行开头、行结尾、以及命令、等号字符及其参数之间。
- 如果参数需要引号，则引号与其它文本之间不允许有空格。
- 在处理每行之前，会删除所有前导和尾部空格。
- 每个命令都以句点开头。
- 每个命令文件都必须以 WindowMaker 命令文件命令 (.WINDOMAKERCOMMANDFILE) 开头。

一个命令文件可以包含多个命令。如果一个命令失败，命令处理器会继续处理文件中的下一个命令。

创建最小命令文件

最小命令文件的示例如下：

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
```

将打印信息发送到文件

您可以将整个 InTouch 应用程序的清单发送到打印机或文本文件。如果 InTouch 窗口包含工业图形，则会将输出发送到 HTML 文件。

以下命令文件示例是将应用程序信息打印到名为 Printer01 的打印机：

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.PRINTAPPLICATIONINFORMATION
.OUTPUTTARGET=Printer
.OUTPUTTARGETNAME=Printer01
.GO
```

下面是打印到文件的命令文件示例。如果该文件存在，则会被覆盖。

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.PRINTAPPLICATIONINFORMATION
.OUTPUTTARGET=TextFile
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
.OUTPUTTARGETNAME=C:\MyApps\AppInfo.txt
.GO
```

将交叉引用发送到文件

您可以创建交叉引用并将其发送到文件。如果该文件存在，则会被覆盖。无需用户交互。

命令文件示例：

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.CROSSREFERENCE
.SEARCHFOR=TagName
.REFERENCETYPE=ByWindow
.OUTPUTFILE=C:\MyApps\AppCrossRef.Csv
.GO
```

创建日志文件

您可以将提示性消息、警告和错误记录到文本文件。如果不指定此命令，则会将处理状态发送到 Logger。如果该文件存在，则会被覆盖。

将错误记录到文件的命令文件示例。

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.COMMANDLOGFILE=C:\MyApps\LogFile.Txt
```

确保文件夹 MyApps 存在。如果文件夹 MyApps 不存在，则不会创建日志文件。

命令语法

下表列出了可以在 WindowMaker 命令文件中使用的命令。

命令	描述
.WINDOWMAKERCOMMANDFILE	将文件识别为 WindowMaker 命令文件。如果找不到此命令，则不会处理文件，并且 WindowMaker 会退出并显示错误码。
.VERSION=1	此命令向 WindowMaker 指示要读取的文件不太新。
.COMMANDLOGFILE=<Full File Path>	将提示性消息、警告和错误记录到此文件。如果该文件存在，则会被覆盖。如果省略此命令，请检查 Logger 以了解有关处理状态的信息。
.GO	运行命令。如果命令包含一个或多个参数，在其之后必须使用 .GO。
.WINDOWCREATE	根据 XML 文件创建窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

命令	描述
.XMLFILEPATH=<Full File Path>	包含窗口规格的 XML 文件的完整文件路径。在使用 WINDOWCREATE 命令时是必需的。
.WINDOWDELETE	按名称删除指定的 InTouch 应用程序窗口。找不到窗口名时，不会生成错误。
.WINDOWNAME=<Window Name>	要删除的窗口的名称。在使用 WINDOWDELETE 命令时是必需的。
.WINDOWRENAME	重命名窗口。如果找不到“旧”名称，命令会发出警告。如果新名称存在，则会生成一条错误消息。如果无法重命名窗口，则会发生错误。如果旧名称与新名称一致，则不执行任何操作。
.OLDWINDOWNAME=<Existing Window Name>	当前窗口名。在使用 WINDOWRENAME 命令时是必需的。
.NEWWINDOWNAME=<New Window Name>	新窗口名。具有此名称的窗口不得存在。在使用 WINDOWRENAME 命令时是必需的。
.PRINTAPPLICATIONINFORMATION	将 InTouch 应用程序数据集打印或转储到文本文件。这等同于在 WindowMaker 中的文件菜单命令上选择打印选项。
.OUTPUTTARGET=Printer|TextFile	将输出发送到打印机或文本文件。
.OUTPUTTARGETNAME=<Printer Name>|<Full Text File Path>	打印机或输出文件的名称。如果该文件存在，则会被覆盖。
.CROSSREFERENCE	以逗号分隔变量 (CSV) 格式生成交叉引用信息。
.SEARCHFOR= TagName|QuickFunctions	按名称搜索标记或 QuickFunction。
。REFERENCETYPE= ByTagName|ByWindow	按标记名或窗口名交叉引用。
。OUTPUTFILE=<Full File Path>	输出文件的完整路径。如果该文件存在，则会被覆盖。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

创建一个应用程序。

您可以使用 WindowMaker 命令文件来创建新的缺省 InTouch 应用程序。要创建空白应用程序，可以使用最小命令文件。但是，无法使用最小命令文件创建空白的托管 InTouch 应用程序。如需详细信息，请参阅创建最小命令文件。

命令文件必须位于要创建应用程序的文件夹中。如果文件夹中包含现有 InTouch.ini 文件，则不会生成应用程序。

新应用程序文件夹的路径不能包含嵌入的“-l”或“-L”字符串序列。例如，无法创建文件夹 `C:\MyApps\App-Large`。

所创建的 InTouch.ini 文件包含类似于以下示例的内容。窗口位置取决于显示应用程序的屏幕分辨率。应用程序具有标题和描述“Generated InTouch Application”。缺省语言为英语。

InTouch.ini 文件内容示例：

```
[InTouch]
AppMode=2
AppName0=Generated InTouch Application
AppName1=
AppName2=
AppName3=
AppDesc0=Generated InTouch Application
AppDesc1=
AppDesc2=
AppDesc3=
LanguageBase=English (United States)
LanguageBaseID=1033
InTouchView=0
SAOConverted=1
WinFullScreen=1
WinLeft=-4
WinTop=-4
WinWidth=1288
WinHeight=1004
SnapOn=1
```

将标记添加到新生成的应用程序

新的 InTouch 应用程序仅包含系统标记。如果要在导入窗口之前将标记添加到应用程序中，则需要：

1. 创建新的应用程序。
2. 运行 DBLoad 以将标记导入到新应用程序中。
3. 导入窗口。

例如：

```
WM.Exe C:\MyApps\App001 COMMANDFILE="C:\BlankFile.Txt"
DBLoad C:\MyApps\App001,C:\TagDumps\App001.Csv,0
WM.Exe C:\MyApps\App01 COMMANDFILE="C:\Commands.Txt"
```

删除窗口

如果要将窗口添加到现有应用程序，但其中已有同名窗口，则必须删除现有窗口。您可以使用 WindowMaker 命令文件来删除现有窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要点：要删除的窗口不存在时，不会出现错误消息。

无法删除现有窗口时，Logger 中会出现一条消息。无需用户交互。

要从 InTouch 应用程序中删除窗口，需要在命令文件中输入命令序列。命令文件可以包含多个删除窗口命令序列。

下例显示了用于从 InTouch 应用程序中删除窗口的命令序列：

```
.WINDOWMAKERCOMMANDFILE
.VERSION=1
.WINDOWDELETE
.WINDOWNAME=Window002
.GO
```

重命名窗口

您可以更改 InTouch 应用程序中窗口的名称。

要重命名 InTouch 应用程序中的窗口，需要在命令文件中输入命令序列。命令文件可以包含多个重命名窗口命令序列。

要重命名的窗口不存在时，不会出现错误消息。

如果新窗口名已存在，则不会执行任何操作，但会记录警告消息。

如果旧名称存在而新名称不存在，但重命名因其他某种原因而失败，则会记录错误消息。

无需用户交互。警告和错误消息会出现在 Logger 中。

如果将窗口重命名为现有窗口的名称，则会在日志文件或 Logger 中记录一条警告消息。在以下示例中，将 Window005 重命名为现有窗口 Window006 后，在日志文件或 Logger 中记录了一条警告消息。

```
.WINDOWRENAME
.OLDWINDOWNAME=Window005
.NEWWINDOWNAME=Window006
.GO
```

导入窗口

您可以将窗口导入到 InTouch 应用程序中。包含单个 InTouch 窗口规格的 XML 文件的完整文件路径是必需的。新窗口名包含在 XML 文件中。

在以下情况下，不会导入窗口：

- 应用程序中存在同名窗口。
- 某个窗口脚本中发生未解决的错误。
- 尝试将对象的一部分添加到窗口时发生未解决的错误。

在以下情况下，可能需要用户交互：

- 脚本或表达式中指定的元素包含错误或者缺失。
- 脚本包含语法错误。

可能会不显示任何内容，而让用户来更正问题。如需详细信息，请检查日志文件和 Logger。

要导入窗口，需要在命令文件中输入命令序列。

例如：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
.WINDOWCREATE
.XMLFILEPATH=C:\WMCommandTest\WMCreateFile.Xml
.GO
```

命令文件可以包含一个导入窗口命令以及多个创建窗口命令序列。

处理错误

您的 XML 文件必须遵循常规 XML 格式规则。如果无法使用 Internet Explorer 打开 XML 文件，说明您的 XML 文件中包含 XML 格式错误。修复所有错误后，可以将该文件用于 WindowMaker。

如果所使用的文件包含常规格式错误，WindowMaker 会在 Logger 中记录错误消息“无法加载 XML 文件”。

脚本中缺少任何元素时、发生任何脚本错误时或发生其它任何元素规格错误时，WindowMaker 都会停止。例如：

- 缺少标记
- 缺少外部 WindowMaker 脚本扩展 DLL
- 缺少 ActiveX 控件
- 缺少向导 DLL

在某些情况下，例如对于特定动画链接或自定义属性覆盖，即使缺少标记，也不会出现消息框。系统会在 Logger 中写入备注，但不会创建动画链接和对象，也不会创建窗口。在其它情况下，脚本和表达式分析会停止。此时，您可以创建标记，如果创建成功，处理将会继续。

缺少 SmartSymbol

如果 SmartSymbol 模板在目标应用程序中不存在，则不会导入窗口。Logger 和输出文本文件中会出现一条错误消息。

如果出于某种原因而未能创建 SmartSymbol 实例，则不会创建窗口。即使 SmartSymbol 模板存在，SmartSymbol 导入也可能会失败。如需有关失败的更多信息，请查看 Logger。

缺少工业图形

如果工业图形引用在 Galaxy 储备库中不存在，则不会创建窗口。Logger 和输出文本文件中会出现一条错误消息。

即使工业图形引用存在，XML 导入过程也可能会失败。如需有关失败的更多信息，请查看 Logger。

表达式、标记名和脚本

在需要表达式、标记名或脚本时，不能使用空文本或仅包含空白的文本。如果这些文本项目的表述不正确，则无法正确分析表达式、标记名或脚本，而且还会导致错误。

以标记名为例，标记类型必须与对象的预期类型一致。在大多数情况下，可以使用点域来访问标记属性。例如，可以按如下方式来访问整型标记 iTag005 的第二位：

```
iTag005.02
```

在本例中，标记和点域的计算结果为离散值，可用于需要离散标记名的情况，而单独使用 iTag005 则会导致错误。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从命令提示符运行 WindowMaker

您可以从命令提示符运行 WindowMaker。

命令行如下：

```
WM.EXE AppPath,Commandfile="Command.txt"
```

参数

AppPath
定义 InTouch 应用程序的完整路径。如果在应用程序的文件夹中运行 WindowMaker，则此参数是可选的。

CommandFile
定义命令文件的完整路径。必须用引号将路径名括起来。等号两侧不允许有额外的空格。

假设 InTouch 应用程序在 `C:\MyApps` 文件夹中，且命令文件为 `C:\WMCommandFile.txt`。下例显示了从命令提示符输入的 WindowMaker 命令：

```
WM.EXE C:\MyApps,COMMANDFILE="C:\WMCommandFile.Txt"
```

注意：不能从命令提示符创建托管的 InTouch 应用程序。

System Platform IDE 扩展

使用 System Platform IDE 扩展，可以执行托管应用程序的 InTouch XML 导入功能。您可以使用 IDE 扩展来选择 XML 导入过程的命令文件。选择命令文件后，会启动 WindowMaker，并会分析关联的 XML 文件。

要点：如果在 XML 导入正在进行时启动新的控制台会话或最大化现有控制台会话，WindowMaker 会停止响应。

System Platform IDE 有一个上下文菜单项用于为 XML 导入选择命令文件。使用鼠标右键单击 InTouchViewApp 衍生模板时，上下文菜单中会显示**处理 InTouch 命令文件**菜单项。但是，在选择以下项目时则不会显示该菜单项：

- 多个 InTouchViewApp 模板
- InTouchViewApp 基本模板
- InTouchViewApp 实例

从较旧版本的 InTouch 迁移托管的 InTouch 应用程序时，必须至少在 WindowMaker 中打开 InTouch 应用程序一次，然后才能处理命令文件。

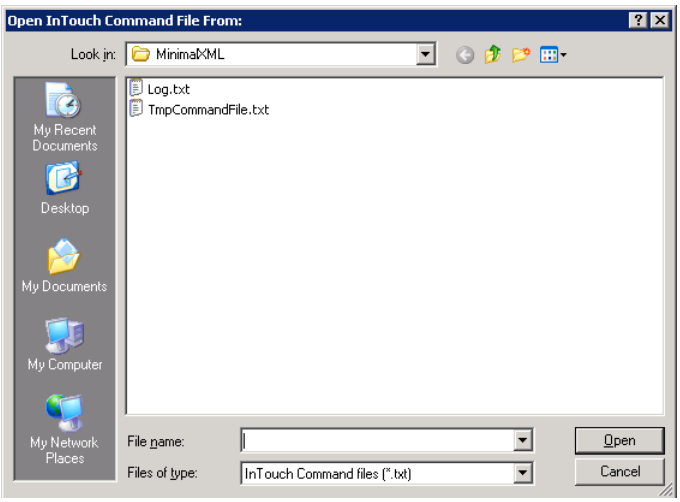
从托管的 InTouch 应用程序运行 WindowMaker

您可以从托管的 InTouch 应用程序运行 WindowMaker。

从托管的 InTouch 应用程序运行 WindowMaker

1. 打开 System Platform IDE。
2. 在**模板工具箱**中，根据 InTouchViewApp 模板创建一个衍生模板。
3. 通过创建新的 InTouch 应用程序或导入现有的独立 InTouch 应用程序，将 InTouch 应用程序与衍生模板关联起来。
4. 使用鼠标右键单击衍生模板，然后选择**处理 InTouch 命令文件**。此时出现**从以下位置打开 InTouch 命令文件：对话**框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



注意：InTouchViewApp 衍生模板必须至少在 WindowMaker 中配置一次，然后才能执行**处理 InTouch 命令文件**命令。

5. 浏览到命令文件的位置，然后**选择打开**。WindowMaker 将启动。

从命令提示符运行 DBDump

使用 DBDump 可以从 InTouch 应用程序中提取标记信息。您可以从命令提示符运行 DBDump。在运行 DBDump 之前，必须停止 WindowMaker。

DBDump 命令参数之间需要用逗号分隔。参数与位置相关。如果要在包含的参数之间省略某个参数，必须加入一个逗号来代表缺失的参数。下例显示了从命令提示符运行 DBDump 命令的语法：

```
DBDump AppPath,CsvPath,GroupTypes,OverwriteCsvFile, MessageBoxes
```

参数

AppPath
定义 InTouch 应用程序的路径。

CsvPath
定义导出文件的路径，该文件包含 InTouch 应用程序的“标记名字典”中的标记定义。

GroupTypes
指定标记在 DBDump 导出文件中是否按 InTouch 标记类型进行分组。1 表示标记数据库名称应当按标记组类型排序。0 表示标记名称不按类型排序。

OverwriteCsvFile
指定是否应覆盖导出文件。1 表示应覆盖导出文件。0 表示不应覆盖导出文件。

MessageBoxes
指定当 DBDump 实用程序导出应用程序标记名目录的内容时是否显示消息。1 表示显示消息框。0 表示不显示消息框。

示例

假设 InTouch 应用程序位于 `C:\MyInTouchApps\App001` 文件夹中，并且您希望将标记数据库的内容写入 `C:\TagDumps\App001.csv` 文件。您不希望显示任何消息框，但希望在目标导出文件存在时将其覆盖。命令如下：

```
DBDump C:\MyInTouchApps\App001,C:\TagDumps\App001.Csv,1,1,0
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

为托管的 InTouch 应用程序运行 DBDump

您可以为托管的 InTouch 应用程序运行 DBDump。

为托管的 InTouch 应用程序运行 DBDump

1. 打开 System Platform IDE。
2. 在模板工具箱中，右键单击 InTouchViewApp 衍生模板，指向导出，然后选择选作 CSV。

   此时出现 **Exporting automation objects to CSV**（将自动化对象导出到 CSV）对话框。
3. 输入 CSV 文件的名称，然后选择保存。应用程序标记数据将成功转储到 CSV 文件。

从命令提示符运行 DBLoad

使用 DBLoad 可以将标记信息导入到 InTouch 应用程序中。您可以从命令提示符运行 DBLoad。在运行 DBLoad 之前，必须停止 WindowMaker。

DBLoad 命令参数与位置相关，如果要在包含的参数之间省略某个参数，必须加入一个逗号来代表缺失的参数。下例显示了从命令提示符运行 DBLoad 命令的语法：

```
DBLoad AppPath,CsvPath,MessageBoxes
```

参数

AppPath
指定 InTouch 应用程序的路径。

CsvPath
指定文件的路径，该文件包含要导入到应用程序的“标记名字典”中的标记定义。

MessageBoxes
指定当 DBLoad 实用程序将导入文件的内容导入到应用程序的“标记名字典”中时是否显示消息。1 表示显示消息框。0 表示不显示消息框。

示例

假设 InTouch 应用程序位于 `C:\MyInTouchApps\App001` 文件夹中，并且您希望从 `C:\TagDumps\App001.Csv` 文件读取标记数据库信息。您不希望显示消息框。下例显示了在命令提示符中输入的 DBLoad 命令：

```
DBLoad C:\MyInTouchApps\App001,C:\TagDumps\App001.Csv,0
```

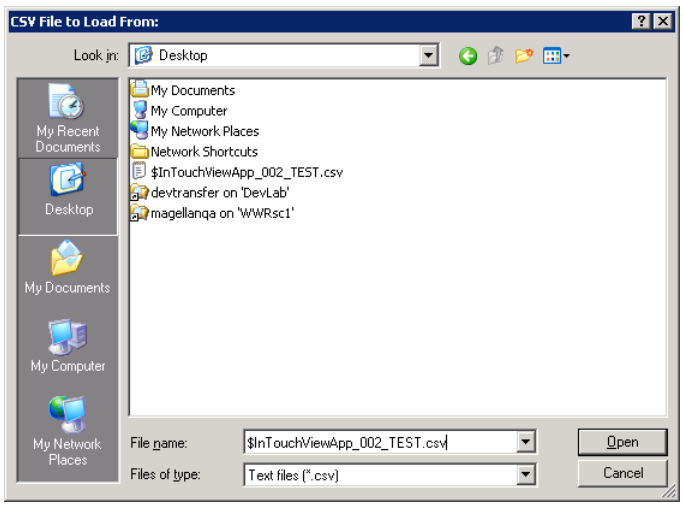
为托管的 InTouch 应用程序运行 DBLoad

您可以为托管的 InTouch 应用程序运行 DBLoad。

为托管的 InTouch 应用程序运行 DBLoad

1. 打开 System Platform IDE。
2. 在模板工具箱中，使用鼠标右键单击 InTouchViewApp 衍生模板，指向导入，然后选择 **DB Load**。此时出现要从中加载的 CSV 文件：对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 浏览以找到 CSV 文件，然后选择打开。标记数据将成功加载到 InTouch 应用程序中。

XML 格式

您可以导入 XML 文件中定义的窗口和大多数窗口元素。但无法从 XML 文件导入标记。必须改为使用 DBLoad 实用程序。

常规 XML 文件格式

XML 文件必须遵循所有常规 XML 格式规则。您的 XML 文件应该能够在 Internet Explorer 中打开。如果不能，说明您的 XML 文件包含格式错误。

使用架构

您可以根据架构文件来验证 XML 文件。使用架构文件来验证 XML 文件可检测到大部分 XML 格式错误。

架构比 XML 输入分析器的限制多。

- 架构要求元素的顺序与各表中列出的顺序一致。
- 元素名称和值区分大小写。
- 在某些情况下，架构要求显式定义元素。

您需要在窗口元素中调用架构验证。如需有关如何指定架构的信息，请参阅窗口定义。

XML 文件标题

XML 声明必须位于文件顶部。最简声明如下：

```
<?xml version="1.0"?>
```

非必需的 XML 元素

InTouch XML 导入功能不会针对未定义的元素、属性或节点生成错误或警告。

您必须正确地输入所有内容。如果您打算使用 <ONRIGHTDOWN> 操作脚本，但不小心输入了 <ONRITEDOWN>，则不会生成警告，并且您的操作脚本不会添加到窗口对象中。

但针对缺少必需元素的 XML 定义，则会创建警告和错误。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用户提供的文本

如果您的脚本使用 XML 字段分隔符，脚本文本必须封装在 CDATA 元素中。XML 文件导入功能不会分析 CDATA 元素中的文本。

输入文本时也可以不将文本封装在 CDATA 元素中，前提是文本不包含任何 XML 字段分隔符。

保留文本空格

在处理不在 CDATA 元素中的文本时，会删除前导和尾部空格。因此，元素（如 <Title> MyWindowName</Title> 会生成没有前导空格的窗口名称“MyWindowName”。

要保留文本中的前导和尾部空格，请将文本封装在 CDATA 元素中。例如，元素 <Title><![CDATA[MyWindowName]]></Title> 会生成窗口名称“MyWindowName”。

通用元素定义

一些元素或定义由许多元素共享。

颜色元素

颜色元素可以按 RGB 值、名称、引用值或整数值来指定。颜色元素包括 R、G、B、Name、Ref 和 Value。这些元素也在其它元素中使用。例如，FillColor、TextColor 和 BGCOLOR。

RGB 元素

使用 RGB 元素可以指定颜色。可指定给 RGB 元素的值范围是 0 到 255。缺少某个元素时，将使用缺省值。缺省值取决于窗口对象。

示例：

```
<FillColor>
    <R>192</R>
    <G>192</G>
    <B>192</B>
</FillColor>
  <TextColor> <R>0</R><G>0</G><B>0</B> </TextColor>
```

颜色名称元素

您可以使用 Internet Explorer V3.0 或更高版本支持的名称来指定颜色。颜色名称区分大小写，并且必须与已知 HTML 颜色一致。

以下网站包含颜色名称和值的列表：

http://www.w3schools.com/html/html_colornames.asp

http://www.learningwebdesign.com/colornames.html

http://www.oreilly.com/catalog/wdnut/excerpt/color_names.html

以下网站包含颜色名称的列表：

http://www.geocities.com/SiliconValley/Pines/6986/colortbl.html

示例：

```
<FillColor>
    <Name>White</Name>
</FillColor>
  <TextColor> <Name>White</Name> </TextColor>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

颜色引用元素

您可以使用十六进制颜色值来指定颜色。颜色值必须始终包含六个十六进制数字。

以下网站按名称列出了颜色及其对应的十六进制值：

http://www.w3schools.com/html/html_colornames.asp

http://www.learningwebdesign.com/colornames.html

以下网站包含非抖动颜色值的列表：

http://www.htmlgoodies.com/tutorials/colors/article.php/3479001

以下网站包含 4096 种颜色及其十六进制代码：

http://yorktown.cbe.wwu.edu/sandvig/MIS314/Assignments/A03/ColorHexCodes.asp

示例：

```
<FillColor>
    <Ref>#FF00FF</Ref>
</FillColor>
  <TextColor> <Ref>#FF00FF</Ref> </TextColor>
```

颜色值元素

您可以使用正整数颜色值来指定颜色。值必须在 0 到 16777215 的范围内。

示例：

```
<FillColor>
    <Value>16711935</Value>
</FillColor>
  <TextColor> <Value>16711935</Value> </TextColor>
```

TextInfo 元素

您可以使用 TextInfo 元素来指定文本如何显示在屏幕上。

以下元素可用于指定文本：

元素	描述
字体	字体系列的名称，例如“System”或“Arial”。
FontStyle	值可以为 {Regular, Italic, Bold, BoldItalic}。
字体大小	需要指示字体大小度量单位。通常以磅为单位。值可以为 0 或更大的值。
下划线	带下划线的文本：{true 或 false}。
删除线	带删除线的文本：{true 或 false}。
TextColor	文本颜色的颜色元素。
TextJustify	文本对齐方式：{Left、Center 或 Right}

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例：

```
<TextInfo>
    <Font>Arial</Font>
    <FontStyle>Regular</FontStyle>
    <FontSize>12</FontSize>
    <Underline>false</Underline>
    <Strikeout>false</Strikeout>
    <TextColor>
      <R>0</R>
      <G>0</G>
      <B>0</B>
    </TextColor>
    <TextJustify>Left</TextJustify>
  </TextInfo>
```

Point 元素

Point 元素用于定义其它元素的位置。Point 元素包含两个元素：X 和 Y。X 和 Y 元素必须包含 -32000 到 32000 范围内的值。

元素	描述
X	左坐标，以像素为单位。必需。
是	上坐标，以像素为单位。必需

示例：

```
<Point>
    <X>10</X>
    <Y>25</Y>
  </Point>
```

Pen 元素

Pen 元素用于指定对象边框的线条特征。

以下元素可用于指定 Pen 元素：

字段	描述
PenColor	使用颜色元素：RGB 元素、名称元素、引用元素或值元素。
PenWidth	笔画宽度，以像素单位。值可以为 1、2、4、6、9 或 11。
PenStyle	值可以为 None、Solid、Dash、Dot、DashDot、DashDotDot。

示例：

```
<Pen>
    <PenColor>
      <R>0</R>
      <G>0</G>
      <B>0</B>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
    </PenColor>
    <PenWidth>4</PenWidth>
    <PenStyle>Solid</PenStyle>
  </Pen>
```

Dimension 元素

Dimension 元素包含用于指定屏幕上对象的左上角坐标的元素。Dimension 元素还包含用于指定长方形对象的宽度和高度的元素。

在 WindowMaker 中，坐标的垂直和水平方向限制都是 -32000 到 +32000。指定 X 和 Y 位置的组合时，如果宽度或高度值使计算的坐标位于 (-32000, -32000, 32000, 32000) 边界之外，则将出现警告，并且值将限定为最大值。高度和宽度值应该为正数。

在 Dimension 元素中，可以使用以下元素来指定屏幕上的区域。

元素	描述
向左	左边缘坐标。必需。
顶端	上边缘坐标。必需。
宽度	宽度，以像素为单位。
高度	高度，以像素为单位。

示例：

```
<Dimension>
    <Left>4</Left>
    <Top>4</Top>
    <Width>632</Width>
    <Height>278</Height>
  </Dimension>
```

表达式

表达式文本封装在 CDATA 部分中。这可防止因 XML 分隔符（在表达式中是有效的）而导致的文件分析失败。

示例：

```
<EXPRESSION>
<![CDATA[
表达式文本的第一行
表达式文本的第二行
表达式文本的第 N 行
]]>
</EXPRESSION>
```

虚拟键代码和虚拟键标记

一些 InTouch 动画链接支持键盘输入。

XML 分析器会将虚拟键的名称转换为虚拟键代码。此外，修饰符键的标帜是使用文本而不是数字位组合来指定的。键名不区分大小写。

InTouch 应用程序可以使用下表中列出的虚拟键名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

所代表的键	虚拟键名
添加	添加
英文字母键	A 到 Z
BACKSPACE	Backspace
取消	CtrlBreak
清除	清除
复制	复制
十进制	十进制
删除	删除
除	除
向下箭头	下
空字符串表示没有分配。	<空白>
END	结束
Enter	返回
ESC	转义
执行	执行
F1	F1
F2	F2
F3	F3
F4	F4
F5	F5
F6	F6
F7	F7
F8	F8
F9	F9
F10	F10
F11	F11
F12	F12
F13	F13

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

所代表的键	虚拟键名
F14	F14
F15	F15
F16	F16
帮助	帮助
HOME	主页
插入	插入
向左箭头	向左
乘	乘
数字键	1 到 9
数字键盘 0	NUMPAD0
数字键盘 1	NUMPAD1
数字键盘 2	NUMPAD2
数字键盘 3	NUMPAD3
数字键盘 4	NUMPAD4
数字键盘 5	NUMPAD5
数字键盘 6	NUMPAD6
数字键盘 7	NUMPAD7
数字键盘 8	NUMPAD8
数字键盘 9	NUMPAD9
NUM LOCK	NumLock
PAGE UP	PageUp
PAGE DOWN	PageDown
PRINT SCREEN	打印
向右箭头	向右
选择	选择
分隔符	分隔符
空格键	空格
减	减

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

所代表的键	虚拟键名
TAB	Tab
向上箭头	上

支持两个修饰符键。对于 CKEYFLAGS 元素，可以单独或以组合形式指定修饰符键。修饰名称是元素值。只要修饰符名称在属性值字符串中，修饰符键就适用。

名称	所代表的键
CTRL	必须将 CONTROL 键与常规键一起按。
Shift	必须将 SHIFT 键与常规键一起按。
CTRL + SHIFT	必须同时按 CONTROL 键和 SHIFT 键。

用于生成 CTRL+A 序列的 XML 为 ：

```
<VirtualKeyType>
   <KeyCode>A</KeyCode>
   <KeyFlags>CTRL</KeyFlags>
</VirtualKeyType>
```

窗口元素

在 XML 文件中，只能包含一个窗口元素。窗口元素必须是 XML 文件中指定的第一个元素。

窗口名不得与 InTouch 应用程序中的现有窗口名相同。

窗口定义

您可以使用以下元素来指定窗口。

元素	描述
标题	窗口名，非空字符串。最多 32 个字符，尾部空格不计算在内。必需。
注释	注释文本。最多 59 个字符。
维数	使用可选架构时必需。不使用架构时，如果未包含 Dimension 元素，则会应用缺省尺寸。缺省值为 4 4 632 278。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
WindowStyle	窗口类型 = {Replace | Overlay| Popup}。
BackgroundColor	颜色元素指定的窗口背景颜色：RGB 元素、名称元素、引用元素或值元素。
FrameStyle	{Single | Double | None}。
TitleBar	启用标题栏 = {true | false}。
CloseButton	启用关闭窗口按钮 = {true | false} 仅当启用标题栏时才能启用。
SizeControls	启用大小控件 = {true | false}。
ScriptOnShow	脚本元素。
ScriptWhileShowing	脚本元素。
ScriptOnHide	脚本元素。
ObjectList	窗口包含的对象，包括 SmartSymbol。

以下示例会创建一个空窗口：

```
<?xml version="1.0" encoding="UTF-8"?>
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/Window
 wwInTouchWindow.xsd" Version="1">
  <Title>ApplicationSecondWindow</Title>
  <Comment>Main application window</Comment>
  <Dimension>
    <Left>4</Left>
    <Top>4</Top>
    <Width>632</Width>
    <Height>278</Height>
  </Dimension>
  <BackgroundColor>
    <R>192</R>
    <G>192</G>
    <B>192</B>
  </BackgroundColor>
  <WindowStyle>Overlay</WindowStyle>
  <FrameStyle>Single</FrameStyle>
  <TitleBar>true</TitleBar>
<CloseButton>True</CloseButton>
  <SizeControls>true</SizeControls>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
</iw:InTouchWindow>
```

最简示例：

```
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/Window
 wwInTouchWindow.xsd" Version="1">
  <Title>ApplicationSecondWindow</Title>
  <Dimension><Left>4</Left><Top>4</Top>
    <Width>632</Width>
     <Height>278</Height></Dimension>
</iw:InTouchWindow>
```

激活架构验证

将特定数据放置在 InTouchWindow 元素中，可以激活架构验证和处理。激活架构时，必须将 InTouchCommon.Xsd 和 InTouchWindow.Xsd 文件放置在 XML 文件所在的文件夹中。

使用架构的示例：

```
<iw:InTouchWindow
xmlns:iw="http://www.wonderware.com/InTouch/Window"
xmlns:itc="http://www.wonderware.com/InTouch/Common"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/Window wwInTouchWindow.xsd"
Version="1">
</iw:InTouchWindow>
```

不使用架构的示例：

```
<InTouchWindow Version="1">
</InTouchWindow>
```

窗口脚本

您可以在 XML 导入文件中创建三种类型的窗口脚本：OnShow、WhileShowing 和 OnHide。可以将脚本元素放置在窗口元素中。

脚本文本可以封装在 CDATA 部分中，以防止脚本中的 XML 分隔符干扰 XML 文件分析。

OnShow 窗口脚本元素

OnShow 脚本元素包含脚本文本。

示例：

```
<ScriptOnShow><![CDATA[
    脚本文本的第一行
    脚本文本的第二行
    脚本文本的第 N 行
  ]]></ScriptOnShow>
```

最简示例：

```
<ScriptOnShow>单行脚本文本</ScriptOnShow>
```

WhileShowing 窗口脚本元素

WhileShowing 脚本包含两个元素。脚本文本可以封装在 CDATA 部分中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
文本	窗口脚本的文本。必需。
频率	脚本执行频率，以毫秒为单位。使用可选架构时必需。缺省值为 1000。

示例：

```
<ScriptWhileShowing>
    <Text><![CDATA[
        脚本文本的第一行
        脚本文本的第二行
        脚本文本的第 N 行
      ]]></Text>
    <Frequency>1000</Frequency>
  </ScriptWhileShowing>
```

最简示例：

```
<ScriptWhileShowing>
    <Text>单行脚本文本</Text>
    <Frequency>1000</Frequency>
  </ScriptWhileShowing>
```

OnHide 窗口脚本元素

OnHide 窗口脚本包含一个元素。脚本文本可以封装在 CDATA 部分中。

示例：

```
<ScriptOnHide>
      <![CDATA[
        脚本文本的第一行
        脚本文本的第二行
        脚本文本的第 N 行
      ]]>
  </ScriptOnHide>
```

最简示例：

```
<ScriptOnHide>脚本文本行</ScriptOnHide>
```

窗口对象

窗口对象放置在 ObjectList 元素定义的对象列表中。

示例：

```
<?xml version="1.0" encoding="UTF-8"?>
<iw:InTouchWindow xmlns:iw="http://www.wonderware.com/InTouch/Window" xmlns:itc="http://
www.wonderware.com/InTouch/Common" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.wonderware.com/InTouch/Window.xsd" Version="1">
<Title>Window0001</Title>
<Comment>Sample simple window</Comment>
<Dimension>
   <Left>10</Left>
   <Top>10</Top>
   <Width>400</Width>
   <Height>400</Height>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
</Dimension>
<BackgroundColor>
   <R>255</R>
   <G>0</G>
   <B>255</B>
</BackgroundColor>
<WindowStyle>Replace</WindowStyle>
<FrameStyle>Double</FrameStyle>
<TitleBar>true</TitleBar>
<SizeControls>true</SizeControls>
<ObjectList>
   <Rectangle>
      <Title>Rectangle1</Title>
      <Pen>
         <PenColor>
            <Name>Black</Name>
         </PenColor>
         <PenWidth>4</PenWidth>
         <PenStyle>Solid</PenStyle>
      </Pen>
      <Dimension>
         <Left>100</Left> <Top>50</Top>
         <Width>270</Width> <Height>80</Height>
      </Dimension>
      <FillColor>
         <R>128</R> <G>128</G> <B>128</B>
      </FillColor>
   </Rectangle>
</ObjectList>
</iw:InTouchWindow>
```

每个窗口对象都可以包含动画链接。包含动画链接的窗口对象具有一个动画链接元素。

```
<AnimationLinks> </AnimationLinks>
```

缺省元素值

未指定窗口对象元素时，会使用下表中的元素值作为缺省值。

元素	缺省值
FILLCOLOR	<R>212</R><G>208</G> <B>200</B>
PENWIDTH	1
PENCOLOR	黑色
PENSTYLE	实线
CORNERDIMENSION	<Width>20</Width> <Height>20</Height>
TEXTCOLOR	黑色
TEXTJUSTIFY	向左

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	缺省值
ROTATION	0
FONT	系统
FONTSTYLE	常规
FONTWEIGHT	忽略
FONTSIZE	10
UNDERLINE	False
STRIKEOUT	False
FLIP	无
TRANSPARENT	<R>0</R><G>255</G> <B>0</B>

笔样式限制

如果将笔样式指定为实线以外的样式，则笔画宽度强制设置为 1。要指定大于 1 的笔画宽度，必须删除笔样式或设置为实线。

笔尺寸

尺寸规格适用于对象的中线。如果对象的笔刷宽度大于 1，则对象可能会超出指定的尺寸。此时，笔刷宽度将跨越对象的边界。一些像素位于对象边界之内，而另一些像素则位于边界之外。

长方形对象

可以为长方形元素指定以下元素。

元素	描述
FillColor	长方形的填充颜色。使用颜色元素来指定。缺省值为 rgb(212, 208, 200)。
笔	Pen 元素。
维数	对象的位置和大小。元素为 top、left、width、height。top、left、width 和 height 值以像素为单位。生成的坐标必须在 -32000 到 +32000 之间。width 和 height 都不能为 0。必需。 如果为 Dimension 元素指定了无效的值或该元素缺少值，则不会创建对象。
标题	对象名。可选。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
动画链接	可选的动画链接列表。

示例：

```
<Rectangle>
    <Title>Rectangle1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>4</PenWidth>
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Dimension>
      <Left>100</Left> <Top>50</Top>
      <Width>270</Width> <Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <AnimationLinks> </AnimationLinks>
  </Rectangle>
```

最简示例：

```
<Rectangle>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Rectangle>
```

圆角长方形对象

可以为圆角长方形对象指定以下元素。

元素	描述
FillColor	填充颜色。使用颜色元素来指定。缺省值为 rgb(212, 208, 200)。
笔	Pen 元素。
维数	对象的位置和大小。元素为 top、left、width、height。top、left、width 和 height 值以像素为单位。生成的坐标必须在 -32000 到 +32000 之间。width 和 height 都不能为 0。必需。 如果为 Dimension 元素指定了无效的值或该元素缺少值，则不会创建对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
CornerDimension	width 和 height 元素。角宽度不能超过长方形的宽度。角高度不能超过长方形的高度。角宽度和角高度必须至少为 1。缺省值为 20, 20。

标题	对象名。可选。
动画链接	可选的动画链接列表。

示例：

```
<RoundedRectangle>
    <Title>RoundedRectangle1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>4</PenWidth>
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Dimension>
      <Left>100</Left>
      <Top>50</Top>
      <Width>270</Width>
      <Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <AnimationLinks>
    </AnimationLinks>
    <CornerDimension>
      <Width>8</Width>
      <Height>8</Height>
    </CornerDimension>
  </RoundedRectangle>
```

最简示例：

```
<RoundedRectangle>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </RoundedRectangle>
```

椭圆对象

可以为椭圆对象指定以下元素。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
FillColor	填充颜色。使用颜色元素来指定。缺省值为 rgb(212, 208, 200)。
笔	Pen 元素。
维数	对象的位置和大小。元素为 top、left、width、height。top、left、width 和 height 值以像素为单位。生成的坐标必须在 -32000 到 +32000 之间。width 和 height 都不能为 0。必需。 如果为 Dimension 元素指定了无效的值或该元素缺少值，则不会创建对象。
标题	对象名。可选。
动画链接	可选的动画链接列表。

示例：

```
<Ellipse>
    <Title>椭圆 1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>4</PenWidth>
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Dimension>
      <Left>100</Left>
      <Top>50</Top>
      <Width>270</Width>
      <Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <AnimationLinks>
    </AnimationLinks>
  </Ellipse>
```

最简示例：

```
<Ellipse>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Ellipse>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

线条对象

可以为线条对象指定以下元素。起点和终点不能是同一个点。

元素	描述
笔	Pen 元素
标题	对象名。可选。
动画链接	可选的动画链接列表。
标记点	Point 元素。必须包含两个点。额外的 Point 元素将被忽略。

示例：

```
<Line>
    <Title>线 1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>150</X><Y>160</Y></Point>
    </Points>
    <AnimationLinks>
    </AnimationLinks>
  </Line>
```

最简示例：

```
<Line>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>150</X><Y>160</Y></Point>
    </Points> </Line>
```

水平线对象

可以为水平线对象指定以下元素。这将生成 WindowMaker 水平/垂直线对象。

元素	描述
笔	Pen 元素。
标题	对象名。可选。
动画链接	可选的动画链接列表。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
标记点	必须包含两个点。将忽略第二个点的 Y 坐标，并将其设置为第一个点的 Y 坐标。额外的 Point 元素将被忽略。必需。 如果为 Points 元素指定了无效的值或该元素缺少值，则不会创建对象。

示例：

```
<HorizontalLine>
    <Title>线 1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>150</X><Y>150</Y></Point>
    </Points>
    <AnimationLinks>
    </AnimationLinks>
  </HorizontalLine>
```

最简示例：

```
<HorizontalLine>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>150</X><Y>150</Y></Point>
    </Points>
  </HorizontalLine>
```

垂直线对象

可以为垂直线对象指定以下元素。这将生成 WindowMaker 水平/垂直线对象。

元素	描述
笔	Pen 元素。
标题	对象名。可选。
动画链接	可选的动画链接列表。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
标记点	必须包含两个点。将忽略第二个点的 X 坐标，并将其设置为第一个点的 X 坐标。额外的 Point 元素将被忽略。必需。 如果为 Points 元素指定了无效的值或该元素缺少值，则不会创建对象。

示例：

```
<VerticalLine>
    <Title>线 1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>50</X><Y>250</Y></Point>
    </Points>
    <AnimationLinks>
    </AnimationLinks>
<VerticalLine>
```

最简示例：

```
<VerticalLine>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>50</X><Y>250</Y></Point>
    </Points>
<VerticalLine>
```

折线对象

必须定义至少两个点，才能加载折线对象。不建议对这两个点使用相同的坐标。

可以为折线对象指定以下元素。

元素	描述
标记点	Point 元素，至少需要两个。 如果为 Points 元素指定了无效的值或该元素缺少值，则不会创建对象。
笔	Pen 元素。
标题	对象名。可选。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
动画链接	可选的动画链接列表。不允许填充颜色、文本颜色、填充百分比、值显示。

示例：

```
<Polyline>
    <Title>polyline1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>60</X><Y>250</Y></Point>
      <Point><X>70</X><Y>350</Y></Point>
      <Point><X>80</X><Y>450</Y></Point>
    </Points>
    <AnimationLinks>
    </AnimationLinks>
  </Polyline>
```

最简示例：

```
<Polyline>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>80</X><Y>450</Y></Point>
    </Points>
  </Polyline>
```

多边形对象

必须定义至少两个点，才能加载多边形对象。不建议对这两个点使用相同的坐标。

可以为多边形对象指定以下元素。

元素	描述
标记点	包含 Point 元素。至少需要两个点。X 和 Y 值以像素为单位。
FillColor	多边形对象的填充颜色。包含一个颜色元素。缺省值为 rgb(212, 208, 200)。
笔	Pen 元素。
标题	对象名。可选。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
动画链接	可选的动画链接列表。不允许文本颜色、值显示动画链接。

示例：

```
<Polygon>
    <Title>polygon1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>1</PenWidth>
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>60</X><Y>250</Y></Point>
      <Point><X>70</X><Y>350</Y></Point>
      <Point><X>80</X><Y>450</Y></Point>
    </Points>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <AnimationLinks>
    </AnimationLinks>
  </Polygon>
```

最简示例：

```
<Polygon>
    <Points>
      <Point><X>50</X><Y>150</Y></Point>
      <Point><X>80</X><Y>450</Y></Point>
    </Points>
  </Polygon>
```

文本对象

可以为文本对象指定以下元素。

元素	描述
标题	对象名。可选。
TextString	显示的文本字符串。必需。如果为 TextString 元素指定了无效的值或该元素缺少值，则不会创建文本对象。
TextInfo	定义文本的显示方式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
旋转	定义文本的方向，以度为单位。可能的值为 :{0, 90, 180, 270}。默认为 0。
动画链接	可选的动画链接列表。
维数	对象的位置和大小。元素为 top、left、width、height。top、left、width 和 height 值以像素为单位。生成的坐标必须在 -32000 到 +32000 之间。width 和 height 都不能为 0。必需。如果为 Dimension 元素指定了无效的值或该元素缺少值，则不会创建对象。

示例：

```
<Text>
    <Title>text1</Title>
    <TextString>This is some text to display</TextString>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <TextInfo>
      <Font>Arial</Font>
      <FontStyle>Regular</FontStyle>
      <FontSize>12</FontSize>
      <Underline>false</Underline>
      <Strikeout>false</Strikeout>
      <TextColor>
        <R>0</R>
        <G>0</G>
        <B>0</B>
      </TextColor>
      <TextJustify>Left</TextJustify>
    </TextInfo>
    <Rotation>0</Rotation>
    <AnimationLinks>
    </AnimationLinks>
  </Text>
```

最简示例：

```
<Text>
    <TextString>This is some text to display</TextString>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Text>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

位图对象

如果指定了源图像，则位图对象必须存在。支持的图形文件格式为：JPG、JPEG、BMP、TIF、PCX、BIF 和 TGA。

如果没有指定源图像，则位图在窗口上显示为缺省长方形元素。

透明色

位图的 InTouch 内部应用程序对象数据结构包含一个布尔值字段，用于指示是否应用透明色。另一个字段用于存储实际的透明色。

在 WindowMaker 中，创建新的位图对象时，位图最初显示为缺省透明色（黑色），且不可用。选择透明色工具后，会将永久透明色应用于位图对象。透明色工具不提供视觉提示来表明已将透明色指定给位图。

对于从 XML 文件导入的位图对象，位图的透明色节点是可选的条目，但如果该节点存在，则会启用透明色，并将其指定给位图对象。对于最终用户，没有机制可以让他们在 WindowMaker 中打开窗口，选择位图对象，然后禁用透明色。

位图对象元素

可以为位图对象指定以下元素。

元素	描述
标题	位图对象名。可选。
FillColor	位图对象的内部 RGB 颜色。包含一个颜色元素。缺省值为 rgb(0, 0, 0)。
SourceImage	位图文件的路径。缺省值为空。
透明	透明色的 RGB 颜色。缺省值为 rgb(0, 255, 0)。
笔	定义位图对象周围的线条。
动画链接	可选的动画链接列表。
维数	对象的位置和大小。元素为 top、left、width、height。top、left、width 和 height 值以像素为单位。生成的坐标必须在 -32000 到 +32000 之间。width 和 height 都不能为 0。必需。 如果为 Dimension 元素指定了无效的值或该元素缺少值，则不会创建对象。

示例：

```
<Bitmap>
    <Title>bitmap1</Title>
    <Pen>
      <PenColor><Name>Black</Name></PenColor>
      <PenWidth>1</PenWidth>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
      <PenStyle>Solid</PenStyle>
    </Pen>
    <Dimension>
      <Left>100</Left>
       <Top>50</Top>
      <Width>270</Width>
       <Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R>
       <G>128</G> <
       B>128</B>
    </FillColor>
    <Transparent>
      <R>0</R> <G>128</G> <B>255</B>
    </Transparent>
    <SourceImage>C:\MyPictures\hello.jpg</SourceImage>
    <AnimationLinks>
    </AnimationLinks>
  </Bitmap>
```

示例：

```
<Bitmap>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <SourceImage>C:\MyPictures\hello.jpg</SourceImage>
  </Bitmap>
```

SourceImage 节点为空的示例：

```
<Bitmap>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <FillColor>
      <R>128</R> <G>128</G> <B>128</B>
    </FillColor>
    <SourceImage></SourceImage>
  </Bitmap>
```

最简示例：

```
<Bitmap>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Bitmap>
```

按钮对象

可以为按钮对象指定以下元素。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
标题	按钮对象名。可选。
TextString	显示的标题。缺省字符串为“Text”。
TextInfo	定义标题的显示方式。
动画链接	可选的动画链接列表。不允许线条颜色、填充颜色、文本颜色、填充百分比、相对方向动画链接。
维数	对象的位置和大小。元素为 top、left、width、height。top、left、width 和 height 值以像素为单位。生成的坐标必须在 -32000 到 +32000 之间。width 和 height 都不能为 0。必需。 如果为 Dimension 元素指定了无效的值或该元素缺少值，则不会创建对象。

示例：

```
Button>
    <Title>button1</Title>
    <TextInfo>
      <Font>Arial</Font>
      <FontStyle>Regular</FontStyle>
      <FontSize>12</FontSize>
      <Underline>false</Underline>
      <Strikeout>false</Strikeout>
      <TextColor>
        <R>0</R>
        <G>0</G>
        <B>0</B>
      </TextColor>
      <TextJustify>Left</TextJustify>
    </TextInfo>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <TextString>Stop All Robots</TextString>
    <AnimationLinks>
    </AnimationLinks>
  </Button>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

最简示例：

```
<Button>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </Button>
```

SmartSymbol

必须在应用程序中定义 SmartSymbol 模板，然后才能导入使用 SmartSymbol 模板的窗口。如果 SmartSymbol 模板不存在，则导入会失败，并且不会创建窗口。

可以为一个窗口指定多个 SmartSymbol。每个 SmartSymbol 在单独的 <SmartSymbol> </SmartSymbol> 元素中声明。

可以为 SmartSymbol 指定以下元素。

元素	描述
符号名称	SmartSymbol 名称。必需。 如果该名称与应用程序中的 SmartSymbol 模板名称不一致，或者缺失，则不会创建对象。
维数	对象的位置和大小。元素为 top、left、width、height。top、left、width 和 height 值以像素为单位。生成的坐标必须在 -32000 到 +32000 之间。width 和 height 都不能为 0。必需。 如果为 Dimension 元素指定了无效的值或该元素缺少值，则不会创建对象。
TagReplace	实例标记替换。
StringReplace	实例字符串替换。

示例：

```
<SmartSymbol>
    <SymbolName>MyCoolSymbol</SymbolName>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <TagReplace>
      <Find>Tag001</Find>
      <Replace>Tag007</Replace>
    </TagReplace>
    <StringReplace>
      <Find>Find This Text</Find>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
      <Replace>Replace it with this text</Replace>
    </StringReplace>
  </SmartSymbol>
```

最简示例：

```
<SmartSymbol>
    <SymbolName>MyCoolSymbol</SymbolName>
    <Dimension>
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
  </SmartSymbol>
```

标记替换

可以替换 SmartSymbol 实例中的标记。只有应用程序中当前存在的 SmartSymbol 实例中的标记可以替换。

标记替换不区分大小写。整个标记名字符串必须匹配。要替换多个标记，可以将一个或多个 <TagReplace> 元素添加到 SmartSymbol 节点。

例如：

```
<TagReplace>
  <Find>Tag1</Find>
  <Replace>DTagA</Replace>
</TagReplace>
<TagReplace>
  <Find><![CDATA[Tag1]]></Find>
  <Replace><![CDATA[DTagA]]></Replace>
</TagReplace>
```

如果指定用于标记替换的标记名在应用程序中不存在，则不会创建 SmartSymbol。此外，也不会创建包含 SmartSymbol 的窗口。

替换标记的类型必须与原始标记的类型相同。

字符串替换

可以替换 SmartSymbol 实例中的字符串。字符串替换区分大小写。整个源字符串必须匹配。在单个 SmartSymbol 元素中，可以使用多个字符串替换元素。

```
<StringReplace>
    <FIND>
      <![CDATA[打开]]>
    </FIND>
    <REPLACE>
      <![CDATA[关闭]]>
    </REPLACE>
  </StringReplace>
  <StringReplace>
    <FIND>Open</FIND>
    <REPLACE>Off</REPLACE>
  </StringReplace>
```

SmartSymbol 示例

这是包含标记和字符串替换元素的 SmartSymbol 元素示例。

```
<SmartSymbol>
    <SymbolName>MyCoolSymbol</SymbolName>
    <Dimension>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
      <Left>100</Left><Top>50</Top>
      <Width>270</Width><Height>80</Height>
    </Dimension>
    <TagReplace>
      <Find>DTag1</Find>
      <Replace>DTagA</Replace>
    </TagReplace>
    <TagReplace>
      <Find>ATag001</Find>
      <Replace>ATag002</Replace>
    </TagReplace>
    <StringReplace>
      <Find><![CDATA[打开]]> </Find>
      <Replace><![CDATA[打开]]> </Replace>
    </StringReplace>
    <StringReplace>
      <FIND>
        <![CDATA[已关闭]]>
      </FIND>
      <REPLACE>
        <![CDATA[关闭]]>
      </REPLACE>
    </StringReplace>
 </SmartSymbol>
```

工业图形

必须在 Galaxy 中定义工业图形，然后才能导入使用工业图形引用的窗口。如果工业图形引用不存在，则导入会失败，并且不会创建窗口。

在符号引用中，可以指定实例中的或图形工具箱中的符号。但是，不能在符号引用中指定模板符号。

可以为一个窗口指定多个工业图形。每个工业图形都在单独的 <Industrial Graphic> </Industrial Graphic> 元素中声明。

可以为工业图形指定以下元素。

元素	描述
SymbolReference	用于在 Galaxy 中查找符号的引用。必需。 如果符号引用指定的工业图形不存在，或者符号引用字段缺失或为空，则不会创建对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
Dimension	对象的位置和大小。Dimension 子元素为 top、left、width 和 height。必需。 指定 width 和 height 是可选的。如果不指定 width 和 height，则会使用工业图形的原始宽度和高度。如果仅指定 width，则会使用纵横比来计算 height，反之亦然。例如，如果原始宽度为 200，原始高度为 100，而您将 width 指定为 100，则 height 将更改为 50。 如果为 Dimension 元素指定了无效的值或该元素不包含值，则不会创建对象。此外，也不会创建包含此符号的窗口。
Title	元素的文本名称。可选。 此名称应该在 XML 文件中唯一；否则将被忽略。
Flip	元素的翻转类型。可选。
Rotation	元素的旋转角度。可选。
StringReplace	一个或多个字符串替换节点。可选。
CustomPropertyOverride	一个或多个自定义属性覆盖。可选。
AnimationLinks	动画链接的列表。可选。 不允许指定线条颜色、填充颜色、文本颜色、填充百分比、相对方向、游标、工具提示、值显示、闪烁或用户输入。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
<ArchestrASymbol>
<Title>EmbedSym1</Title>
<Dimension>
      <Left>200</Left>
      <Top>200</Top>
      <Width>150</Width>
      <Height>150</Height>
   </Dimension>
   <Flip>None</Flip>
   <Rotation>0</Rotation>

   <SymbolReference>ButtonChromeMomentaryRed</SymbolReference>
   <AnimationLinks>
   </AnimationLinks>
   <StringReplace>
      <Find>LABEL</Find>
      <Replace>OFF</Replace>
   </StringReplace>
   </ArchestrASymbol>
```

最简示例：

```
<ArchestrASymbol>
<Dimension>
      <Left>200</Left>
      <Top>200</Top>
   </Dimension>
   <SymbolReference>ButtonChromeMomentaryRed</SymbolReference>
</ArchestrASymbol>
```

CustomPropertyOverride

只能覆盖那些已为工业图形定义的自定义属性。

要覆盖多个自定义属性，请将一个或多个 <CustomPropertyOverride> 元素添加到工业图形节点。

示例：

```
<ArchestrASymbol>
   <Title>EmbedSym1</Title>
   <Dimension>
   <Left>200</Left>
   <Top>200</Top>
   <Width>150</Width>
   <Height>150</Height>
   </Dimension>
   <Flip>None</Flip>
   <Rotation>0</Rotation>
   <SymbolReference>ButtonChromeMomentaryRed</SymbolReference>
   <AnimationLinks>
   </AnimationLinks>
   <CustomPropertyOverride> <CustomPropertyName>cp1</CustomPropertyName>
      <OverrideValue>DTagA</OverrideValue>
      <IsConstant>false</IsConstant>
   </CustomPropertyOverride>
</ArchestrASymbol>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在此示例中，cp1 是现有自定义属性的名称。将使用设置为 DTagA 的新值来覆盖自定义属性。IsConstant 是可选字段，用于指示是否应将值解释为常量。仅当自定义属性的类型为“字符串”、“时间”或“经过时间”时，IsConstant 标帜才适用。缺省条件下，IsConstant 标帜设置为 False。

备注：如果为 OverrideValue 指定的标记名在标记数据库中不存在，则不会在窗口上创建工业图形引用，并且该特定窗口的导入会失败。错误消息将记录在日志文件或 Logger 中。

工业图形字符串替换类型

可以替换工业图形实例中的现有字符串。字符串替换区分大小写。在单个工业图形节点中，可以使用多个字符串替换节点。

```
  <StringReplace>
    <FIND>
      <![CDATA[打开]]>
    </FIND>
    <REPLACE>
      <![CDATA[关闭]]>
    </REPLACE>
  </StringReplace>
<StringReplace>
    <FIND>Open</FIND>
    <REPLACE>Off</REPLACE>
  </StringReplace>
```

不支持的窗口对象

无法导入单元、符号、实时趋势和历史趋势对象。如果 XML 文件中包含不支持的对象的元素，则会将其忽略。

使用 XML 实用程序导入窗口

使用 XML 实用程序导入窗口时，您可以向 XML 脚本中添加关闭按钮脚本。

注意：批处理文件 (wm.bat) 中已指定了您要导入的窗口。

```
<CloseButton>True</CloseButton>
```

如果您的 XML 文件包含该脚本，那么您导入的窗口的标题栏上会带有关闭窗口按钮。

窗口动画链接

窗口对象动画链接在 <AnimationLinks> </AnimationLinks> 元素中声明。

一个窗口元素中可以有零个或更多个动画链接。

仅部分元素支持所有动画链接类型。工业图形支持以下动画链接：

- ObjSize_Height
- ObjSize_Width
- Location_Vert
- Location_Hori
- TPushB_Disc
- TPushB_Action
- TPushB_ShowWin

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- TPushB_HideWin
- Misc_Visib
- Misc_Disable

如需有关窗口动画链接的详细信息，请参阅 InTouch HMI 文档。

某些动画链接类型会阻止您创建其它动画链接类型。动画链接按 XML 文件中指定的顺序进行处理。

脚本/表达式/标记名需求矩阵

每个动画链接都有一个控制字段。控制字段可以是脚本、表达式或标记名。某些控制字段受允许的标记或表达式类型的限制。下表列出了每个动画链接所需的控制字段以及控制字段的任何限制。

动画链接类型	控制字段	控制字段限制
离散用户输入	标记	离散标记
模拟用户输入	标记	模拟标记
字符串用户输入	标记	字符串标记
离散线条颜色	表达式	离散表达式
模拟线条颜色	表达式	模拟表达式
离散报警线条颜色	标记	离散标记报警状态
模拟报警线条颜色	标记	模拟标记报警状态
离散填充颜色	表达式	离散表达式
模拟填充颜色	表达式	模拟表达式
离散报警填充颜色	标记	离散标记报警状态
模拟报警填充颜色	标记	模拟标记报警状态
离散文本颜色	表达式	离散表达式
模拟文本颜色	表达式	模拟表达式
离散报警文本颜色	标记	离散标记报警状态

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

动画链接类型	控制字段	控制字段限制
模拟报警文本颜色	标记	模拟标记报警状态
垂直游标	标记	有效模拟标记
水平游标	标记	有效模拟标记
对象大小高度	表达式	模拟值
对象大小宽度	表达式	模拟值
垂直位置	表达式	有效表达式
水平位置	表达式	有效表达式
垂直填充百分比	表达式	模拟值
水平填充百分比	表达式	模拟值
离散触动按钮	标记	离散值
动作触动按钮	“”脚本	有效脚本
显示窗口触动按钮	窗口名	窗口必须存在。
隐藏窗口触动按钮	窗口名	窗口必须存在。
可见性	表达式	离散值
闪烁	表达式	离散值
相对方向	表达式	模拟值
禁用	表达式	离散值
工具提示	表达式	字符串标记
离散值显示	表达式	离散表达式
模拟值显示	表达式	模拟表达式
字符串值显示	表达式	字符串表达式

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

离散用户输入

可以为离散用户输入动画链接指定以下元素：

元素	描述
标题	对象名。可选。
消息	给用户的消息文本。缺省值为无消息文本。使用可选架构时必需。
InputOnly	是否仅输入：{true, false}。缺省值为 false。
OnMessage	用户的开启消息文本。缺省值为 On。不能为空文本。
OffMessage	用户的关闭消息文本。缺省值为 Off。不能为空文本。
SetPrompt	用户的设置提示消息文本。缺省值为 On。不能为空文本。
ResetPrompt	用户的重置提示消息文本。缺省值为 Off。不能为空文本。
KeyAssignment	虚拟键元素。缺省值为无分配。空字符串表示无分配。
表达式	离散标记。必需。如果无效或缺失，则不会创建对象。

示例：

```
<UserInputDiscrete>
    <Title>UserInputDiscrete1</Title>
    <InputOnly>false</InputOnly>
    <KeyAssignment>
      <KeyCode>F1</KeyCode>
      <KeyFlags>Ctrl</KeyFlags>
    </KeyAssignment>
    <Message>Pump Valve State</Message>
    <Expression><![CDATA[dTag001]]></Expression>
    <OnMessage>On Message Text</OnMessage>
    <OffMessage>Off Message Text</OffMessage>
    <ResetPrompt>
      <![CDATA[复位提示]]>
    </ResetPrompt>
    <SetPrompt>Set Prompt Text</SetPrompt>
  </UserInputDiscrete>
```

最简示例：

```
<UserInputDiscrete>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
    <Message>Pump Valve State</Message>
    <Expression>dTag001</Expression>
  </UserInputDiscrete>
```

模拟用户输入

可以为模拟用户输入动画链接指定以下元素。

元素	描述
标题	对象名。可选。
消息	给用户的消息文本。使用可选架构时必需。缺省值为无消息文本。
InputOnly	是否仅输入：true, false。缺省值为 false。
MinAnalogValue	允许的最小浮点值。使用可选架构时必需。缺省值为 0.0
MaxAnalogValue	允许的最大浮点值。使用可选架构时必需。缺省值为 100.0。必须大于指定给 MINANALOGVALUE 元素的值。
KeyPadEnabled	指定数字小键盘是否可见。可能的值包括 True 或 False。缺省值为 false。
KeyAssignment	虚拟键元素、空字符串或空缺。缺省值为无分配。空字符串表示无分配。
表达式	模拟类型标记名。必需。如果无效或缺失，则不会创建对象。

示例：

```
<<UserInputAnalog>
    <Title>UserInputAnalog1</Title>
    <InputOnly>false</InputOnly>
    <KeyAssignment>
      <KeyCode>F1</KeyCode>
      <KeyFlags>Ctrl</KeyFlags>
    </KeyAssignment>
    <Message>Flush Pump Speed</Message>
    <Expression>aTag001</Expression>
    <KeyPadEnabled>false</KeyPadEnabled>
    <MinAnalogValue>0.0</MinAnalogValue>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
    <MaxAnalogValue> 100.0</MaxAnalogValue>
  </UserInputAnalog>
```

最简示例：

```
<UserInputAnalog>
    <Message>Pump Valve State</Message>
    <Expression><![CDATA[aTag001]]></Expression>
    <MinAnalogValue>0.0</MinAnalogValue>
    <MaxAnalogValue>100.0</MaxAnalogValue>
  </UserInputAnalog>
```

字符串用户输入

您可以使用 EchoEnabled 或 EchoMode，但不能同时使用两者。EchoMode 允许指定口令模式，而 EchoEnabled 只允许启用或禁用状态。

回显字符元素用于控制在运行时如何显示用户输入。可能的值包括 Yes、No 和 Password。

- 如果将该元素设置为 Yes，则在运行时，字符串字符会显示在输入编辑框中。输入为启用状态。口令字符和加密为禁用状态。
- 如果将该元素设置为 No，则在运行时，输入字符不会显示在输入编辑框中。输入为启用状态。口令字符和加密为禁用状态。
- 如果将该元素设置为 Password，则在运行时会显示口令字符，而不是用户输入的口令。口令字符是可选的，但如果选择指定，则不能为空。缺省口令字符是星号。加密是可选的，在缺省条件下是关闭的。“仅输入”是必需的，会被强制开启。

如果元素不符合这些准则，将使用缺省选项。

可以为字符串用户输入动画链接指定以下元素。

元素	描述
标题	对象名。可选。
消息	给用户的消息文本。使用可选架构时必需。缺省值为无消息文本。
InputOnly	是否仅输入：{true, false}。缺省值为 false。
EchoCharacters	回显字符 : no, yes, password。使用可选架构时必需。缺省值为 yes。
KeyPadEnabled	数字小键盘是否可见：true, false。缺省值为 false。
PasswordCharacter	口令字符。缺省值为“*”。
EncryptEnabled	启用加密：yes, no。缺省值为 no。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
KeyAssignment	虚拟键元素、空字符串或空缺。缺省值为无分配。空字符串表示无分配。
表达式	消息类型标记。必需。如果无效或缺失，则不会创建对象。

示例：

```
<UserInputString>
    <Title>UserInputString1</Title>
    <InputOnly>false</InputOnly>
    <KeyAssignment>
      <KeyCode>F1</KeyCode>
      <KeyFlags>Ctrl</KeyFlags>
    </KeyAssignment>
    <Message>Select Pump</Message>
    <Expression>mTag001</Expression>
    <EchoCharacters>true</EchoCharacters>
    <EncryptEnabled>false</EncryptEnabled>
    <PasswordCharacter>*</PasswordCharacter>
  </UserInputString>
```

最简示例：

```
<UserInputString>
    <Message>Select Pump</Message>
    <Expression>mTag001</Expression>
    <EchoCharacters>true</EchoCharacters>
  </UserInputString>
```

离散线条颜色

可以为离散线条颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
Oncolor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Offcolor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	离散标记或表达式。必需。如果 Expression 元素值缺失或无效，则不会创建对象。

示例：

```
<LineColorDiscrete>
    <Title>LineColorDiscrete1</Title>
    <Expression>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
        <![CDATA[dTag001]]>
    </Expression>
    <OnColor><Name>Green</Name></OnColor>
    <OffColor><Name>Red</Name></OffColor>
  </LineColorDiscrete>
```

最简示例：

```
<LineColorDiscrete>
    <Expression>dTag001</Expression>
  </LineColorDiscrete>
```

模拟线条颜色

所需的断点值必须为递增的值。如果不是这样，将记录一条警告，指出违规的值是 1 加上前一个值。

如果所需的值缺失或无效，则不会创建包含动画链接的对象。

可以为模拟线条颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
Color1	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color2	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color3	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color4	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color5	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color6	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color7	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color8	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color9	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color10	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值	常量模拟值。必须包含九个 Value 元素。必需。
表达式	模拟标记或表达式。必需。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例：

```
<LineColorAnalog>
    <Title>LineColorAnalog1</Title>
    <Expression>aTag001</Expression>
    <Colors>
      <Color1><Name>White</Name></Color1>
      <Color2><Name>Red</Name></Color2>
      <Color3><Name>Orange</Name></Color3>
      <Color4><Name>Yellow</Name></Color4>
      <Color5><Name>Green</Name></Color5>
      <Color6><Name>Blue</Name></Color6>
      <Color7><Name>Cyan</Name></Color7>
      <Color8><Name>Magenta</Name></Color8>
      <Color9><Name>Violet</Name></Color9>
      <Color10><Name>Black</Name></Color10>
    </Colors>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </LineColorAnalog>
```

最简示例：

```
<LineColorAnalog>
    <Expression>aTag001</Expression>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </LineColorAnalog>
```

离散报警线条颜色

可以为离散报警线条颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
AlarmColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	离散标记。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<LineColorDiscreteAlarm>
    <Title>LineColorDiscreteAlarm1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <NormalColor><Name>Black</Name></NormalColor>
    <AlarmColor><Name>Red</Name></AlarmColor>
  </LineColorDiscreteAlarm>
```

最简示例：

```
<LineColorDiscreteAlarm>
    <Expression>dTag001</Expression>
  </LineColorDiscreteAlarm>
```

模拟报警线条颜色

可以为模拟报警线条颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
值：LOLOCOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：LOCOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：NORMALCOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：HICOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：HIHICOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：NORMALCOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：MINORCOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：MAJORCOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
ROC：NORMALCOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
ROC：ROCCOLOR	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	模拟标记。必需。如果 Expression 无效或缺失，则不会创建对象。

值报警的示例：

```
<LineColorAnalogAlarm>
    <Title>LineColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <Value>
      <LoLoColor><Name>Red</Name><LoLoColor>
      <LoColor><Name>DarkRed</Name><LoColor>
      <NormalColor><Name>Black</Name></NormalColor>
      <HiColor><Name>DarkGreen</Name></HiColor>
      <HiHiColor><Name>Green</Name></HiHiColor>
    </Value>
  </LineColorAnalogAlarm>
```

偏差报警的示例：

```
<LineColorAnalogAlarm>
    <Title>LineColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
   <Deviation>
      <NormalColor><Name>Black</Name></NormalColor>
      <MinorColor><Name>Green</Name></MinorColor>
      <MajorColor><Name>Red</Name></MajorColor>
    </Deviation>
  </LineColorAnalogAlarm>
```

ROC 报警的示例：

```
<LineColorAnalogAlarm>
    <Title>LineColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <ROC>
      <NormalColor><Name>Black</Name></NormalColor>
      <ROCColor><Name>Red</Name></ROCColor>
    </ROC>
  </LineColorAnalogAlarm>
```

离散填充颜色

可以为离散填充颜色动画链接指定以下元素。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
标题	对象名。可选。
OnColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
OffColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	离散标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<FillColorDiscrete>
    <Title>FillColorDiscrete1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <OnColor><Name>Green</Name></OnColor>
    <OffColor><Name>Red</Name></OffColor>
  </FillColorDiscrete>
```

最简示例：

```
<FillColorDiscrete>
    <Expression>dTag001</Expression>
  </FillColorDiscrete>
```

模拟填充颜色

如果所需的值缺失或无效，则不会创建包含动画链接的对象。

可以为模拟填充颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
Color1	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color2	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color3	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color4	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color5	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color6	包含一个颜色元素。缺省值为 rgb(0,0,0)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
Color7	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color8	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color9	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color10	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值	常量模拟值。必须包含九个 Value 元素。必需。
表达式	模拟标记或表达式。必需。

示例：

```
<FillColorAnalog>
    <Title>FillColorAnalog1</Title>
    <Expression>aTag001</Expression>
    <Colors>
      <Color1><Name>White</Name></Color1>
      <Color2><Name>Red</Name></Color2>
      <Color3><Name>Orange</Name></Color3>
      <Color4><Name>Yellow</Name></Color4>
      <Color5><Name>Green</Name></Color5>
      <Color6><Name>Blue</Name></Color6>
      <Color7><Name>Cyan</Name></Color7>
      <Color8><Name>Magenta</Name></Color8>
      <Color9><Name>Violet</Name></Color9>
      <Color10><Name>Black</Name></Color10>
    </Colors>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </FillColorAnalog>
```

最简示例：

```
<FillColorAnalog>
    <Expression>aTag001</Expression>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </FillColorAnalog>
```

离散报警填充颜色

可以为离散报警填充颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
AlarmColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	离散标记。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<FillColorDiscreteAlarm>
    <Title>FillColorDiscreteAlarm1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <NormalColor><Name>Black</Name></NormalColor>
    <AlarmColor><Name>Red</Name></AlarmColor>
  </FillColorDiscreteAlarm>
```

最简示例：

```
<FillColorDiscreteAlarm>
    <Expression>dTag001</Expression>
  </FillColorDiscreteAlarm>
```

模拟报警填充颜色

可以为模拟报警填充颜色动画链接指定以下元素。如果所需的值缺失或无效，则不会创建包含动画链接的对象。

元素	描述
标题	对象名。可选。
值：LoLoColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
值：LoColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：HiColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：HiHiColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：MinorColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：MajorColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
ROC：NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
ROC：ROCColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	模拟标记。必需。如果 Expression 无效或缺失，则不会创建对象。

值报警的示例：

```
<FillColorAnalogAlarm>
    <Title>FillColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <Value>
      <LoLoColor><Name>Red</Name><LoLoColor>
      <LoColor><Name>DarkRed</Name><LoColor>
      <NormalColor><Name>Black</Name></NormalColor>
      <HiColor><Name>DarkGreen</Name></HiColor>
      <HiHiColor><Name>Green</Name></HiHiColor>
    </Value>
  </FillColorAnalogAlarm>
```

偏差报警的示例：

```
<FillColorAnalogAlarm>
    <Title>FillColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
    <Deviation>
      <NormalColor><Name>Black</Name></NormalColor>
      <MinorColor><Name>Green</Name></MinorColor>
      <MajorColor><Name>Red</Name></MajorColor>
    </Deviation>
  </FillColorAnalogAlarm>
```

ROC 报警的示例：

```
<FillColorAnalogAlarm>
    <Title>FillColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <ROC>
      <NormalColor><Name>Black</Name></NormalColor>
      <ROCColor><Name>Red</Name></ROCColor>
    </ROC>
  </FillColorAnalogAlarm>
```

离散文本颜色

可以为离散文本颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
OnColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
OffColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	字符串标记或表达式。必需。

示例：

```
<TextColorDiscrete>
    <Title>TextColorDiscrete1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <OnColor><Name>Green</Name></OnColor>
    <OffColor><Name>Red</Name></OffColor>
  </TextColorDiscrete>
```

最简示例：

```
<TextColorDiscrete>
    <Expression>dTag001</Expression>
  </TextColorDiscrete>
```

模拟文本颜色

可以为模拟文本颜色动画链接指定以下元素。如果所需的值缺失或无效，则不会创建包含动画链接的对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
标题	对象名。可选。
Color1	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color2	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color3	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color4	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color5	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color6	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color7	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color8	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color9	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Color10	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值	常量模拟值。必须包含九个 Value 元素。必需。
表达式	模拟标记或表达式。必需。

示例：

```
<TextColorAnalog>
    <Title>TextColorAnalog1</Title>
    <Expression>aTag001</Expression>
    <Colors>
      <Color1><Name>White</Name></Color1>
      <Color2><Name>Red</Name></Color2>
      <Color3><Name>Orange</Name></Color3>
      <Color4><Name>Yellow</Name></Color4>
      <Color5><Name>Green</Name></Color5>
      <Color6><Name>Blue</Name></Color6>
      <Color7><Name>Cyan</Name></Color7>
      <Color8><Name>Magenta</Name></Color8>
      <Color9><Name>Violet</Name></Color9>
      <Color10><Name>Black</Name></Color10>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
    </Colors>
    <Values>
      <Value>10.0</Value> <Value>20.0</Value>
      <Value>30.0</Value> <Value>40.0</Value>
      <Value>50.0</Value> <Value>60.0</Value>
      <Value>70.0</Value> <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </TextColorAnalog>
```

最简示例：

```
<TextColorAnalog>
    <Expression>aTag001</Expression>
    <Values>
      <Value>10.0</Value>
      <Value>20.0</Value>
      <Value>30.0</Value>
      <Value>40.0</Value>
      <Value>50.0</Value>
      <Value>60.0</Value>
      <Value>70.0</Value>
      <Value>80.0</Value>
      <Value>90.0</Value>
    </Values>
  </TextColorAnalog>
```

离散报警文本颜色

可以为离散报警文本颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
AlarmColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	离散标记。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<TextColorDiscreteAlarm>
    <Title>TextColorDiscreteAlarm1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <NormalColor><Name>Black</Name></NormalColor>
    <AlarmColor><Name>Red</Name></AlarmColor>
  </TextColorDiscreteAlarm>
```

最简示例：

```
<TextColorDiscreteAlarm>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
    <Expression>dTag001</Expression>
  </TextColorDiscreteAlarm>
```

模拟报警文本颜色

可以为模拟报警文本颜色动画链接指定以下元素。

元素	描述
标题	对象名。可选。
值：LoLoColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：LoColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：HiColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
值：HiHiColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：MinorColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
偏差：MajorColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
ROC：NormalColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
ROC：ROCColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
表达式	模拟标记。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<TextColorAnalogAlarm>
    <Title>TextColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <Value>
      <LoLoColor><Name>Red</Name><LoLoColor>
      <LoColor><Name>DarkRed</Name><LoColor>
      <NormalColor><Name>Black</Name></NormalColor>
      <HiColor><Name>DarkGreen</Name></HiColor>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
      <HiHiColor><Name>Green</Name></HiHiColor>
    </Value>
  </TextColorAnalogAlarm>
```

偏差类型报警的示例：

```
<TextColorAnalogAlarm>
    <Title>TextColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <Deviation>
      <NormalColor><Name>Black</Name></NormalColor>
      <MinorColor><Name>Green</Name></MinorColor>
      <MajorColor><Name>Red</Name></MajorColor>
    </Deviation>
  </TextColorAnalogAlarm>
```

ROC 类型报警的示例：

```
<TextColorAnalogAlarm>
    <Title>TextColorAnalogAlarm1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <ROC>
      <NormalColor><Name>Black</Name></NormalColor>
      <ROCColor><Name>Red</Name></ROCColor>
    </ROC>
  </TextColorAnalogAlarm>
```

垂直游标

可以为垂直游标动画链接指定以下元素。

元素	描述
标题	对象名。可选。
ReferenceLocation	垂直参考：top, middle, bottom。缺省值为 bottom。
TopValue	顶部值。不能与 BottomValue 相同。默认为 0。
BottomValue	底部值。缺省值为 100。
UpwardMovement	向上移动。必须在 0 到 32767 的范围内。如果值超出范围，则会对其进行限定，并记录一条错误消息。缺省值为 50。
DownwardMovement	向下移动。必须在 0 到 32767 的范围内。如果值超出范围，则会对其进行限定，并记录一条错误消息。缺省值为 50。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
表达式	模拟标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<SliderVertical>
    <Title>SliderVertical1</Title>
    <ReferenceLocation>Bottom</ReferenceLocation>
    <TopValue>10.0</TopValue>
    <BottomValue> 110.0</BottomValue>
    <UpwardMovement> 20.0</UpwardMovement>
    <DownwardMovement> 120.0</DownwardMovement>
    <Expression> <![CDATA[aTag001]]> </Expression>
  </SliderVertical>
```

最简示例：

```
<SliderVertical>
    <Expression>aTag001</Expression>
  </SliderVertical>
```

水平游标

可以为水平游标动画链接指定以下元素。

元素	描述
标题	对象名。可选。
ReferenceLocation	参考位置：left, center, right。缺省值为 left。
LeftValue	游标的左侧值。默认为 0。
RightValue	游标的右侧值。缺省值为 100。
LeftMovement	向左水平移动。必须在 0 到 32767 的范围内。如果值超出范围，则会对其进行限定，并记录一条错误消息。缺省值为 50。
RightMovement	向右水平移动。必须在 0 到 32767 的范围内。如果值超出范围，则会对其进行限定，并记录一条错误消息。缺省值为 50。
表达式	模拟标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<SliderHorizontal>
    <Title>SliderHorizontal1</Title>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
    <ReferenceLocation>Left</ReferenceLocation>
    <LeftValue>10.0</LeftValue>
    <RightValue>120.0</RightValue>
    <LeftMovement>20.0</LeftMovement>
    <RightMovement>150.0</RightMovement>
    <Expression>
      <![CDATA[aTag001]]>
    </Expression>
  </SliderHorizontal>
```

最简示例：

```
<SliderHorizontal>
    <Expression><![CDATA[aTag001]]></Expression>
  </SliderHorizontal>
```

对象高度

可以为对象高度动画链接指定以下元素。对象高度动画链接不能与相对方向动画链接一起使用。

元素	描述
标题	对象名。可选。
SizeAnchor	定义对象上定位点的位置。值可以为 bottom, middle, top。缺省值为 bottom。
SizeMin	最小高度值。默认为 0。
SizeMax	最大高度值。缺省值为 100。必须大于指定给 SizeMin 元素的值。
MinPercent	最小高度百分比。缺省值为 0。范围是从 0 到 100。
MaxPercent	最大高度百分比。缺省值为 100。必须大于最小百分比。范围是从 0 到 100。
表达式	模拟标记名或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<ObjectSizeHeight>
    <Title>ObjectSizeHeight1</Title>
    <Expression> <![CDATA[aTag001]]> </Expression>
    <SizeMin>0.0</SizeMin>
    <SizeMax>100.0</SizeMax>
    <MinPercent>0.0</MinPercent>
    <MaxPercent>100.0</MaxPercent>
    <SizeAnchor>Top</SizeAnchor>
  </ObjectSizeHeight>
```

最简示例：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
<ObjectSizeHeight>
    <Expression>aTag001</Expression>
  </ObjectSizeHeight>
```

对象宽度

可以为对象大小宽度动画链接指定以下元素。对象宽度动画链接不能与相对方向动画链接一起使用。

元素	描述
标题	对象名。可选。
SizeAnchor	定义对象上定位点的位置。值可以是：{Left、center、right}。缺省值为 left。
SizeMin	最小宽度值。默认为 0。
SizeMax	最大宽度值。缺省值为 100。必须大于 SizeMin。
MinPercent	最小宽度百分比。缺省值为 0。范围是从 0 到 100。
MaxPercent	最大宽度百分比。缺省值为 100。必须大于 MinPercent。范围是从 0 到 100。
表达式	模拟标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
< ObjectSizeWidth>
    <Title>ObjectSizeWidth1</Title>
    <Expression> <![CDATA[aTag001]]> </Expression>
    <SizeMin>0.0</SizeMin>
    <SizeMax>100.0</SizeMax>
    <MinPercent>0.0</MinPercent>
    <MaxPercent>100.0</MaxPercent>
    <SizeAnchor>Center</SizeAnchor>
  </ObjectSizeWidth>
```

最简示例：

```
<ObjectSizeWidth>
    <Expression>aTag001</Expression>
  </ObjectSizeWidth>
```

垂直位置

可以为垂直位置动画链接指定以下元素。对象垂直位置动画链接不能与相对方向动画链接一起使用。

元素	描述
标题	对象名。可选。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
MinValue	顶部值。缺省值为 0。不能与底部值相同。
MaxValue	底部值。缺省值为 100。不能与顶部值相同。
DecreaseMovement	向上垂直移动。缺省值为 0。不能小于 0。
IncreaseMovement	向下垂直移动。缺省值为 100。不能大于 100。
表达式	模拟标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<LocationVertical>
    <Title>LocationVertical1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <MinValue>0.0</MinValue>
    <MaxValue>100.0</MaxValue>
    <DecreaseMovement>0</DecreaseMovement>
    <IncreaseMovement>100</IncreaseMovement>
  </LocationVertical>
```

最简示例：

```
<LocationVertical>
    <Expression>aTag001</Expression>
  </LocationVertical>
```

水平位置

可以为水平位置动画链接指定以下元素。对象水平位置动画链接不能与相对方向动画链接一起使用。

元素	描述
标题	对象名。可选。
MinValue	左侧值。缺省值为 0。不能与右侧值相同。
MaxValue	右侧值。缺省值为 100。不能与左侧值相同。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
DecreaseMovement	向左水平移动。缺省值为 0。不能小于 0。
IncreaseMovement	向右水平移动。缺省值为 100。不得大于 100。
表达式	模拟标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<LocationHorizontal>
    <Title>LocationHorizontal1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
    <MinValue>0.0</MinValue>
    <MaxValue>100.0</MaxValue>
    <DecreaseMovement>0</DecreaseMovement>
    <IncreaseMovement>100</IncreaseMovement>
  </LocationHorizontal>
```

最简示例：

```
<LocationHorizontal>
    <Expression>aTag001</Expression>
  </LocationHorizontal>
```

垂直填充百分比

可以为垂直填充百分比动画链接指定以下元素。垂直填充百分比动画链接不能与相对方向动画链接一起使用。

元素	描述
标题	对象名。可选。
FillDirection	定义运动的方向。可能的值有：{Up、down}。缺省值为 up。必需。
FillColor	纯色填充颜色元素。缺省值为 rgb(0, 0, 0)。
FillMin	定义最小值。默认为 0。
FillMax	定义最大值。缺省值为 100。必须大于最小填充。
FillMinPercent	将最小值定义为百分比。缺省值为 0。范围是从 0 到 100。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
FillMaxPercent	将最大值定义为百分比。缺省值为 100。范围是从 0 到 100。必须大于最小百分比。
表达式	模拟标记或表达式。必需。

示例：

```
<PercentFillVertical>
    <Title>PercentFillVertical1</Title>
    <Expression><![CDATA[aTag001]]></Expression>
    <FillMin>0.0</FillMin>
    <FillMax>100.0</FillMax>
    <FillMinPercent>0</FillMinPercent>
    <FillMaxPercent>100</FillMaxPercent>
    <FillColor><Name>Purple</Name></FillColor>
    <FillDirection>Up</FillDirection>
   </PercentFillVertical>
```

最简示例：

```
<PercentFillVertical>
    <Expression>aTag001</Expression>
    <FillDirection>Up</FillDirection>
   </PercentFillVertical>
```

水平填充百分比

可以为水平填充百分比动画链接指定以下元素。水平填充百分比动画链接不能与相对方向动画链接一起使用。

元素	描述
标题	对象名。可选。
FillDirection	left, right。缺省值为 right。
FillColor	包含一个颜色元素。缺省值为 rgb(0, 0, 0)。
FillMin	最小填充值。默认为 0。
FillMax	最大填充值。缺省值为 100。必须大于最小填充。
FillMinPercent	填充对象的最小百分比。缺省值为 0。范围是从 0 到 100。
FillMaxPercent	填充对象的最大百分比。缺省值为 100。范围是从 0 到 100。必须大于最小填充百分比。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
表达式	模拟标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<PercentFillHorizontal>
    <Title>PercentFillHorizontal1</Title>
    <Expression><![CDATA[aTag001]]></Expression>
    <FillMin>0.0</FillMin>
    <FillMax>100.0</FillMax>
    <FillMinPercent>0</FillMinPercent>
    <FillMaxPercent>100</FillMaxPercent>
    <FillColor><Name>Purple</Name></FillColor>
    <FillDirection>Right</FillDirection>
   </PercentFillHorizontal>
```

最简示例：

```
<PercentFillHorizontal>
    <Expression>aTag001</Expression>
    <FillDirection>Left</FillDirection>
   </PercentFillHorizontal>
```

离散按钮

可以为离散按钮动画链接指定以下元素。

元素	描述
标题	对象名。可选。
ButtonType	离散按钮的类型：{direct、reverse、toggle、reset、set}。必需。如果无效或缺失，则不会创建对象。
表达式	离散类型标记或表达式。必需。如果 Expression 元素无效或缺失，则不会创建对象。
KeyAssignment	虚拟键元素、空字符串或空缺。缺省值为无分配。空字符串表示无分配。

示例：

```
<ButtonDiscreteValue>
    <Title>ButtonDiscreteValue1</Title>
    <ButtonType>Reverse</ButtonType>
    <KeyAssignment>
      <KeyCode>F2</KeyCode>
      <KeyFlags>Shift</KeyFlags>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
    </KeyAssignment>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
  </ButtonDiscreteValue>
```

最简示例：

```
<ButtonDiscreteValue>
    <ButtonType>Reverse</ButtonType>
    <Expression><![CDATA[dTag001]]></Expression>
  </ButtonDiscreteValue>
```

显示窗口按钮

如果在生成链接时不存在命名的窗口，则会记录一条警告。所生成的链接将不带有任何窗口操作。必须至少有一个命名的窗口，否则不会导入 ShowWindow 动画链接。

可以为显示窗口按钮动画链接指定以下元素。

元素	描述
标题	对象名。可选。
WindowName	要显示的窗口的名称。必须至少指定一个窗口名，否则不会将此动画链接添加到对象。

示例：

```
<ButtonShowWindow>
    <Title>ButtonShowWindow1</Title>
    <WindowName>
      <![CDATA[Window006]]>
    </WindowName>
    <WindowName>Window007</WindowName>
    <WindowName><![CDATA[SliderWindow]]></WindowName>
  </ButtonShowWindow>
```

隐藏窗口按钮

如果在生成链接时不存在命名的窗口，则会记录一条警告。所生成的链接将不带有任何窗口操作。必须存在有一个命名的窗口，否则不会导入 HideWindow 动画链接。

可以为隐藏窗口按钮动画链接指定以下元素。

元素	描述
标题	对象名。可选。
WindowName	要隐藏的窗口的名称。必须至少指定一个窗口名，否则不会将此动画链接添加到对象。

示例：

```
<ButtonHideWindow>
    <Title>ButtonHideWindow1</Title>
    <WindowName>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
      <![CDATA[Window001]]>
    </WindowName>
    <WindowName>Window002</WindowName>
    <WindowName> <![CDATA[SliderWindow]]> </WindowName>
  </ButtonHideWindow>
```

可见性

可以为可见性动画链接指定以下元素。

元素	描述
Title	对象名。可选。
State	定义对象是否可见。可能的值为：{on, off}。缺省值为 On。使用可选架构时必需。
Expression	离散标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<Visibility>
<Title>Visibility1</Title>
<Expression>dTag001</Expression>
<State>On</State>
</Visibility>
```

最简示例：

```
<Visibility>
<Expression>dTag001</Expression>
<State>On</State>
</Visibility>
```

闪烁

可以为闪烁动画链接指定以下元素。当表达式为 true 时，动画链接会闪烁。

元素	描述
Title	对象名。可选。
BlinkAttribute	{Invisible, Visible}。缺省值为 Visible。
BlinkSpeed	定义对象闪烁的速率。可能的值为：{Slow, Medium, Fast}。缺省值为 Medium。
TextColor	定义闪烁文本的颜色。包含一个颜色元素。缺省值为 rgb(0,0,0)。
LineColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
FillColor	包含一个颜色元素。缺省值为 rgb(0,0,0)。
Expression	离散标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<Blink>
<Title>Blink1</Title>
<Expression>dTag001</Expression>
<TextColor><R>0</R><G>0</G><B>0</B></TextColor>
<LineColor><R>0</R><G>0</G><B>0</B></LineColor>
<FillColor><R>0</R><G>255</G><B>0</B></FillColor>
<BlinkAttribute>Invisible</BlinkAttribute>
<BlinkSpeed>Slow</BlinkSpeed>
</Blink>
```

最简示例：

```
<Blink>
<Expression>dTag001</Expression>
<BlinkAttribute>Visible</BlinkAttribute>
<BlinkSpeed>Fast</BlinkSpeed>
</Blink>
```

相对方向

可以为相对方向动画链接指定以下元素。相对方向动画链接不能与游标、大小、位置或填充百分比动画链接一起使用。

元素	描述
Title	对象名。可选。
X	从对象中心点开始的水平偏移量。可选。缺省值为 0。
Y	从对象中心点开始的垂直偏移量。可选。缺省值为 0。
CWMax	最大顺时针旋转值。缺省值为 100。
CWRotation	顺时针旋转。缺省值为 360。必须在 0 到 360 的范围内。CWROTATION+CCWROTATION 不能超过 360。
CCWMax	最大逆时针旋转值。缺省值为 0。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
CCWRotation	逆时针旋转。缺省值为 0。必须在 0 到 360 的范围内。CWRotation+CCWRotation 不能超过 360。
Expression	模拟标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<Orientation>
<Title>Orientation1</Title>
<Expression> </Expression>
<X>0</X> <Y>0</Y>
<CWMax>100.0</CWMax>
<CWRotation>360.0</CWRotation>
<CCWMax>0.0</CCWMax>
<CCWRotation>0.0</CCWRotation>
</Orientation>
```

最简示例：

```
<Orientation>
<Expression>aTag001</Expression>
</Orientation>
```

禁用

可以为禁用动画链接指定以下元素。

元素	描述
Title	对象名。可选。
State	{on, off}。缺省值为 On。使用可选架构时必需。
Expression	离散标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<Disable>
<Title>Disable1</Title>
<Expression>dTag001</Expression>
<State>On</State>
</Disable>
```

最简示例：

```
<Disable>
<Expression>dTag001</Expression>
<State>On</State>
</Disable>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

静态工具提示

可以为静态工具提示动画链接指定以下元素。

元素	描述
标题	对象名。可选。
消息	静态文本消息。必需。如果无效或缺失，则不会创建对象。

示例：

```
<TooltipStatic>
    <Title>TooltipStatic1</Title>
    <Message>
      <![CDATA[ Click here to win a million dollars!
      ]]>
    </Message>
  </TooltipStatic>
```

动态工具提示

可以为动态工具提示动画链接指定以下元素。

元素	描述
标题	对象名。可选。
表达式	Expression 元素。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<TooltipTag>
    <Title>TooltipTag1</Title>
    <Expression>
      <![CDATA[sTag001]]>
    </Expression>
  </TooltipTag>
```

离散值显示

可以为离散值动画链接指定以下元素。

元素	描述
标题	对象名。可选。
OnMessage	动画链接值为 True 时要显示的字符串。缺省文本为 On。
OffMessage	动画链接值为 False 时要显示的字符串。缺省文本为 Off。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
表达式	离散标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<ValueDisplayDiscrete>
    <Title>ValueDisplayDiscrete1</Title>
    <Expression>
        <![CDATA[dTag001]]>
    </Expression>
    <OnMessage>
      <![CDATA[Pump is On]]>
    </OnMessage>
    <OffMessage>
      <![CDATA[Pump is Off]]>
    </OffMessage>
  </ValueDisplayDiscrete>
```

最简示例：

```
<ValueDisplayDiscrete>
    <Expression>dTag001</Expression>
  </ValueDisplayDiscrete>
```

模拟值显示

可以为模拟值动画链接指定以下元素。

元素	描述
标题	对象名。可选。
表达式	模拟标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<ValueDisplayAnalog>
    <Title>ValueDisplayAnalog1</Title>
    <Expression>
        <![CDATA[aTag001]]>
    </Expression>
  </ValueDisplayAnalog>
```

字符串值显示

可以为字符串值动画链接指定以下元素。

元素	描述
标题	对象名。可选。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
表达式	消息标记或表达式。必需。如果 Expression 无效或缺失，则不会创建对象。

示例：

```
<ValueDisplayString>
    <Title>ValueDisplayString1</Title>
    <Expression>
        <![CDATA[sTag001]]>
    </Expression>
  </ValueDisplayString>
```

按钮动作脚本

按钮动作脚本元素在 <ButtonActionScripts> </ButtonActionScripts> 元素中声明，后者在动画链接元素中声明为 0。

示例：

```
<AnimationLinks>
  <ButtonActionScripts>
    <OnLeftDown>
        <![CDATA[
          脚本文本的第一行
          脚本文本的第二行
          脚本文本的第 N 行
        ]]>
    </OnLeftDown>
  </ButtonActionScripts>
</AnimationLinks>
```

鼠标左键按下时/按下时

在同一对象中，不要同时使用 OnLeftDown 和 OnKeyDown 动画链接。

示例：

```
<OnLeftDown>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</OnLeftDown>
```

鼠标左键按下期间/按下期间

可以为 WhileLeftDown 或 WhileKeyDown 动画链接指定以下元素。

元素	描述
Text	脚本文本。必需。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

元素	描述
Frequency	脚本执行频率，以毫秒为单位。范围是从 1 到 360000。使用可选架构时必需。缺省值为 1000。

在同一对象中，不要同时使用 WhileLeftDown 和 WhileKeyDown 动画链接。

示例：

```
<WhileLeftDown>
<Text>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</Text>
<Frequency>1000</Frequency>
</WhileLeftDown>
```

鼠标左键放开时/放开时

可以将脚本文本放置在 OnLeftUp 元素中。在同一对象中，不要同时使用 OnLeftUp 和 OnKeyUp 动画链接。

示例：

```
<OnLeftUp>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</OnLeftUp>
```

双击鼠标左键时

可以将脚本文本放置在 OnLeftDoubleClick 动画链接中。

示例：

```
<OnLeftDoubleClick>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</OnLeftDoubleClick>
```

鼠标右键按下时/右键按下时

可以将脚本文本放置在 OnRightDown 元素中。在同一对象中，不要同时使用 OnRightDown 和 OnRightKeyDown 动画链接。

示例：

```
<OnRightDown>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
脚本文本的第 N 行
]]>
</OnRightDown>
```

鼠标右键按下期间/右键按下期间

可以为 WhileRightKeyDown 或 WhileRightDown 动画链接指定以下元素。

元素	描述
Text	脚本文本。必需
Frequency	脚本执行频率，以毫秒为单位。1 到 360000 之间的一个数。缺省值为 1000。

在同一对象中，不要同时使用 WhileRightDown 和 WhileRightKeyDown 动画链接。

示例：

```
<WhileRightDown>
<Text>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</Text>
<Frequency>1000</Frequency>
</WhileRightDown>
```

鼠标右键按下时/右键按下时

可以将脚本文本放置在 OnRightUp 和 OnRightKeyUp 元素中。在同一对象中，不要同时使用 OnRightUp 和 OnRightKeyUp 动画链接。

示例：

```
<OnRightUp>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</OnRightUp>
```

双击鼠标右键时

可以将脚本文本放置在 OnRightDoubleClick 元素中。

示例：

```
<OnRightDoubleClick>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</OnRightDoubleClick>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

鼠标中键按下时/中键按下时

可以将脚本文本放置在 OnMiddleKeyDown 元素中。在同一对象中，不要同时使用 OnMiddleDown 和 OnMiddleKeyDown 动画链接。

示例：

```
<OnMiddleDown>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</OnMiddleDown>
```

鼠标中键按下期间/中键按下期间

可以为 WhileMiddleKeyDown 或 WhileMiddleDown 动画链接指定以下元素。

元素	描述
Text	脚本文本。必需
Frequency	脚本执行频率，以毫秒为单位。1 到 360000 之间的一个数。使用可选架构时必需。缺省值为 1000。

在同一对象中，不要同时使用 WhileMiddleDown 和 WhileMiddleKeyDown 动画链接。

示例：

```
<WhileMiddleDown>
<Text>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</Text>
<Frequency>1000</Frequency>
</WhileMiddleDown>
```

鼠标中键放开时/中键放开时

可以将脚本文本放置在 OnMiddleKeyUp 和 OnMiddleUp 动画链接中。在同一对象中，不要同时使用 OnMiddleUp 和 OnMiddleKeyUp 动画链接。

示例：

```
<OnMiddleUp>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</OnMiddleUp>
```

双击鼠标中键时

可以将脚本文本放置在 OnMiddleDoubleClick 元素中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例：

```
<OnMiddleDoubleClick>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</OnMiddleDoubleClick>
```

鼠标悬停时

可以为 OnMouseOver 动画链接指定以下元素。

元素	描述
Text	脚本文本。必需。
Timeout	超时，以毫秒为单位。使用可选架构时必需。缺省值为 250。

示例：

```
<OnMouseOver>
<Text>
<![CDATA[
脚本文本的第一行
脚本文本的第二行
脚本文本的第 N 行
]]>
</Text>
<Timeout>250</Timeout>
</OnMouseOver>
```

鼠标左键等价键

可以为鼠标左键等价键动画链接指定以下元素。

元素	描述
KeyCode	虚拟键代码名或空字符串。必需。空字符串表示无分配。
KeyFlags	虚拟键标帜组合或空字符串。必需。空字符串表示无分配。未指定 KeyCode 时禁用。

“CTRL+B”的示例：

```
<LeftKey>
    <KeyCode>B</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </LeftKey>
```

“L”的无键标帜修饰符示例：

```
<LeftKey>
    <KeyCode>L</KeyCode>
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
  </LeftKey>
```

CTRL+SHIFT+F7 的两个键标帜修饰符示例：

```
<LeftKey>
    <KeyCode>F7</KeyCode>
    <KeyFlags>CtrlShift</KeyFlags>
  </LeftKey>
```

鼠标右键等价键 - 不支持

InTouch XML 导入功能不支持鼠标右键等价键。

可以为鼠标右键等价键动画链接指定以下元素。

元素	描述
KeyCode	虚拟键代码名或空字符串。必需。空字符串表示无分配。
KeyFlags	虚拟键标帜组合或空字符串。必需。空字符串表示无分配。未指定 KeyCode 时禁用。

示例：

```
<RightKey>
    <KeyCode>B</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </RightKey>
```

鼠标中键等价键 - 不支持

InTouch XML 导入功能不支持鼠标中键等价键。

可以为鼠标中键等价键动画链接指定以下元素。

元素	描述
KeyCode	虚拟键代码名或空字符串。必需。空字符串表示无分配。
KeyFlags	虚拟键标帜组合或空字符串。必需。空字符串表示无分配。未指定 KeyCode 时禁用。

```
<MiddleKey>
    <KeyCode>B</KeyCode>
    <KeyFlags>Ctrl</KeyFlags>
  </MiddleKey>
```

不支持的 InTouch 功能

使用 XML 导入功能无法导入向导和 ActiveX 控件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用 XML 导入功能无法导入标记或标记数据库。您可以创建包含标记定义的文件，然后使用 DBLoad 实用程序将其导入到应用程序的“标记名字典”。如需有关 DBLoad 的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序维护*》。如需有关从命令提示符运行 DBLoad 的信息，请参阅从命令提示符运行 DBLoad。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 15 在 CONNECT 中使用工业图形

CONNECT 是通用的云储备库，允许您管理常见的内容，如工业图形、控件和云中的小组件。

可以按需下载和上传图形。图形存储在 CONNECT 的“仓库”中；有三种类型的仓库 – 全局、租户和用户特定。在每个仓库内，用户都可以配置多个驱动器。您必须拥有 CONNECT 用户帐户才能管理云中的图形。每个驱动器都可以为不同的用户配置不同的访问级别。

用户将拥有只读或读写访问权限，具体取决于管理员授予的访问权限。

- 具有只读访问权的用户可以在云中查看图形。他们可以将图形下载到云，但这些图形将以只读方式提供。他们无法修改或上传图形。
- 具有读写访问权的用户可以查看、下载、修改图形，并将其上传到云。

多个用户可以访问同一驱动器中的图形，但一次只有一个用户可以编辑或保存图形。

登录到 CONNECT

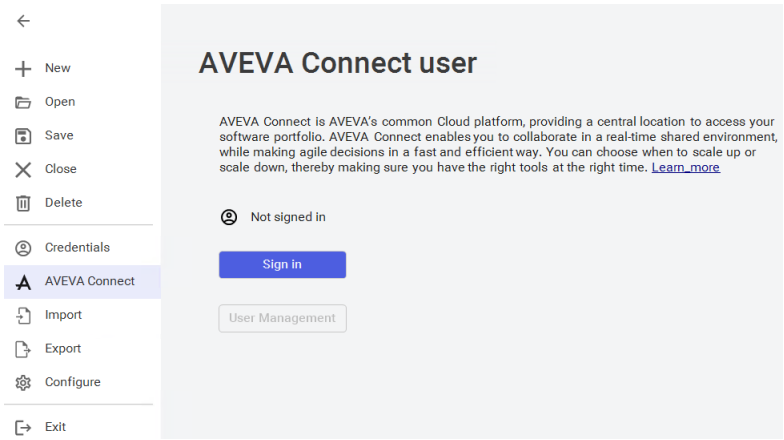
您必须拥有 CONNECT 帐户和管理员的授权才能访问 CONNECT。

1. 如果使用的是 InTouch，请启动 WindowMaker，然后从文件菜单中选择 **CONNECT**。

   如果使用的是 System Platform IDE，请选择 **Galaxy** 菜单，然后选择 **CONNECT**。

   注意：如果在配置器的许可模式下选择了连接体验模式，则 **CONNECT** 选项将不可用。

   此时出现“CONNECT 用户”配置屏幕。



2. 选择登录。
3. 输入您的 CONNECT 电子邮件地址。
4. 输入您的密码。

   如果凭证通过验证，则您已登录。

   注意：您必须拥有“内容参与者”角色，才能访问 CONNECT 工业图形驱动器。此要求适用于 Operations Control 连接体验和非连接体验使用。

5. 选择 **AVEVA Cloud Integration Studio** 解决方案。例如：Pym Technologies。
6. AVEVA Drive 显示为“可视化”文件夹内的单独储备库。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：选择 Integration Studio 解决方案后，它会缓存在浏览器内存中。要切换 Integration Studio 解决方案，必须清除浏览器缓存。

在共享驱动器之间导航

登录到特定仓库后，您可以在不同的驱动器间移动，并可以上传和下载图形。

1. 在视图菜单上的云驱动器组中，选择共享驱动器。
2. 从可用选项中选择一个驱动器。

   “可视化”文件夹将刷新以显示所选驱动器。

注意：您无法移动由另一个用户锁定的图形。

将图形上传/下载到云

AVEVA Drive 功能与您的本地“可视化”文件夹类似。您可以在 AVEVA Drive 上上传或创建文件夹来组织图形。通过读写访问权，可以将图形下载到本地“可视化”文件夹，并将其用于您的应用程序。在下载或上传图形时，将根据所选的覆盖选项覆盖内容。

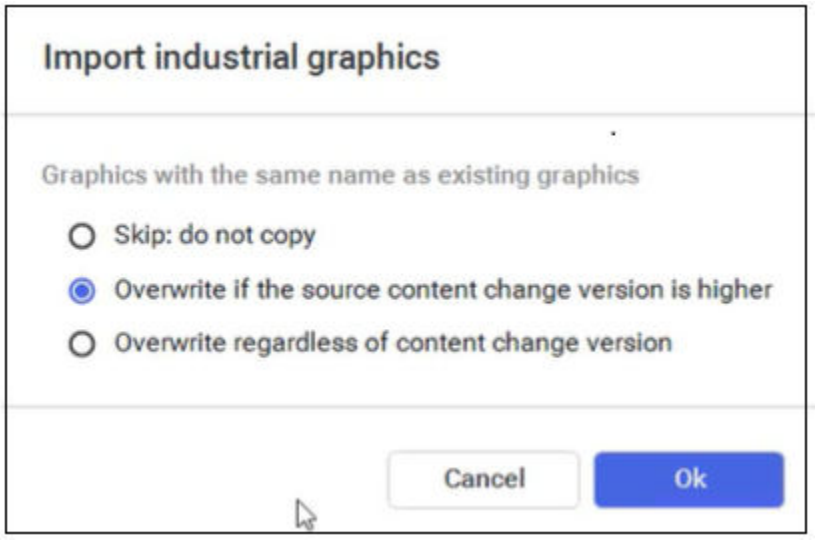
- 如果目标驱动器中已经存在这些内容，它们将保留在相同的工具集位置。否则，内容将添加到与源相同的工具集路径中。

对于所选的嵌入内容

- 如果目标驱动器中已经存在嵌入内容，这些内容将保留在目标中相同的工具集位置。
- 如果嵌入内容与其父内容位于相同的工具集位置，或位于相同的文件夹层次结构中，则会将嵌入内容添加到与父内容相同的工具集路径中。否则，嵌入内容将添加到与源相同的工具集路径中。

将图形上传到 CONNECT

1. 将图形从本地“可视化”文件夹拖放到 AVEVA Drive 文件夹。
2. 如果图形已存在于该驱动器中，将会出现一个对话框，要求覆盖或跳过上传。



3. 选择适当的选项，然后选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 此时出现上传内容对话框并显示操作进度。选择查看详细信息可查看进度。
5. 选择关闭。

您还可以上传包含多个图形的文件夹。

将图形下载到本地可视化文件夹

具有读写访问权的用户可以将图形从 AVEVA Drive 下载到他们的本地“可视化”文件夹。

1. 将图形或文件夹从 AVEVA Drive 拖放到本地“可视化”文件夹。

   下载内容对话框要求确认下载。

   - 选择是以继续下载。
   - 选择否以取消操作。
   - 选择查看详细信息以查看所用控件和域名空间的列表。下载期间的任何错误也都会显示。
2. 如果图形已存在，则出现“导入工业图形”对话框。
3. 选择适当的选项。
4. 图形将下载到本地“可视化”文件夹中。此外还会复制嵌入图形。

对各种云内容的上传/下载支持

下表中列出了对各种云内容的上传/下载支持。

云内容	上传支持	下载支持
嵌入式客户端控件和 HTML5 小组件	上传会将嵌入式图形内容（包括客户端控件和 HTML5 小组件）打包为 aaPKG 格式，然后复制/覆盖到云。	下载将提取嵌入式图形内容（包括客户端控件和 HTML5 小组件），并将控件导入/覆盖到本地应用程序中。
嵌入式应用程序对象图形	不支持	不支持
自定义脚本库	不支持	不支持
应用程序样式库	不支持	不支持

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

管理 CONNECT 中的图形

您还可以直接在 AVEVA Drive 上创建、重命名、更新、复制和删除图形，就像管理本地图形一样。

- 右键单击 AVEVA Drive 或任一文件夹可创建新的图形或工具集。
- 双击可使用图形编辑器编辑图形。
- 右键单击并选择相应的选项可重命名、复制或删除图形，以及更新缩略图。

  云不支持的上下文菜单选项显示为灰色。例如，“设置 Web 客户端主页符号”、“设置 Web 客户端根文件夹”、“导出符号”和“导出本地化”。

  注意：直接在 AVEVA Drive 中更新嵌入图形的缩略图会产生大量云流量。因此，我们建议您在本地更新缩略图。

多个用户管理图形

多个用户可以同时使用 AVEVA Drive 上可用的图形。但是，图形一次只能由一个用户进行编辑。当一个用户正在编辑云储备库中的图形时，该图形将被签出并锁定，从而防止其它用户进行任何编辑。所有其它用户只能在签出时查看只读的未修改图形。仅当修改用户将其签入时，最新的更改才会反映出来。

- 文件夹中图形图标旁边的锁定图标表示该图形已由登录的用户签出。
- 文件夹中图形图标旁边的编辑图标表示该图形正在由另一个用户进行编辑。

第一个用户保存并签入图形后，第二个用户可以签出并对其进行更改。在保存并签入图形之后，所有用户都可以查看已修改的图形。

关于云图形版本

AVEVA Drive 图形支持工业图形、客户端控件，以及从具有不同图形子系统版本的不同产品发布的云中小组件。

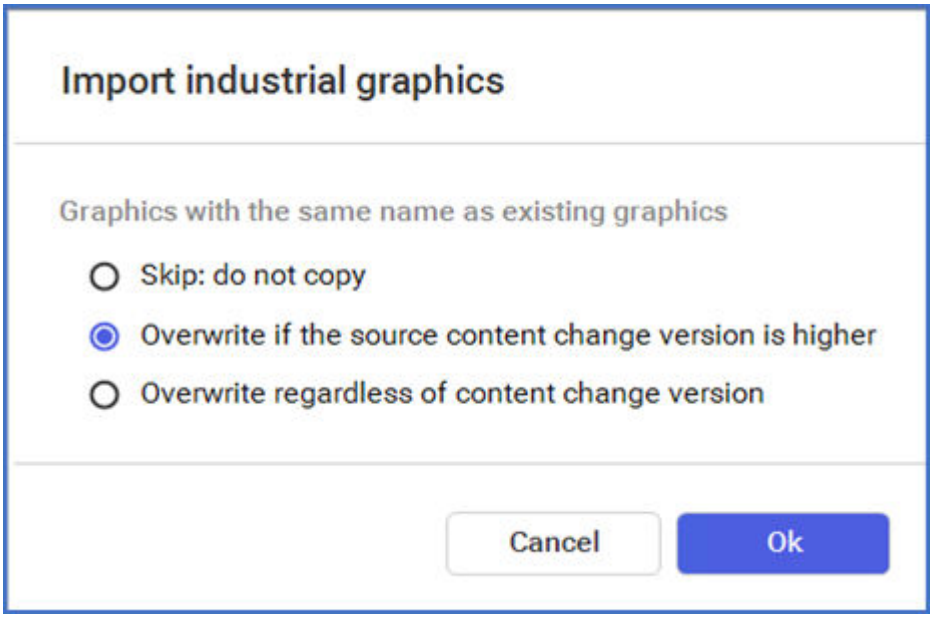
云图形版本支持的关键概念

- 工业图形以特定格式存储在本地和云中，该格式支持图形子系统中可用的所有特性和功能。
- 当向图形子系统中添加新的特性和功能时，需要保存并签入图形，以供其它用户使用。
- 工业图形子系统的不同版本由唯一的内部版本号标识，用户不可见。在发行时，每个产品都将使用最新版本的工业图形子系统。
- 工业图形子系统的版本号包括在每个存储的工业图形中，并且工业图形子系统使用该版本号来确保只有具有匹配或更高版本工业图形子系统的产品才允许编辑、嵌入或下载该工业图形。
- 连接到同一个云储备库的所有用户将看到同一组图形。编辑、嵌入或下载云图形的能力将取决于用户连接所用的产品中工业图形子系统的版本。
- 由具有较新版本工业图形子系统的产品保存到云储备库的工业图形，将在具有访问相同云的较旧版本的产品的“可视化”文件夹中显示为禁用。旧产品版本中的用户将无法编辑、嵌入或下载图形储备库。
- 就像本地“可视化”文件夹一样，图形名称在任何单个云储备库中都必须是唯一的。即使图形是由具有不同工业图形子系统版本的产品存储的，也是如此。

覆盖云中的图形内容

如果上传或下载的图形内容已存在，系统会提示覆盖该图形。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



覆盖取决于源和目标的图形子系统版本，以及图形的更改日志版本。

- 选项 1 将跳过覆盖，无论源和目标的图形子系统版本或更改日志版本如何。
- 如果源和目标之间的图形子系统版本不同，无论更改日志版本如何，选项 2 和 3 都将覆盖图形。
- 如果源和目标之间的图形子系统版本相同：
  - 仅当源具有更高的更改日志版本时，选项 2 才会覆盖。
  - 无论更改日志版本如何，选项 3 都将覆盖。

覆盖云中的自定义客户端控件和小组件

云图形版本管理的指南不适用于此版本发行时的自定义客户端控件和小组件。支持对客户端控件和小组件进行下载或上传操作。但是，不支持覆盖。如果自定义客户端控件或小组件已在目标位置可用，则无论客户端控件内部版本如何，都将跳过覆盖。要覆盖源储备库客户端控件，必须先在源储备库中删除该客户端控件。之后，上传/下载客户端控件即会成功。

注意：如果将新的自定义客户端控件或小组件上传到云，则连接到相同云储备库的所有其它用户必须重新启动 WindowMaker，才能使用最新的自定义客户端控件或小组件。

云图形版本控制 - 情形和示例

本节提供云图形版本处理的高级情形和期望值的非排他性列表。

让我们考虑两种不同版本的产品 P1_Old 和 P2_New，它们连接到 AVEVA 云中的相同租户。

- P1_Old 具有较旧版本的图形子系统，而 P2_New 具有较新版本的图形子系统。
- 首先，云储备库具有由 P1_Old 上传的 S1 和 S2 图形。

情形 / 步骤	期望值
情形 1： P2_New 创建 S3。	S3 保存在较新的图形版本中，因此无法在较旧的版本中对其进行编辑、嵌入或下载。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

P1_Old 试图编辑 S3。	P1_Old 收到一条消息，表明 S3 保存在较新版本中，无法打开。
情形 2： P2_New 打开 S2。 P2_New 将 S2 保存在云中。 接着，P1_Old 试图打开 S2。	在较旧的图形版本中创建的 S2 可以在较新的图形版本中打开和保存。 S2 保存在较新的图形版本中，因此无法在较旧的版本中对其进行编辑、嵌入或下载。 P1_Old 收到一条消息，表明 S3 保存在较新版本中，无法打开。
情形 3： P1_Old 保存 S1 和 S2，其中 S2 嵌入到 S1 中。 P2_New 仅保存和更新 S2。 P1_Old 试图编辑 S1。	因为 S1 仍然是 P1_Old 中的相同版本，所以 S1 被打开。 嵌入的 S2 将无法加载。
情形 4： P2_New 在云中创建一个新的 S4。 P2_New 在 S4 中嵌入 S1、S2 和 S3。	虽然 S1 位于较旧的图形版本中，但仍然可以在 S4（较新版本）中加载。 可以嵌入 S2 和 S3，因为它们与 S4 是相同的图形版本。
情形 5： P1_Old 在云中创建一个新的 S5。 P1_Old 在 S5 中嵌入 S1、S2 和 S3。	可以嵌入 S1，因为它们是相同的图形版本。 由于 S2 和 S3 是较新的版本，因此无法嵌入。
情形 6： P1_Old 下载 S3。	因为 S3 位于较新的图形版本中，所以无法下载。
情形 7： 云储备库具有一个较新图形版本的图形 S3。 P1_Old 试图上传具有相同名称 S3 的图形。	通过选择“选项 3”，可以覆盖该图形。
情形 8： P2_New 具有图形 S8。 P1_Old 在本地创建 S8、S9，并将 S8 嵌入到 S9 中。 P1_Old 将 S9 上传到云。	S9 上传到了云。 S8 被阻止上传，因为云中存在较新的版本。通过选择“选项 3”，可以覆盖内容。

键盘快捷方式

文件菜单

键盘快捷键	描述

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Ctrl + N	新建窗口
Ctrl O	打开窗口
CTRL + S	保存窗口

主页菜单

组	**键盘快捷键**	**描述**
剪贴板		
	CTRL+X	剪切
	Ctrl+C	复制
	Ctrl+D	重复
	Ctrl+V	粘贴
对齐		
	Ctrl+F4	垂直顶部
	Ctrl+F6	垂直中心
	Ctrl+F8	垂直底部
	Ctrl+F3	水平左侧
	Ctrl+F5	水平中心
	Ctrl+F7	水平右侧
排列		
	Shift + F9	置于顶层
	F9	置于底层
	F6	顺时针旋转
	Shift + F6	逆时针旋转
	F7	水平翻转
	Shift + F7	垂直翻转
间距		
	Ctrl+I	水平
	Ctrl+K	垂直
	Ctrl+F9	中心点
图形		
	F5	执行构建

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

	Shift+F5	中断
单元		
	F3	执行构建
	Shift+F3	中断
报警组		
	Shift+A	打开报警组列表
	Alt+A	添加报警组
标记字典		
	Ctrl+T	打开标记名字典
	Ctrl+P	打开访问名对话
	Ctrl+R	打开“交叉引用”屏幕
	Ctrl+U	更新使用计数
	Ctrl+G	删除未使用标记

绘图菜单

组	键盘快捷键	描述
模式		
	CTRL+A	动画链接
	Shift+W	选择元素
格式		
	A+ 或 Ctrl+加号键	字体放大
	A- 或 Ctrl+减号键	字体缩小
对齐		
	Ctrl+F4	垂直顶部对齐
	Ctrl+F6	垂直中心对齐
	Ctrl+F8	垂直底部对齐
	Ctrl+F3	水平左对齐

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

	Ctrl+F5	水平中心对齐
	Ctrl+F7	水平右对齐
排列		
	Shift + F9	置于顶层
	F9	置于底层
	F6	顺时针旋转
	Shift + F6	逆时针旋转
	F7	水平翻转
	Shift + F7	垂直翻转
间距		
	Ctrl+I	水平
	Ctrl+K	垂直
	Ctrl+F9	中心点
智能符号		
	Shift+G	生成
	Shift+S	开始编辑
	Shift+M	管理
	Shift+E	结束编辑
	Shift+R	恢复

动画菜单

组	键盘快捷键	描述
模式		
	Shift+W	选择
	CTRL+A	动画链接
链接		
	Shift+X	剪切
	Shift+C	复制
	Shift+L	清空
图形		
	Shift+B	执行构建

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

	Shift+F5	中断
单元		
	F3	执行构建
	Shift+F3	中断
对象		
	Alt+F5	调整形状
	F2	全选
位图		
	Alt+F7	导入
	Alt+F8	原始大小
	Alt +F9	编辑
	Ctrl+V	粘贴
替换		
	Ctrl+E	替换标记
	Ctrl+L	替换字符串

属性菜单

组	**键盘快捷键**	**描述**
图形		
	Shift +O	原始大小
	Alt+B	编辑
	Shift+D	动态大小传播
Symbol Factory		
	Shift+V	从剪贴板粘贴
	Shift+P	从文件导入
	Shift+Y	创建新类别
	Alt+J	添加所选 InTouch 对象
客户端		
	Shift+N	立即通知
	Shift+F	关闭时通知

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

视图菜单

组	键盘快捷键	描述
查看		
	Alt+Y	恢复
	Alt+W	窗口
	Alt+A	SuperTag
	Alt+O	工具
	Alt+I	脚本
	Alt+G	图形
	Alt+E	属性
	Alt+D	对齐网格
	Alt+R	标尺
	Alt+T	状态栏
更新		
	Alt+M	窗口略图
应用程序		
	Alt+N	打开文件夹
主题		
	Alt+L	在深色模式和浅色模式之间切换

其它

键盘快捷键	描述
Shift+H	打开帮助主题
Alt+U	打开运行时
Ctrl + W	打开属性网格
Shift + 加号键	放大半径（圆角长方形）
Shift + 减号键	缩小半径（圆角长方形）
Ctrl + H	水平间距

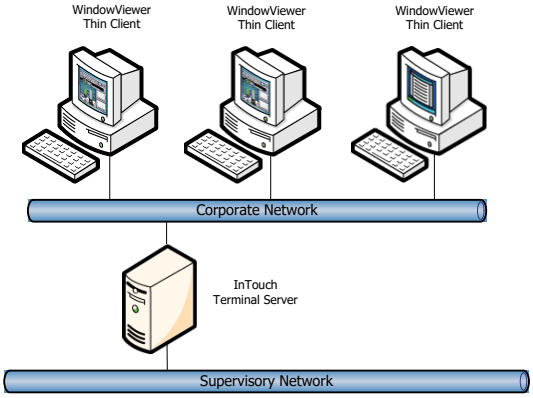
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Deploy

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 16 部署和使用终端服务与远程桌面服务

“终端服务”是 Microsoft Windows Server 操作系统中包含的一种可配置的服务，该服务从服务器集中运行基于 Windows 的应用程序。在“终端服务”中，客户端计算机访问服务器节点，其中有多个 InTouch 应用软件的实例在同时运行。



“终端服务”环境由三个部分组成：

- **终端服务服务器**服务器管理每个客户端会话的计算资源，并为客户端用户提供唯一的环境。服务器接收并处理远程客户端上执行的所有键击与鼠标操作，并将操作系统与应用程序的所有显示输出传递给相应的客户端。所有的“终端服务”应用程序处理都在服务器上发生。
- **远程桌面协议（Remote Desktop Protocol，简称 RDP）**。“远程桌面协议”(RDP) 客户端应用程序将输入数据（如键击、鼠标移动等）传递给服务器。
- **客户端**。“终端服务”客户端不执行本地应用程序处理，它仅显示应用程序输出。通过运行 Windows 程序菜单上的**终端服务客户端**命令，可以从客户端访问“终端服务”。与“终端服务器”连接之后，客户端环境看起来与 Windows 服务器的一样。由于应用程序不在本机运行，因此是完全透明的。

如需有关“终端服务”的详细信息（包括功能与优点），请参阅 Microsoft 文档。

规划终端服务器应用程序

重要事项：我们建议先在测试服务器上安装应用程序，然后再在生产环境中部署。

在安装“终端服务”之前

- 确定需要在服务器上安装的客户端应用程序（如 InTouch）。
- 确定客户端的硬件要求。
- 确定支持客户端所需的服务器配置。
- 确定“终端服务”以及要运行的其它应用程序所需的许可证。
- 理解 InTouch 应用程序的某些方面（如报警、安全性、I/O 及脚本）在“终端服务”下如何运行。

在终端服务环境中部署 InTouch 应用程序

在终端服务环境中部署 InTouch 应用程序时，应该为每个节点部署一个单独的 InTouch 应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

终端服务环境中的报警

通过给 Terminal Services for InTouch 使用“分布式报警系统”，不同终端会话上运行的客户端可以选择显示哪些报警以及如何显示。

“报警供应器”通过唯一确定应用程序及应用程序实例的名称来识别它们。“报警供应器”或“报警接收器”向“分布式报警系统”注册之后，此信息将可供“分布式报警系统”使用。

运行“报警供应器”的节点则通过在系统中唯一确定该计算机节点的名称进行识别。在该计算机节点上启动“分布式报警系统”的实例之后，此信息便会提供给它。

在记录报警事件时，节点与完整的“报警供应器”名可以确定报警的来源。

在“终端服务”环境中确认报警时，记录的“操作员节点”是运行操作员所使用的“终端服务”会话的客户计算机的名称。如果无法检索计算机的节点名，则使用节点的 IP 地址。

注意：“报警供应器”在“终端”会话上不受支持。它们只在“终端控制台”上受支持。

终端服务环境中的安全性

使用应用程序安全性来保证 InTouch 应用程序、IndustrialSQL Server 及其它机密信息系统的安全。

- 使用 $Operator 系统标记来确保应用程序安全性。然后，您可以通过将这些函数链接到内部标记，以控制操作员对特定功能的访问。

  如需有关使用 $Operator 系统标记的详细信息，请参阅关于保护 InTouch 安全。

- 使用更新的 TseGetClientId() 函数取代 GetNodeName() 函数，来确定客户端计算机。使用“终端服务”时，GetNodeName() 函数返回终端服务器的名称，而不是客户端计算机的名称。

使用安全性跟踪来监视非法入侵企图。如果怀疑系统遭受任何形式的攻击，则可以启用日志功能，以记录一系列可跟踪的事件。通常安全性日志/跟踪功能要求使用大量的处理资源，所以缺省条件下它是禁用的。

警示：安全性跟踪要求使用大量的资源。在评估试验服务器时启用跟踪功能，以准确估计 InTouch 应用程序硬件要求。

终端服务环境中的 I/O

InTouch HMI 无法在“终端服务”环境中启动“I/O 服务器”。根据 view 会话的启动顺序，可能需要使用 IOReinitialize() 函数。在启动从这些服务器读取数值的应用程序之前，必须确保所有的服务器（I/O 设备或 view 应用程序）都在运行。

避免在 WindowViewer 启动时接收“正在初始化 I/O”的错误消息

1. 打开 WindowMaker。
1. 在文件菜单上，选择配置，然后选择 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。
1. 在首选项选项卡上的 **I/O** 部分中，清除启动本地服务器复选框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

终端服务环境中的脚本执行

“终端服务”中运行的所有应用程序都使用相同的定时参考（服务器时钟），因此在存在大量的 CPU 加载操作期间，脚本可能不会运行。CPU 加载异常可能会由视频处理负载过重引起，在多个应用程序定义了相同的脚本触发器（例如 End-of-Shift 事件）时，也可能会发生此种情形。因此，如果服务器正忙于处理许多客户端的脚本，则在定时器通常可以启动脚本的时间间隔之内，它可能无法启动其它客户端上的脚本。这可能会使脚本无法在客户端上运行。

为确保脚本正确运行，请将使用公用触发器的脚本合并到一起，并将它们移到一个应用程序（如标记服务器）中。这是执行试验布署的主要原因之一。通过试验部署，您可以有机会执行压力测试，以确定所选的硬件配置是否够用。

正确登录到终端会话以运行 InTouch

每个会话都必须使用唯一帐户登录。这可以手工完成，也可以将终端服务配置为实施唯一登录。

注意：使用同一登录帐户在多个会话上运行可导致损坏和其它意外结果。

终端服务环境中的报警查询语法

会话报警的报警查询语法为：

```
\\ServerNodename\InTouch!$System
```

控制台报警的报警查询语法在节点名称后包括一个冒号 (:)，例如：

```
\\ServerNodename:\InTouch!$system
```

终端服务环境中的其它限制

下表介绍在终端服务器上运行应用程序时的一些限制以及建议的解决方案。

功能	是否支持？	注释
WindowViewer	是	WindowViewer 在“终端服务”下作为服务运行时不受支持。
到 I/O 设备或 MS Office（例如 Excel）的 DDE	否	使用标记服务器（控制台或单机）。这包括 DDE QuickScript：WWExecute()、WWpoke() 以及 WWRequest()
自 MS Office 的 DDE（例如 Excel 中配置的热链接）	是	Excel 与 InTouch HMI 必须在相同的会话中运行。
历史趋势	是	使用标记服务器或 NAD 来记录数值。多个会话可能会读取相同的历史文件，但仅有一个控制台可以写入历史文件。
InTouch 报警记录器	是	--
MEM OLE 自动化	是	--
打印报警	否	--

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

功能	是否支持？	注释
保留标记	是	必须使用 NAD。
SQL 访问 (ODBC)	是	数据库必须位于单独的计算机上。
到 I/O 设备或其它 InTouch 应用程序的 SuiteLink。	是	与另一个视图会话通讯时，应包含“终端服务器”节点名，并将所需会话的 IP 地址附加到该应用程序名。例如，view10.103.25.6.，“I/O 服务器”在客户端会话中不受支持。

使用脚本检索关于 InTouch 客户端会话的信息

您可以给“终端服务”使用以下 InTouch QuickScript 函数。

- TseGetClientId() 函数
- TseGetClientNodeName() 函数
- TseQueryRunningOnConsole() 函数
- TseQueryRunningOnClient() 函数

TseGetClientId() 函数

如果 View 应用程序在“终端服务器”客户端上运行，则返回字符串形式的客户端 ID（客户端的 TCP/IP 地址）。此客户端 ID 供内部使用，以生成 SuiteLink 服务器名与 logger 文件名。否则 TseGetClientId() 函数返回一个空字符串。

语法

```
MessageResult=TseGetClientId();
```

示例

客户端 IP 地址 10.103.202.1 保存到 MsgTag 标记中。

```
MsgTag=TseGetClientID();
```

TseGetClientNodeName() 函数

如果给运行 View 应用程序的“终端服务器”客户端指定的名称 Windows 能够识别，则返回客户端节点名。否则 TseGetClientNodeName() 函数返回一个空字符串。

语法

```
MessageResult=TseGetClientNodeName();
```

示例

客户端节点名作为指定给 MsgTag 标记的值返回。

```
MsgTag=TseGetClientNodeName();
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

TseQueryRunningOnConsole() 函数

TseQueryRunningOnConsole() 函数可以从脚本中运行，以显示 View 应用程序是否正在“终端服务”控制台上运行。

语法

```
Result=TseQueryRunningOnConsole();
```

返回值

如果 View 应用程序在“终端服务”控制台上运行，则返回非零整数值。否则 TseQueryRunningOnConsole() 函数返回零。

示例

如果 WindowViewer 在“终端服务”控制台上运行，则 IntTag 设置为 1。

```
IntTag=TseQueryRunningOnConsole();
```

TseQueryRunningOnClient() 函数

如果 View 应用程序在“终端服务”客户端上运行，则返回非零整数值。否则返回零。

语法

```
Result=TseQueryRunningOnClient();
```

返回值

如果 View 不在“终端服务”客户端上运行，则返回 0。

示例

如果 WindowViewer 在“终端服务”客户端上运行，则 IntTag 设置为 1。

```
IntTag=TseQueryRunningOnClient;
```

远程桌面服务概述

远程桌面服务（以前称为终端服务）是 Windows Server® 2008 R2 及更高版本中的一种服务器角色，它提供了相关技术，使用户能够访问安装在远程桌面会话主机（RD 会话主机）服务器上的基于 Windows 的程序，或访问完整的 Windows 桌面。通过远程桌面服务，用户可以从企业内部网络或互联网 访问 RD 会话主机服务器。

当用户访问 RD 会话主机服务器上的程序时， 该程序会在服务器上运行。每个用户只能看到自己的个人会话。会话由服务器作系统透明地进行管理，并且独立于任何其它客户端会话。此外，您可以配置远程桌面服务，使用 Hyper-V™ 为用户分配虚拟机，或者让远程桌面服务在连接时为用户动态分配一个可用的虚拟机。

有关远程桌面服务的详细信息，请参阅 Windows Server 2008 R2 TechCenter (http://go.microsoft.com/fwlink/?LinkId=138055) 上的“远程桌面服务”页面 。

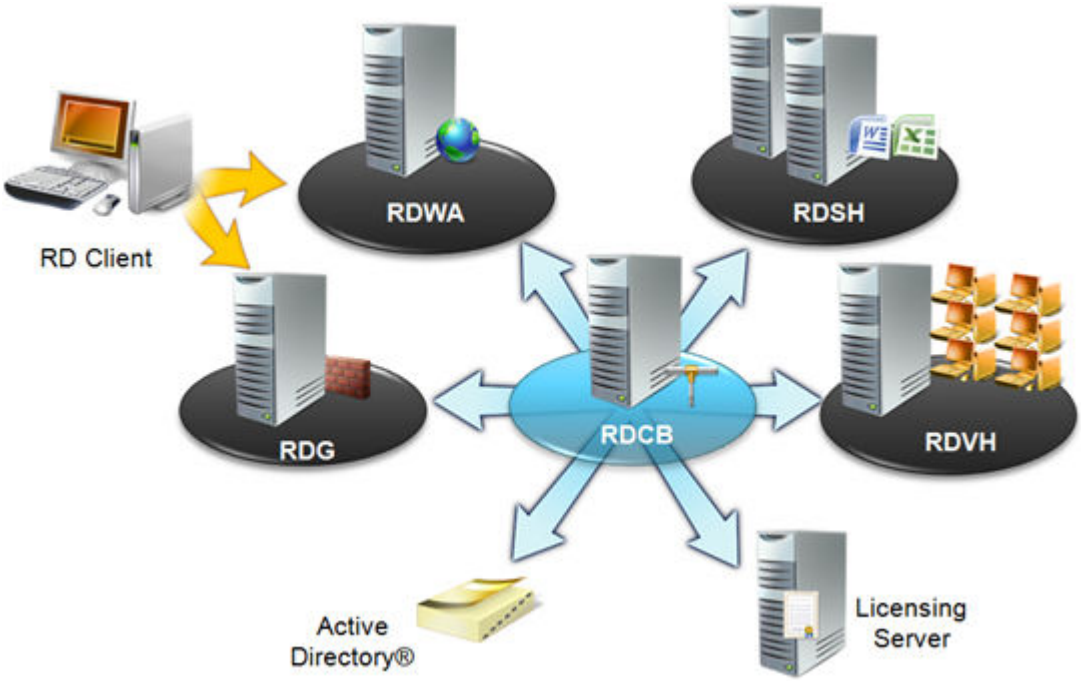
“远程桌面服务”角色服务

远程桌面服务是由多个角色服务组成的服务器角色。在 Windows Server 2008 R2 及更高版本中，远程桌面服务由以下角色服务组成：

- **RD 会话主机**：远程桌面会话主机（简称 RD 会话主机，以前称为“终端服务器”），使服务器能够托管基于 Windows 的程序或完整的 Windows 桌面。用户可以连接到 RD 会话主机服务器，以运行程序、保存文件和使用该服务器上的网络资源。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- **RD Web 访问**： 远程桌面 Web 访问（简称 RD Web 访问，以前称为 TS Web 访问），允许用户通过运行 Windows 7 的计算机上的“开始”菜单或通过 Web 浏览器访问 RemoteApp 和桌面连接。RemoteApp 和桌面连接为用户提供 RemoteApp 程序和虚拟桌面的自定义视图。
- **RD 许可**： 远程桌面许可（简称 RD 许可，以前称为 TS 许可）管理远程桌面服务客户端访问许可证 (RDS CAL) , 这些许可证是每个设备或用户连接到 RD 会话主机服务器所必需的。您使用 RD 许可在远程桌面许可服务器上安装、发放 RDS CAL，并跟踪这些许可证的可用性。
- **RD 网关**： 远程桌面网关（简称 RD 网关，以前称为 TS 网关）允许经授权的远程用户通过任何连接到互联网的设备，连接到企业内部网络中的资源。
- **RD 连接代理**： 远程桌面连接代理（简称为 RD 连接代理，以前称为 TS 回话代理）支持负载均衡的 RD 会话主机服务器集群中进行会话负载均衡和 会话重新连接。RD 连接代理还用于通过 RemoteApp 和桌面连接为用户提供对 RemoteApp 程序和虚拟桌面的访问权限。
- **RD 虚拟化主机**： 远程桌面虚拟化主机（RD 虚拟化主机）与 Hyper-V 集成，用于托管虚拟机并将其作为虚拟桌面提供给用户。您可以为组织中的每个用户分配唯一的虚拟桌面，或为其提供对虚拟桌面池的共享访问权限。



确保远程桌面服务 (RDS) 连接的安全

通过以下安全措施防范攻击：

1. 使用强密码

   对有权访问远程桌面的所有账户使用强密码。

2. 更新软件

   通过启用和审核自动 Microsoft 更新，确保正在运行的客户端和服务器软件均为最新版本。

3. 设置帐户锁定策略

   通过设置计算机在多次尝试输入错误密码后锁定账户一段时间，这将有助于防止“暴力破解”攻击。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 使用双因素身份验证

   RD 网关支持智能卡双因素身份验证。

5. 更改远程桌面的监听端口

   防止试图访问默认远程桌面端口 (TCP 3389) 的网络攻击和蠕虫。

6. 使用 RD 网关

   RD 网关限制对远程桌面端口的访问，同时通过单个“网关”服务器支持远程连接。使用 RD 网关服务器时，桌面和工作站上的所有远程桌面服务均通过 RD 网关进行路由。RD 网关服务器通过 HTTPS（端口 443）监听远程桌面请求，并将客户端连接到目标计算机上的远程桌面服务。有关详细步骤，请参阅以下链接：http://technet.microsoft.com/en-us/library/cc770601.aspx

7. 为远程桌面服务连接配置网络级身份验证

   网络级身份验证要求用户在创建会话之前，先通过 RD 会话主机服务器的身份验证。网络级身份验证可提高 RD 会话主机服务器的可用性（降低 RD 会话主机服务器遭受拒绝服务攻击的风险）。https://technet.microsoft.com/en-us/library/hh831778.aspx

8. 配置服务器身份验证和加密级别

   缺省条件下，“终端服务”会话使用远程桌面协议 (RDP) 内置加密功能。但是，RDP 并不提供用于验证终端服务器身份的身份验证。通过使用传输层安全性协议 (TLS) 1.0 进行服务器身份验证和加密终端服务器通信，可以增强终端服务会话的安全性。RDS 和客户端计算机必须正确配置 TLS，才能提供增强的安全性。默认情况下，客户端和服务器之间的 RDS 连接以可用的最高安全级别（128 位）进行加密，确保传输数据的完整性和机密性。

Windows Server 2016 远程桌面服务最佳实践

在 Windows Server 2016 环境中，可以实施以下远程桌面服务最佳实践

- 使用多因素身份验证

  利用 Active Directory 的强大功能，结合多因素身份验证，实施高级别的安全性保护。请参阅 Microsoft 文档：https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-plan-mfa

- 使用用户配置文件磁盘 (UPD) 确保数据存储安全

  用户配置文件磁盘 (UPD) 允许用户数据、自定义设置和应用程序配置跟随用户在一个单一集合中。UPD 是按用户、按集合保存在中央共享位置的 VHD 文件，用户登录时会挂载到用户的会话中 - UPD 在该会话期间被视为本地驱动器。请参阅 Microsoft 文档：https://docs.microsoft.com/en-us/windows-server/remote/remote-desktop-services/rds-plan-secure-data-storage

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 17 分发应用程序

通常，分布式应用程序有一个中央开发工作站、中央数据存储区以及多个客户端工作站。您可以使用 InTouch 的“网络应用程序开发”（Network Application Development，简称 NAD）来构建与维护分布式应用程序。NAD 可以在不限制应用程序开发的情况下，让多个客户端工作站分别维护一个应用程序副本。通过使用应用程序的单独副本，可以提供 Viewer 冗余配置。在应用程序发生更改时，客户端工作站自动收到通知。您可以创建单机、基于客户端以及基于服务器的 InTouch 应用程序。

您也可以使用 System Platform IDE 来管理与部署 InTouch 应用程序。

支持的 InTouch 架构

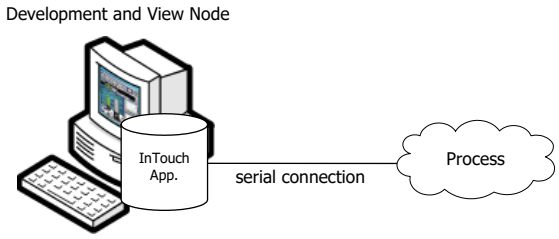
支持的 InTouch 网络架构包括：

- 单机
- 基于客户端
- 基于服务器
- NAD

单机架构

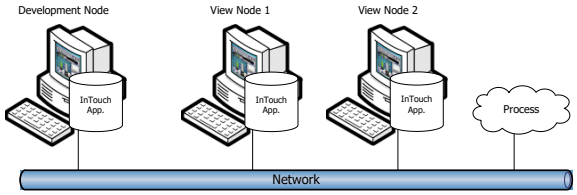
单机应用程序通常由一个充当主操作员界面的非联网计算机组成。此计算机直接连接（如使用串行电缆）到工业过程。

在这种架构中，您在一台计算机上开发 InTouch 应用程序。您可以将应用程序复制到另一台计算机上进行修改，然后再复制回原来的计算机上。



基于客户端的架构

在基于客户端的架构中，运行 WindowViewer 的每台计算机（View 节点）上，或网络服务器某个独特的位置上，都有某个 InTouch 应用程序的一份唯一的副本。在下例中，应用程序在开发节点上进行开发与测试，然后再复制到每个 View 节点上。



由于每个节点均可自足，因此具有内在的冗余特性；并且，在可以使用的 View 节点数上，不存在任何限制。

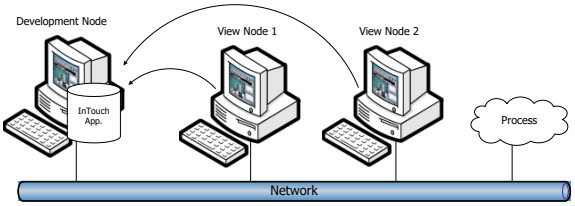
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

每个 View 节点都必须有完全相同的一份应用程序，并且对任何网络数据源（如“I/O 服务器”或 IndustrialSQL Server）都必须有相同的访问权限。不过，每个 View 节点都需要与共享的服务器维持一个单独的对话，这样就可能导致增加了网络负载。

您可以在开发节点上修改与测试应用程序，而不致影响正在运行的过程。不过，您必须将修改后的应用程序分发到各个 View 节点。您必须先在本地关闭每个 View 节点，将新的应用程序复制到其中，然后再重新启动。

基于服务器的架构

基于服务器的架构将一个公用的 InTouch 应用程序分发到多个 View 节点。在下图中，两个 View 节点从开发节点访问相同的应用程序。



对于每个 View 节点：

- 逻辑驱动器必须映射到开发节点的共享网络驱动器上。
- 共享的应用程序必须向 InTouch 程序注册。
- 计算机对应用程序引用的任何数据源都必须有完全相同的访问权限。通过使用脚本组合来确定节点名，并基于该名称来更改每个数据位置，可以有多种方法来定义数据源的位置。

在这种架构中，只需维护单个应用程序。应用程序发生变化并重新启动 WindowViewer 时，View 节点会自动更新。

这种架构的缺点包括：

- 应用程序的开发受到限制
- 如果开发节点崩溃，则没有冗余备份可以使用
- 所有节点都必须使用相同的屏幕分辨率

网络应用程序开发 (NAD)

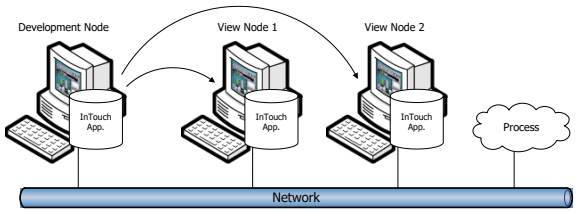
在“网络应用程序开发”(NAD) 架构中，在中央网络位置（通常是开发节点）上维护应用程序的 NAD 储备库副本。每个 View 节点都会将应用程序复制到用户定义的位置并从那里运行它。

在通知客户端应用程序有更改（使用 WindowMaker 特别菜单上的通知客户端命令）时，会在应用程序目录中设置一个标帜，然后由 View 节点读取它。

您可以配置 View 节点处理应用程序更改的方式。这可以是忽略更改，也可以是自动关闭并重新启动 View 节点，从而重新加载 NAD 储备库应用程序。

在下图中，两个 View 节点都从开发节点注册了 NAD 储备库应用程序，但实际上还是在各自的本地计算机上运行它。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



注意：如果将应用程序配置成将历史数据写入 NAD 储备库应用程序节点的应用程序目录，则所有的 NAD 节点都会尝试将它们的历史数据写入 NAD 储备库应用程序。为避免这种情况，在每个 NAD 节点上，将历史数据配置成写入本地目录，而不是 NAD 储备库应用程序节点。

如果将一个复杂的大型应用程序分发到许多个节点，则在最初下载时，系统响应时间可能会明显变慢。不过对于更新，这种情况会得到优化。对于较慢的网络或串行连接，应用程序传输可能会是个问题。

此外请注意其它的一些网络限制，如过滤掉特定类型的网络通信与地址的路由器用户。

规划网络化应用程序

在构建 InTouch 应用程序时，无论采取何种架构，都必须慎重考虑：

- 对 I/O 数据源的访问。
- 对共享文件的访问。
- 记录数据的位置。
- 任何特殊的网络要求。

网络化应用程序的 I/O 数据访问

InTouch HMI 使用“访问名”来引用实时 I/O 数据。每个“访问名”都等价于一个由节点名、应用程序以及主题组成的 I/O 地址。在分布式应用程序中，I/O 引用可以设置为网络“I/O 服务器”的全局地址，也可以是本地“I/O 服务器”的本地地址。

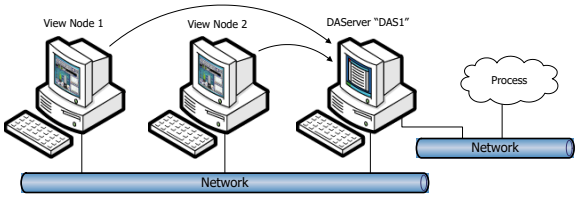
注意：InTouchView 只能使用单一 Galaxy“访问名”。您无法为 InTouchView 创建其它“访问名”。如需有关 InTouchView 限制的详细信息，请参阅在运行时查看应用程序。

View 节点对数据源的访问权限必须与开发节点相同。

使用全局 I/O 地址

I/O 数据的全局地址允许所有 View 节点共享一个基于网络的公共“I/O 服务器”。这样便没有必要使用多个“I/O 服务器”，但容错能力会下降，并可能导致整体性能降低。

在下图中，两个 View 节点在运行一份相同的应用程序。这两个 View 节点引用相同的 I/O 数据源。由于每个应用程序都使用该数据源的全限定 I/O 地址，因此所有的引用都指向同一个“I/O 服务器”。



您可以设置 InTouch 应用程序，使之能够使用第三方寻址惯例在“访问名”中确定另一个节点上存储的数据元素。<1>“</1>访问名<2>”</2>寻址惯例包括远程数据所在位置的节点名<3>、</3>应用程序名以及主题名

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

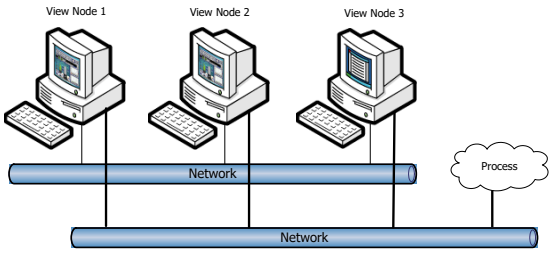
<4>。</4> InTouch 应用程序通过结合使用“访问名”与项目名来获取远程数据。如需有关为远程“I/O 服务器”定义“访问名”的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序开发*》中的使用 I/O 进行数据访问。

注意：在 WindowMaker 中创建“访问名”时，如果“访问名”使用 SuiteLink 协议，软件会阻止“访问名”去访问相同的节点、应用程序及主题。在运行时，请勿使用 IOSetAccessName() 函数将“访问名”重定向到重复的对象，否则重定向的“访问名”将无法正常工作。

使用本地 I/O 地址

每个 View 节点都有自己的“I/O 服务器”时，使用 I/O 数据的本地地址。这种架构提供容错操作，在网络中断时，每个 View 节点都能继续独立运行。

在下图中，两个 View 节点运行相同应用程序的不同副本，并引用各自的 I/O 数据源。由于每个应用程序都给该数据源使用本地 I/O 地址，因此每个引用都指向本地“I/O 服务器”。



使用本地“I/O 服务器”会显著增加过程连接网络的负载。例如，由于必须单独处理每个节点的请求，所以三个节点产生的流量将三倍于单个节点所产生的流量。

如需有关为本地“I/O 服务器”定义“访问名”的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序开发*》中的使用 I/O 进行数据访问。

SuiteLink

SuiteLink 通讯协议基于 TCP/IP 协议。由于提供以下功能，SuiteLink 通常用于一些高速工业应用：

- “数值时间质量”（Value Time Quality，简称 VTQ），其中的时间标签及质量指示符与传递给支持 VTQ 的客户端的所有数据值关联。InTouch HMI 是支持 VTQ 的客户端，其标记数据在传输时带有 VTQ 指示符。
- 通过 Microsoft Windows 操作系统的性能监视器，可以对数据吞吐量、服务器负载、计算机资源消耗以及网络传输等进行非常全面的诊断。
- 不管应用程序是在单个节点，还是分布在大量的节点上，都可以在应用程序之间始终维持很高的数据吞吐量。

SuiteLink 并非 DDE、FastDDE 或 NetDDE 的替代品。客户端与服务器之间的每个连接都取决于网络要求。

访问共享的文件

在分布式应用程序中，文件引用可以设置为

- 网络文件服务器的全局地址。
- 本地文件的本地地址。

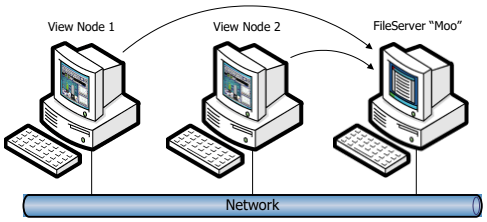
View 节点对数据源的访问权限必须与开发节点相同。

使用文件数据的全局地址

您可以设置文件数据的全局地址，以便所有的 View 节点都共享一组基于公共网络的文件。这就给这些文件提供了单源维护，但它的容错性能要比本地副本差。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在下图中，两个 View 节点分别运行一份相同的应用程序，但引用相同的配方文件。由于每个应用程序都使用映射成该文件全限定网络路径的盘符，因此所有的引用都指向相同的文件。



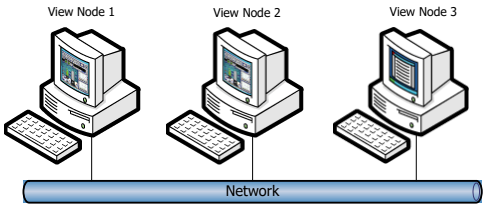
设置共享的文件

1. 将网络驱动器映射到包含所引用的文件的共享路径。例如，G:\Directory\Recipe.csv，其中"G:\"是指向 \\Moo\Share 的映射盘符。您必须在每个 View 节点上映射这个相同的驱动器。
2. 在脚本中，引用共享的路径。例如：

```
RecipeSelectRecipe("G:\Directory\Recipe.csv", "review", "RecipeName");
```

使用文件数据的本地地址

每个 View 节点都有自己的一份文件副本时，可以使用文件数据的本地地址。在下图中，三个 View 节点分别运行一份相同的应用程序，并引用配方文件的本地副本。



在本例中，本地地址是：

```
C:\Directory\Recipe.csv
```

其中"C:\"是本地驱动器。

在脚本中，引用本地路径。例如：

```
RecipeSelectRecipe("C:\Directory\Recipe.csv", "review", "RecipeName");
```

这种架构有容错功能。不过，您必须将任何文件更改都复制到所有的 View 节点上。

任何文件访问权限都只能为"只读"，并且不允许修改本地文件。

通过 UNC 访问共享的文件

在通常输入文件路径的任何地方（如应用程序目录项、配置项以及分布式报警），都可以使用"通用命名惯例"（Universal Naming Convention，简称 UNC）。如果使用 UNC 名，则不需要创建映射的驱动器。

UNC 地址采用 \\节点\共享\路径形式，其中：

- "节点"是包含文件共享的计算机的名称。
- "共享"是指定给该计算机上共享文件夹的逻辑名称。
- "路径"是跟共享有关的该文件的普通路径。

注意：如果使用 SuiteLink，则节点名的长度限制为 15 个字符。

在通过 UNC 访问文件之前，必须在要访问的计算机上创建文件共享。如需有关详细信息，请参阅 Windows 文档。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

例如，假设有一台网络名是 "EngineRm" 的计算机，并且您已使用共享名 "Root" 共享了该计算机的根驱动器 "C:\"。要设置到 "C:\IT\Apps\Boiler" 应用程序的 UNC 路径，必须使用以下 UNC：

\\EngineRm\Root\IT\Apps\Boiler

如果 "Boiler" 目录自身已作为 "Boiler" 共享，则 UNC 可以简化为：

\\EnginerRm\Boiler

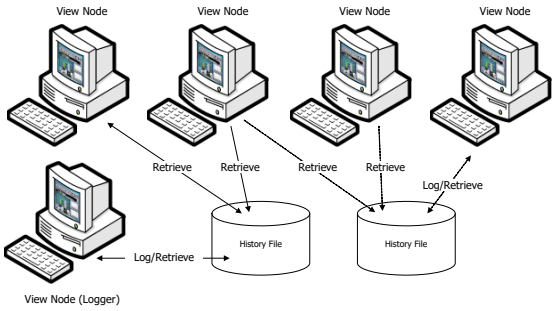
如果该共享是 PATH 环境变量指定的路径，则不要求提供路径。

注意：如果需要写入由 UNC 地址引用的文件，则该共享必须是可读写的共享，即使它位于本地节点。如果创建由口令保护的共享，则除非先设置了网络驱动器映射，否则将不能使用 UNC 来访问该共享。您可以使用 Windows 的<1>“</1>资源管理器<2>”</2>从远程节点来设置驱动器映射<3>。</3>

在分布式环境中记录数据

通过使用 InTouch 分布式历史系统，可以从网络上的任何 InTouch 应用程序中检索历史数据。此系统还允许同时从多个历史数据库远程检索数据。这些数据库都称为历史供应器。

只有一个 InTouch 节点可以写入分布式历史文件。不过，查看该文件内容的 InTouch 节点数则没有任何限制。



从历史文件检索数据的远程节点可能看不到最近一个小时的数据（基于 Logger 节点的时间）。远程趋势只能查看已写入记录节点的磁盘的数据。

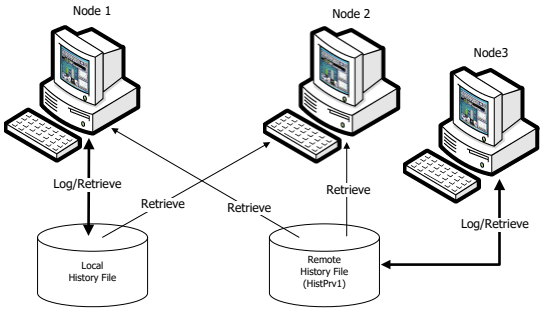
对于选择了“记录数据”的每个标记名，在给该标记名采集 22 个样本之后，其数据会自动写入磁盘。如果调用 HTUpdateToCurrentTime() 函数，则无论采集多少个样本，数据都会写入磁盘。缺省条件下，数据每小时写入磁盘一次。通过将下面这行代码添加到 INTOUCH.ini 文件，可以更改此时间间隔：

```
ForceLogging=X;
```

其中 X 是分钟数，可设置为 5 和 120 之间的任意时间间隔。

注意：使用分布式历史系统时，NetDDE Helper 服务必须正在运行。

下图显示一个典型的分布式历史系统的配置，它使用“网络应用程序开发”(NAD) 来分发应用程序。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

节点 1 与 2 包含同一份 InTouch 应用程序；不过该应用程序配置为只允许节点 1 写入本地历史文件，而任何一个节点都可以从本地历史文件或远程历史文件中检索。节点 3 也正在写入远程历史文件位置并从中进行检索。给节点 3（历史供应器）指定的名称为 HistPrv1。节点 1 既是开发工作站又是运行时工作站，而节点 2 只是运行时工作站。

创建此类型的应用程序

1. 创建一个历史供应器列表。请参阅配置远程历史供应器。。
2. 创建并配置历史趋势对象。如需详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序开发*》中的绘制标记数据的趋势。
3. 为应用程序配置分布式记录。请参阅配置分布式历史记录。
4. 分发应用程序。请参阅针对 NAD 配置 InTouch 应用程序。

您可以手动或通过 NAD 来分发应用程序。在分发应用程序时，历史供应器列表文件作为应用程序的一部分进行分发。

分发应用程序之后，可以运行 View 节点，并同时检索本地标记名和远程历史供应器的标记名。尽管应用程序将在所有的 View 节点上运行，但只有记录节点才将数据记录到历史日志文件；而其它节点则只能从中读取。

配置远程历史供应器。

您必须给要用于 InTouch HMI 的每个远程历史供应器指定一个名称与网络位置。您可以使用远程 InTouch 历史供应器，也可以使用远程 IndustrialSQL Server 历史供应器。

注意：您无法给 InTouchView 应用程序配置远程历史供应器 。如需有关 InTouchView 应用程序限制的详细信息，请参阅 InTouchView 应用程序。

在本地 InTouch 应用程序被认为是历史供应器的情况下，不必为应用程序去定义它。

如果在应用程序中引用了未定义的历史供应器，则 WindowViewer 会忽略该引用并将一条错误消息写入 Logger。

HistData 实用程序无法从 Historian 供应器中检索历史信息。

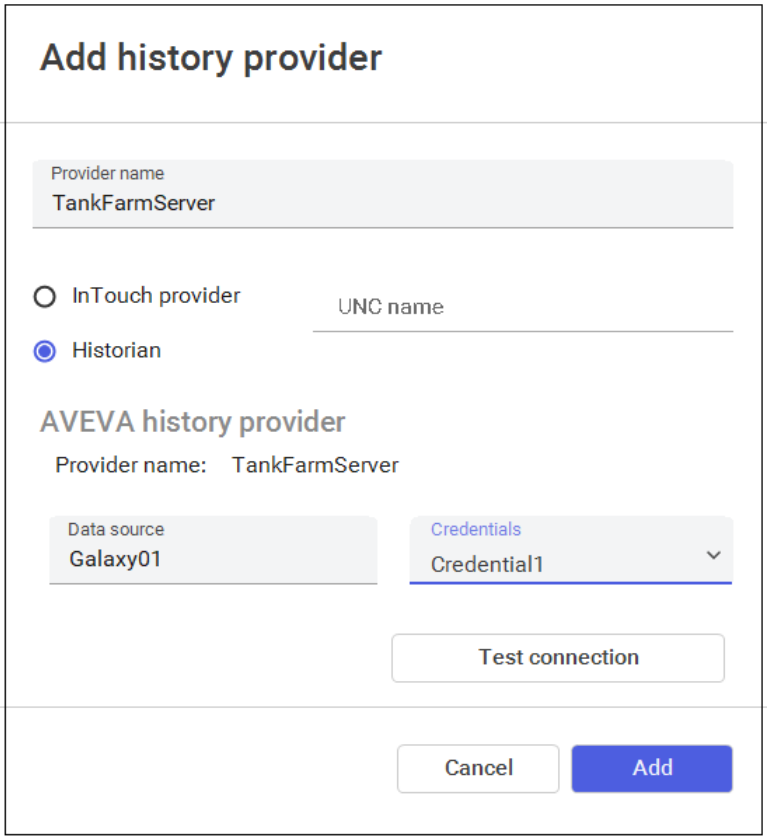
配置历史供应器

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择分布式名称管理器。

   此时出现“分布式名称管理器”配置屏幕，其中包含用于分布式报警和分布式历史的单独选项卡。
3. 在分布式历史选项卡上，选择 + 图标以添加历史供应器，或按 Alt+A。

   此时出现添加历史供应器对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**供应器名**框中，为新的历史供应器输入希望使用的名称。

   供应器名的长度不得超过 16 个字母数字字符。

5. 配置 InTouch 历史供应器

   a. 选择 **InTouch 供应器**。

   b. 在 **UNC 名**框中，输入 InTouch 应用程序目录的 UNC 路径，然后选择**添加**。

   UNC 路径格式为：

   ```
   \\节点名\卷名\目录\
   ```

   如果 UNC 位置受密码保护，必须先使用 Windows 资源管理器来建立节点连接。

6. 配置 AVEVA 历史供应器

   a. 选择 **Historian**。

   b. 在**数据源**框中，输入安装了 Historian 服务器的服务器的节点名。

   c. 从**凭据**下拉列表中，选择用于身份验证的凭据。

注意：对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发》中的使用凭据管理器。

7. 选择**测试连接**，以验证与 Historian 服务器数据库的连接。此时出现一条消息，指出与数据库的连接是否成功。

8. 选择**添加**以关闭对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Historian 服务器节点会出现在历史供应器列表中。

动态配置远程历史供应器

在运行时，您也可以动态配置历史趋势的远程历史供应器，方法是创建一个脚本，在 HTSetPenName() 函数中指定远程历史供应器标记引用。例如：

```
HTSetPenName("HistTrendTag", 1, "HistPrv1.Boiler1");
```

其中，1 指定将绘制指定的远程历史供应器标记的趋势笔。

对于远程历史供应器，运行时历史趋势设置对话框与 .Pen 点域不受支持。

配置分布式历史记录

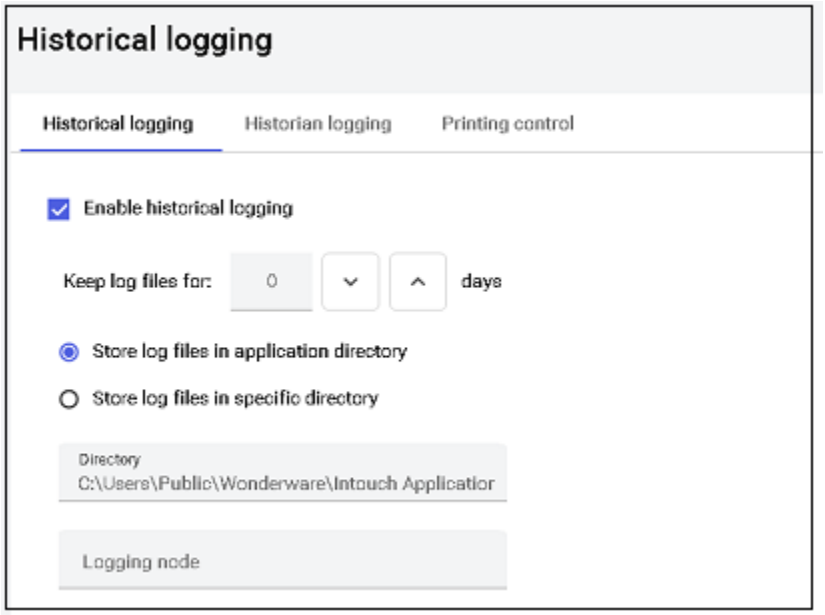
只有一个 InTouch 节点可以写入历史文件。不过可以有多个 InTouch 节点查看该文件。

注意：您无法为 InTouchView 应用程序配置历史记录。如需有关 InTouchView 应用程序限制的详细信息，请参阅 InTouchView 应用程序。

配置分布式历史记录

1. 在文件菜单上，选择配置，然后选择历史记录。

   此时出现历史记录配置屏幕。



2. 选中启用历史记录复选框以打开全局标记记录功能。
3. 选择在指定目录中存储日志文件，然后在输入框中输入日志文件存储位置的路径。

   您必须输入有效的“通用命名惯例”(UNC) 路径。例如，`\\Node\Share\Path`

   如果使用 NAD，必须确保路径指向应用程序文件夹之外的某个文件夹。
4. 在记录节点名框中，输入要记入历史日志文件的节点的名称。

   此设置只允许将此处指定的节点记录到文件中。
5. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：在选择了启用历史记录选项的应用程序分发到 WindowViewer 节点时，该节点会检查此选项以确定是否应该记录它。如果选择了启用历史记录，则可能的设置是：字段等于节点的名称 - 启用记录的字段不等于节点的名称 - 禁用记录。

特殊网络的注意事项

如果使用低速网络，并因此导致 InTouch HMI 启动或保存信息要花费很长时间，则可以在 NAD 客户端上修改 `win.ini` 设置：

```
ViewNadClearNADCopyDirectory=0
ViewNADCopyApplicationOnStartup=1
ViewNADOnApplicationChanged=3（或 4）
ViewNADThreadPriority=2
```

对于 ViewNADOnApplicationChanged 参数，设置为 3 对应于“InTouch 应用程序管理器”中节点属性对话框上的将更改加载到 WindowViewer 选项。设置为 4 对应于提示用户将更改加载到 WindowViewer 选项。这些设置可以让应用程序继续执行，而 NAD 下载则在单独的执行线程上同时进行。

NAD 对应用程序执行更新时，它只是从 NAD 储备库应用程序复制更改过的文件。NAD 不复制用于运行时语言切换功能的 SmartSymbol 设计时字典文件。

针对 NAD 配置 InTouch 应用程序

“网络应用程序开发”或简称 NAD 是一种特殊架构，它兼备基于客户端和基于服务器的架构的优点。NAD 不仅提供自动通知应用程序更改的功能，而且还可以自动将更新的应用程序分发到各个 View 节点。

针对 NAD 配置应用程序时，必须指定 WindowViewer 要将 NAD 储备库应用程序复制到哪个文件夹。

- 如果这是开发节点，则可以输入本地文件夹路径，如 c:\InTouch\NAD。您也可以输入远程网络 UNC 路径，如 \\node\share\path。对于大多数文件存储都在一个中央位置的基于文件服务器的网络，这非常方便。
- 如果这是客户端节点（仅限运行时），则通常使用本地文件夹路径。

我们建议尽量使用本地文件夹，以避免因网络延迟与故障而影响 WindowViewer 的操作。

警示：不要使用根文件夹或指向根文件夹的 UNC 路径名。在复制 NAD 储备库应用程序目录之前，View 节点将删除指定的目标应用程序文件夹中的所有文件与子文件夹。因此，切勿使用 NAD 储备库应用程序文件夹的路径或指向 NAD 储备库应用程序文件夹的 UNC 路径。

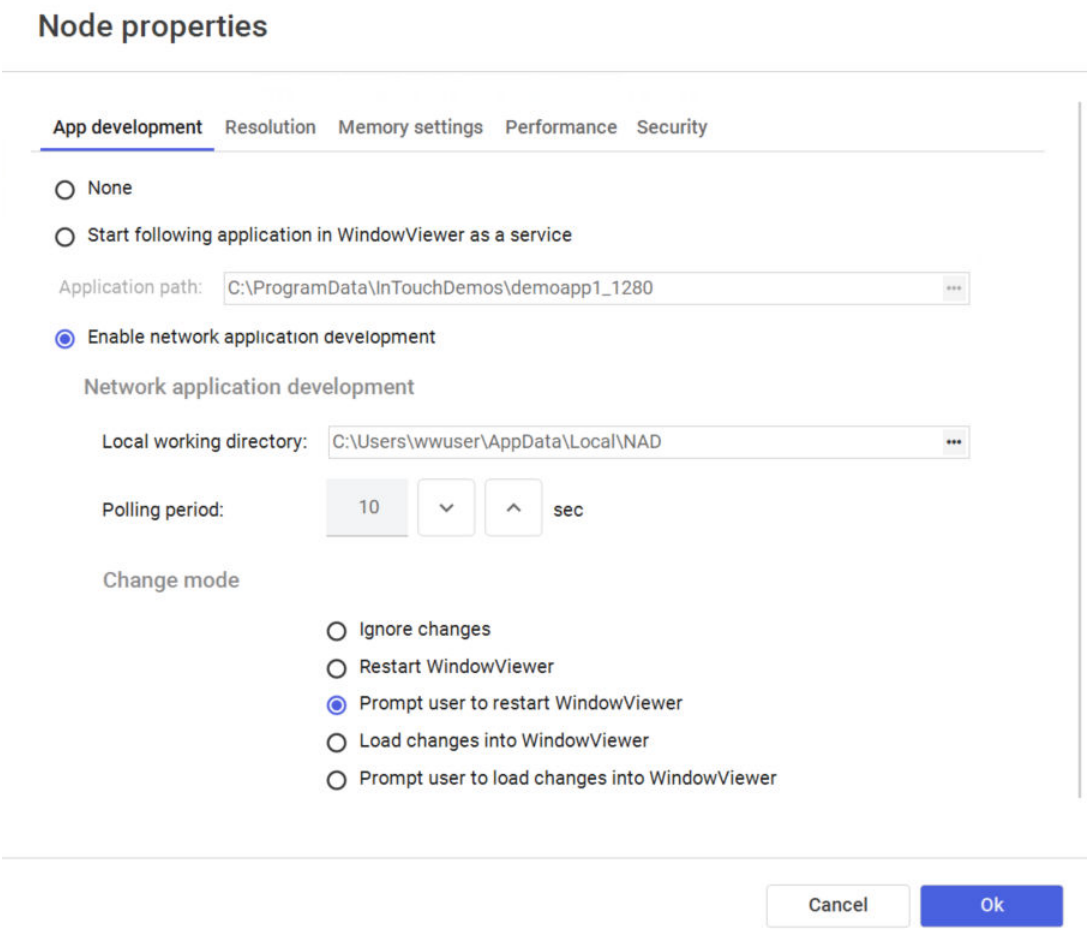
如果不指定文件夹，WindowViewer 会在 WindowViewer 启动的文件夹中自动创建一个名称为 NAD 的本地子文件夹。此 NAD 文件夹应视为临时文件夹，并且除了由 NAD 自身复制的那些文件之外，不得将其它文件保存到在其中。

针对 NAD 配置应用程序

1. 启动“应用程序管理器”。
2. 在工具选项卡上，选择节点属性。

   此时出现节点属性屏幕。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 选择允许网络应用程序开发单选按钮。
4. 在本地工作目录框中，输入希望 WindowViewer 将 NAD 储备库应用程序所复制到的文件夹的路径。
5. 在轮询周期（秒）框中，输入以秒计的间隔，View 节点将按照此间隔来检查开发节点是否有更新。
   - 注意不要将这个值设置得过小。如果 WindowViewer 过于频繁地检查 NAD 储备库应用程序更改，则会干扰它为运行中应用程序提供服务。
6. 在更改模式区域中，选择相应的选项确定在 NAD 储备库应用程序发生更改时 WindowViewer 所要采取的操作。
   - 选择忽略变化，以让 WindowViewer 节点忽略开发节点上所进行的任何更改。
   - 选择重新启动 **WindowViewer**，让 WindowViewer 节点复制更新的 NAD 储备库应用程序（如果配置成如此），然后重新启动自身。
   - 选择提示用户重新启动 **WindowViewer**，以便向操作员显示应用程序已发生更改的消息。操作员可以重新启动 WindowViewer 并更新应用程序，也可以继续使用当前的应用程序。
   - 选择将更改加载到 **WindowViewer** 以便将开发节点上发生的更改动态加载到 WindowViewer 中。对于大规模的更新，这可能会影响性能。

注意：建议您只在应用程序变更很细微且数量很少时才使用将更改加载到 WindowViewer 选项。少量更改的示例包括现有窗口内进行的更改、调整图形工具栏元素的大小、添加新的图形工具栏元素和引

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用替换。如果所进行的更改需要重新启动 WindowViewer（如添加新标记、添加新窗口或更改配置）或如果存在疑问，请改用某个重新启动选项。

- 选择提示用户将更改加载到 **WindowViewer**，以便向操作员显示应用程序已发生更改的消息。此消息提示操作员加载这些更改。

7. 选择确定。

执行 NAD 自动更新

在应用程序开发期间，您可以启动 NAD 自动更新。

运行通知客户端命令时，会设置一个标帜，向所有的远程 View 节点通知 NAD 储备库应用程序已经发生更改。客户端可以根据为每个节点定义的改变模式选项自动开始更新过程。

在 View 节点上首次打开独立应用程序（含有嵌入的工业图形）时，图形可能不显示，并且会在 Logger 中记录错误。为避免出现这种情况，请从 NAD 储备库节点运行通知客户端命令，工业图形将根据改变模式选项加载到 View 节点上。

执行自动更新

1. 在 WindowMaker 中打开应用程序。
2. 在属性菜单上的客户端组中：
   a. 选择立即通知以立即通知客户端。
   b. 选择关闭时通知，以在关闭 WindowMaker 时提醒通知 NAD 客户端。

注意：如果选择关闭时通知选项，每次关闭 WindowMaker 时，它都会验证自上次通知以来是否有任何更改。如果有任何更改，将出现一个对话框，提示“是否要通知 NAD 客户端？”。选择是将通知客户端，选择否将忽略更改。

执行 NAD 手工更新

您可以编写脚本供操作员在他们工作的 View 节点上手工启动 NAD 更新。

要使用 NAD 手工更新应用程序，必须在节点属性对话框中，将改变模式选项设置为忽略变化。如需详细信息，请参阅针对 NAD 配置 InTouch 应用程序。

在脚本中使用以下系统标记与函数执行 NAD 手工更新：

- $ApplicationChanged 系统标记
- $ApplicationVersion 系统标记
- RestartWindowViewer() 函数
- ReloadWindowViewer() 函数

$ApplicationChanged 系统标记

发出信号，指出“网络应用程序开发”(NAD) 架构中的 NAD 储备库应用程序已经更改。

类别

应用程序

用法

```
$ApplicationChanged
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

每次通过选择 WindowMaker 特别菜单上的通知客户端生成更新信号时，此系统标记便更改为 1。当应用程序更新时，$ApplicationChanged 重置为 0。此标记可用于生成一条消息，通知操作员 NAD 储备库应用程序已经发生更改。

您也可以在数据改变脚本中使用 $ApplicationChanged 系统标记，以构建节点更新通知脚本。此脚本可以启动您自己的对话框或停止正在运行的进程。然后您可以使用 ReloadWindowViewer() 函数来启动更新过程。

数据类型

离散（只读）

示例

在数据改变脚本的标记名框中使用以下语句会导致运行脚本的主体。脚本主体可能显示一个窗口，提醒用户重新启动 WindowViewer 以便使更改生效。

```
$ApplicationChanged
```

另请参阅

$ApplicationVersion

$ApplicationVersion 系统标记

包含应用程序的当前版本号。每次发生可以保存或撤消的更改时，此数字都会更改。

类别

应用程序

用法

```
$ApplicationVersion
```

附注

与 $ApplicationVersion 系统标记关联的值设置为 InTouch 应用程序的当前版本。每次应用程序发生可以保存或撤消的更改时，版本都会更改。此标记可用于生成一条消息，通知操作员 NAD 储备库应用程序已经发生更改。

数据类型

实型（只读）

示例

如果用在模拟显示链接中，则此系统标记显示 WindowViewer 中运行的应用程序的当前版本。

```
$ApplicationVersion
```

另请参阅

$ApplicationChanged

RestartWindowViewer() 函数

关闭 WindowViewer，复制更新的 NAD 储备库应用程序（如果配置成如此），然后重新启动 WindowViewer。

类别

系统

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
RestartWindowViewer();
```

附注

在未使用“网络应用程序开发”(NAD) 自动更新函数时，可以使用此函数来更新应用程序。

使用 $ApplicationChanged 系统标记来确定何时发生 NAD 更新。

您使用通知客户端命令来启动 NAD 更新。不过，操作员可能需要延迟更新时间。您可以将此函数用于某个按钮动作脚本，以便操作员可以在方便时重新启动 WindowViewer。

您也可以使用 ReloadWindowViewer() 函数，它可以在不关闭 WindowViewer 的情况下更新 View 节点。

另请参阅

$ApplicationChanged，ReloadWindowViewer()

ReloadWindowViewer() 函数

使用更新的 NAD 储备库应用程序来动态更新 WindowViewer，而无需中断服务。

类别

系统

语法

```
ReloadWindowViewer();
```

可供用户控制 WindowViewer 的重新加载。

附注

在未使用“网络应用程序开发”(NAD) 自动更新函数时，可以使用此函数来更新应用程序。

使用 $ApplicationChanged 系统标记来确定何时发生 NAD 更新。

您使用通知客户端命令来启动 NAD 更新。不过，操作员可能需要延迟更新时间。您可以将此函数用于按钮动作脚本，以便操作员可以在方便时在 WindowViewer 中重新加载应用程序。

另请参阅

$ApplicationChanged

应用程序编辑锁定

为防止多个开发人员同时编辑某个应用程序，在编辑会话中 WindowMaker 锁定该应用程序。如果试图打开锁定的应用程序，则会显示一条错误消息。当前编辑该应用程序的节点的名称包含在消息中。

如果 WindowMaker 在加载应用程序时异常关闭，则可能不会删除 appedit.lok 文件。您可以通过从应用程序目录中删除 appedit.lok 文件来手工解除锁定。

在 NAD 更新期间对应用程序更改

WindowViewer 节点更新应用程序时，它会尽可能地在复制过程中保留 NAD 储备库应用程序的属性（只读、系统、隐藏，等等）。

WindowViewer 也会复制 NAD 储备库应用程序的所有文件与子文件夹，以下文件除外：*.WVW、*.DAT、*.LGH、*.IDX、*.LOG、*.LOK、*.FSM、*.STG、*.DBK、*.CBK、*.HBK、*.KBK、*.LBK、*.NBK、*.OBK、

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.TBK、.WBK、*.XBK、*.$$$、RETENTIV.X、RETENTIV.D、RETENTIV.A、RETENTIV.S、RETENTIV.H、RETENTIV.T、SSD_、WM.INI、DB.INI、LINKDEFS.INI、TBOX.INI、GROUP.DEF 以及 ITOCX.CFG。

注意：WindowViewer 以递归方式删除目标应用程序文件夹中的所有文件与子文件夹（运行时语言切换功能所需的那些除外）。此文件夹应视作临时文件夹。不能存放其它文件。

开始更新之后，NAD 客户端会创建一个列表，列出客户端应用程序目录中出现的本地文件与子目录。NAD 客户端一边在 NAD 储备库文件列表中查找更新，一边从本地列表中删除每个 NAD 储备库文件对应的客户端文件。本地列表中剩余的项目是废弃的文件与子目录，应该从应用程序中删除掉。

下载的所有文件将复制到临时子目录 NAD_Temp 中。仅在重试次数限制范围内成功复制所有的新文件以及更新的文件之后，才会将这些文件从 NAD_Temp 复制到应用程序目录。如果 NAD 客户端不得不放弃更新，正在运行的应用程序并不会因为只引入部分新文件或更新的文件而损坏。

如果下载所有的新文件与更新的文件之后无法联系到 NAD 储备库应用程序，则通过从 NAD_Temp 中复制更新并删除废弃的文件，仍可以完成更新。这确保不会仅仅因为丢失连接而无法确认文件是否存在于 NAD 储备库应用程序中便将它们删除掉。

NAD 可以检测出在下载应用程序期间是否对 NAD 储备库应用程序进行过其它的更改。如果发生这种情况，NAD 将放弃下载应用程序。如果在最近的更新之后运行通知客户端命令，NAD 将在下一个轮询周期自动开始下载最新的应用程序。否则，它将等到发出下一条通知客户端命令之后再开始下载应用程序。

在运行时调整应用程序分辨率

通过使用“动态分辨率转换”（Dynamic Resolution Conversion，简称 DRC），可以使创建的分布式应用程序能够在不同的屏幕分辨率下运行。

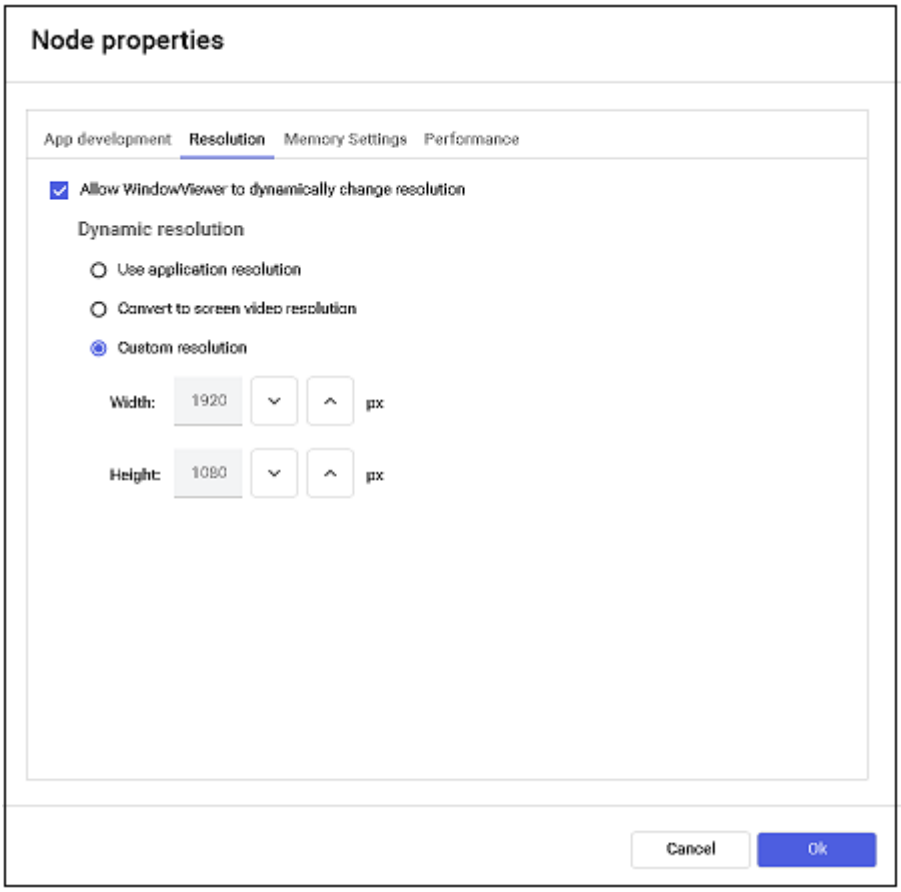
每个 View 节点都可以适当地调整应用程序，包括调整到自定义的分辨率。这种比例调整可以在 WindowViewer 编译应用程序时进行，并不要求使用 WindowViewer。每个 View 节点都可以使用不同的 DRC 设置，因此每个 View 节点都必须配置自己的设置。

警示：如果不使用 DRC 来调整应用程序，则仅当节点的屏幕分辨率与应用程序开发节点的屏幕分辨率完全相同时，WindowViewer 才会运行应用程序。如果分辨率不同，WindowViewer 会提示操作员运行 WindowMaker，以便按该节点的分辨率转换应用程序。如果已设置 NAD 储备库应用程序目录的 UNC 路径，请务必小心，因为这只会修改原始应用程序。

针对 DRC 配置应用程序

1. 启动“应用程序管理器”。
2. 在工具菜单上，选择节点属性。此时会出现“节点属性”对话框。
3. 选择分辨率选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 如果希望 WindowViewer 在本地缩放 NAD 储备库应用程序，请选择**“允许 WindowViewer 动态改变分辨率”复选**框。
5. 在**动态分辨率**区域中，**选择**以下操作之一：
   - 如果希望 WindowViewer 以**开发时**的分辨率运行应用程序而忽略节点的分辨率，**请选择使用应用程序分辨率**。例如，如果应用程序在 800x600 下**开发**，而节点的分辨率是 1024x768，则 WindowViewer 不**动态缩**放应用程序。相反，应用程序分辨率保持为 800x600。
   - 如果希望 WindowViewer 以节点分辨率运行应用程序而忽略**开发时**应用程序的分辨率，**请选择转换为屏幕视频分辨率**。例如，如果节点以 800x600 运行，而应用程序是在 1280x1024 下**开发**的，则 WindowViewer 动态调整应用程序，使之适合节点的 800x600 分辨率。
     - 如果目标分辨率与创建应用程序时的屏幕分辨率不同，那么 WindowViewer 将从原始应用程序分辨率缩放为当前的屏幕分辨率。原始应用程序分辨率是创建应用程序时的屏幕分辨率，与目标分辨率设置无关。例如，如果开发应用程序时的分辨率为 1920x1080，目标分辨率为 1280x1024，而查看节点以 800x600 的分辨率来运行应用程序，那么 WindowViewer 将动态缩放应用程序使用原始应用程序分辨率 1920x1080。如需详细信息，请参阅原始应用程序分辨率。
   - 如果希望 WindowViewer 以**宽度 (X)** 与**高度 (Y)**（必须是整数值）框中指定的特定分辨率运行应用程序，**请选择自定义分辨率**。应用程序的分辨率和节点的分辨率都会被忽略。例如，如果**宽度 (X)** 与**高度 (Y)** 分别设置为 512 和 384，则应用程序将动态缩放，以适合节点屏幕上的 512x384 像素区。
     - 如果目标分辨率与创建应用程序时的屏幕分辨率不同，那么 WindowViewer 将从原始应用程序分辨率缩放为当前的屏幕分辨率。原始应用程序分辨率是创建应用程序时的屏幕分辨率，与目标分辨率设置无关。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 选择确定。

锁定应用程序分辨率

您可以通过配置 WindowMaker 属性来锁定 InTouch 应用程序窗口的大小。这样可让您在不调整窗口和图形大小的情况下转换应用程序分辨率。

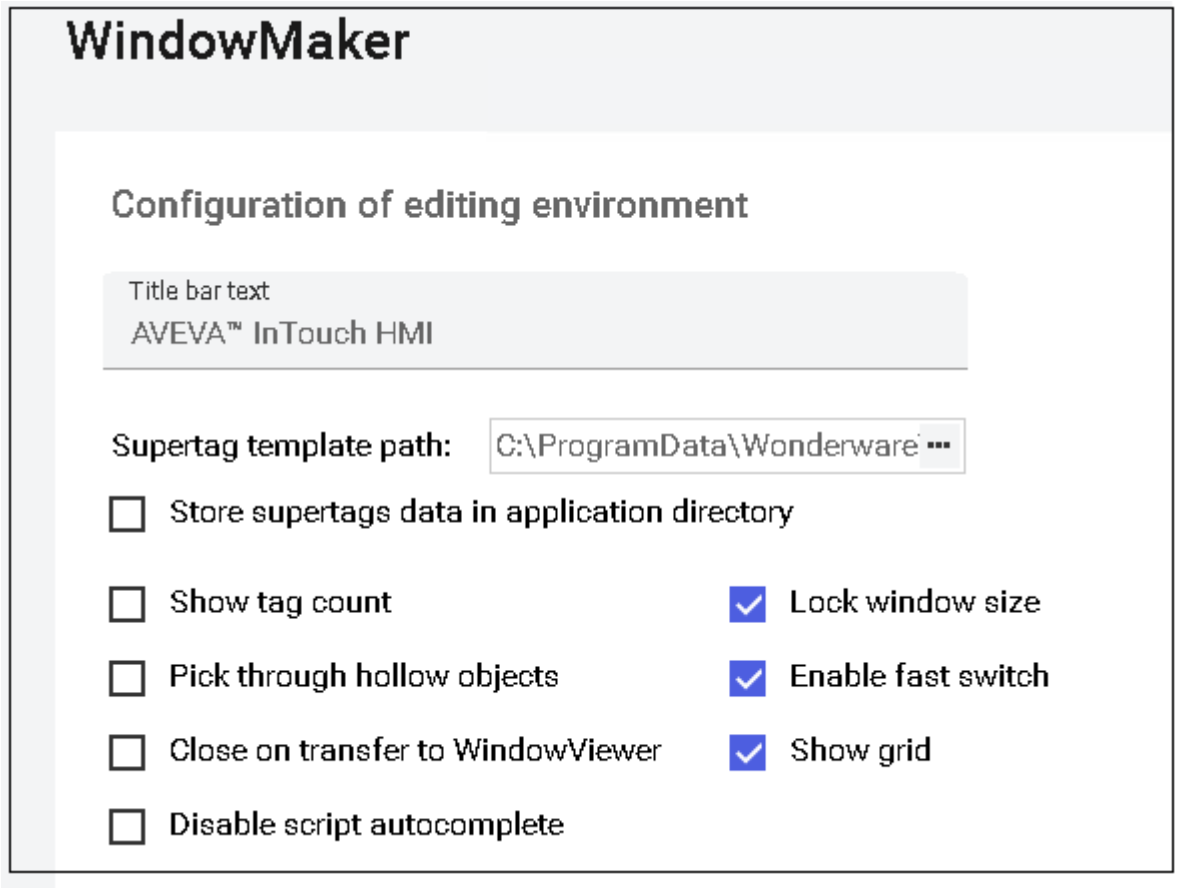
如果选择该选项，那么您下次在分辨率不同的计算机上打开应用程序时，系统会提示您指定是否要将应用程序的分辨率转换成新的分辨率，而不调整窗口和图片的大小。

可以从 WindowMaker 内部或从应用程序管理器锁定应用程序分辨率。

从 WindowMaker 锁定应用程序分辨率

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择 **WindowMaker**。

   此时出现 **WindowMaker 配置**屏幕。



3. 选中锁定窗口大小复选框。缺省条件下，该复选框处于未选中的状态。
4. 选择保存。

从应用程序管理器锁定应用程序分辨率

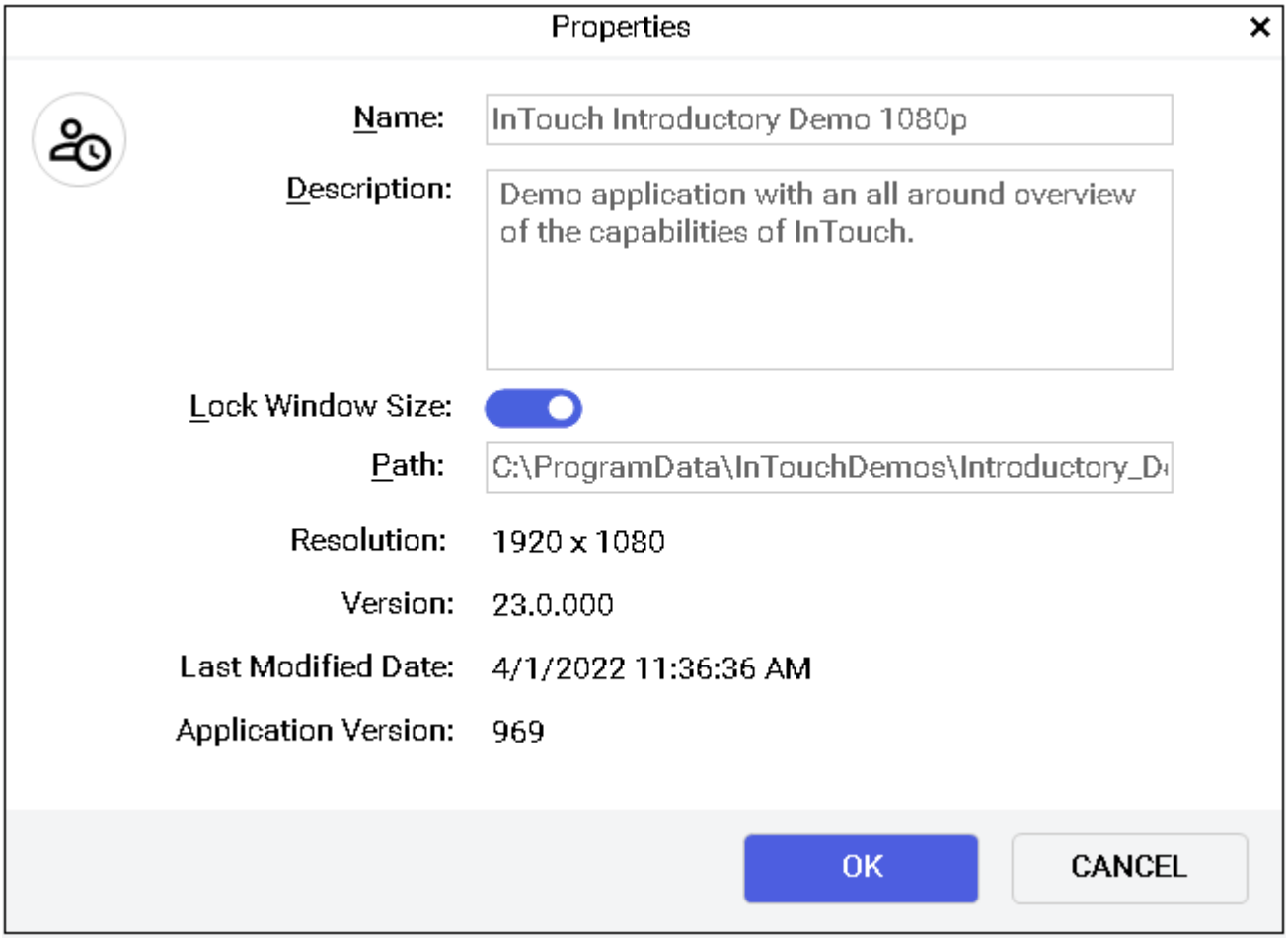
1. 打开“应用程序管理器”。
2. 选择要配置的应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 选择菜单栏上的文件，然后选择属性。

   此时出现属性对话框。

4. 选择锁定窗口大小开关。缺省条件下，该复选框处于未选中的状态。



5. 选择确定。

将应用程序发布到远程节点

通过使用“应用程序发布器”，可以创建压缩的自解压数据包文件，包含在另一台计算机上安装 InTouch 应用程序所需的所有相关文件与安装过程。您使用“应用程序发布器”发布独立的 InTouch 应用程序。您可以使用 System Platform IDE 发布托管的 InTouch 应用程序。

有两个选项可供发布应用程序：

- **仅限运行时。**在仅限运行时的软件包中，包含运行应用程序而不是编辑应用程序所需的文件。
- **设计时与运行时。**在设计时与运行时的软件包中，包含编辑与运行应用程序所需的所有文件。由于某些运行时文件（如编译的 *.wvw 文件）可以从设计时文件重新创建，因此它们被排除在外。

您可以将已发布的应用程序发到 Web 服务器上，以便可以下载并安装它们。对于发布的应用程序，会显示以下软件包信息：

- 软件包描述

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 发布者姓名
- 发布的文件名（可执行文件）
- 应用程序分辨率

例如：

描述	乳品加工应用程序
发布者	Navin Johnson
文件名	Dairy.exe / Video Resolution…(1024x768)

描述	乳品加工应用程序
发布者	Navin Johnson
文件名	Dairy_2.exe / Video Resolution…(800x600)

发布的文件的内容

下表列出对于发布的所有独立的 InTouch 应用程序，都会包含的文件夹、文件以及会排除的文件。

包含的文件夹	包含的文件	排除的文件
主应用程序文件夹	全部	备份文件。这些文件有 .?bk 文件扩展名。
	使用这些扩展名的文件： .win、.dat、.lgh、.idx、.log、.fsm、.stg、.$$$	不在“特殊目录”列表中的子文件夹
	retentiv.x retentiv.d retentiv.a retentiv..s（两个点） retentiv.h wm.ini db.ini linkdefs.ini tbox.ini group.def itocx.cfg	appedit.lok 文件，表示在 WindowMaker 中打开了应用程序。
	名称为 SSD_*.xml 形式的任何文件。	编译的窗口文件，使用 .wvw 文件扩展名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

包含的文件夹	包含的文件	排除的文件
Dictionary 子文件夹，用于运行时语言切换功能	使用 .xml 扩展名的所有文件。	
符号子文件夹	所有文件与子文件夹。	
	wiz.ini 文件，如果安装了向导。	
	向导可执行文件的副本。	
	.dll 文件、 .wdo 文件、 .wdf 文件	

对于仅限运行时的应用程序，会排除文件名为 SSD_*.xml 的所有文件。

发布独立的 InTouch 应用程序

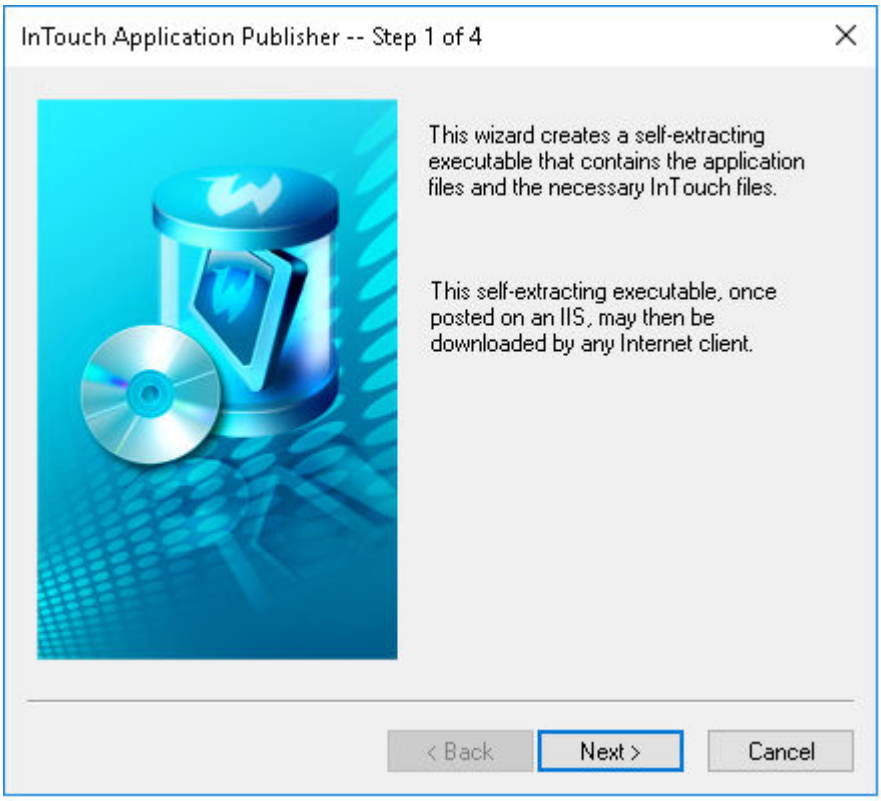
使用“应用程序发布器”发布独立的 InTouch 应用程序。如果希望发布的应用程序在特定的屏幕分辨率下运行，在发布之前，请将原始应用程序设置成该分辨率。要发布托管的 InTouch 应用程序，请使用 System Platform IDE。

发布独立的 InTouch 应用程序

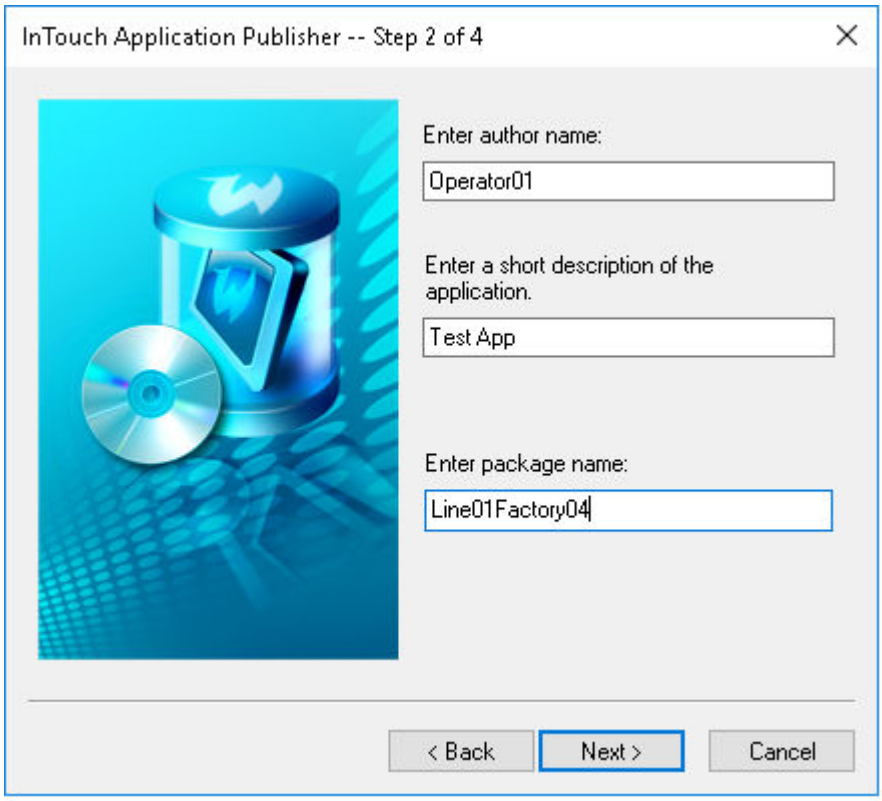
1. 启动“应用程序发布器”。
   a. 打开 WindowMaker。
   b. 展开工具窗格，然后展开应用程序。
   c. 双击应用程序发布器。

      此时出现 InTouch 应用程序发布器 – 第一步/共四步对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

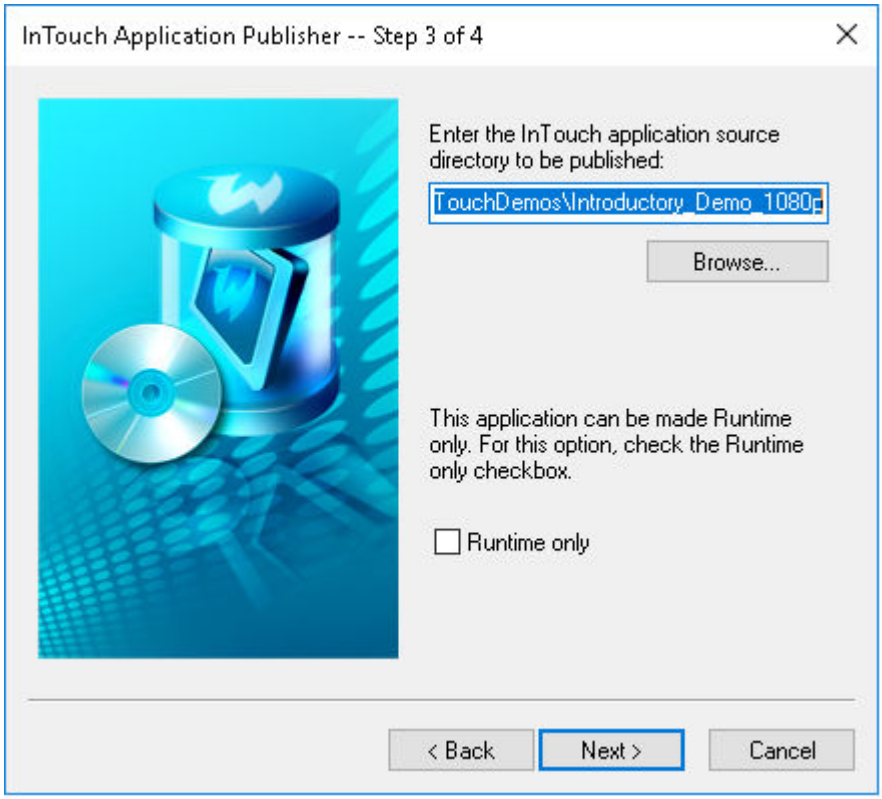


2. 选择下一步。此时出现 InTouch 应用程序发布器 – 第二步/共四步对话框。



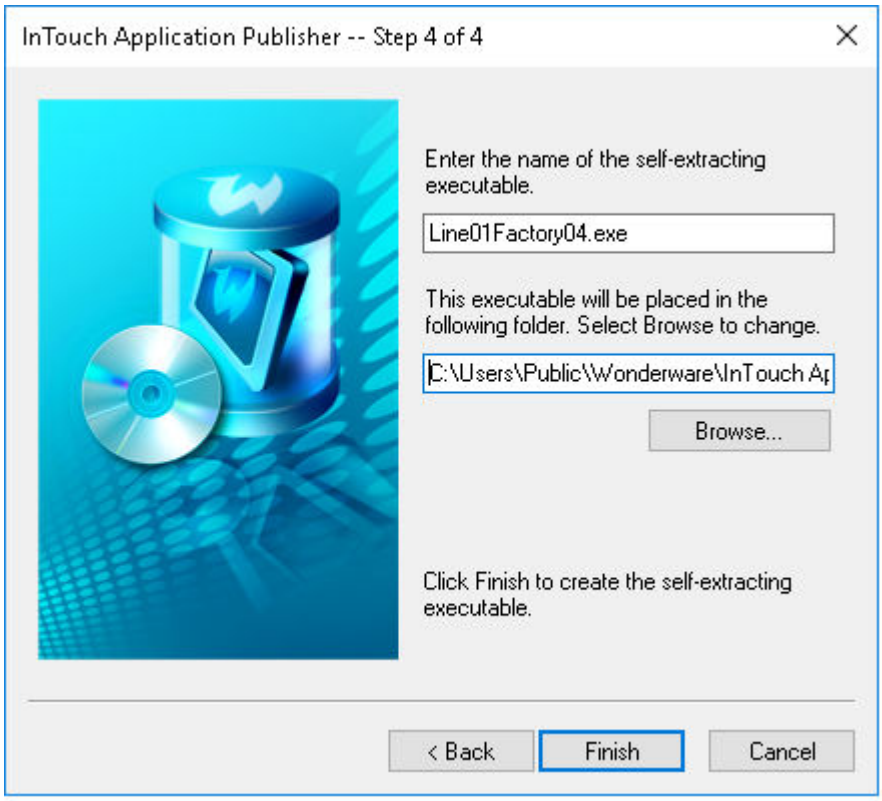
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1. 配置软件包详细信息。
   - 在输入作者名框中，输入应用程序联系人的姓名。姓名限制为 256 个字符。
   - 在描述框中，输入应用程序的描述。限制为 256 个字符。
   - 在软件包名框中，为发布的应用程序软件包输入唯一的名称。限制为 32 个字符。如果使用现有软件包的名称，则会覆盖掉现有的软件包。
2. 选择下一步。此时出现 InTouch 应用程序发布器 – 第三步/共四步对话框。



3. 配置关于发布的详细信息。
   - 在文本框中，输入 InTouch 应用程序文件夹的路径。缺省路径是 WindowMaker 应用程序文件夹。
   - 选择仅限运行时复选框，以便在发布的文件中排除 WindowMaker 开发文件。
4. 选择下一步。此时出现 InTouch 应用程序发布器 – 第四步/共四步对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 配置应用程序软件包可执行文件的详细信息。
   - 在第一个框中，验证可执行文件名是否正确。缺省条件下，可执行文件名与软件包名相同。
   - 在第二个框中，输入要保存可执行文件的文件夹的路径；或是选择浏览以选择不同的文件夹。缺省条件下，可执行文件保存在当前的临时文件夹中。
6. 选择完成。

发布托管的 InTouch 应用程序

您可以发布托管的 InTouch 应用程序。发布的 InTouch 应用程序与 InTouchViewApp 模板不再关联。

发布的应用程序无法在 IDE 中编辑，也无法导入到另一个 InTouchViewApp 模板中。换句话说，您无法使用 IDE 管理或重新发布它。

发布的 InTouch 应用程序仍然可以通过任何内嵌的工业图形与 Galaxy 通讯。例如，您可以将数据写回 Galaxy，或以可视化方式显示 Galaxy 数据。

您可以使用基本的 InTouch 操作编辑工业图形，如复制、剪切、粘帖、创建副本、移动、调整大小、翻转、旋转、及配置 InTouch 动画链接。

不过，工业图形无法修改，新的工业图形也无法嵌入 InTouch 应用程序。只有托管的 InTouch 应用程序才允许使用这些操作。

您可以在不支持 ArchestrA 处理要求的环境中执行这些操作。例如，在远程工厂或小型网络中。

发布托管的 InTouch 应用程序

您可以从与托管的 InTouch 应用程序关联的 InTouchViewApp 对象中发布该应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

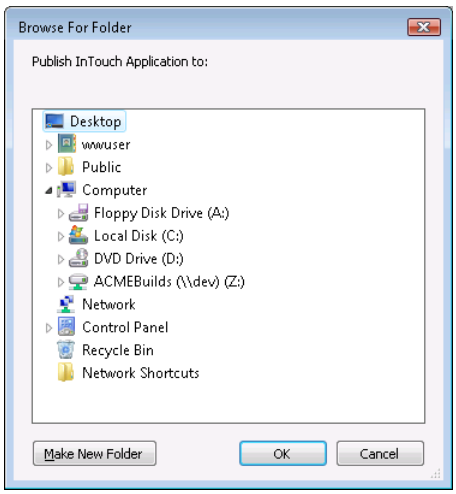
导出的内容由包含对象信息的文件夹与托管的 InTouch 应用程序组成。

这与导出 InTouchViewApp 对象本身并不相同。如需详细信息，请参阅导入与导出 InTouchViewApp 对象。

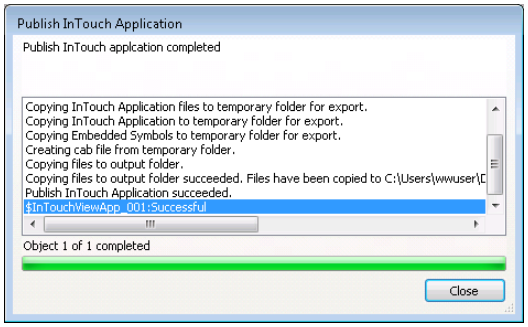
无法将发布的 InTouch 应用程序重新导入 InTouchViewApp 对象中。无法编辑发布的 InTouch 应用程序。

发布托管的 InTouch 应用程序

1. 打开 System Platform IDE。
2. 找到包含要发布的托管 InTouch 应用程序的 InTouchViewApp 对象。
3. 使用鼠标右键单击对象，然后选择 **Publish Intouch Application**（发布 **InTouch** 应用程序）。此时出现浏览文件夹对话框。



4. 指定要将 InTouch 应用程序发布到的文件夹。执行以下任意操作：
   - 浏览到现有的文件夹。
   - 选择 **Make New Folder**（新建文件夹）以创建一个新的文件夹或文件夹结构。
5. 选择确定。此时出现 **Publish InTouch Application**（发布 InTouch 应用程序）进度对话框。



6. 发布完成时，选择 **Close**（关闭）。此时会在所选文件夹中创建一个包含新发布的 InTouch 应用程序的目录。您现在可以将它复制到任何运行时节点上。

将应用程序发布到 Insight

您可以使用 Insight Publisher 将应用程序发布到 Insight 网站。您可以使用“应用程序管理器”或 WindowMaker。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将应用程序发布到 Insight

1. 打开“应用程序管理器”。
2. 在工具菜单上的工具组中，选择 **Insight Manager**。

   此时出现 **Insight Publisher** 窗口。

在 InTouch WindowMaker 中，可以从应用程序下面的工具窗格使用 **AVEVA Insight Publisher**。

您可以选择以下选项之一并按照屏幕说明继续。

- **发布** - 从现有 InTouch 应用程序创建新的 Insight 数据源。
- **导入** - 从 OPC、MQTT 或 OI 服务器导入 Excel 电子表格列表项目。
- **授权** - 创建数据源。

如需详细信息，请参阅 Historian 文档。

注意：您需要有 Insight 账户才能发布该应用程序。

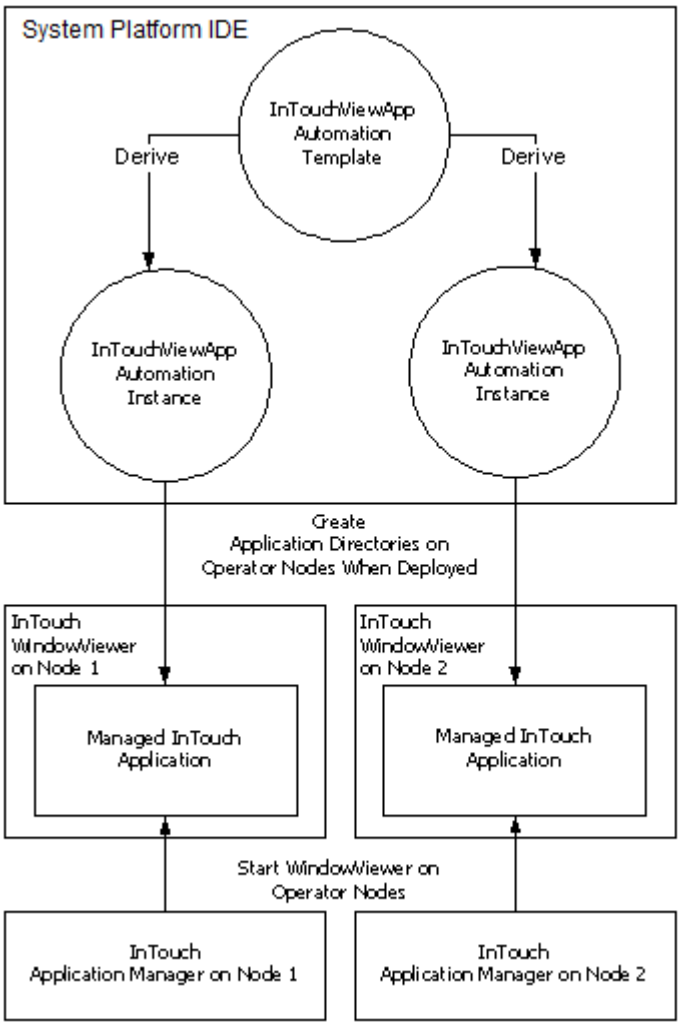
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 18 运行时使用托管的 InTouch 应用程序

通过将 InTouchViewApp 实例部署到远程节点，可以在这些节点上运行托管的 InTouch 应用程序。

您也可以将 InTouch 应用程序的更改与包含的工业图形部署到这些节点上；并选择对于每个节点，是接受还是拒绝更改。

下图显示托管的 InTouch 应用程序如何部署到运行时节点。



部署托管的 InTouch 应用程序

您可以从 System Platform IDE 将托管的 InTouch 应用程序部署到本地节点或远程节点。部署应用程序之后，可以在远程节点上的 WindowViewer 中运行它。

注意：WindowViewer 一次只能运行一个应用程序。如果在本地节点上部署平台，则 Galaxy 的已配置样式将优先于任何其它独立或托管应用程序中的任何已配置样式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

第一次部署 InTouchViewApp 对象

第一次部署 InTouchViewApp 对象时，会将关联的 InTouch 应用程序复制到存放该对象的平台节点上。此节点称为操作员节点。

部署托管的 InTouch 应用程序

1. 打开 System Platform IDE。
2. 选择 InTouchViewApp 实例，这是您要部署其托管的 InTouch 应用程序的实例。
3. 在 **Object**（对象）菜单上，选择 **Deploy**（部署）。此时出现部署对话框。
4. 选择确定。此时完整的 InTouch 应用程序会复制到操作员节点上。

部署对托管的 InTouch 应用程序所作的更改

注意：覆盖 WindowViewer 缺省配置选项尚未在非英语操作系统上进行测试。Web 客户端不支持此选项。

您可以按以下方式更改托管的 InTouch 应用程序：

- 更改托管的 InTouch 应用程序中使用的工业图形的引用、内容或大小。
- 通过从 InTouchViewApp 模板中打开 WindowMaker 来更改托管的 InTouch 应用程序本身。

在这两种情况下保存更改时，这些更改都会从更新的模板传播到衍生的实例中。这些更改通过“待处理的更改”图标标识。

这些更改不立即反映到正在运行的 WindowViewer 会话中。每个节点的操作员可以选择接受或拒绝更改。如需详细信息，请参阅在操作员节点上接受新的应用程序版本。

部署对托管的 InTouch 应用程序所作的更改

1. 打开 System Platform IDE。
2. 选择 InTouchViewApp 实例，这是您要部署对其托管的 InTouch 应用程序所作的更改的实例。
3. 在 **Object**（对象）菜单上，选择 **Deploy**（部署）。此时出现部署对话框。
4. 选择确定。此时这些更改复制到操作员节点上。

启动托管的 InTouch 应用程序

从操作员节点中，可以启动“应用程序管理器”，并选择要运行的托管的 InTouch 应用程序。

您也可以在“应用程序管理器”中设置分辨率，以便使用不同的分辨率来运行托管的 InTouch 应用程序。

启动托管的 InTouch 应用程序

1. 在部署了 InTouchViewApp 对象的节点上，启动“InTouch 应用程序管理器”。
2. 在应用程序列表中，选择要在 WindowViewer 中运行的托管的 InTouch 应用程序。
3. 选择 WindowViewer 图标。一小段时间之后，应用程序在 WindowViewer 中启动。

设置动态分辨率转换设置

1. 打开 InTouch 应用程序管理器。
2. 在工具菜单上，选择节点属性。此时会出现节点属性对话框。
3. 选择分辨率选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 选择允许 WindowViewer 动态改变分辨率复选框。

   如果未选择允许 WindowViewer 动态改变分辨率，则托管的应用程序将使用开发时的分辨率来运行。

5. 配置希望如何运行应用程序。执行以下任意操作：
   - 选择使用应用程序分辨率以便使用开发时的分辨率来运行托管的应用程序。
   - 选择转换为屏幕视频分辨率以转换托管的应用程序，使它可以在屏幕分辨率下运行。
   - 选择自定义分辨率将托管的应用程序转换为指定的分辨率。
6. 选择确定。

在部署托管应用程序时控制 WindowViewer 重新启动等待时段

当部署需要重新启动的托管应用程序时，您可以控制 WindowViewer 重新启动前的延迟。实施延迟是为了确保报警子系统正常运行。

WindowViewer 重新启动延迟由位于本地 C:\Windows 文件夹的 win.ini 文件中的下列参数控制。所有参数值均以秒为单位。

- `ViewManagedRestartWaitPeriod=0`

  此参数控制 WindowViewer 重新启动等待时段。缺省值为 0，或者无等待时段。

- `ViewNADShutdownWaitPeriod=30`

  对于使用网络应用程序开发 (NAD) 托管的应用程序，此参数控制 NAD 关闭前的延迟。缺省值为 30 秒。

- `ViewNADRestartWaitPeriod=90`

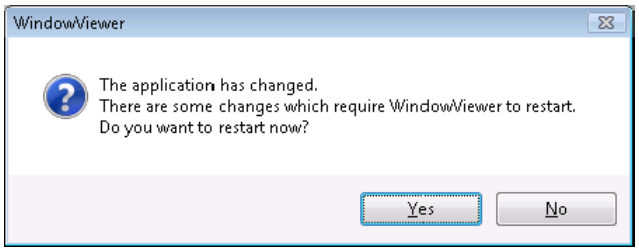
  对于使用 NAD 托管的应用程序，此参数控制 NAD 重新启动前的延迟。缺省值为 90 秒。

在操作员节点上接受新的应用程序版本

如果托管的 InTouch 应用程序发生了更改，并且部署与它关联的 InTouchViewApp 实例，则可以在运行时节点上选择接受还是拒绝更改。

此时出现一条消息，提示您是否要接受对托管的 InTouch 应用程序所作的更改：

- 从“InTouch 应用程序管理器”中启动 WindowViewer 时。
- WindowViewer 正在运行时。



根据更改的性质，可能提示您重新启动 WindowViewer 应用程序，或只是重新加载它。您也可以设置 WindowViewer 处理应用程序更改的行为，例如：

- WindowViewer 如何接受或拒绝这些更改。
- 存在待处理的更改时，WindowViewer 多长时间提醒您重新加载或重新启动 WindowViewer 一次。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

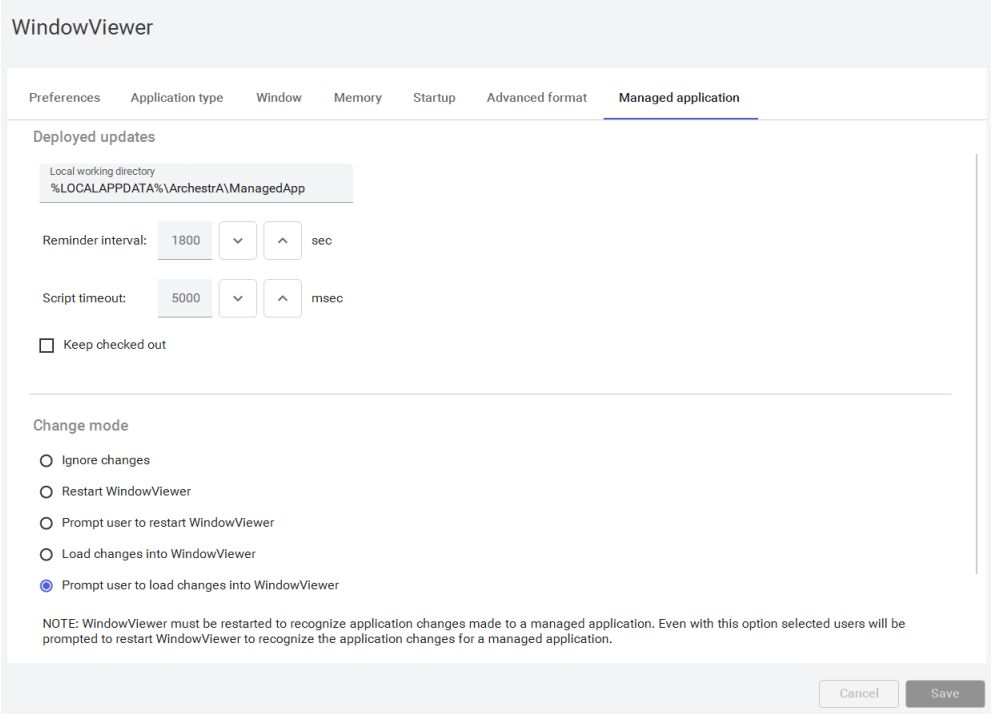
在操作员节点上接受新的应用程序版本

- 选择是。此时对托管的 InTouch 应用程序所作的更改会复制到操作员节点上，WindowViewer 重新启动或重新加载。

设置 WindowViewer 处理应用程序更改的行为

1. 在 WindowMaker 中打开托管的 InTouch 应用程序。
2. 在文件菜单上，指向配置，然后选择 **WindowViewer**。

   此时出现 **WindowViewer** 配置屏幕。
3. 选择托管的应用程序选项卡。



4. 在改变模式区域中，配置部署更改时 WindowViewer 如何响应。执行以下任意操作：
   - 选择忽略变化让 WindowViewer 忽略任何部署的更改。您可以手动配置 RestartWindowViewer() 与 ReloadWindowViewer() 脚本函数，以便根据 ApplicationChanged 系统标记来接受更改。
   - 选择重新启动 **WindowViewer** 让 WindowViewer 自动重新启动。
   - 选择提示用户重新启动 **WindowViewer** 让 WindowViewer 提示用户重新启动 WindowViewer。
   - 选择将更改加载到 **WindowViewer** 让 WindowViewer 自动加载更改。
   - 选择提示用户将更改加载到 **WindowViewer** 让 WindowViewer 提示用户将更改加载到 WindowViewer。

注意：如果选择将更改加载到.. 或 提示用户将更改加载到.. 选项，必须重新启动 WindowViewer 才能识别对托管的应用程序的更改。即时选择这些选项，系统也将提示您重新启动 WindowViewer。

1. 在提醒间隔（秒）框中，输入多久（以秒计）提醒用户一次将更改加载到 WindowViewer 或重新启动 WindowViewer。此选项仅在设置适当的更改模式之后才可用。将间隔设置为 0 时不会再提醒用户。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 选择确定。

设置 WindowViewer 的缺省行为

1. 在 WindowMaker 中打开托管的 InTouch 应用程序。
2. 在文件菜单上，指向配置，然后选择 **WindowViewer**。此时出现 **WindowViewer** 配置屏幕。
3. 选择托管的应用程序选项卡。
4. 选择恢复缺省值。此时这些设置重置为缺省值。
5. 选择确定。

运行嵌入的工业图形中的 ArchestrA 脚本

在 WindowViewer 中运行托管的 InTouch 应用程序时，与元素关联的任何 ArchestrA 脚本或符号脚本自身都可以按预期运行。

不过，符号中包含的有些脚本可能会运行很长时间，使您不能与其它 InTouch 元素交互。

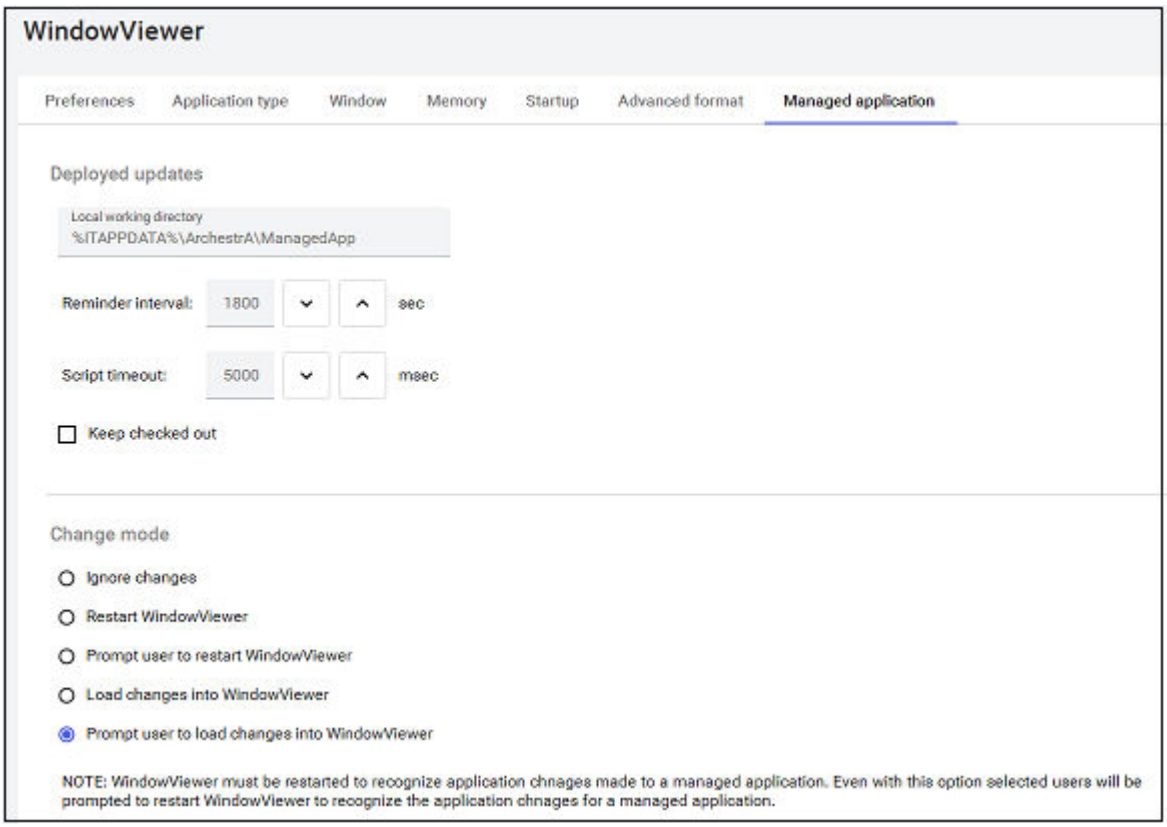
为防止出现这种情况，您可以设置一个适合托管的 InTouch 应用程序中所有脚本的脚本超时。脚本超时会让脚本停止执行，并将控制权返回给操作员。

缺省条件下，脚本 5 秒后超时。

设置脚本超时

1. 在 WindowMaker 中打开托管的 InTouch 应用程序。
2. 在文件菜单上，指向配置，然后选择 **WindowViewer**。此时出现 **WindowViewer** 配置屏幕。
3. 选择托管的应用程序选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**脚本超时**（毫秒）框中，输入以毫秒为单位的值。
5. 选择**确定**。

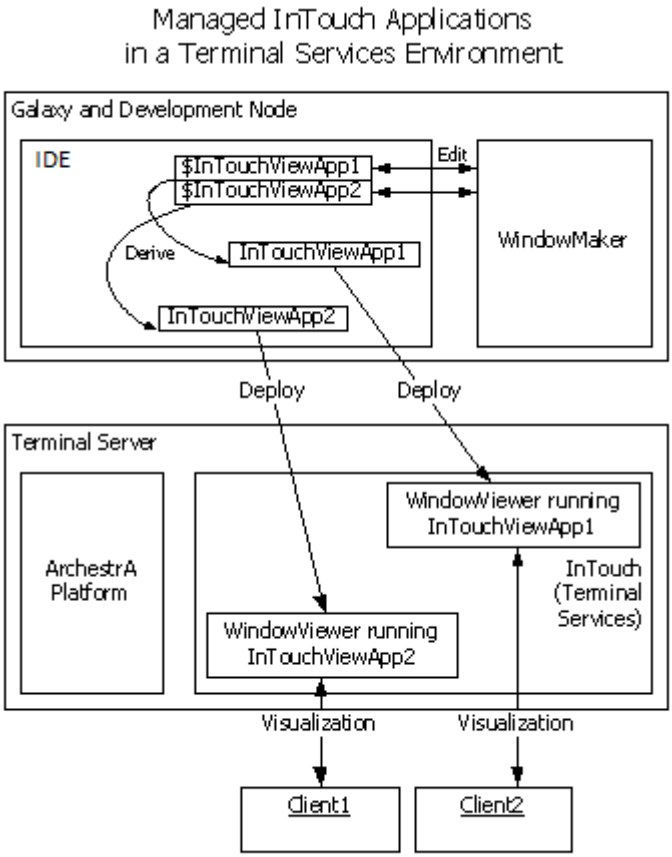
在终端服务环境中部署 InTouchViewApp 对象

您可以在“终端服务”环境中运行托管的 InTouch 应用程序。此架构的主要优势是可以同时在一台计算机上运行多个 InTouch 应用程序。

在终端服务环境中运行托管的 InTouch 应用程序

- 使用 InTouch Terminal Services Edition。
- 如果终端服务器节点上有多个 InTouchViewApplication 实例，则部署每个 InTouchViewApp 实例及其自身的 ViewEngine 宿主。
- 在每个托管的 InTouch 应用程序自身的终端服务客户端会话中运行该应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



注意：在终端服务器节点上只需要部署一个 InTouchViewApp 对象即可使应用程序可用于多个终端会话客户端。不需要在终端服务器节点上为每个将使用该应用程序的客户端都部署一个单独的 InTouchViewApp 对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 19 在 AVEVA Edge 中使用 InTouch HMI 应用程序

AVEVA Edge 为用户提供了可用于部署和复用 InTouch HMI 应用程序的选项。使用“应用程序管理器”，您可以将应用程序下载到 AVEVA Edge 或将它们上传到 CONNECT。

已下载的 zip 文件是 InTouch HMI 应用程序的压缩版本，由于它们的文件大小很低，因此适用于部署到边缘设备和容器。也可以将同一个 zip 上传到 CONNECT，在那里可以将它下载到其它设备。

注意：如果未启用与 AVEVA Edge 相关的选项，请启动 WindowMaker。

下载 AVEVA Edge Zip 文件

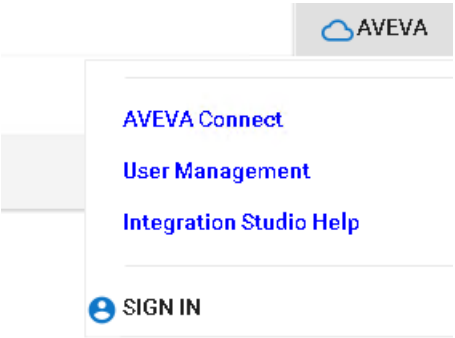
1. 选择应用程序。
2. 选择 图标。

   压缩的 (.zip) 文件将准备好可供下载。
3. 保存该文件。
4. 使用 AVEVA Edge Management 门户并添加新设备。在项目配置选项的下面上传该 zip 文件。

   如需进一步的部署和配对说明，请参阅“AVEVA Edge 帮助”。
5. 用户可以使用 InTouch HMI Web 客户端来验证应用程序是否在正确地运行。如需有关 Web 客户端的详细信息，请参阅*在 Web 浏览器中查看应用程序图形*。

上传 AVEVA Edge Zip 文件

1. 选择“应用程序管理器”窗口最右侧的 **AVEVA** 按钮。

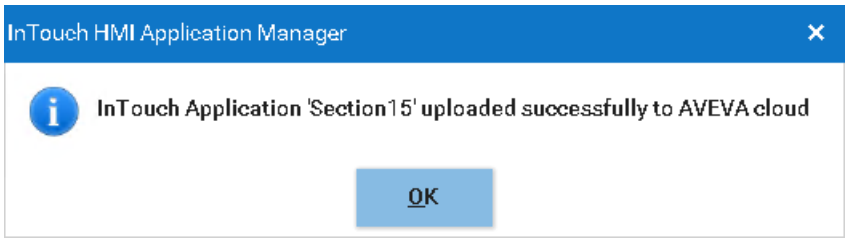


2. 选择登录。
3. 输入 CONNECT 凭证。
4. 登录后，选择要上传的应用程序。

   作为可选项，您可以更改“驱动器”，但是无法在驱动器之间移动应用程序。
5. 选择 。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在将应用程序上传到 CONNECT 之后，即会显示一条成功上传的消息。

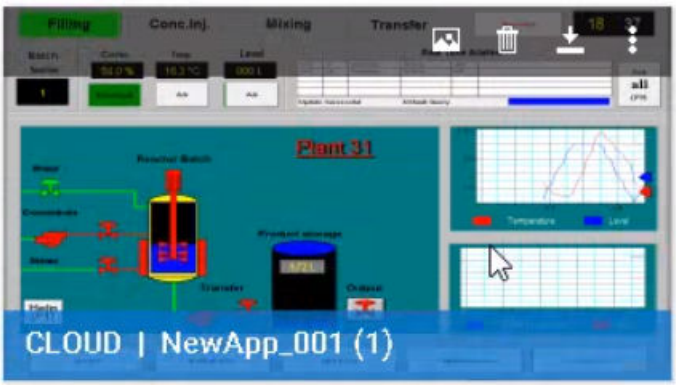


已上传的应用程序显示在应用程序列表中，带有 *Cloud | <应用程序名称>* 前缀。

云应用程序具有以下选项：

- 略图
- 删除
- 下载

所有其它选项都处于禁用状态。



将应用程序上传到“共享驱动器”之后，所有用户即可对其进行访问。随后，用户可以从云存储下载和删除该应用程序。

为边缘设备配置用户

必须在“应用程序管理器”中配置那些需要从边缘设备访问 InTouch Web 客户端的用户。Web 客户端的相应用户组中必须存在相同的用户。您可以将用户列表导出或导入到 csv 文件中。文件的模板如下所示：

```
用户名，密码，IsWrite
TestUser,TestPass#1,1
```

新增用户

1. 选择 。

   此时出现“用户信息”对话框。

2. 输入用户名和密码。
3. 密码必须遵守以下规则：

   长度至少为 6 个字符

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1 个小写字符

1 个大写字符

1 个数字

1 个特殊字符

4. 选择访问类型。
5. 选择保存。
6. 选择 + 可添加新用户。
7. 选择 🗑 可删除用户。
8. 配置完所有用户后，选择 导出已配置的列表。
9. 在 AVEVA Edge Management 中，在“用户配置”选项的下面于“添加设备”程序期间上传该列表。

   对边缘设备进行配置和配对之后，即会在边缘设备上创建用户的列表。这些用户将能够访问 Web 客户端并查看应用程序图形。

导入现有用户列表

如果您具有要包含的现有用户的列表，请按照所需的格式准备一个 .csv 文件。

1. 选择 并提供 .csv 文件的路径。
2. 文件中的项目将包含在用户的列表中。
3. 使用 AVEVA Edge Management 门户并添加新设备。在用户配置选项的下面上传该 .csv 文件。

   如需进一步的部署和配对说明，请参阅“AVEVA Edge 帮助”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Operate

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 20 在运行时查看应用程序

您可以使用 WindowViewer 运行 InTouch 应用程序。专为在 Application Server 环境中使用而设计的应用程序称为 InTouchView 应用程序。您可以在任何支持 HTML5 的 Web 浏览器中使用 InTouch Web 客户端查看工业图形。这些应用程序在 WindowViewer 中运行，但由 Application Server 提供大多数 HMI 功能。

关于在运行时查看应用程序

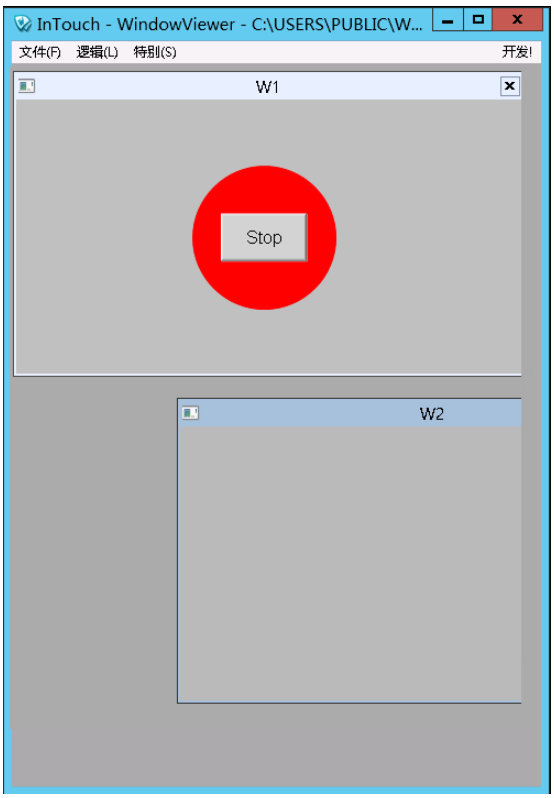
您可以使用 WindowViewer 运行 InTouch 应用程序。专为在 Application Server 环境中使用而设计的应用程序称为 InTouchView 应用程序。您可以在任何支持 HTML5 的 Web 浏览器中使用 InTouch Web 客户端查看工业图形。这些应用程序在 WindowViewer 中运行，但由 Application Server 提供大多数 HMI 功能。

在运行时以不同的目标分辨率大小查看应用程序

如果您指定的应用程序目标分辨率与屏幕分辨率不同，那么 WindowViewer 会以指定的目标分辨率显示应用程序。只有在画出目标分辨率大小轮廓的画布边界内开发的窗口、窗口控件和图形会在运行时显示。在运行时会自动调整目标分辨率大小，以适应 WindowViewer 的菜单和标题栏控件。

嵌入的控件的纵横比在运行时保持不变。

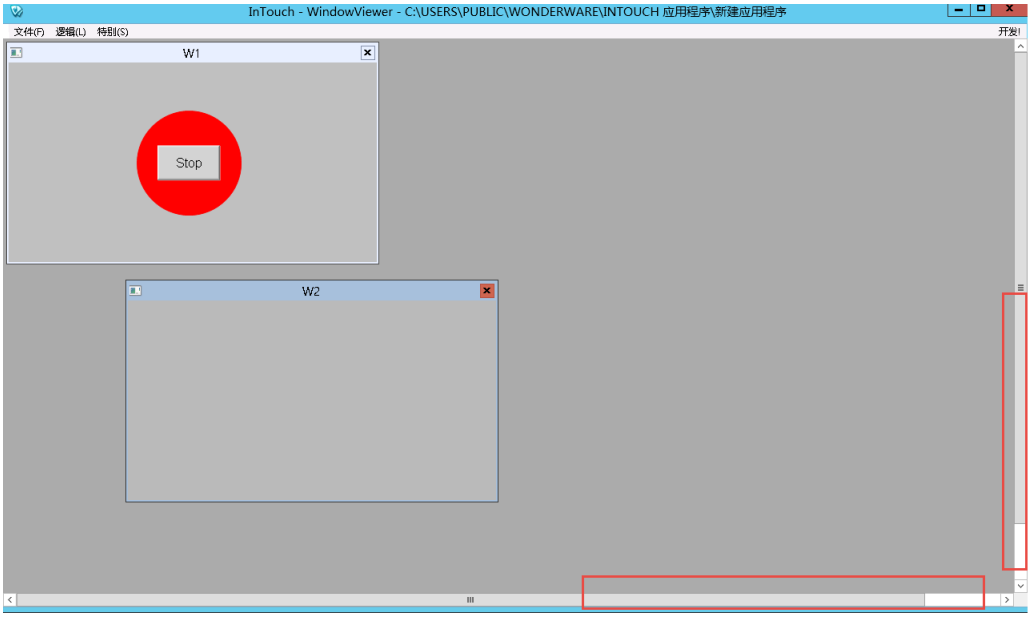
例如：



注意：如果指定的目标分辨率小于屏幕分辨率，那么运行时窗口的宽度和高度无法调整为超过指定的目标分辨率。最大化 WindowViewer 窗口会将其放大到最大目标分辨率。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果指定的目标分辨率大于屏幕分辨率，在运行时将显示垂直和水平滚动条。窗口将相应滚动。ShowSymbol 动画和 ShowGraphic 脚本中的弹出窗口和弹出图形将不会滚动。



错误消息和弹出对话框将显示在采用目标分辨率的应用程序中心，而不是屏幕中心。这同样适用于键盘和数字小键盘。

关于 InTouch Web 客户端

要使用鼠标手势缩放：通过 InTouch Web 客户端功能，您可以在任何支持 HTML5 的 Web 浏览器上查看 InTouch HMI 应用程序内使用的工业图形。通过内置的 Web 服务器提供的 Web 浏览器访问权，您无需使用 Microsoft Windows® Server 的远程桌面协议 (RDP) 或 Internet 信息服务 (IIS)，即可从任何 Microsoft Windows 客户端或服务器操作系统访问应用程序图形。您可以在 Web 浏览器中查看独立应用程序和托管应用程序的应用程序图形。独立应用程序窗口必须转换为工业图形，才能在 Web 客户端上查看。

如需有关 InTouch Web 客户端的详细信息，请参阅*在 Web 浏览器中查看应用程序图形*。Web 客户端作为 System Platform 安装的一部分安装。如需有关配置 Web 客户端的信息，请参阅《System Platform 安装指南》。

原始应用程序分辨率

原始应用程序分辨率是创建应用程序时的屏幕分辨率，与目标分辨率设置无关。

只有在以下条件下，才能更新原始应用程序分辨率：

- 应用程序创建时
- 对应用程序进行转换时

如果稍后将该应用程序切换回屏幕分辨率，那么当目前屏幕分辨率与原始应用程序分辨率不同时，会发生应用程序转换。否则，不会发生转换。如果创建应用程序时使用目标分辨率，将会影响“动态分辨率转换”的行为。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在运行时使用键盘、鼠标和触摸手势进行平移和缩放

框架窗口允许您在运行时平移和缩放工业图形。此功能通过 WindowMaker 中的 **InteractionMode** 属性启用。

在运行时缩放

在运行时可以放大和缩小框架内容。请确保框架已启用正确的平移和缩放属性。缩放范围在 100% 到 5000% 之间。

编辑符号的 **ZoomPercent** 属性可以更改符号或元素在运行时的可见性。例如，向可见性动画添加以下内容后，你就可以根据缩放百分比级别动态更改符号。

```
ZoomPercent => 200
```

注意：您可以在运行时写入此属性。

为符号设置 **ZoomPercent** 后，该符号将在可见区域的中心缩放至设置的百分比。

为某个符号的元素设置 **ZoomPercent** 后，该符号将缩放至设置的百分比，但以其元素为中心。

以下脚本是为元素设置的 **ZoomPercent** 的示例：

```
TextBox1.ZoomPercent = 500
```

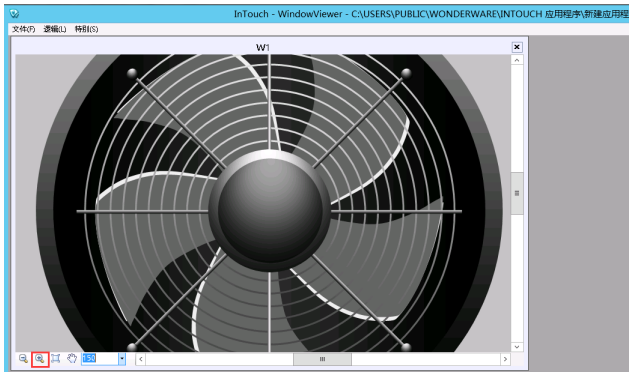
使用鼠标手势缩放

执行以下操作：

- 按“Ctrl”键并向上滚动鼠标滚轮可放大框架内容。内容将从鼠标指针的当前位置放大。

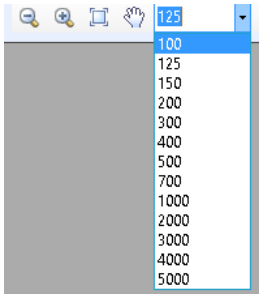
  注意：如果鼠标指针在框架外部，则无法放大框架内容。

- 向下滚动鼠标滚轮可缩小框架内容。
- 从“平移与缩放”控件工具栏选择放大和缩小图标。然后必须左键单击框架的内容才能放大。



- 在框架内容上双击鼠标左键可将缩放级别恢复为 100%。
- 使用“缩放级别”组合框选择预定义的缩放级别。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



- 从“平移与缩放”控件工具栏中选择“橡皮圈缩放”图标可选择要放大的特定区域。

使用键盘手势缩放

执行以下操作：

- 同时按“Ctrl”和“+”键可放大
- 同时按“Ctrl”和“-”键可缩小

使用触摸手势缩放

执行以下操作：

- 将两个手指放在屏幕上，展开手指可放大
- 将两个手指放在屏幕上，合并手指可缩小
- 双击可将缩放级别恢复为 100%

缩放限制

运行时缩放功能有以下限制：

- Windows 公共控件和客户端控件的缩放级别只能在 500% 以内。这些控件包括：
  - 自定义控件：单选按钮组、复选框、编辑框、组合框、日历、日期时间选取器和列表框
  - 嵌入的报警客户端和趋势客户端控件
  - 第三方客户端控件
- Windows 控件可覆盖鼠标、键盘和触摸输入
- InTouch 不会覆盖 Windows 控件中的自定义字体
- 在多笔趋势中，缩放符号时不会缩放光标读取数据。

在运行时平移

配置适用的符号属性以在运行时启用框架内的平移功能。可以使用鼠标和触摸手势在运行时平移。不支持使用键盘手势进行平移。在运行时，图形上的缩放级别必须大于 100% 才能平移。

使用鼠标手势进行平移

执行以下任一操作：

- 按住中间鼠标滚轮。

  此时显示平移手形图。
- 从“平移与缩放”控制工具栏中选择平移手形图。按住鼠标左键并移动平移手形图标来平移显示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

显示将平移，直到释放鼠标按键为止。

使用触摸手势进行平移

1. 将一个手指按在框架内容上。
2. 根据需要在屏幕上移动手指进行平移。

注意：对于这两种平移方法，水平和垂直滚动条将根据平移方向进行调整。

使用触摸手势同时平移和缩放

1. 将两个手指放在屏幕上，将手指向下、向左和向右移动来调整缩放内容的中心。
2. 使用一个手指在框架内容上平移。
3. 将第二个手指放在框架内容上可同时缩放。

平移限制

以下限制适用：

- 不支持使用键盘手势进行平移。
- Windows 控件可覆盖鼠标、键盘和触摸输入。因此，在 Windows 控件区域上方可能禁用平移。

对触摸手势的动画支持

无论缩放级别为何，为触摸支持配置的所有动作脚本都应该以相同方式执行功能。所有交互动画和可视化动画的行为在框架内容缩放时与框架内容在标准视图中时相同。使用触摸时，交互动画将正常执行功能。

注意：**ShowSymbol** 动画或 **ShowSymbol** 脚本函数显示的工业图形弹出窗口在缺省情况下将启用平移和缩放。但是，您不能禁用此配置。

下表列出了通常为触摸支持配置的动作脚本：

动作脚本	触发的触摸
鼠标左键按下时	触摸按下
鼠标左键按下期间	触摸按下并滑动
鼠标左键放开时	触摸放开
双击鼠标左键时	双击
单击鼠标右键时	触摸按下并按住
鼠标右键放开时	触摸按下并适当按住，放开时执行
双击鼠标右键时	不支持
鼠标右键按下期间	触摸按下并适当按住，按住时滑动一点
鼠标中键单击时	不支持
鼠标中键按下期间	不支持
鼠标中键放开时	不支持

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

鼠标中键双击时	不支持

注意：在动画区域上使用触摸交互时，动画将优先于平移动作，平移动作将被忽略。如果一个手指保持触摸交互，任何后续触摸点都将被忽略。

要在这种情形下启用平移，请在“平移与缩放”工具栏中选择平移图标。

触摸手势限制

对于运行时平移与缩放，某些限制适用于触摸手势功能。不支持下面的功能：

- 缩放 Windows 公共控件的字体

注意：此外，具有嵌入式 Windows 控件的符号与不带控件的符号使用不同的缩放机制。具有嵌入式控件的符号的最大缩放限制为 500%，而不带控件的符号最多可放大 5000%。不带控件的符号的优势是缩放更平滑。使用带嵌入式控件的符号的额外限制是，当缩放正在进行时，控件将闪烁。当使用触摸手势进行缩放时，这尤其明显。

对框架窗口使用 ShowGraphic() 函数

您可以在运行时运行 ShowGraphic() 脚本函数来更改与框架窗口关联的符号。请按以下示例的方式来指定框架窗口名称和关联的符号：

```
Dim graphicInfo as aaGraphic.GraphicInfo;
graphicInfo.Identity = "InTouch:FrameWindow01";
graphicInfo.GraphicName = "Symbol_002";
ShowGraphic( graphicInfo );
```

无论该框架窗口的设计时配置如何，“Symbol_002”都将在运行时显示在“FrameWindow01”中。您可以使用 ShowGraphic 脚本托管同一窗口中的不同符号。

注意：对于 InTouch HMI 2020 R2 SP1 P01 以下的版本，仅当 Y 值超过 200 时才会发生明显的位置变化。如果 Y 值低于 200，则图形的位置没有变化或只有非常轻微的偏移，因此很难进行观察和验证。对于 InTouch HMI 2023 以上的版本，仅当 Y 值超过 300 时才会发生明显的位置变化。如果 Y 值低于 300，则图形的位置没有变化或只有非常轻微的偏移，因此很难进行观察和验证。

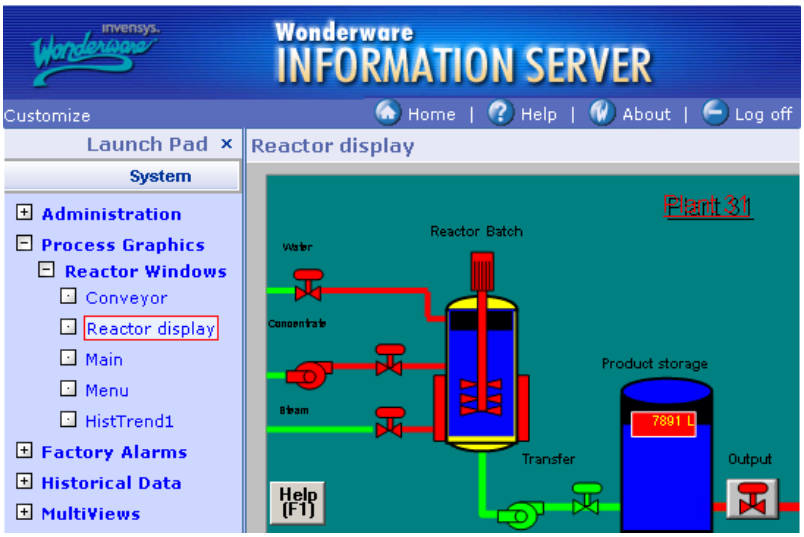
限制：

关闭和缓存窗口时，仅会对当前显示在框架窗口中的符号进行图形缓存。不会缓存已替换的符号。

在 Internet 上运行 InTouch 窗口

Information Server 是可以在 Web 或企业内部网中聚集与呈现工厂生产数据的 Web 门户应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您可以按照以下方式使用 InTouch 与 Information Server：

- 过程可视化

  您可以将 InTouch 应用程序发布到 Information Server 门户，以呈现生产过程及通过 Web 浏览器实施控制。

- 数据交互

  您可以使用 Information Server 门户读取和写回 InTouch 标记值。这可以实现与工厂过程的交互，而不必使用 InTouch 客户端。

- 报警显示

  您可以使用 Information Server 门户显示 InTouch 实时与历史报警数据。

- 历史数据显示

  您可以使用 Information Server 门户显示 Historian 数据库中保存的 InTouch 历史数据。

- Table Weaver 显示

  您可以使用 InTouch 创建 Table Weaver 内容单元的显示。

如需有关详细信息，请参阅 Information Server 文档。

将窗口设置为在运行时显示

“起始”窗口是用户从图标或菜单命令直接启动 WindowViewer 时，出现在 WindowViewer 中的窗口。

如果使用运行时快速切换来启动 WindowViewer，则不会出现起始窗口。

通过在脚本中使用 ShowHome() 函数，可以在运行时随时显示起始窗口。

设置起始窗口

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择 **WindowViewer**。此时出现 **WindowViewer 配置**屏幕。
3. 单击启动选项卡。
4. 选择要在 WindowViewer 启动时打开的一个或多个窗口。

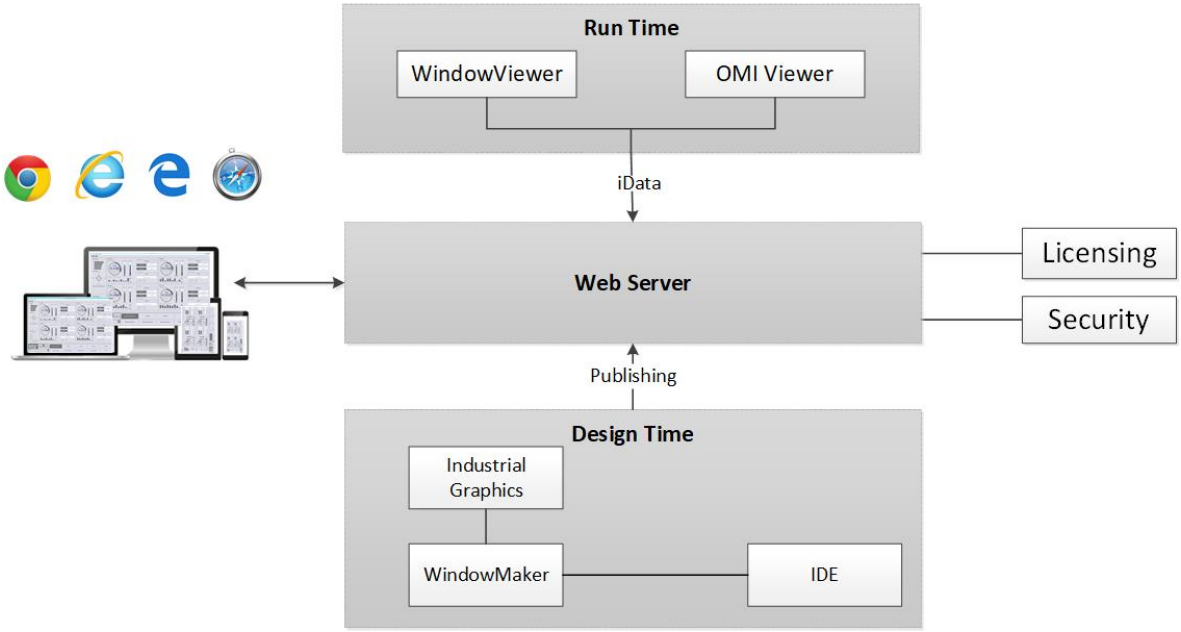
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5. 单击确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 21 在 Web 客户端中查看应用程序图形

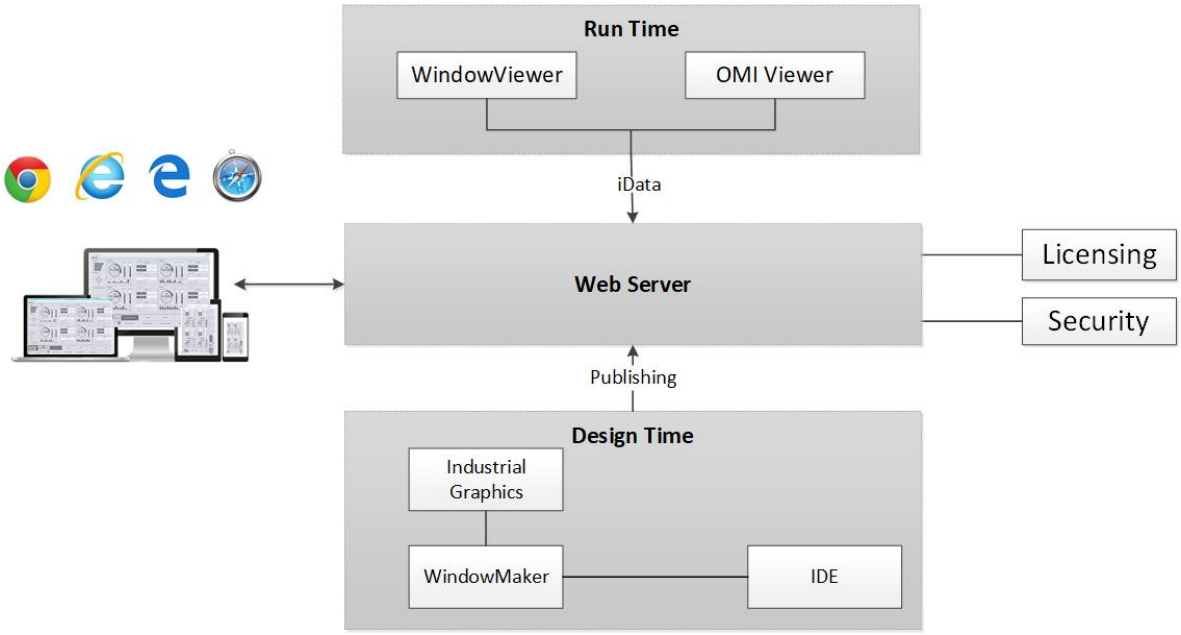
Web 客户端允许您在任何支持 HTML5 的 Web 浏览器上查看 AVEVA InTouch HMI 应用程序和 AVEVA OMI ViewApp。Web 客户端也可在移动平台上使用。内置的 Web 服务器提供了从任意 Microsoft Windows 客户端或服务器操作系统通过 Web 浏览器访问应用程序的权限，无需使用 Microsoft Windows® Server 的远程桌面协议 (RDP) 或 Internet Information Services (IIS)。



使用 Web 客户端

Web 客户端允许您在任何支持 HTML5 的 Web 浏览器上查看 AVEVA InTouch HMI 应用程序和 AVEVA OMI ViewApp。Web 客户端也可在移动平台上使用。内置的 Web 服务器提供了从任意 Microsoft Windows 客户端或服务器操作系统通过 Web 浏览器访问应用程序的权限，无需使用 Microsoft Windows® Server 的远程桌面协议 (RDP) 或 Internet Information Services (IIS)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



设计 InTouch Web 客户端应用程序

设计 InTouch Web 客户端应用程序包括以下几个阶段：

配置（可选）

1. 配置安全协议。
   a. 使用 System Platform 配置器连接到系统管理服务器 - 在远程或本地服务器之间进行选择。
2. 配置 AVEVA Identity Manager。
3. 配置反向代理服务器。

设计

1. 创建和/或编辑应用程序。
2. 标识将包含您希望在 Web 客户端上显示的所有图形的根文件夹。
3. 标识并设置“Web 客户端根”文件夹。
4. 标识并设置 Web 客户端主页符号。
5. 快速切换到 InTouch WindowViewer。
   - 要访问 InTouch 标记，需要运行 WindowViewer，但对于 Application Server 对象属性引用则不需要。
6. 使用 Web 客户端快速切换按钮或启动浏览器，并导航到 http://localhost/InTouch。

迭代开发

1. 主页符号显示在网页上。
2. 使用导航叠加导航到符号/窗口。
3. 在 WindowMaker 中进行任何与图形相关的更改，浏览器上会自动刷新这些更改。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

仅当进行以下图形更改或重新启动 WindowViewer 后，对“质量和状态样式”、“元素样式”和“格式样式”的更改才会传播。

- 更新并保存符号的内容
- 创建、导入或删除符号
- 将符号移动到不同的工具包文件夹
- 更新根文件夹或主页符号分配

4. 您可以继续构建应用程序并在浏览器中测试输出。

部署

1. 将用户分配到其中一个与 Web 客户端相关的用户组（本地或域）或选择匿名访问。
2. 发布或部署应用程序。
3. 在 InTouch WindowViewer 中运行应用程序。WindowViewer 必须运行才能从 InTouch 标记接收实时数据。

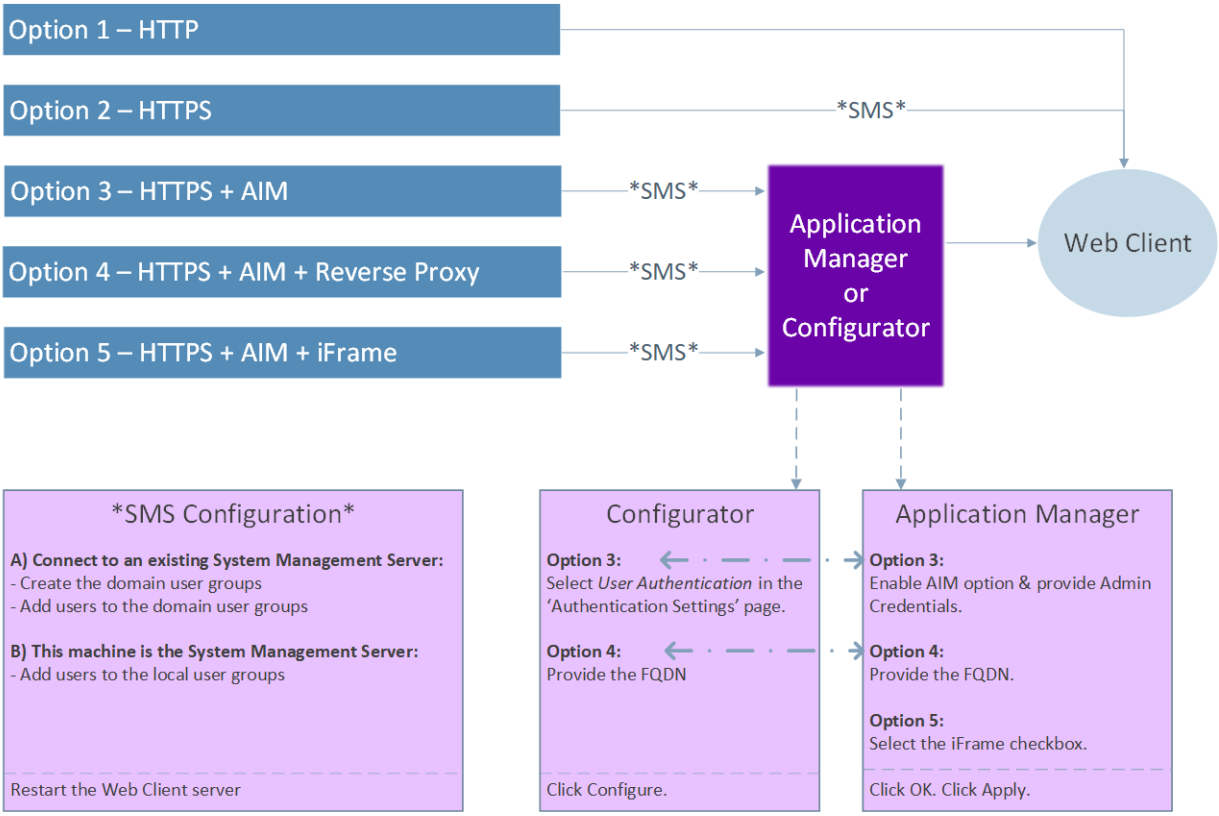
   发布或部署应用程序后，将可以从 Intranet 上的任意计算机访问该应用程序。

   您可以指向 http://<IP 地址>/InTouch 或 http://<节点名>/InTouch，其中 <IP 地址> 或 <节点名> 对应于发布或部署应用程序的计算机。

保护 InTouch Web 客户端访问安全

通过配置系统管理服务器 (SMS)，可以安全地将 InTouch Web 客户端与 https 协议结合使用。此外，在配置 SMS 之后，可以将 InTouch Web 客户端与 AVEVA Identity Manager (AIM) 结合使用，以支持基于非 Windows 的安全性和单一登录。作为管理员，针对用户身份验证和安全性有多个可用的配置选项。安全性和用户管理应该协同工作，如需详细信息，请参阅在 Web 客户端中配置用户访问权限。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



选项 1：缺省条件下，可以使用 http 协议和基于 Windows 的身份验证来访问 InTouch Web 客户端。

选项 2：使用配置器连接到本地或远程系统管理服务器。此处 InTouch Web 客户端将结合使用 https 协议与基于 Windows 的安全性（缺省条件下）。如果您连接到远程服务器，那么在访问 Web 客户端时必须为域用户提供凭证。

选项 3：在配置系统管理服务器之后，可以选择性地启用 AVEVA Identity Manager (AIM)。AIM 是一种可提供 OpenID Connect 端点的独立身份验证服务器。要使用 AIM，必须向标识服务器注册客户端。您可以使用系统管理服务器配置器或 InTouch HMI 应用程序管理器来配置 AIM。如需详细信息，请参阅向 AVEVA Identity Manager 进行注册。

在运行时，当"InTouch Web 客户端"页面加载时，如果没有有效的安全令牌，那么：

- Web 客户端将重定向到 AIM 的登录页面。
- AIM 将从 Active Directory 中检查用户凭证。
- 如果凭证有效，Active Directory 将会提供一个安全令牌，并将它返回给 Web 客户端。
- Web 客户端接着将使用该令牌授予用户访问权限。

如果安全令牌已经存在，即会授予用户访问权限。AIM 仅支持可从 Active Directory 进行验证的用户。

选项 4："反向代理"选项允许 OT 网络外部的用户访问 Web 客户端。在配置器的"安全网关"字段中或 InTouch HMI 应用程序管理器的"Web 客户端"选项卡中提供反向代理服务器的 FQDN。

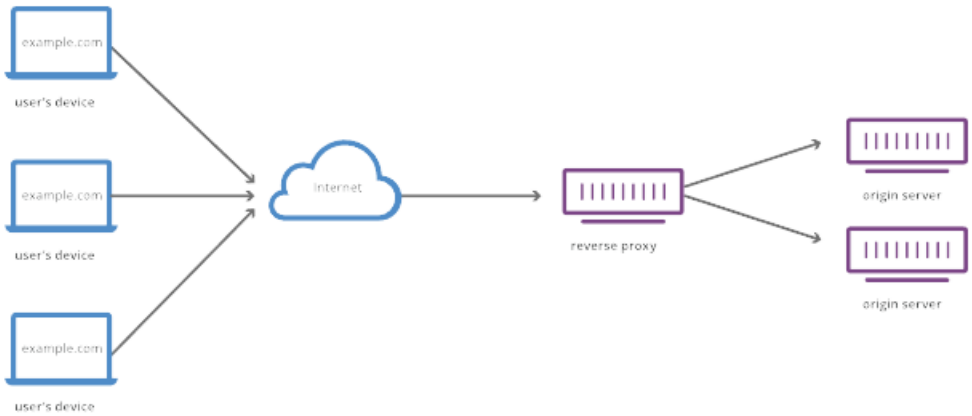
选项 5：在将 AIM 用于用户身份验证时，它会阻止将图形显示在 iFrame 中。选中**允许将 Web 客户端嵌入任何网站复选**框后，用户可以在运行时在 iFrame 中显示图形。在运行时，使用"共享"图标并选择要在 iFrame 中插入的代码段。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从 OT 网络的外部访问 InTouch Web 客户端

反向代理是位于一个或多个 Web 服务器的前面，可拦截来自客户端的请求的服务器。通过反向代理，在客户端向网站的源服务器发送请求时，反向代理服务器即会在网络边缘拦截那些请求。然后，反向代理服务器再将请求发送给源服务器，并接收来自源服务器的响应。如果这些 Web 服务器中的任何一个出现故障或其中一个发生故障转移，那么反向代理会将来自客户端的请求发送到冗余 Web 服务器。

正向代理位于客户端的前面，可确保没有任何源服务器能与该特定客户端直接通讯。另一方面，反向代理位于源服务器的前面，并确保没有任何客户端曾经与该源服务器直接进行通讯。



必须先在 DMZ 服务器上配置反向代理，OT 网络外部的设备才能访问 InTouch Web 客户端。有关更多信息，请参阅 AVEVA DMZ 安全链接。必须启用 AVEVA Identity Manager，反向代理才会起作用。有许多反向代理解决方案的提供商。请参阅相关的反向代理解决方案文档、有关冗余支持的信息以及有关在基础结构中设置反向代理解决方案的操作说明。

向 AVEVA Identity Manager 进行注册

使用 AVEVA Identity Manager (AIM)，可以将 InTouch Web 客户端配置为使用单一登录，而不是缺省的基于 Windows 操作系统的身份验证。

注意：InTouch Web 客户端仅通过 AIM 配置支持基于 ArchestrA 的安全性。

配置标识服务器

1. 在 System Platform 配置器中，配置**通用平台 > 系统管理服务器**。
2. 在 AVEVA 应用程序管理器中，使用用户凭证注册 AIM 服务器。

“AIM 注册”对话框也可以用来配置反向代理服务器：

1. 设置反向代理服务器：
2. 在 System Platform 配置器中，配置**通用平台 > 系统管理服务器**。
3. 在“AIM 注册”对话框中提供“安全网关”地址。

您可以选择远程或本地系统管理服务器。如需有关配置“系统管理服务器”的详细信息，请参阅 *System Platform 安装*。

向标识服务器进行注册

1. 在应用程序管理器中，**选择 Web 客户端选项**卡。
2. 依次**选择工具**和 **Identity Manager**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现**标识服务器设置**屏幕。

3. 选择使用 **AIM 服务器作为身份验证服务器**复选框，以允许使用 AVEVA Identity Manager。

   标识服务器字段显示在配置器中所配置的标识服务器。

4. 更新以下设置：

   a. **用户名：提供用户名以连接到标识服务器。**用户必须属于 Administrators 用户组。

   b. **密码**：提供相应用户名的密码。

5. 要完成反向代理设置，请在**安全网关**字段中提供反向代理或 DMZ 服务器的 URL。

   这是一个可选设置，只有将 Web 客户端托管在反向代理服务器的后面时，才需要进行此设置。有关更多信息，请参阅 AVEVA DMZ 安全链接。

6. 或者，也可以选择**允许将工业图形嵌入任何网站**复选框，以在运行时在 HTML iframe 中查看 Web 客户端。

7. 选择**确定**。

8. 选择**应用**。

注意：如果网站名称或地址发生更改，您必须配置 AVEVA Identity Manager 才能注册新网站。

使用单一登录访问 InTouch Web 客户端

如果在配置器中已启用“系统管理服务器”选项，您可以使用单一登录来访问 InTouch Web 客户端。这将与缺省的基于 Windows 的身份验证有所不同。此方法将使用 HTTPS 协议，并且仅支持可从 Microsoft Active Directory 进行验证的域用户（使用远程 SMS 时）。如果用户选择使用 HTTP 协议，Web 客户端将回退到仅使用基于 Windows 的身份验证。

如需有关配置“系统管理服务器”的详细信息，请参阅 *AVEVA System Platform 安装*。

1. 启动 Web 客户端。

   此时出现登录页面。

2. 输入您的用户名和口令。

3. 选择**登录**。

   作为可选项，还可以选择 **Windows 集成**，以使用当前登录的用户的凭证。

   您将使用 Identity Manager 凭证进行登录。

4. 使用**注销**可注销会话。

将本地标记用于 AIM

InTouch Web 客户端中本地标记的行为独立于 WindowViewer。如果启用 AIM，本地标记的行为会有所不同。如果启用 AIM，则会针对每个用户的每个会话使用唯一的令牌。

	用户 1 - 会话 1	**用户 1 - 会话 2**	**用户 2 - 会话 1**
禁用 AIM	相同标记值	相同标记值	不同标记值
启用 AIM	不同标记值	不同标记值	不同标记值

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在 InTouch Web 客户端中配置用户访问权限

匿名访问 InTouch Web 客户端

InTouch Web 客户端既提供了匿名访问，也提供了通过身份验证的用户访问。缺省条件下，匿名访问处于禁用状态。要启用此选项，请选中 InTouch HMI 应用程序管理器的“Web 客户端”选项卡中的允许匿名访问复选框。在运行时，用户无需提供任何用户凭证，并且将作为“Guest”用户进行登录。Guest 用户将对 Web 客户端具有只读访问权限。

为 InTouch Web 客户端配置通过身份验证的用户

身份验证和许可工作流程都使用用户组来确定哪些用户获得访问权限，以及为 Web 会话获得的许可证的类型。根据 SMS 配置，可以在本地或远程节点上配置用户组。需要访问权限的所有用户都必须包含在其中一个组内，具体取决于他们的访问级别。

如果将 Web 客户端配置为使用远程系统管理服务器，请在远程服务器上创建以下域用户组：aaInTouchUsers 和 aaInTouchRWUsers。在部署应用程序之前，将所有相关用户添加到这些组。

使用 InTouch Web 客户端虚拟帐户

为了提高安全性，我们降低了 Web 客户端虚拟服务帐户的权限。要继续对 InTouch 应用程序使用 Web 客户端，InTouch Web 客户端虚拟服务帐户必须具有对 InTouch 应用程序文件夹的读/写访问权限。InTouch Web 服务的虚拟服务帐户称为“NT SERVICE\AIGWebServer”。

InTouch Web 客户端虚拟服务帐户将具有对以下应用程序的读/写访问权限：

- 所有现有 InTouch 应用程序。
- 所有通过“应用程序管理器”创建/导入的 InTouch 应用程序。
- 所有通过“查找”操作找到的 InTouch 应用程序；仅限用户在提示对话框中选择“是”以授予 Web 客户端虚拟帐户对找到的应用程序的读/写访问权限时。

要手动授予 InTouch Web 客户端虚拟帐户对 InTouch 应用程序的访问权限：

1. 使用文件资源管理器导航到 InTouch 应用程序。
2. 使用鼠标右键单击该文件夹，然后选择属性。
3. 在属性对话框的安全选项卡下，选择编辑。
4. 在组或用户名部分中，选择添加。

   此时出现选择用户、计算机、服务帐户或组对话框。
5. 选择位置。

   此时出现位置对话框。
6. 选择计算机名，然后选择确定。
7. 在输入要选择的对象名字段中，输入 NT SERVICE\AIGWebServer。
8. 选择确定。
9. 在经过身份验证的用户的权限部分的允许列下，选中以下复选框：读取和执行、列出文件夹内容、读取、写入。
10. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：InTouch Web 客户端虚拟服务帐户不会具有对共享位置中应用程序文件夹的访问权限。如果应用程序路径以“\\”开头，则为文件共享的路径。

启用 InTouch Web 客户端功能

缺省条件下，安装时会禁用 InTouch Web 客户端功能。您必须先启用 InTouch Web 客户端功能才能使用它。如果在配置程序中配置下列其中一个选项，将会自动启用 InTouch Web 客户端功能：

- **通用平台 > 系统管理服务器**
- **应用程序管理器 > Web 客户端**

如需有关配置器的详细信息，请参阅《*System Platform 安装*》。要手动启用或禁用 Web 客户端，必须具有管理权限。您可以从“应用程序管理器”或 InTouch WindowMaker 启用 Web 客户端功能。

从应用程序管理器启用 InTouch Web 客户端功能

1. 启动“应用程序管理器”。
2. 选择 Web 客户端选项卡。
3. 在文件菜单上的主组中，选择 **Web 客户端**。
4. 再次选择 **Web 客户端**即可禁用。

从 WindowMaker 启用 InTouch Web 客户端功能

具有管理权限的用户可以从 WindowMaker 启用 Web 客户端功能。

1. 在 WindowMaker 中启动一个应用程序。
2. 选择 **Web 客户端**快速切换按钮。

   将出现以下对话框：“InTouch Web 客户端功能当前在此系统上为禁用状态。是否要启用？”
3. 选择是。

   此时出现 InTouch Web 客户端页面。

如果您尝试在 Web 客户端处于禁用状态时启动它，您将收到以下错误消息：**HTTP 错误 404.0 – 找不到**。

如果非管理用户试图从 WindowMaker 启用 Web 客户端功能，将会引导他们从“应用程序管理器”启用该功能。

自定义 InTouch Web 客户端

AVEVA 应用程序管理器的“Web 客户端”选项卡为用户提供了用于配置与 InTouch Web 客户端相关的设置的选项。

更新 InTouch Web 客户端设置

1. 启动“应用程序管理器”，然后选择 **Web 客户端选项**卡。

   此时出现 **Web 客户端设置**屏幕。
2. 配置以下设置：
   - *当前应用程序*：从列表中选择一个应用程序以修改 Web 客户端设置。
   - *图形刷新率（毫秒）*：设置 Web 浏览器将在 Web 服务器中查询图形数据的频率。缺省值为 1 秒。如需详细信息，请参阅《System Platform 安装》。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- *报警刷新率（毫秒）*：设置 Web 浏览器将在 Web 服务器中查询报警数据的频率。缺省值为 1 秒。如需详细信息，请参阅《System Platform 安装》。
- *显示标题*：选择该复选框，以显示标题栏。
- *启用导航栏*：选中该复选框以显示“导航栏”。

  如果未选择显示标题设置，将会禁用此设置。
- *允许匿名访问*：选中该复选框以允许用户无需经过身份验证即可访问 Web 客户端。
- *启用工作区*：选中该复选框以启用工作区。

3. 在“高级设置”部分下，选择自定义以配置以下设置：
   - *网站名称*：提供将替换标准 URL 的字符串。
   - *应用程序标题*：提供将在应用程序栏中显示为应用程序标题的字符串。
   - *网站标题*：提供将在浏览器的标题栏上显示为标题的字符串。
   - *网站图标*：提供将替换浏览器标题栏上的图标的图像文件。
     i. 选择选择文件...，以选择图标图像。
     ii. 选择一个文件。
     iii. 选择打开。
4. 选择保存，以保存这些高级设置。
5. 修改设置后，选择应用。
6. 要查看 InTouch Web 客户端，请选择启动。

注意：如果网站名称或地址发生更改，请确保配置 AVEVA Identity Manager 以注册新网站。如需详细信息，请参阅向 AVEVA Identity Manager 进行注册, 向 AVEVA Identity Manager 进行注册。

如需有关这些设置的详细信息，请参阅*在 Web 浏览器中查看应用程序图形*。

配置图形，以在 InTouch Web 客户端中查看

使用图形工具箱，您可以对包含将在 InTouch Web 浏览器上显示的应用程序图形的文件夹进行配置。只有设置为“Web 客户端根”文件夹的文件夹内存储的图形会显示在 Web 浏览器上。

注意：在安装过程中，将在 InTouch 运行时节点上创建操作系统用户组 **aaInTouchUsers** 和 **aaInTouchRWUsers**。只有 aaInTouchUsers 或 aaInTouchRWUsers 用户组中的用户有权在 Web 浏览器中查看应用程序的图形。在配置应用程序前，请将相关用户添加到该用户组。缺省条件下，会将安装用户添加到这两个组。

---

1. 启动 InTouch WindowMaker。
2. 打开应用程序。

   在工业图形工具箱中，创建或标识将包含您想要在 Web 浏览器上显示的文件夹和图形层次结构的文件夹。
3. 右键单击该文件夹，然后选择设置 **Web 客户端根文件夹**。

   图标缩略图将更改以反映设置。

图标	描述

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

	当根文件夹设置为“Web 客户端根”文件夹时
	当根文件夹以外的文件夹设置为“Web 客户端根”文件夹时

缺省条件下，会将图形工具箱的根文件夹设置为“Web 客户端根”文件夹。

设置 InTouch Web 客户端主页符号

您可以通过设置 Web 客户端主页符号来设置 InTouch Web 客户端页面在启动时显示的缺省符号。

1. 要设置“Web 客户端根”文件夹，请参阅配置 InTouch 应用程序图形以在 Web 浏览器中查看。

   主页符号必须是“Web 客户端根”文件夹的子项。

2. 在“Web 客户端根”文件夹下，找到主页符号。
3. 使用鼠标右键单击符号，然后选择**设置 Web 客户端主页符号**。

图标缩略图 ( ) 将根据设置进行相应地更改。如果未设置任何主页符号，那么 Web 客户端页面在启动时将显示空白页。在浏览器中查看 Web 客户端时，选择应用程序栏上的“主页”图标可导航到主页符号。

了解 InTouch Web 客户端主页图标的行为

独立的应用程序可以在 Web 客户端上查看。独立应用程序窗口必须转换为工业图形，才能在 Web 客户端上查看。只能在 Web 客户端上查看框架窗口和工业图形。

对于 InTouch 应用程序，可以在 WindowMaker 中使用“起始窗口”选项卡来设置缺省条件下 WindowViewer 中将出现的窗口。请参阅*设置要在运行时出现的窗口*。类似地，您可以设置当 Web 客户端启动时作为缺省符号显示的图形，请参阅设置 InTouch Web 客户端主页符号。

如果您已使用 InTouch ShowHome 脚本函数，当窗口转换为图形后，脚本将转换为 ShowHome() 函数，并将显示相应的图形作为 WindowViewer 中的主页符号。

这些设置将影响在 InTouch Web 客户端启动时缺省条件下会显示哪个窗口或图形。对于 InTouch HMI 应用程序，有两种可能性：

1. 如果已配置起始窗口

   即使用户已配置主页符号，在所有场合，缺省也只显示起始窗口。选择“主页”图标将显示起始窗口。

2. 如果未配置起始窗口

   并且已配置主页符号，缺省条件下会显示主页符号。

   如果未配置主页符号，缺省条件下不会显示任何符号或起始窗口。选择主页符号不会显示任何图形或窗口。

在 Web 浏览器中查看应用程序

在配置 InTouch Web 客户端应用程序后，可以在 Web 浏览器中查看该应用程序。有关经过测试的受支持浏览器的列表，请参阅浏览器和移动支持。

注意：为让 InTouch Web 客户端正常工作，建议始终安装最新版本的浏览器或将浏览器升级到任何可用的最新版本。

---

1. 启动支持 HTML5 的浏览器，输入 URL：`http://<hostname>/InTouch` or `http://<IPAddress>/InTouch`。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果您的凭据已通过身份验证，并获得了有效的 Web 服务器许可证，将显示 InTouch Web 客户端页面。缺省条件下，将显示主页符号。

您可以通过以下 URL 访问 Web 客户端：https://<hostname>/InTouch or https://<IPAddress>/InTouch。为通用平台系统管理服务器配置了证书（本地或远程服务器）时，Web 客户端将自动使用 HTTPS 协议在服务器与客户端之间进行加密通讯。如果未显示 Web 客户端页面，请参阅在 Web 浏览器中查看图形的疑难排解。

如需有关自定义 Web 客户端 URL 的信息，请参阅自定义 InTouch Web 客户端。

2. 选择汉堡图标。

左侧的覆盖窗口会显示图形的层次结构，这些图形位于运行中应用程序内的所选“Web 客户端根”文件夹下。

3. 选择文件夹名称。

此时会显示该文件夹内的图形。

4. 选择图形。

所选图形会显示在 InTouch Web 客户端上。

只有设置为“Web 客户端根”文件夹的文件夹内存储的图形会显示在 Web 浏览器上。如果禁用了匿名访问，那么访问图形的用户帐户必须经过身份验证和授权，如需详细信息，请参阅获得 InTouch 应用程序图形的许可证。

注意：为了改进图形渲染并获得更好的整体性能，我们建议启用 GPU 硬件加速。

InTouch WindowViewer 必须在主计算机上运行并具有读/写 InTouch 许可证，Web 客户端中的符号才能从 InTouch 标记接收实时数据。WindowViewer 还可以作为服务运行。Application Server 必须处于运行状态才能从应用程序对象接收实时数据。

注意：只读 InTouch 许可证不支持远程数据连接，因此不能提供足够的功能来充当 InTouch Web 客户端的数据源。所有向 InTouch Web 客户端提供数据的 WindowViewer 应用程序都必须在读/写 InTouch 许可证下运行。

InTouch Web 客户端快速切换

在应用程序的开发过程中，您可以使用 Web 客户端快速切换功能在 WindowMaker 和 InTouch Web 客户端之间切换。

- 要使用 Web 客户端快速切换功能，请在 InTouch WindowMaker 中选择 **Web 客户端**按钮。

如果在 WindowMaker 中编辑的应用程序和在 Web 客户端中查看的应用程序相同，则 WindowMaker 中的任何更改都会自动反映到 Web 客户端上。为了使自动刷新起作用，一次只能编辑和查看一个应用程序。InTouch 网页会刷新为先前显示的符号。如果针对另一个应用程序运行 WindowViewer，则 Web 客户端页面将自动切换到该应用程序。

- 您可以从配置器中指定**图形刷新率**。缺省条件下，**轮询频**率为 1000 毫秒。
- 您还可以从配置器中指定**报警刷新率**。缺省条件下，**轮询频**率为 1000 毫秒。

报警刷新率应高于图形刷新率，并且图形刷新率的值应小于 180 秒。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

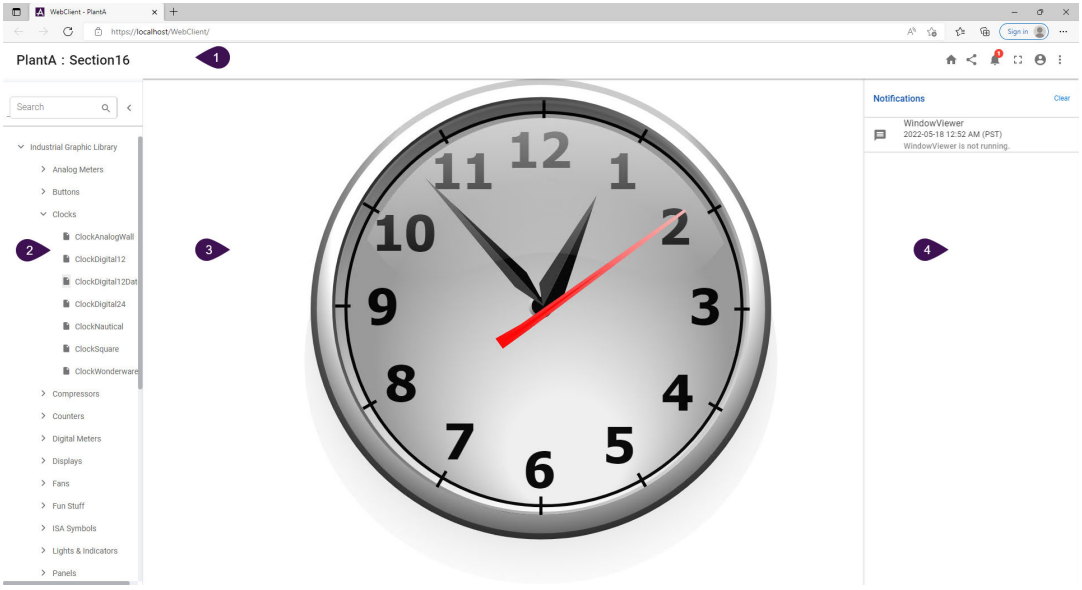
注意：要在从远程 IDE 编辑托管的 InTouch 应用程序的同时使 InTouch Web 客户端在快速切换模式下运行，请先在 GR 和远程 IDE 计算机中将 **InTouch Web Windows** 服务配置为以属于 **aaConfigTools** 用户组成员的用户账户身份运行。如需详细信息，请参阅 Application Server 用户文档。

对于给定的 InTouch 应用程序，缺省条件下会禁用 WindowMaker 中的 Web 客户端快速切换按钮。 在 WindowViewer 中运行应用程序时，该按钮将自动启用。

注意：Web 客户端快速切换功能仅在以下情况下可用：
- 在开发期间，
- 在安装了 InTouch WindowMaker 和 InTouch WindowViewer 的计算机上，
- 在通过指向 *http://localhost/InTouch* 的 URL 访问 Web 客户端的 Web 浏览器上。

了解 InTouch Web 客户端页面

InTouch Web 客户端页面提供多种不同选项，通过页面上的图标来为用户提供信息，并丰富用户的使用体验。下表介绍了这些选项。



元素	描述
1	应用程序栏显示 InTouch 应用程序名称、符号名、主页图标、通知图标、全屏图标和配置文件图标。
应用程序名称：符号名称	应用程序栏将显示应用程序名和符号名。 例如 – *PlantA: Section16* - 其中 PlantA 是应用程序名称，Section16 是符号名。
	选择可查看作为 Web 客户端主页符号的符号集。
	选择可查看所选图形的 iFrame 代码段。

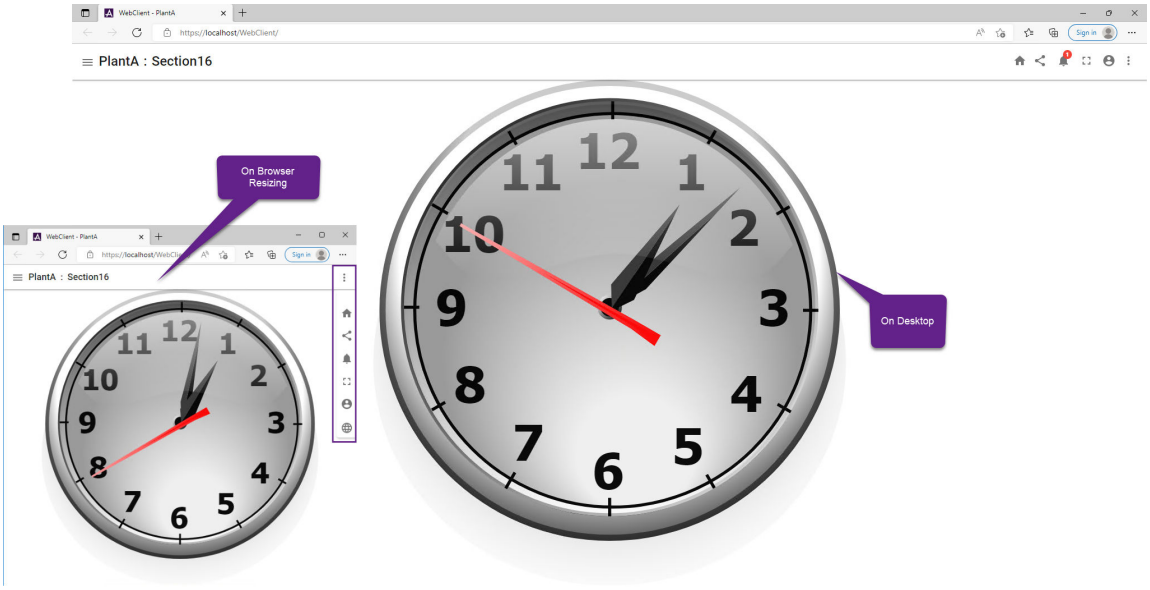
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

	选择与许可证状态变更和 WindowViewer 状态变更消息有关的“通知”图标消息。通知数量将叠加在图标上。
	选择可以使画布区域最大化，并以全屏模式显示网页。您也可以使用 F11 键。应用程序栏和导航叠加窗口将不可见。 **要查看标题栏：** • 将光标移至页面顶部，应用程序栏将向下展开。 • 选择向下箭头可将应用程序栏固定在页面上。 • 选择向上箭头可隐藏应用程序栏。
	选择可以使画布区域最小化，并退出全屏模式。
	配置文件图标显示登录的用户。
Sign Out	选择配置文件图标，然后选择注销图标。仅当配置“系统管理服务器”选项时，“注销”才可用。如需详细信息，请参阅使用单一登录访问 InTouch Web 客户端。
	选择应用程序栏中的 ⋮ ，然后选择语言图标。选择语言。工具提示、翻译的静态文本和自定义属性将以所选语言进行显示。仅显示在 WindowMaker 中配置的语言。 **注意：**当用户更改语言时，将会创建一个新会话，从而导致本地标记中的数据丢失。
2	选择 ☰ 图标可查看文件夹导航叠加窗口。所选“Web 客户端根”文件夹下的所有文件夹和符号都会显示。 您可以使用叠加窗口顶端的搜索框来搜索符号。使用自动完成功能可启用搜索框，其中将列出与您所输入的搜索项相符的符号名称。选择该符号名称可在画布区域查看该符号。
3	画布区域显示从导航叠加窗口中选择的符号。该符号将缩放，以适应浏览器视区大小，同时保持纵横比。导航和通知叠加窗口不会占用画布上的任何空间，也不会影响图形的大小。
4	选择通知图标可查看与连接和许可证问题有关的通知。
	选择警告图标可查看 InTouch Web 客户端不支持的动画或属性的列表。此图标仅在应用程序开发期间以及浏览器指向的 URL 为 http://localhost/InTouch 时可用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

调整应用程序图形和浏览器的大小

InTouch Web 客户端页面可以用不同的屏幕大小查看。调整浏览器时，图形会调整大小以适应新的浏览器视区尺寸。在横向和纵向之间旋转移动设备时，图形也将调整大小。



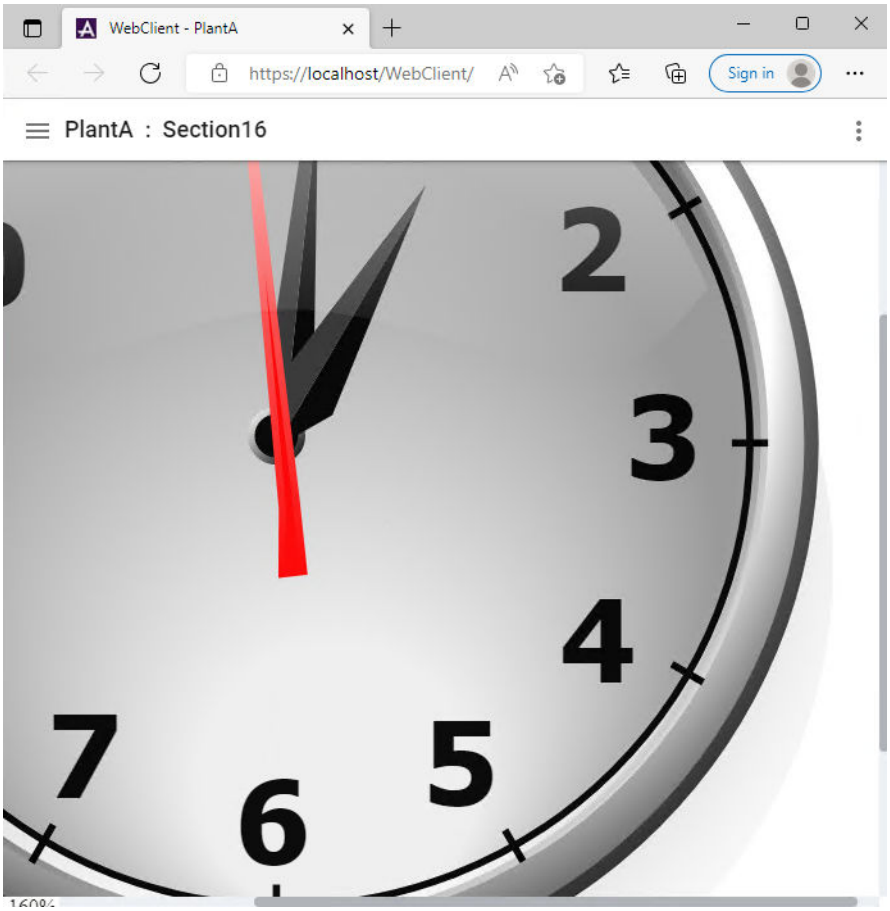
应用程序栏和较小的屏幕

对于非常小的屏幕，右侧的图标将折叠到垂直条中。图标将隐藏以节省屏幕空间。您可以在任意移动设备上查看工业图形，并使用平移与缩放手势查看较大图形。

平移与缩放支持

对于所有支持的浏览器和设备，InTouch Web 客户端都支持平移与缩放手势。在非移动设备上的 Web 浏览器中放大图形时，缩放百分比显示在水平滚动条的左下角。缩放百分比上限为 500%。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



通过触摸进行平移与缩放

您可以使用两个手指在聚焦的显示画面上放大和缩小。

- 要进行放大，请将两个手指放在屏幕上并展开。要进行缩小，请用两个手指捏合显示画面。
- 双击可将显示画面恢复为 100% 缩放。

要平移显示画面，请将一个手指放在显示画面上，然后移动手指。

使用鼠标进行平移与缩放

您可以使用鼠标滚轮在聚焦的显示画面上放大和缩小。

- 按住 **Ctrl** 键，向上或向下滚动滚轮来放大或缩小。如果鼠标指针在窗格外部，则不允许缩放。
- 双击鼠标左键可将显示画面恢复为 100% 缩放。

要平移显示画面，请按住鼠标中键，将鼠标移动至区域，然后松开中键。

使用键盘进行缩放

您可以使用键盘在聚焦的显示画面上放大和缩小。

要进行放大，请同时按 **Ctrl** 和 **+** 键。要进行缩小，请同时按 **Ctrl** 和 **-** 键。

按位置缩放

您还可以从选择的位置放大图形。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

不支持“平移与缩放”工具栏。对于使用 ShowSymbol 动画或 ShowGraphic 脚本、按钮、选择和游标显示的图形，不支持平移与缩放手势。

将 InTouch 应用程序图形托管在外部网站上

您可以在 InTouch WindowMaker 中配置图形，并使用图形的直接 URL 将它们托管在外部网站上。

1. 使用“工业图形工具箱”创建一个符号。
2. 在 Web 客户端上测试该符号。
3. 在外部网站中，创建一个 HTML **<iframe>** 标记。
4. 在 URL 属性中，包含指向 http://<hostname>/InTouch/?iframe=true&symbol=<graphicname> 或 http://<IPAddress>/InTouch/?iframe=true&symbol=<graphicname> 的链接，
   其中 <hostname> 或 <IPAddress> 对应于发布或部署应用程序的计算机，<graphicname> 是您希望显示的图形的名称。
5. 如果符号采用非英语语言进行显示，URL 格式将会有所不同。例如，法语的 URL 将为：
   http://<hostname>/InTouch/?localeid=1036&iframe=true&symbol=<graphicname>

   更高版本的 Web 客户端允许在运行时进行语言切换，从而导致 URL 格式发生更改。在升级到最新版本的 Web 客户端之后，较早的 URL 格式不会显示针对非英语语言配置的符号。
6. 发布或部署应用程序。
7. 在 InTouch WindowViewer 中运行应用程序。WindowViewer 必须运行才能从 InTouch 标记接收实时数据。

语法：

```
<iframe src="http://localhost/InTouch/?iframe=true&symbol=YourGraphicName"></iframe>
或
<iframe src="http://NodeName/InTouch/?iframe=true&symbol=YourGraphicName"></iframe>
```

示例：以本地计算机上的 Web 客户端应用程序中的符号 ClockAnalogWall 为例。

```
<iframe src="http://localhost/InTouch/?iframe=true&symbol=ClockAnalogWall"></iframe>
<iframe src="http://localhost/InTouch/?localeid=1036&iframe=true&symbol=ClockAnalogWall"></
iframe>
```

要使用 URL 直接访问该符号，必须使用相同的 InTouch Web 客户端安全性和许可证。相关错误消息将返回到 <iframe> 部分，与 InTouch Web 客户端错误消息相似。

备注：由于 Google Chrome 80 以及更新版本中的策略更改，在使用 http 协议时不会显示放在 <iframe> 标记中的图形。要使用 http 协议显示图形，请禁用 **chrome://flags** 下面的以下可用 cookie 设置
- *SameSite by default cookies*
- *Enable removing SameSite=None cookies*
- *Cookies without SameSite must be secure*

如需详细信息，请参阅 https://www.chromestatus.com/feature/5633521622188032。建议对 Web 客户端使用 https 协议，以防止发生任何安全问题。

生成 iFrame 代码块

您可以直接从浏览器为符号生成 <iframe> 块。然后可以在外部网站中使用该代码块在运行时查看工业图形。

1. 导航到图形。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 选择共享 图标 。
3. 在父对象字段中，指定此图形的父对象。
4. 将显示支持的自定义属性的列表。选择一个自定义属性，然后提供一个值。

   提供的所有值都将针对数据类型进行验证。例如，如果某个自定义属性需要整数值，将不允许输入字符值。
5. 选择支持跨源以在本地或远程计算机上的任何 Web 客户端中使用脚本。

   嵌入的 HTML 代码将根据所选选项进行更改。
6. 选择复制。
7. 您现在可以将网站中生成的代码粘贴到一个 .html 文件中。

```
<iFrame src="http://<hostname>/InTouch/?iframe=true&symbol=Sym1?crossOrigin=1"
id="symbol" width="640px" height="480px" customProperty='[{"n":"CP1","v":"True","t":1},
{"n":"CP2","v":"24.5","t":6},{"n":"CP3","v":"Orange","t":7}]' onload="postMsg()"></
iFrame>
<script type="text/javascript">
函数 postMsg(){
    var iframe=document.getElementById('symbol');
    var obj={};
    for (var i=0;i<iframe.attributes.length;i++) {
        obj[iframe.attributes[i].nodeName]=iframe.attributes[i].nodeValue;
    }
    var win=iframe.contentWindow;
    win.postMessage(obj,obj['src']);
};
</script>
```

   If Language = French (France)

```
<iFrame src="http://<hostname>/InTouch/?localeid=1036&iframe=true&symbol=Sym1"
id="symbol" width="640px" height="480px" onload="postMsg()"></iFrame>
<script type="text/javascript">
函数 postMsg(){
    var iframe=document.getElementById('symbol');
    var obj={};
    for (var i=0;i<iframe.attributes.length;i++) {
        obj[iframe.attributes[i].nodeName]=iframe.attributes[i].nodeValue;
    }
    var win=iframe.contentWindow;
    win.postMessage(obj,'*');
};
</script>
```

支持的图形元素和已知限制

本节介绍支持的图形元素和已知问题。

已知限制

每个产品版本都会增强 InTouch Web 客户端。以下列表汇总了当前版本的主要限制。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 不支持 RDS 会话中运行的 InTouch WindowViewer 与 Web 服务器进行数据通讯。
- Web 客户端不支持从 System Platform 2017 及更早版本发布的应用程序。您必须将原始应用程序迁移到当前版本，然后重新发布应用程序才能配合 Web 客户端使用。
- 不支持窗口名中含有特殊（不支持的）字符的框架窗口。
- 不支持间接标记。
- 不支持 PlaySound() 脚本函数。
- 对于 InTouch HMI 托管的应用程序，不支持基于 Galaxy 的安全性。
- 系统标记不会返回每个用户的用户名和相关详细信息。
- 使用脚本设置图形属性：使用脚本读取或写入图形元素属性仅支持以下属性：
  - 位置（X、Y）
  - 大小（高度、宽度）
  - 角度
  - StartAngle
  - SweepAngle
- 符号库中的几个符号包含不受当前版本的 Web 客户端支持的图形元素或颜色设置。
- Windows 公共控件 (WCC) 是指“单选按钮组”、“复选框”、“编辑框”、“组合框”、“日历”、“日期时间选择器”和“列表框”。

  注意：组合框控件无法在输入字符串时将值传递给引用对象。无论选择类型如何，您都只能从组合框中的预定义值中进行选择。
- 不支持多笔趋势图形元素。
- 不支持位状态动画。
- 不支持以下真值表动画：禁用、可见性、闪烁和值显示。
- 不支持历史摘要数据类型的自定义属性。

以下几节详细介绍了这些限制。

要点：在 Web 客户端中加载图形时，不受支持的功能将记录器中记录为警告。生成的报告将包括警告路径。

所有受支持的图形元素支持的属性

在 InTouch Web 客户端中查看时，符号中所有受支持的图形元素均支持下列属性：

- 角度
- X
- 是
- 宽度
- 高度
- 启用

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 可见
- AbsoluteAnchor
- RelativeAnchor
- 透明度
- ElementStyle
- OwningObject
- 开始
- 结束
- Radius
- Tension
- 自定义属性覆盖

注意：ElementStyle 属性允许用户选择已定义的元素样式并将其应用于图形元素。元素样式包括用于定义 FillColor、LineColor、TextColor 和 OutlineColor 等颜色属性的选项。

所有受支持的图形元素均支持的属性以及一些限制

在 InTouch web 客户端中查看时，以下图形元素具有一些渲染限制：

图形元素属性	属性限制
FillColor	不支持纹理。FillColor 的一些其它限制可能适用于单个元素。
UnFilledColor	FillColor 属性的限制同样适用于 UnFilledColor 属性。
LineColor	不支持纹理。
字体	仅支持 Font 属性的“字体大小”、“字体类型”、“粗体”或“常规字体样式”选项。
TextColor	仅支持 TextColor 属性的“纯色”选项。
FillOrientation	仅当“颜色”设置为“纯色”、“填充”或“未填充时，才支持 RelativeToScreen。
FillBehavior	将始终设置为 Both。
运行时行为	TabOrder、TabStop 和 ZoomPercent 在任何图形元素上都不受支持。
相对引用	支持 MyPlatform、MyEngine、MyHost、MyArea、MyContainer 和 Me。不支持 MyViewApp。
自动调整刻度	不受支持。
显示图形	支持。
线条	如果将“线条”图形配置成符号边框上的 LineWeight 属性值超过 1px，它将会被截断。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

图形元素属性	属性限制
组合/嵌入图形	支持“可见性”、“闪烁”、“禁用”、“按钮”、“用户输入/值显示”、ShowSymbol 和 ActionScript 动画。 不支持通过脚本更改“Treat as icon”和“Enabled”属性。 %FillHorizontal 和 %FillVertical 动画： - 仅在组上适用。不支持用于子元素。 - 不支持用于“按钮”和“图像”。 - 不支持将“相对于屏幕”用于组和子元素。 - 当子元素的角度发生更改时，不支持。

支持的图形元素及其它属性

仅下面列出的图形元素可以在 InTouch web 客户端中显示。**此表中未列出的任何图形元素在 InTouch web 客户端中均不受支持。**

该表列出了每个图形元素支持的属性以及这些属性的任何限制。当使用包含颜色属性的图形时，颜色的限制同样适用于 FillColor、LineColor 和其它相关图形属性。

注意：除了此处列出的属性之外，在所有图形元素支持的属性和所有图形元素均支持的属性以及一些限制中列出的所有属性均可与任何支持的图形元素一起使用。

图形元素	支持的属性	限制说明
线条 折线 曲线 2 点弧形 3 点弧形 连接线	线条样式 LineWeight LinePattern StartCap EndCap	StartCap、EndCap：支持，但在呈现上可能会稍有不同。
按钮	ButtonStyle 文本 WordWrap FillOrientation FillColor 对齐	ButtonStyle：仅支持标准按钮样式。 WordWrap：任何超出按钮宽度的标题都将被截断。 FillOrientation：“RelativeToScreen”仅在颜色设置为特定值时受支持。 FillColor：仅支持纯色。如果选择了多色渐变，这些渐变将转换为单一颜色（第一个颜色）。 Alignment：仅支持居中对齐。
文本	对齐	Alignment：仅支持左上。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

图形元素	支持的属性	限制说明
图像	HasTransparentColor TransparentColor ImageStyle	不支持 FillColor、FillPercent 和 unfilledColor。
文本框	文本 TextFormat WordWrap LineWeight LinePattern 对齐 字体	不支持翻转。
状态	图形 表达式	Status Style：仅支持缺省配置。
Web 报警客户端 (EAC) （只有某些 HMI/ SCADA 软件支持此元素。）	报警模式 颜色 列明细 查询和过滤 时间设置 运行时行为	**报警模式** • Client Mode：仅支持“当前报警”、“最新报警与事件”。 • Alarm Query： 支持 InTouch 和 Application Server 报警 • 使用缺省确认注释：仅支持显示“报警与事件”的注释。 **报警与事件：** 颜色：不支持“搁置”和“闪烁未确认报警”。 列明细：不支持排序。 查询过滤：仅支持“缺省”查询。不支持自定义查询或过滤。 数据绑定：不支持。 事件：不支持。 时间设置：不支持。 运行时行为：仅支持“显示标题”、“显示网格”和“允许调整列尺寸”。仅支持所选记录上的用户确认或所有报警记录上的确认。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

图形元素	支持的属性	限制说明
Web 趋势客户端 （只有某些 HMI/ SCADA 软件支持此元素。）	笔 外观 选项 历史来源	**笔** 仅支持“显示”、“笔名”（作为描述）、“表达式”或“引用”。 笔详细信息：最小值、最大值 笔选项：颜色绘图类型（全部被视为线条）。 **外观** PlotArea：单个标记模式，网格（显示垂直网格、显示水平网格、颜色）。 X time axis：显示光标（Cursor1:颜色）、值的个数。 Y value axis：值的个数、值轴标签（多个刻度、单个刻度）。 **选项** 仅支持“检索：趋势持续时间” 数据绑定：不支持 事件：不支持 历史来源：不支持“连接超时（以秒计）”、“查询超时（以秒计）”和“使用 HTTP”。 注意：不支持趋势客户端的方法和属性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

图形元素	支持的属性	限制说明
Web SQL 数据网格 （只有某些 HMI/ SCADA 软件支持此元素。）		数据绑定动画 事件动画 自定义属性 • CmdCopy_dg • CmdPaste_dg • CmdWrite_dg • RowsChanged_dg • WriteButtonHide_dg • ColumnAggregateEnable_dg • SupportThemes_dg - 不支持 SQLDataGrid 控件。 - 无法使用自定义属性检索 selectedRowNumber 属性值。 - 由于不支持数据绑定，因此如果在自定义属性中引用了某个属性并且缺省值为空，则无法检索该属性的缺省值。 - 由于 Kendo 网格的限制，RowCount 属性无法显示分组计数。它将始终显示记录总数。 - 当在本地数据库中使用“Windows”身份验证时，添加虚拟用户 NT Service\AIGWebServer，以允许它访问该数据库。不支持使用“Windows”身份验证和“SQLServer”模式访问远程数据库。 - 当 SQLGrid 在轮播小组件中使用并在 WindowViewer 中进行查看时，它不受支持。 - 将组应用到记录数超过 2000 条且有 5 个列的 SQL 网格时，通过滚动和刷新各行可能会看到一些性能问题。
趋势笔 （只有某些 HMI/ SCADA 软件支持此元素。）	数据源	趋势时间段：不支持固定类型。
复选框		选定值发生变化：无论用户设置如何，仅支持立即提交的值。 Checked：不支持 Checked 属性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

图形元素	支持的属性	限制说明
单选按钮组	静态值与标题 状态	选定值发生变化：无论用户设置如何，仅支持立即提交的值。
编辑框		选定值发生变化：无论用户设置如何，仅支持立即提交的值。
组合框		选定值发生变化：无论用户设置如何，仅支持立即提交的值。 类型：不支持 MaximumLength
日历		选定值发生变化：无论用户设置如何，仅支持立即提交的值。 Bold Dates：不支持 ShowToday：即使被签出，也将始终显示。 Calendar Colors：不支持
日期时间选择器		选定值发生变化：无论用户设置如何，仅支持立即提交的值。 Configuration：不受支持。 Calendar DropDown Colors：不支持
列表框		选定值发生变化：无论用户设置如何，仅支持立即提交的值。
ConnectorPoint		不支持移动 ConnectorPoint。

支持的动画

下表列出在 InTouch Web 客户端中查看图形时在运行时支持的动画。当使用包含颜色属性的动画时，所选动画颜色的限制与 FillColor、LineColor 和其它相关图形元素属性的限制相同。此外，在浏览器中查看不支持写入值。任何类型的可更新特性值的动画均不受支持。

注意：下表列出了所有受支持的动画。**对于所有未列出的动画，均不支持 InTouch Web 客户端中查看。**

动画	限制
报警边框	“报警边框”动画在图形元素周围显示报警边框，其可视外观表示图形元素的报警类型和确认状态。每个报警边框的外观都由关联的元素样式定义。并非所有元素样式属性都受到支持。例如，如果在报警边框元素样式中将“渐变”指定为元素样式的“线条颜色覆盖”选项的值，它将不受支持。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

动画	限制
元素样式	请参阅关于颜色、线条和字体的图形元素限制。适用于图形元素的限制同样适用于“元素样式”动画。 表达式或引用：不支持“时间”和“经过时间”。 只有重新部署或发布应用程序后，对元素样式定义的更改才会生效。 不支持 Windows 公共控件。
填充样式 线条样式 文本样式	请参阅关于颜色、线条和字体的图形元素限制。适用于图形元素的限制同样适用于“元素样式”动画。 表达式或引用不支持“时间”和“经过时间”状态。 不支持 Windows 公共控件。 TextColor：仅支持纯色。
水平填充百分比 垂直填充百分比	仅支持 RelativeToGraphic。
水平游标 垂直游标	游标不能将数据写入标记，但可以支持与自定义属性绑定的数据。 游标在写入自定义属性时仅支持“在鼠标放开时”。 游标不支持光标定位。 游标不支持“显示工具提示”。
宽度 高度	不支持 Windows 公共控件。
相对方向	设置“相对方向”动画时，不支持“相对定位”或“相对原点”。（仅支持定位在中心）。
禁用	RadioButton 图形元素不支持“禁用”动画。
点	不支持“曲线”和“封闭曲线”上的点动画。
工具提示	支持工具提示动画，但图像、单选按钮、按钮和文本图形元素除外。 组或嵌入符号不支持工具提示动画。 不支持 Windows 公共控件。
动作脚本	支持动作脚本，但调用对话框的函数除外。只有框架窗口支持 ShowGraphic 动画。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

动画	限制
ShowSymbol	标题：仅支持缺省选项。 类型：支持模式和无模式符号，其中“关闭”按钮选项已选中。 位置：仅支持中心，其中客户端区域 x,y 值为 0,0。 大小：仅支持“相对于符号”选项。 快捷方式：不支持。
显示图形	标题：仅支持缺省选项。 类型：支持模式和无模式符号，其中“关闭”按钮选项已选中。 位置：仅支持中心、客户端区域，且 x,y 值为 0,0。 大小：仅支持“相对于符号”选项。 快捷方式：不支持。
超链接	不支持 Windows 公共控件、按钮、文本和文本框。

Web 客户端移动应用程序

操作系统要求

从 Apple App Store 或 Google Play Store 下载移动应用程序。搜索 AVEVA 移动操作，然后选择安装。

最低 Android 版本：Android 6.0（API 级别 23 – Marshmallow）

目标 Android 版本：Android 9.0（API 28 级 – Pie）

最低 iOS 版本：iOS 9.0 或更高版本

如需有关设备支持的信息，请参阅浏览器和移动支持。

登录到移动应用程序

安装应用程序后，提供用户凭证并登录到应用程序以查看图形。

1. 启动移动应用程序。
2. 输入正在运行应用程序的节点的服务器名称或 IP 地址。
3. 选择或输入网站名称。提供的网站名称应该与在 InTouch HMI 应用程序管理器中指定的*自定义 URL* 相匹配。
4. 输入用户名与密码。如果需要，提供用户名的域名。
5. 选中全屏模式复选框，以使用全屏模式查看应用程序。

   此设置用于显示电视或更大显示画面的全尺寸图形，其中导航栏和标题栏在缺省条件下处于隐藏状态。
6. 选择登录。

   成功登录后，将出现 Web 客户端主页。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

首次登录需要用户凭证。应用程序将记住用户凭证供后续使用。使用清除按钮可清除登录屏幕中的详细信息。只有属于 Web 客户端用户组的用户才能访问移动应用程序。使用注销按钮可退出应用程序。

注意 ：Web 客户端移动应用程序支持匿名登录以及通过 AVEVA Identity Manager (AIM) 登录。

使用 InTouch Web 客户端移动应用程序

如果您的凭据已通过身份验证，并获得了有效的 Web 服务器许可证，将显示该应用程序。缺省情况下，将显示主页符号（如果已配置）。

1. 选择 图标。

   左侧叠加窗口显示正在运行的应用程序内的图形的层次结构。

2. 选择文件夹名称。

   此时会显示该文件夹内的图形。

3. 选择图形。

   将显示所选图形。

   您还可以平移和缩放以便以不同比例查看图形。

InTouch Web 客户端移动应用程序支持运行时语言切换。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 22 在运行时切换语言

您可以开发能够在运行时切换为另一种语言的应用程序。

要启用运行时语言切换功能，必须完成以下任务：

- 为应用程序配置多种语言。
- 导出应用程序文本进行脱机翻译。
- 翻译一个或多个导出的字典文件。
- 导入一个或多个已翻译的字典文件。

关于运行时语言切换

您可以开发能够在运行时切换为另一种语言的应用程序。

要启用运行时语言切换功能，必须完成以下任务：

- 为应用程序配置多种语言。
- 导出应用程序文本进行脱机翻译。
- 翻译一个或多个导出的字典文件。
- 导入一个或多个已翻译的字典文件。

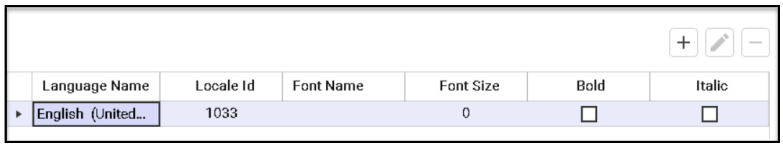
为运行时语言切换配置语言

每个 InTouch 应用程序都与用于开发该应用程序的基本语言关联。您必须配置希望支持的任何其它语言。

注意：如果结合使用语言切换与“网络应用程序开发”(NAD)，建议将 NAD 客户端节点的更改模式设置为“重新启动 WindowViewer”或“提示用户重新启动 WindowViewer”，而不是“将更改加载到 WindowViewer”或“提示用户将更改加载到 WindowViewer”。

要为运行时语言切换配置语言

1. 在 WindowMaker 中，打开要配置语言的应用程序。
2. 在文件菜单上的配置组中，选择语言。此时出现语言窗口。



   配置语言对话框显示应用程序的基本语言。

3. 选择添加。此时出现添加语言对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 指定语言与字体设置。通过配置字体设置，可以为翻译的文本定义缺省字体属性。
   - 在按名称或语言环境 ID 列表中，选择要添加的语言。如果按名称选择语言，则对应的语言环境 ID 出现在语言环境 ID 列表中，反之亦然。
   - 选择字体。此时出现字体对话框。配置字体，然后单击确定。
5. 选择确定以关闭添加语言对话框。此时配置的语言列在配置语言对话框中。
6. 要添加更多的语言，重复步骤 3 到 5。
7. 完成后，选择关闭。

删除语言：

a. 从配置语言对话框中选择要删除的语言。
b. 选择删除。

   此时出现确认删除对话框，验证您是否要从应用程序中删除语言。
c. 选择是。

   配置语言对话框将刷新并显示已删除的语言。

设置缺省语言：

a. 选择要设置为配置语言对话框上的应用程序缺省语言的语言。
b. 选择设置缺省值。

   配置语言对话框将刷新并在左下角显示应用程序的缺省语言。

为配置的语言更改字体设置

所有语言的缺省字体都是 Tahoma。字体和大小取决于基本语言中各个短语的相应设置。您可以为已配置的语言更改字体设置。由于不同语言的文本显示效果之间存在差异，您可以指定适当的字体，以确保翻译好的文本能在按钮及其它对象上正确显示。

为配置的语言更改字体设置

1. 在 WindowMaker 中，打开要为其配置的语言更改字体设置的应用程序。
2. 在文件菜单上，指向配置，然后选择语言。

   此时出现语言屏幕。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 在**常规选项**卡下的**语言**列表中，**选择**目**标语**言。
4. 双**击字体名称**和**字体大小**条目以**编辑值**。
5. 根据需要，**选**中或清除**粗体**和**斜体复选**框。
6. **进**行更改，然后**选择确定**。

添加运行时语言切换功能

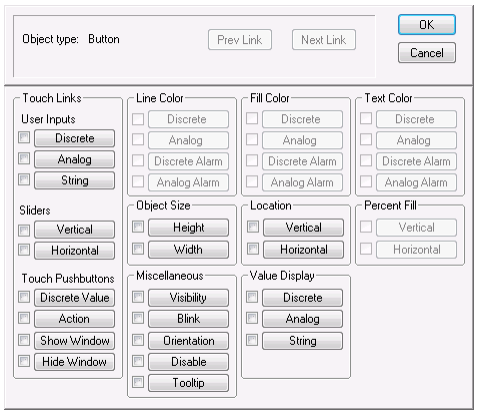
通过使用 WindowViewer **特别**菜单上的**语言**命令，运行时用户可以切换应用程序界面的语言。

您也可以在应用程序中添加一个按钮，供运行时用户切换语言。在开始之前，确保已经为应用程序配置其它语言，并且您知道该语言的语言环境 ID。如需有关为应用程序配置语言的详细信息，请参阅为运行时语言切换配置语言。

添加用于在运行时切换语言的按钮

1. 在 WindowMaker 中，打开要添加语言切换按钮的应用程序窗口。
2. 绘制按钮，并为该按钮指定文本标签，指明选中时所要切换到的目标语言。
3. 双击该按钮。

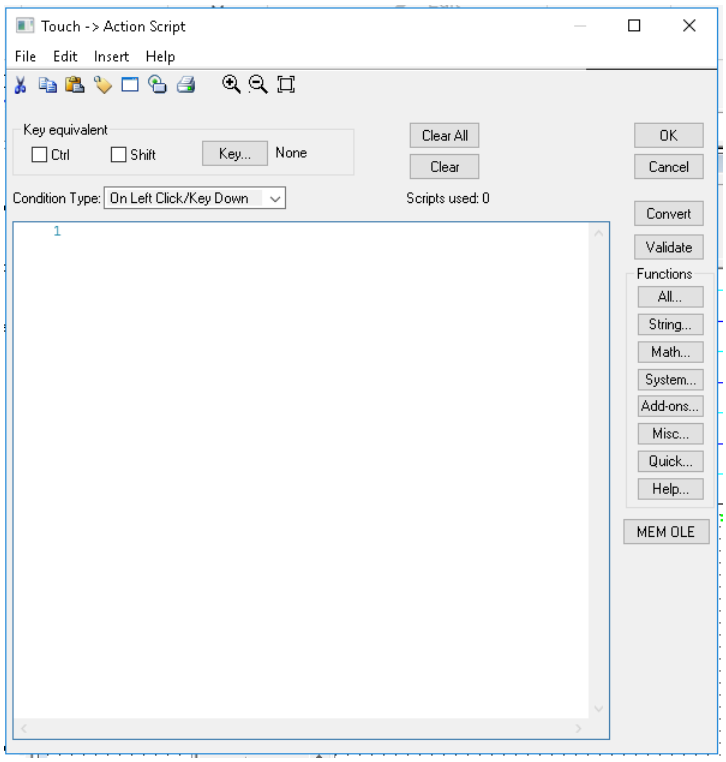
   此时出现动画选择对话框。



4. 在**触动按钮**区域中，**选择动作**。

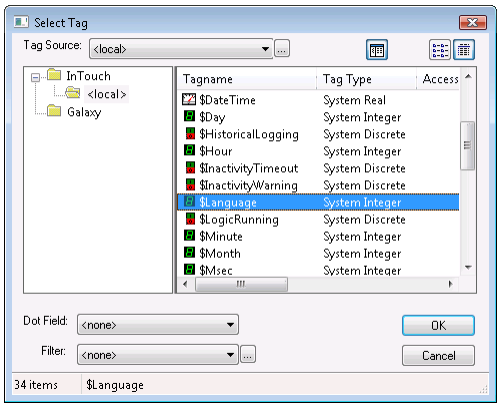
   此时出现**触动 -> 动作脚本**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 双击**触动 -> 动作脚本对话**框脚本区域中的任何位置。

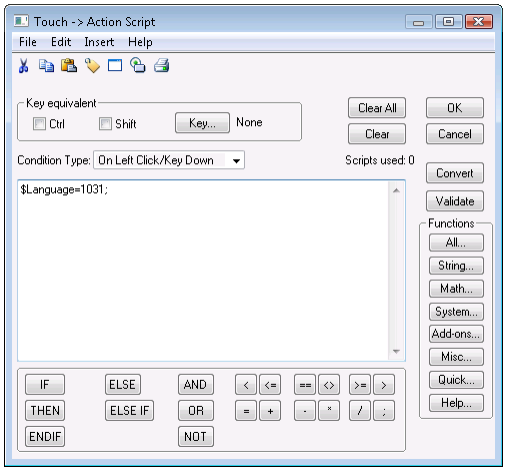
   此时出现**选择标记**对话框。



1. **选择 $Language** 系**统标记**，然后**选择确定**。

   将 **$Language** 系**统标记设**置成要指定**给**按**钮**的**语**言的**语**言**环**境 ID，然后**选择确定**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



注意：您也可以使用脚本函数 SwitchDisplayLanguage(LocaleID) 来取代 $Language 标记。

1. 选择确定以关闭对话框。

SwitchDisplayLanguage() 函数

将可以看到的静态文本与报警域的显示语言切换成提供了翻译字符串的所需语言。

类别

其它

语法

```
SwitchDisplayLanguage(LocaleID);
```

参数

LocaleID
运行时显示静态文本字符串与报警域所用的语言。

示例

在本例中，在运行时显示德语。

```
SwitchDisplayLanguage(1031);
```

另请参阅

$Language 系统标记

$Language 系统标记

如果为 InTouch 应用程序定义了多种语言，则 $Language 系统标记反映当前显示的语言的“语言 ID”值。缺省条件下，这是基本 InTouch 系统 (E/F/G/J/SC) 的语言 ID（语言环境 ID）。将它设置为另一个 ID 时，会将字符串与报警域切换为新语言中定义的值。

注意：$Language 标记被配置为本地标记，以允许在 Web 客户端中进行独立的语言切换。多个用户会话可以采用不同的语言查看 Web 客户端。Web 客户端中语言上的更改不会影响在 WindowViewer 中选择的语言，反之亦然。

类别

系统

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

整型（可读写）

导出应用程序文本进行脱机翻译

如果 InTouch 应用程序包含许多字符串，则通常发送文本字符串进行批量翻译。您可以导出应用程序的字符串进行翻译，并使用文本编辑器、XML 编辑器或 Microsoft Excel 这样的电子表格程序来组织管理它们。

您可以从以下位置导出静态文本：

- 文本对象
- 按钮文本
- SmartSymbol 中的文本
- 工具提示静态文本
- 用户消息
- 输入链接中的“打开/关闭消息”
- 输出链接中的“打开/关闭消息”
- 向导上的文本

您只有在关闭 WindowMaker 中的所有窗口之后才能导出字典。如果导出字典文件之后对应用程序进行了更改，则必须再次导出字典文件。如需详细信息，请参阅将文本导出到现有的字典文件。

一次只能导出一种语言的文本字符串。缺省条件下，InTouch HMI 打开“我的 InTouch 应用程序”文件夹。如果选择任何其它文件夹，则 InTouch HMI 将该路径设置为缺省值。为每种语言创建一个用于导出短语的新文件夹，可以便于对字典文件的管理。例如，...\My InTouch Applications\My German Files\。

InTouch HMI 为应用程序创建字典文件，并且为应用程序中的每个 SmartSymbol 创建一个单独的字典文件。应用程序字典名的格式为 application name_localeID，而 SmartSymbol 字典文件的格式则是 SSD_Name of the Symbol_localeID_GUID。

导出应用程序的字典时，文件是 .xml 文件，您可以使用 Microsoft Excel 2003 或更高版本对其进行编辑。

导出应用程序文本进行脱机翻译

1. 启动 WindowMaker，打开要导出文本字符串进行脱机翻译的应用程序。
2. 在文件菜单上，指向配置，然后选择语言。

   此时出现语言屏幕。
3. 选择导出选项卡。此时出现导出字典对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 配置导出设置。
   - 在定义的语言列表中，选择要导出的语言字典。
   - 在路径框中，输入要将字典导出到的文件夹。选择浏览以选择现有的文件夹或创建一个新文件夹。
5. 选定导出。此时显示导出进度。如果导出成功，则出现导出成功对话框。
6. 选择关闭以返回 WindowMaker 窗口，或选择关闭并启动“资源管理器”以打开包含字典文件的文件夹。

将文本导出到现有的字典文件

导出应用程序文本进行脱机翻译之后，可能需要对应用程序进行更改。如果更改应用程序，您需要再次导出文本。如需详细信息，请参阅导出应用程序文本进行脱机翻译。

如果多次导出到相同的目录，则会出现确认文件替换消息。

如果单击是，则使用自上次导出以来所添加的任何新字符串和语言信息来更新现有的 .xml 文件。如果现有的字典文件包含任何短语的翻译，并且先前已导入 InTouch HMI，则那些翻译会保留下来。如果自上次导出以来从应用程序中删除了任何短语，则也会从字典文件中删除它们。

翻译导出的字典文件

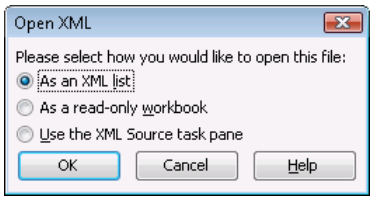
导出包含应用程序文本的字典文件之后，使用 Microsoft Excel 2003 或更高版本来编辑文本。

InTouch HMI 为导出的每种语言创建一个单独的字典文件。InTouch HMI 还为应用程序中的每个 SmartSymbol 创建一个单独的字典文件。务必确保翻译所有语言与 SmartSymbol 的所有字典文件。

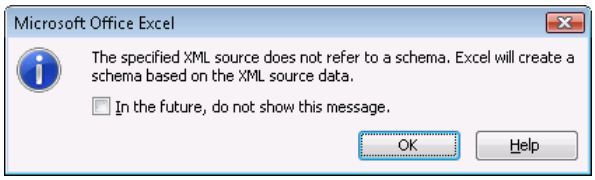
要翻译导出的字典文件

1. 在 Excel 中打开 XML 文件。此时出现打开 XML 对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



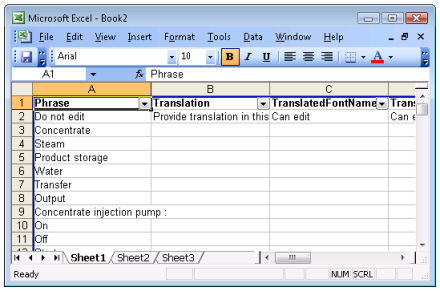
2. 选择作为 XML 列表，然后选择确定。此时可能出现一条消息。



3. 选择确定。

   此时 XML 文件在 Excel 中打开，各个列分别是：

   - 应用程序中的短语
   - 翻译人员翻译好的短语
   - 翻译的字体名
   - 翻译的字体属性
   - 翻译的字体大小
   - 基本字体属性
   - 基本字体大小
   - 上下文、短语 ID、语言 ID 以及外语 ID



要点：只要修改 **Translation**、**TranslatedFontSize**、**TranslatedFontName** 以及 **TranslatedFontProperty** 列中的数据。请勿更改任何列标题。请勿插入或删除行。

1. 在与 **Phrase** 列中的基本语言字符串对应的行的 **Translation** 列中，输入特定于语言的文本。
2. 如果需要，为翻译的字符串更改字体参数，使文本适合 WindowViewer 中提供的空间。
   - 在 **TranslatedFontName** 列中，输入字体名。
   - 在 **TranslatedFontProperty** 列中，输入字体属性的代号：

     B = **粗体**

     I = *斜体*

     U = 下划线

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

例如，如果希望文本为粗体，请在 **TranslatedFontProperty** 列中输入 **B**。如果希望文本为粗体并且带下划线，请在 **TranslatedFontProperty** 列中输入 **BU**。

3. 保存文件，将“XML 数据”用作文件类型。

要点：如果另存为其它文件类型（如 XML 电子表格），则 Excel 会更改构架，并且会导致 InTouch HMI 无法加载该文件。如果更改 XML 文件的名称，则运行时语言切换功能无法正常工作。

导入翻译的字典文件

InTouch HMI 为导出的每种语言创建一个字典文件。InTouch HMI 还为应用程序中的每个 SmartSymbol 创建一个单独的字典文件。翻译之后，必须导入每种语言的字典文件，这样才能进行这些语言的运行时语言切换。给定语言的所有字典文件都应该放置在同一文件夹中。

导入翻译的字典文件

1. 启动 WindowMaker，打开要导入翻译的字典文件的应用程序。
2. 在文件菜单上，指向配置，然后选择语言。
1. 在语言屏幕中，选择导入选项卡。

1. 配置导入设置。
   - 在定义的语言列表中，选择要导入的语言字典。
   - 在路径框中，输入要导入的字典文件所在的路径。选择浏览以浏览并选择文件。
2. 选择导入。
3. 如果重新导入 SmartSymbol 字典文件，则会提示替换现有的文件。

   如果导入成功，则出现导入成功对话框。

运行时语言切换进行报警

作为运行时语言切换设置的一部分，您也可以将报警组名称、报警注释、报警域、内部状态消息等内容本地化。除切换文本字符串的运行时语言之外，还可以配置 Alarm Viewer 与 Alarm DB View 控件中报警注释、报警状态、报警类型以及报警类的运行时语言切换。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您还可以使用翻译的警报组名称编写脚本、查询或过滤。

导出报警注释进行翻译

您可以导出报警注释进行翻译。

您可以导出以下标记的“报警状态”、“报警类型”以及“报警类”等字段：

- 包含报警注释的所有标记。
- 包含标记注释的所有标记。
- 系统标记，这样便可以将系统标记激活事件时在客户端中显示的注释进行本地化。

理解两个英文字符的应用程序 ID

在导出报警与标记注释进行本地化时，必须指定两个英文字符的应用程序 ID。该 ID 供系统内部使用，以区分同名的应用程序所生成的报警。

由于标记同时可以包含标记注释与报警注释，因此在两个英文字符的应用程序 ID 的后面添加 1 与 2 来区分这两个字段。标记注释在 ID 与标记名之间有一个 1。报警注释在 ID 与标记名之间有一个 2。例如，AA1TankLevel 是标记注释，AA2TankLevel 是报警注释。

如果导出应用程序，将删除应用程序 ID 信息。

如果报警数据库包含不带两个英文字符应用程序 ID 的旧数据，并且新记录以某个 ID 为前缀，则 Alarm DB View 控件中的报警注释查询不处理以下运算符：<、<=、> 和 >=。

导出报警注释

您可以导出报警注释进行翻译。

警示：在导出报警与标记注释之前，请备份目标目录中的任何文件，以防数据损坏或发生错误。

导出报警注释进行脱机翻译

1. 启动 WindowMaker，打开要导出报警注释进行脱机翻译的应用程序。
2. 在文件菜单上，指向配置，然后选择语言。

   此时出现语言屏幕。
3. 在语言屏幕中，选择导出报警选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 在**路径**框中，**输**入要将字典**导**出到的文件**夹**。**选择浏览**以**选择现**有的文件**夹**或**创**建一个新文件**夹**。
5. 在**两个英文字符代表唯一的应用程序**框中，**输**入两个英文字符。ID 只能包含字母数字字符，并且区分大小写。

警示：如果先前从**这**个**应**用程序中**导**出了**报**警或**标记**注**释**，**则**在下次**导**出它**们时必须**使用相同的两个英文字符的**应**用程序 ID。如果**输**入两个英文字符的新**应**用程序 ID，InTouch HMI 会**为**所有的**报**警与**标记**重新生成 ID，**这样**会**导**致**丢**失**现**有的全部翻**译**。

1. **选择导出**将信息**导**出到 XML 字典文件。

   InTouch HMI **为**配置的**每种语**言**创**建一个**单**独的**导**出文件。不同**语**言的所有字典文件都**导**出到指定的**单**个目**录**。

   对于**导**出的任何**语**言，如果存在重**复**的文件，**则**会提示文件的名称。您可以取消**导**出或**继续执**行**导**出操作。

   如果**导**出成功，**则**出**现导出成功对话**框。

   注意：如果**标记**字典中配置的**报**警注**释**的**长**度超**过** 127 个字符，或是**标记**注**释**的**长**度超**过** 46 个字符，**则**不**导**出**该报**警或**标记**注**释**。在**导**出**过**程**结**束**时**，会通知您注**释**没有**导**出到字典文件，并在**导**出目**录**中**创**建了 AlarmComment.log 或 TagComment.log 文件。

2. **选择关闭**以返回 WindowMaker 窗口，或**选择关闭并启动资源管理器**以打**开**包含字典文件的文件**夹**。

导出到现有的报警注释文件

导出**报**警与**标记**注**释**供脱机翻**译**之后，可能需要**对应**用程序**进**行更改，**这**可能又会要求再次**导**出**报**警与**标记**注**释**。如需**详细**信息，**请**参**阅**导出报警注释进行翻译。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果多次导出到相同的目录，则出现确认文件替换对话框。

选择是使用自上次导出以来所添加的任何新字符串和语言信息来更新现有的字典文件。如果现有的字典文件包含任何短语的翻译，并且先前已导入 InTouch，则那些翻译会保留下来。如果自上次导出以来从应用程序中删除了任何短语，则也会从字典文件中删除它们。

选择全部为是以更新 InTouch HMI 中配置的所有语言的现有字典文件。

选择否或全部为否以分别防止覆盖现有的文件或所有语言的现有文件。

如果再次导出，所有报警注释、报警域以及标记注释的现有翻译都会保留下来。

编辑字典文件

创建字典文件之后，需要编辑字符串。

字典文件的名称根据两个英文字符的应用程序 ID 与导出的语言进行创建。例如，如果配置的语言是“中文 (PRC)-2052”，两个英文字符的应用程序 ID 是 **AA**，则产生的文件名是 **AA_2052_ AlarmComment.xml**。编写该文件所使用的 XML 架构与运行时语言切换文件的相同。

字典文件的一般结构如下：

```
  <?xml version="1.0" encoding="utf-8" ?>
- <ArrayOfAlarmCommentPhraseItem xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  - <AlarmCommentPhraseItem Phrase="Do not edit">
      <Translation>Provide translation in this column</Translation>
      <Context>Do not edit</Context>
      <PhraseID>0</PhraseID>
      <LanguageID>0</LanguageID>
      <ForeignLanguageID>0</ForeignLanguageID>
    </AlarmCommentPhraseItem>
  - <AlarmCommentPhraseItem Phrase="System">
      <Translation />
      <Context>$System : System Tag Comment</Context>
      <PhraseID>AA1$System</PhraseID>
      <LanguageID>1033</LanguageID>
      <ForeignLanguageID>2052</ForeignLanguageID>
    </AlarmCommentPhraseItem>
  </ArrayOfAlarmCommentPhraseItem>
```

输入翻译字符串的翻译。请勿更改其它任何信息。

您可以覆盖有些“报警状态”、“报警类型”以及“报警类”的值。这些值的最大允许长度是 50 个字符。

对于 InTouch 生成的报警，以下“报警状态”值可以覆盖：

要覆盖的值	要显示的缺省字符串
未确认但已恢复正常	未确认但已恢复正常
已确认且已恢复正常	已确认且已恢复正常
未确认报警	未确认报警
已确认报警	已确认报警

对于 InTouch 生成的报警，以下“报警类型”值可以覆盖：

要覆盖的值	要显示的缺省字符串
SPC	SPC
HIHI	HIHI
HI	HI
LO	LO

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要覆盖的值	要显示的缺省字符串
LOLO	LOLO
MINDEV	MINDEV
MAJDEV	MAJDEV
ROC	ROC
DSC	DSC
OPR	OPR
LGC	LGC
DDE	DDE
SYST	SYST
USER	USER
PRO	PRO
LOGON_FAILED	LOGON_FAILED
对于 InTouch 生成的报警，以下“报警类”值可以覆盖：	
要覆盖的值	**要显示的缺省字符串**
DEV	DEV
ROC	ROC
DSC	DSC
事件	事件
值	**值**

导入翻译的报警注释

翻译字符串之后，必须导入每种语言的字典文件，这样才能进行这些语言的运行时语言切换。

导入翻译的报警注释字典文件之后，它们会复制到应用程序目录内各自语言的文件夹中。

现有应用程序的任何已翻译的报警注释都会被导入的文件的内容覆盖掉，应用程序版本 ($AppVersion) 按 1 进行递增。

要从其它节点导入多个字典文件以支持多个节点上报警域的本地化，请将其它节点上翻译的字典文件复制到单个目录。选择此目录作为导入路径。多个字典文件可以在单个导入操作中进行导入。InTouch HMI 基于所导入的语言自动创建文件路径。

导入已翻译的报警注释文件

1. 启动 WindowMaker，打开要导入翻译的字典文件的应用程序。
1. 在文件菜单上，选择配置，然后选择语言。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现语言对话框。

1. 在语言屏幕中，选择导入报警选项卡。

2. 在路径框中，输入要导入的字典文件的路径，或者选择浏览以浏览并选择文件。
3. 选择导入。如果字典文件中没有完成任何翻译，则会出现一个对话框，指示没有翻译的字典文件可以导入。
4. 选择确定。如果导入成功，则出现导入成功对话框。

导出报警组名称进行翻译

您可以导出报警组名称进行翻译。

警示：在导出报警与标记注释之前，请备份目标目录中的任何文件，以防数据损坏或发生错误。

导出报警注释进行脱机翻译

1. 启动 WindowMaker，打开要导出报警注释进行脱机翻译的应用程序。
2. 在文件菜单上，指向导出。
3. 选择本地化，然后选择报警消息。
4. 此时出现导出报警域屏幕
5. 在“选择要导出的语言”列表中，选择要导出以进行翻译的语言。
6. 在“选择目录”框中，选择要导出文件的位置。
7. 在两个英文字符代表唯一的应用程序框中，输入两个英文字符。ID 只能包含字母数字字符，并且区分大小写。

   警示：如果先前从这个应用程序中导出了报警或标记注释，则在下次导出它们时必须使用相同的两个英文字符的应用程序 ID。如果输入两个英文字符的新应用程序 ID，InTouch HMI 会为所有的报警与标记重新生成 ID，这样会导致丢失现有的全部翻译。

8. 选择导出将信息导出到 XML 字典文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch HMI 为配置的每种语言创建一个单独的导出文件。不同语言的所有字典文件都导出到指定的单个目录。

对于导出的任何语言，如果存在重复的文件，则会提示文件的名称。您可以取消导出或继续执行导出操作。

如果导出成功，则出现导出成功对话框。

注意：如果标记字典中配置的报警注释的长度超过 127 个字符，或是标记注释的长度超过 46 个字符，则不导出该报警或标记注释。在导出过程结束时，会通知您注释没有导出到字典文件，并在导出目录中创建了 AlarmComment.log 或 TagComment.log 文件。

9. 选择**关闭**以返回 WindowMaker 窗口，或选择**关闭并启动资源管理器**以打开包含字典文件的文件夹。

导入翻译的报警组名称

翻译字符串之后，必须导入每种语言的字典文件，这样才能进行这些语言的运行时语言切换。

导入翻译的报警注释字典文件之后，它们会复制到应用程序目录内各自语言的文件夹中。

现有应用程序的任何已翻译的报警注释都会被导入的文件的内容覆盖掉，应用程序版本 ($AppVersion) 按 1 进行递增。

要从其它节点导入多个字典文件以支持多个节点上报警域的本地化，请将其它节点上翻译的字典文件复制到单个目录。选择此目录作为导入路径。多个字典文件可以在单个导入操作中进行导入。InTouch HMI 基于所导入的语言自动创建文件路径。

导入已翻译的报警组文件

1. 启动 WindowMaker，打开要导入翻译的字典文件的应用程序。
1. 在文件菜单上，选择导入。
2. 选择“本地化”，然后选择报警消息。

   此时出现导入报警域对话框。
1. 在“选择要导入的语言”列表中，选择所需的语言。
2. 在路径框中，输入要导入的字典文件的路径，或者选择浏览以浏览并选择文件。
3. 选择导入。如果字典文件中没有完成任何翻译，则会出现一个对话框，指示没有翻译的字典文件可以导入。
4. 选择确定。如果导入成功，则出现导入成功对话框。

在运行时测试语言切换功能

在应用程序中启用运行时语言切换之后，需要测试语言切换功能。报警与标记注释以及报警域的语言切换只能在 Alarm Viewer 与 Alarm DB View 控件中查看。

在处理本地化版的报警与标记注释时，应注意以下问题：

- 如果报警或标记注释没有翻译成 $Language 所指定的语言，则报警客户端显示缺省的注释。
- 如果 Alarm Viewer 控件从多个供应器进行查询，则远程节点上的报警注释、标记注释和报警域也会显示为翻译的版本（前提是应用程序具有翻译的远程节点应用程序字典文件）。
- 如果确认报警并提供注释，则报警客户端显示此注释，而不是本地化版的报警注释。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- Alarm Viewer 控件处于冻结模式时，即使进行语言切换，也不会显示切换后的语言。取消控件的冻结时，控件使用已翻译的字符串进行更新。
- Alarm Viewer 控件的本地化仅用于控件的显示。即便切换了语言，所有脚本函数仍返回缺省的字符串。
- Alarm DB Logger 只将数据的缺省语言字符串存储在数据库中。本地化版的字符串不存储在数据库中。
- 报警域的唯一 ID（如 EVENT 与 ACK）是预定义的，在不同节点上的多个字典文件中使用的是相同的 ID。报警客户端从第一个加载的字典文件中选择翻译，并忽略其它字典文件中的翻译。理想的情况是，所有字典文件中的报警域在一种语言中都应该有相同的翻译。对于给定语言的相同报警状态，多个报警客户端（Alarm DB View 与 Alarm Viewer 控件）使用相同的翻译。
- 对于报警注释，翻译的文本截断为 131 个字符；对于标记注释，截断为 160 个字符。

要试语言切换功能

1. 在 WindowViewer 中打开应用程序。
2. 在文件菜单上，指向配置，选择语言，然后选择要切换到的语言的名称。

   此时会相应地加载并显示翻译的字典文件（如果存在）中的信息。
3. 如果添加了用于切换语言的按钮，则可以选择此按钮来测试脚本。

将本地化版的文件分发到网络应用程序开发客户端

包含本地化版的报警注释、标记注释以及报警域的文件作为 InTouch 应用程序的一部分，分发到各个“网络应用程序开发”(NAD) 客户端。接收到包含报警注释的更新文件时，必须重新启动 WindowViewer 才能在支持的报警客户端中看到翻译的报警注释。

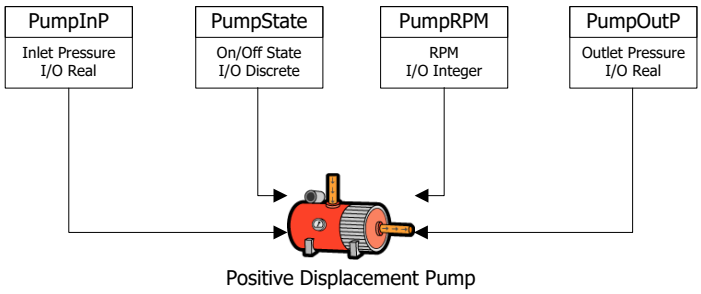
如果结合使用语言切换与“网络应用程序开发”(NAD)，请将 NAD 客户端节点的更改模式设置为“重新启动 WindowViewer”或“提示用户重新启动 WindowViewer”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 23 标记

InTouch 人机界面 (HMI) 应用程序是生产环境中各种组件的图形化表示。工厂操作员使用这种图形界面来监视和管理生产过程。

下图显示一个泵的示例，它是生产过程中的一个组件。该泵有一些属性及关联的值。压力、RPM 及状态都是泵的属性，这些属性的值通过 HMI 进行监控。



在 InTouch HMI 应用程序中，*标记*代表数据项。通过使用标记，可以将特定的组件属性当作数据项从生产环境中访问。在上图中，PumpState 标记指出泵是打开还是关闭的。对于生产环境中的各种组件，如果要在 InTouch 应用程序中监视或控制其属性，则可以为它们创建标记。

对于从生产组件中采集的不同类型的数据，可以使用不同类型的标记。例如，PumpState 标记返回 On/Off 布尔值，以指出泵是正在运行还是已经停止。对于要成为应用程序一部分的数据类型，需要指定适当类型的 InTouch 标记。

标记查看器入门

标记查看器是一款外部应用程序，可让您在运行时观察、监视标记并修改标记的值。它会根据其所在的报警组向您提供应用程序中的所有可用标记的列表，这些标记将按照层次结构排列。只有在运行 WindowViewer 时，您才可以运行“标记查看器”。“标记查看器”只能显示本地 InTouch 应用程序中的可用标记且不支持远程引用。

要使用“标记查看器”，您首先需要在 WindowMaker 中启用它。然后您就可以在运行时启动该应用程序了。一次只能打开一个“标记查看器”实例。

启用 Tag Viewer

您可以在设计时配置 WindowViewer 属性，以便运行“标记查看器”。您还可以配置 WindowViewer 菜单，这样“标记查看器”选项就会显示在特别菜单中。

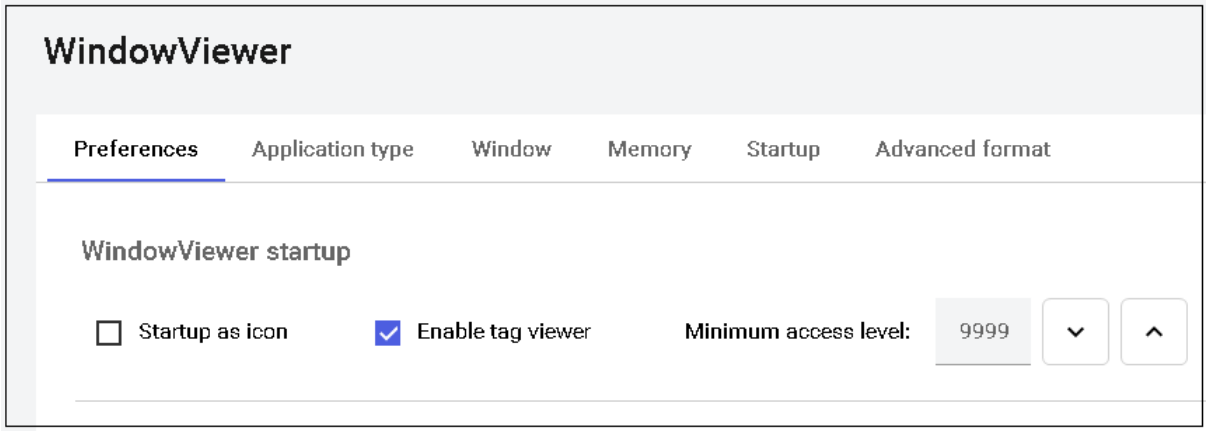
修改这些属性后，如果 WindowViewer 已打开，那么您必须重新启动 WindowViewer 以便应用这些更改。

启用 Tag Viewer

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择 **WindowViewer**。

   此时出现 **WindowViewer** 配置屏幕。
3. 选择首选项选项卡。

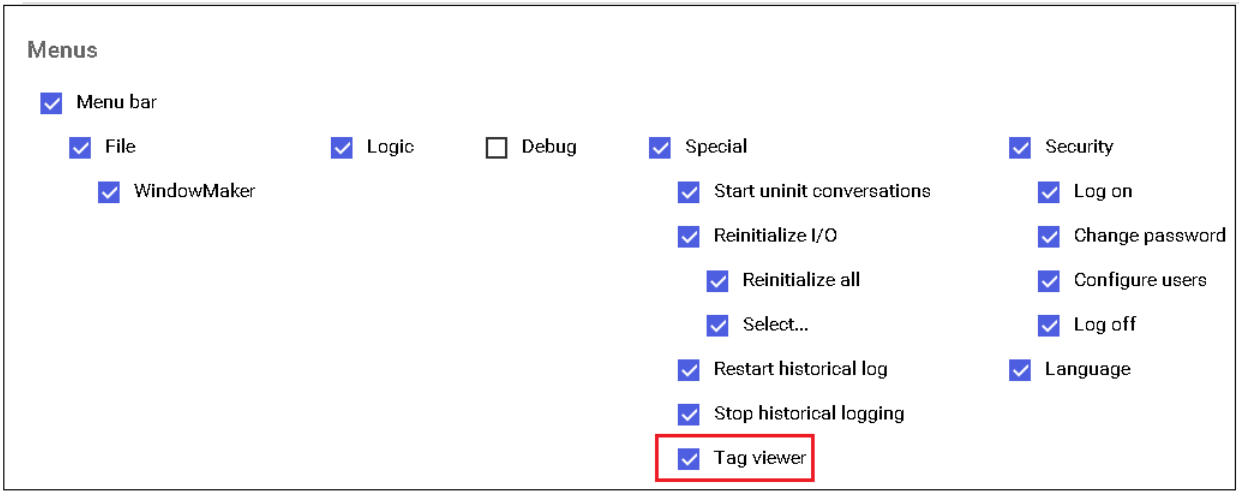
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在 **WindowViewer 启动**区域中，选中**启用标记查看器**复选框，这样即可启动“标记查看器”。缺省条件下，该复选框处于未选中状态。
5. 在**最小访问级别**框中，配置运行“标记查看器”应用程序所需的安全访问级别。缺省值为 9999，您可以键入 0 至 9999 之间的任意值。

   注意：如果您的访问级别低于最低访问级别，则无法在运行时启动“标记查看器”。

6. 选择窗口选项卡。



1. 在**菜单**区域中选中**标记查看器**复选框，以便在 WindowViewer 的**特别**菜单中启用**标记查看器选项**。
2. 选择**确定**。
3. 更改任何参数后，必须重新启动 WindowViewer 才能应用更改。

启动标记查看器

只有在 WindowViewer 正在运行，且已在设计时启用了“标记查看器”的情况下，您才可以启动“标记查看器”。您可以从 WindowViewer 菜单中或通过调用 LaunchTagViewer() 函数的脚本启动“标记查看器”。该函数可以通过应用程序脚本（OnStartup 和 OnShutdown）以外的任意类型脚本执行。如果未启用“标记查看器”，那么您将无法通过调用函数启动“标记查看器”且记录器中会记录一条警告消息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

安全性

要运行“标记查看器”，您必须拥有足够的安全权限。下表介绍了可用的 InTouch 用户安全级别。

安全级别	描述
ArchestrA	如果选择此选项，则必须执行以下操作： 1. 在 **WindowViewer** 属性屏幕中，设置**访问级别 >= 最小访问级别**。 2. 登录到 WindowViewer，然后启动“标记查看器”。 如果您正在运行托管的 InTouch 应用程序，则适用该选项。
InTouch	如果选择此选项，则必须执行以下操作： 1. 在 **WindowViewer** 属性屏幕中，设置**访问级别 >= 最小访问级别**。 2. 登录到 WindowViewer，然后启动“标记查看器”。 如果您正在运行独立的 InTouch 应用程序，则适用该选项。
无	如果选择此选项，则任意用户都可在运行时启动“标记查看器”，而无需登录 WindowViewer。
操作系统	如果选择此选项，则必须执行以下操作： 1. 在 **WindowViewer** 属性屏幕中，设置**访问级别 >= 最小访问级别**。 2. 登录到 WindowViewer，然后启动“标记查看器”。 如果您正在运行独立的 InTouch 应用程序，则适用该选项。
CONNECT	如果您正在运行独立 InTouch 应用程序，且已在**许可证模式选项**卡下选择了**连接体验选项**，则该选项适用。

如果您打开的是 InTouch 应用程序，那么您必须通过再次登录来重新验证自己的身份，然后才能运行“标记查看器”。如需有关配置用户的详细信息，请参阅《AVEVA™ InTouch HMI 管理指南》中的管理用户并设置其授权级别。

注意 ：如果 InTouch 的安全级别为“无”，则无需再次登录。

登录

1. 打开 WindowViewer。
2. 在特别菜单上，选择安全性，然后选择登录。
3. 通过有效的用户 ID、口令和需要的最低访问级别登录。

注意：您可以作为管理员登录，也可以创建拥有足够权限的用户 ID。

启动标记查看器

1. 打开 WindowViewer。
2. 在特别菜单上，选择标记查看器。此时会出现“标记查看器”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

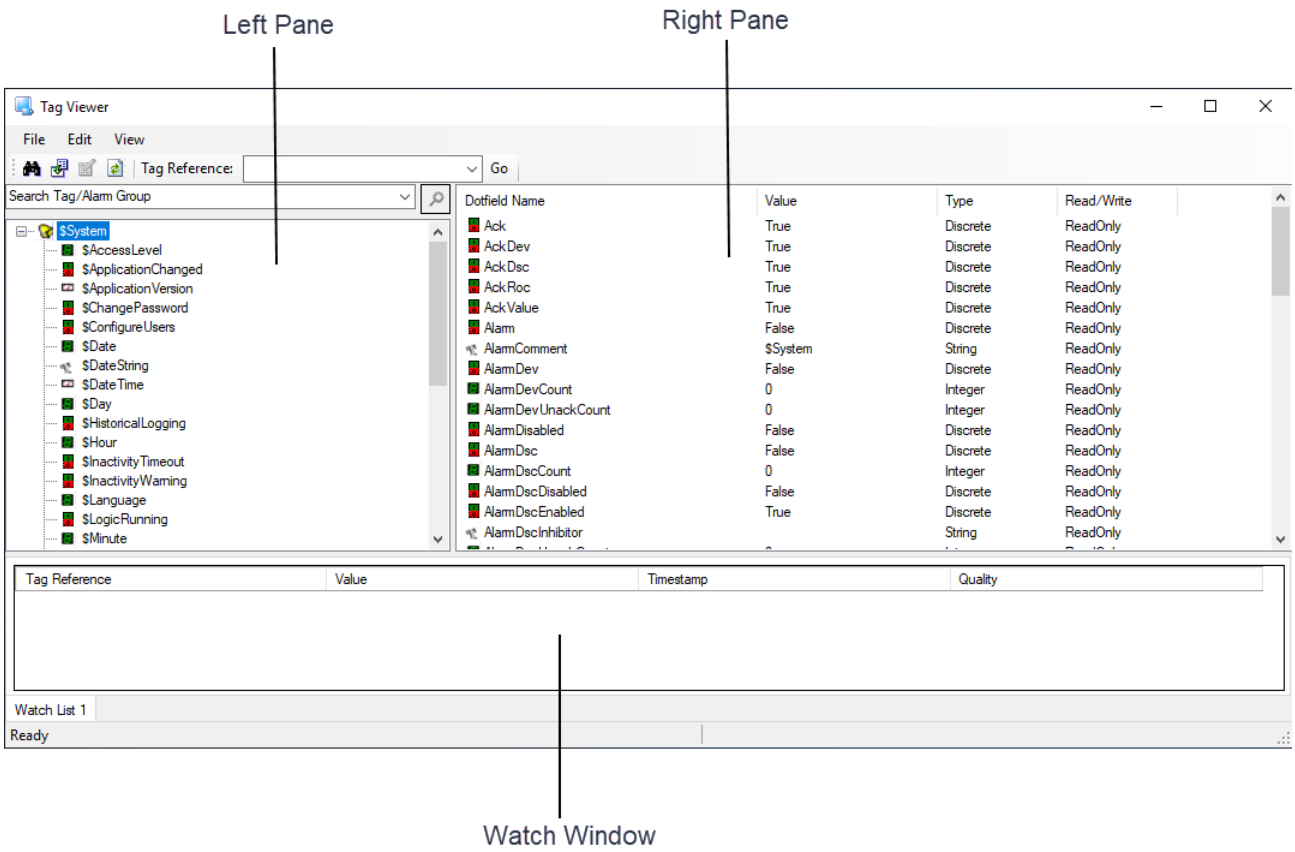
通过脚本启动标记查看器

1. 打开 WindowMaker。
2. 使用 LaunchTagViewer() 函数配置任意脚本（应用程序脚本“启动时”和“关闭时”除外）。
3. 打开 WindowViewer。
4. 执行脚本以启动“标记查看器”。

在标记查看器中导航

“标记查看器”窗口包括以下三部分：

- 左侧窗格，它会根据标记所在的报警组按层次结构显示标记。
- 右侧窗格，它会显示左窗格中所选标记或报警组的所有可用点域的列表。
- 底部的观察窗口，它会显示您要监视的标记的运行时值。



关闭标记查看器

您可以通过以下方法关闭“标记查看器”：

- 关闭“标记查看器”
- 关闭 WindowViewer
- 更改登录的用户

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用标记查看器

您可以在运行时使用“标记查看器”执行以下操作：

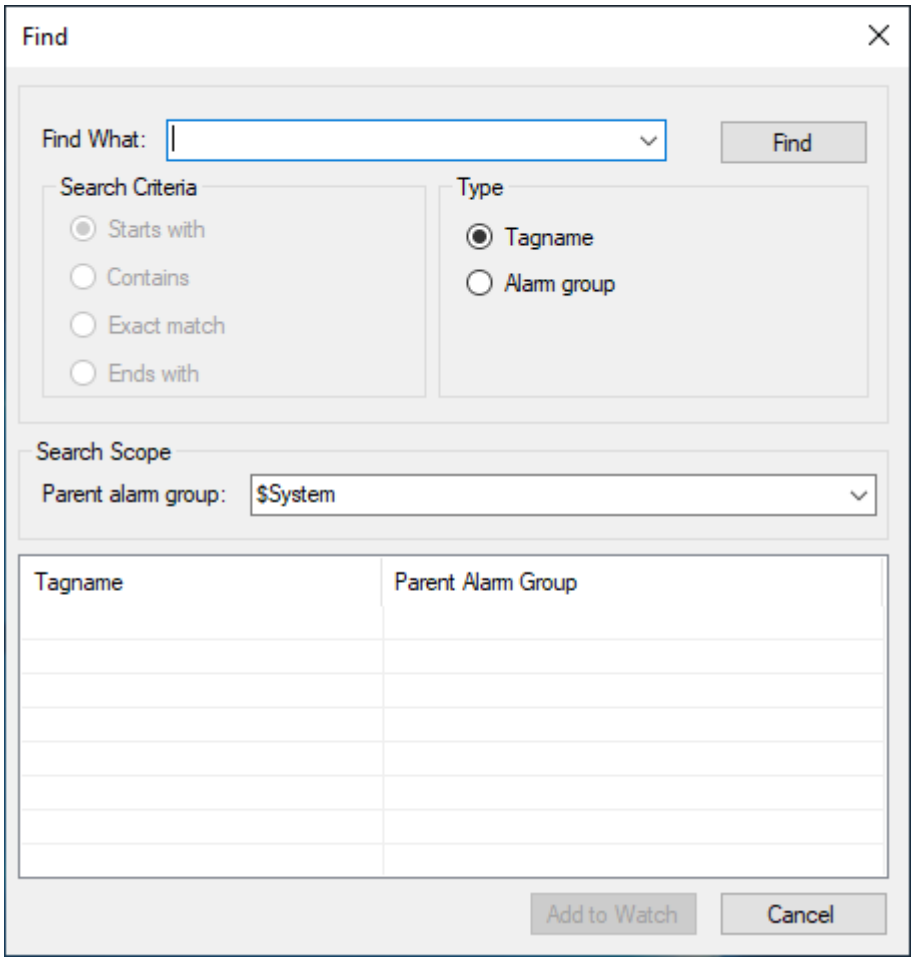
- 搜索标记
- 查看标记
- 修改标记属性
- 添加和填充观察窗口
- 管理观察窗口

搜索标记

您可以使用查找对话框在 WindowViewer 中搜索特定标记。

搜索标记

1. 在编辑菜单上，选择查找标记/报警组。此时出现查找对话框。



2. 在查找内容框中，输入要查找的标记或报警组的名称。缺省条件下，该框会显示您上一次搜索的标记或报警组。查找内容框还会保留搜索历史。如果您是首次搜索标记或报警组，则查找内容框将为空。

注意：关闭“标记查看器”后，搜索历史会被删除。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 在搜索标准区域中，选择以下某个选项：
   - 开头是：选择该选项可以查找以您在查找内容框中输入的文本开头的标记或报警组。缺省条件下，该选项处于已选择状态。
   - 其中包括：选择该选项可以查找包含您在查找内容框中输入的文本的标记或报警组。
   - 完全匹配：选择该选项可以查找与您在查找内容框中输入的文本相匹配的标记或报警组。
   - 结尾是：选择该选项可以查找以您在查找内容框中输入的文本结尾的标记或报警组。
4. 在类型区域中，选择以下某个选项：
   - 标记名：选择该选项可以查找标记。缺省条件下，该选项处于已选择状态。
   - 报警组：选择该选项可以查找报警组。
5. 在父报警组框中，输入标记所属的报警组的名称。如果您保留缺省值 **$System**，那么系统将不会按任何报警组过滤搜索结果。
6. 选择查找。此时会出现标记或报警组列表。
7. 在搜索结果中，双击标记名以便在“标记查看器”中选择该标记。

注意：要将标记添加到观察窗口中，请选择添加到观察。

执行快速搜索

如果需要，您可以通过“标记查看器”快速搜索标记。

执行快速搜索

1. 在左侧窗格上方的搜索框中，输入标记名的全部或部分文本。

   注意：搜索框还会保留搜索历史，直到您关闭“标记查看器”。

2. 按 **Enter** 键或单击搜索图标。系统会选择一个以您输入的文本开头的标记。

   注意：如果您输入了以通配符 **(*)** 开头的文本，则系统会查找包含该文本的任意标记名。

3. 再次选择搜索图标可继续搜索。

   注意：您还可以通过在输入文本后按 **F3** 来执行快速搜索。

管理标记

“标记查看器”可让您在运行时观察标记并监视标记的值。您可以查看应用程序中的所有可用标记。这些标记会根据其报警组按树结构排列。

查看标记

您可以在“标记查看器”中查看标记的详细信息。

查看标记

1. 打开“标记查看器”。
2. 在左侧窗格中，查看应用程序中的可用标记的列表。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：“标记查看器”只能显示 InTouch 应用程序中的可用标记，不支持远程引用。

3. 在左侧窗格中，选择某个标记可在右侧窗格中查看它的详细信息。此时会出现以下信息：
   - 点域名称：显示所选标记或报警组的点域。
   - 值：显示点域的值。如果您选择点域，则该值会被更新。
   - 类型：显示点域类型（如整型、离散型）。
   - 读取/写入：指明标记是只读标记还是可供修改。
4. 将标记添加到观察窗口中，以便在运行时查看它的点域值。如需有关将标记添加到观察窗口中的详细信息，请参阅本附录中的将标记或点域添加到观察窗口中。

修改标记属性

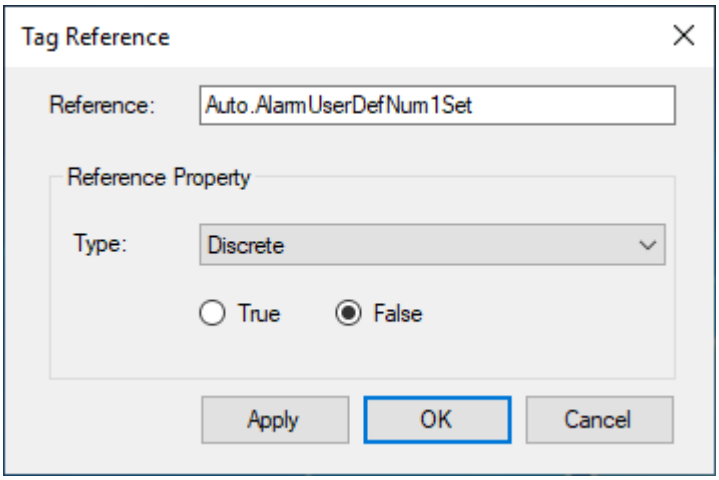
您可以在“标记查看器”中修改标记的属性。只有在点域已标记为“可读写”或“只写”时，您才可以修改标记的属性。如果您是通过只读许可证运行 WindowViewer 的，则无法修改任何标记。

如果您尝试修改以下内容，则系统会要求您重新验证自己的身份：

- 指向使用“安全写入”配置的 Galaxy 属性的间接标记。
- 指向使用“验证写入”配置的 Galaxy 属性的间接标记。在这种情况下，还必须提供其他用户的登录详细信息。

修改标记属性

1. 打开“标记查看器”。
2. 在左侧窗格中，选择标记可修改其属性。标记的详细信息会显示在右侧窗格中。
3. 执行以下任意操作：
   - 在右侧窗格中选择相应属性，然后选择工具栏上的**继续**。此时出现标记引用对话框。

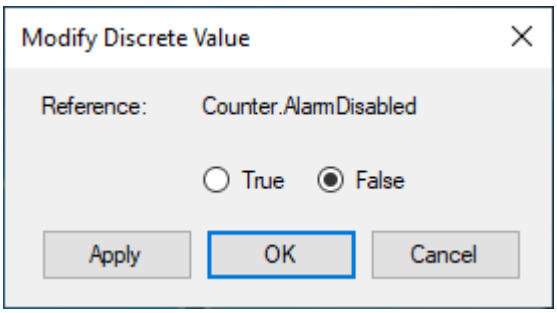


   - 在类型列表中，选择标记类型并更改相应的值。

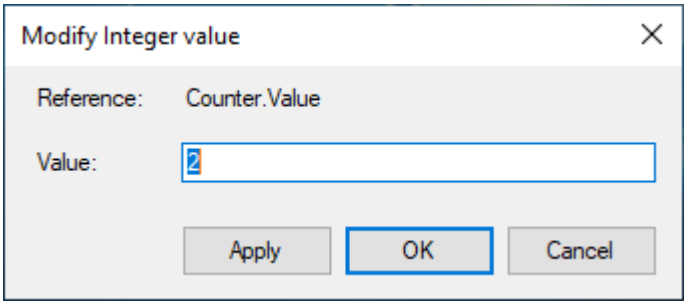
     如果您要更改点域的数据类型，请双击右侧窗格中的任意值。此时出现修改对话框。对话框会根据点域的数据类型而有所不同。您可以修改以下数据类型的值：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

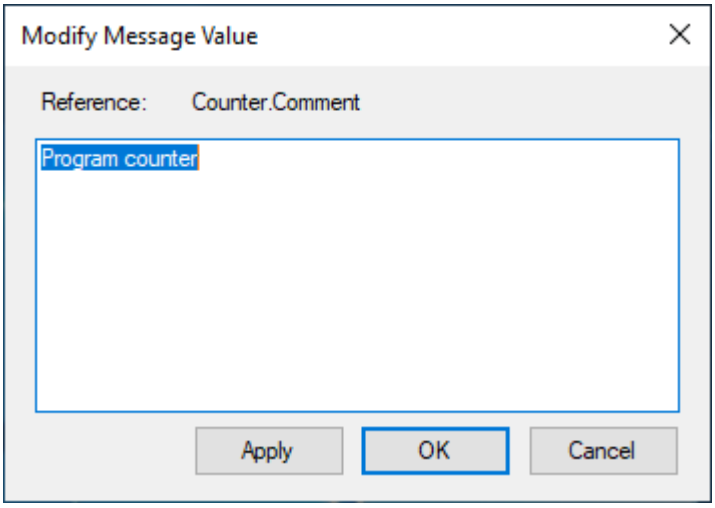
离散值：您可以选择 **True** 或 **False**。



数字：您可以输入
-2147483648 到 2147483647 之间的任意数字。



消息：您可以输入不超过 131 个字符的任意文本消息。



- 使用鼠标右键单击项目，然后修改修改。此时出现修改对话框。修改相应值。

4. 选择应用。相应值会更改。

管理观察窗口

缺省条件下，观察窗口会显示在“标记查看器”的底部。您可以使用该窗口在运行时查看标记的值。您可以通过创建其它观察窗口将相关标记组合在一起。如果需要，您还可以删除观察窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

添加观察窗口

如果您要对标记进行分组并查看这些标记，可以创建其它观察窗口。新观察窗口会以选项卡的形式添加到缺省观察窗口旁。您至多可以创建 50 个观察窗口。

添加观察窗口

1. 使用鼠标右键单击观察窗口，然后选择添加观察窗口。此时会添加新观察窗口。缺省条件下，新观察窗口称为“观察列表 <n>”。

   注意：要重命名观察窗口，请使用鼠标右键单击观察窗口选项卡，然后单击重命名选项卡。

2. 根据需要将标记添加到观察窗口中。
3. 通过选择选项卡在多个观察窗口之间移动。

将标记或点域添加到观察窗口中

您可以将标记或点域添加到观察窗口中，以便在运行时监视它们的值。最多可以将 2000 个标记或点域添加到观察窗口中。

注意：如果您将报警组添加到观察窗口中，那么该报警组下的所有标记都会添加到观察列表中。

将标记添加到观察窗口中

1. 执行以下任意操作：
   - 在左侧窗格中，使用鼠标右键单击标记，然后选择添加到观察。此时标记会添加到观察窗口中。
   - 在左侧窗格中选择标记，然后将其拖放到观察窗口中。
   - 使用鼠标右键单击观察窗口，然后选择添加标记引用。
   - 在右侧窗格中，使用鼠标右键单击任意点域，然后选择添加到观察。此时点域会添加到观察窗口中。

   注意：您可以选择多个标记或点域并将其添加到观察窗口中。

2. 选择任意列标题以便根据列对表格进行排序（升序或降序）。

向组标记中添加分隔符

您可以通过在标记组后添加分隔符将类似标记组合在一起。如果您保存观察窗口，则观察窗口中的分隔符会自动保存。

向组标记中添加分隔符

1. 在观察窗口中，选择要在其后添加分隔符的行。
2. 使用鼠标右键单击观察窗口，然后选择添加分隔符。分隔符会添加到所选行之后。
3. 选择分隔符以及标记引用，然后将它们拖放到其它位置以改变观察窗口中的标记顺序。

注意：如果您在观察窗口中对信息进行排序，则系统会删除分隔符。

备份与恢复观察窗口

您可以将观察窗口组保存为备份并在需要时重新加载这些窗口。

保存观察窗口组

如果需要，您可以保存已打开的所有观察窗口。您可以稍后加载这组观察窗口并监视标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

保存观察窗口组

- 在文件菜单上，选择保存观察列表并保存观察窗口。缺省条件下，InTouch 会将观察窗口保存在当前的应用程序目录下的观察列表文件夹中。

注意：要将观察窗口保存到其它位置，请选择文件菜单上的观察列表保存为。

加载观察窗口组

您可以打开之前已保存的任何观察窗口。

加载观察窗口

1. 在文件菜单上，选择加载观察列表。此时出现选择文件对话框。

   注意：您还可以使用鼠标右键单击观察窗口，然后选择加载观察列表。

2. 浏览到保存文件的位置并打开文件。

   注意：在打开观察窗口时，现有窗口会替换成新窗口。只有当前应用程序中的有效引用会添加到观察窗口中。

I/O 引用

“标记查看器”中显示的 I/O 引用或间接标记的行为与 WindowViewer 中窗口的行为类似。

- 如果您从观察窗口中删除 I/O 或间接标记，则会取消预订 I/O 引用。
- 如果您未向观察窗口中添加 I/O 或间接标记，但其点域已刷新，或您已选择左侧窗格中的标记，那么“标记查看器”将会预订 I/O 引用。只要标记处于选定状态，这一预订就不会改变。
- 如果关闭“标记查看器”，那么“标记查看器”之间对所有 I/O 或间接标记的预订都会取消。
- 关闭“标记查看器”或从观察窗口中删除标记不会对 WindowViewer 中的标记产生任何影响。

标记查看器的持续性

就其 InTouch 应用程序而言，“标记查看器”在以下方面应具有持续性：

- 主窗口的位置和大小。
- 左侧窗格中标记树结构的大小、左侧窗格中的点域列表以及观察窗口。

就其 WindowViewer 会话而言，“标记查看器”在以下方面应具有持续性：

- 左侧窗格中的所选标记
- 观察窗口
- 状态栏和工具栏

注意：关闭 WindowViewer 后，“标记查看器”与其 WindowViewer 会话的持续性将不复存在。

观察窗口文件格式

您保存的观察窗口文件会采用 XML 文件格式。XML 架构为：

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="InTouchTagViewer">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="Watch" minOccurs="1" maxOccurs="unbounded">
          <xs:complexType>
```

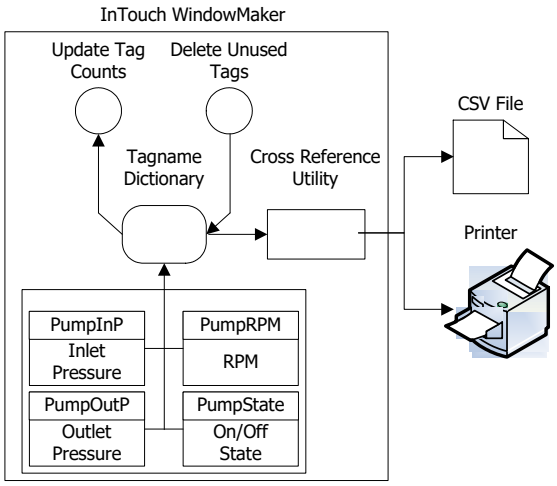
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
            <xs:sequence>
              <xs:element name="ReferenceString" type="xs:string" minOccurs="0"
maxOccurs="unbounded"></xs:element>
              <xs:element name="Separator" minOccurs="0" maxOccurs="unbounded"></
xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

减少标记使用

InTouch 应用程序中可使用的最大标记数取决于您的许可证。您的许可证会阻止正在运行的应用程序的并发标记计数超出最大限制。

您可以跟踪 InTouch 应用程序中的标记使用。下图显示"交叉引用"实用程序如何通过分析"标记名字典"的内容来生成标记使用报告。



您应该维持运行 InTouch 应用程序所需的最少标记数。维持最少标记计数的一种方法是删除未使用的标记。您必须在删除未使用的标记之前更新标记计数。

InTouch 与 Historian

Historian 是储存工厂过程数据的一种实时与历史数据库。它结合了关系型数据库的储存容量与实时系统的速度。作为 Microsoft SQL Server 的增强，Historian 获取工厂数据的速度获得了极大的提高。它还将工厂数据与 InTouch HMI 及其它相关产品的事件、摘要、生产和历史数据集成到一起。

InTouch HMI 可以将历史标记数据存储到远程历史储备库。在这种情况下，要访问 InTouch 应用程序，必须共享 InTouch HMI 应用程序目录并配置 Historian。如果使用 Historian 存储 InTouch HMI 历史数据，则必须使用 WindowMaker 中的"分布式名称管理器"来指定与数据库的连接。InTouch HMI 可以在"趋势向导"中显示 Historian 数据库中存储的历史数据。您需要安装 SQL Server 客户端组件，并浏览 Historian 计算机以访问历史标记数据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

确定标记使用

InTouch HMI 会为本地“标记名字典”中定义的所有标记维护一份使用计数。标记计数不包括内部系统标记。

远程标记不在“标记名字典”中定义。每次在应用程序内引用远程标记时，InTouch 标记计数均会递增。如需有关 InTouch 如何统计远程标记引用的详细信息，请参阅根据许可证确定最大远程标记数。

请确保已完成以下操作后，再删除未使用的标记：

- 关闭 WindowViewer。
- 更新本地标记计数和对远程标记的引用。
- 生成“交叉引用”标记报告。
- 使用“交叉引用”实用程序来查找应用程序中的标记使用。
- 在 WindowMaker 中保存应用程序。

确定标记计数

您可以更新应用程序“标记名字典”中当前定义的本地标记的计数。您也可以更新应用程序中引用其它节点上的远程标记的计数。

显示本地标记计数

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在文件菜单上，选择配置，然后选择 **WindowMaker**。

   此时出现 WindowMaker 配置屏幕。
3. 选择显示标记计数。

   选择此选项时，必须读取整个“标记名字典”才能更新显示的标记计数。更新“标记名字典”可能需要更长的时间。
4. 选择保存。此时出现一条消息，指出必须重新启动 WindowMaker 才能使配置更改生效。
5. 关闭 WindowMaker。
6. 在 WindowMaker 中重新启动应用程序。

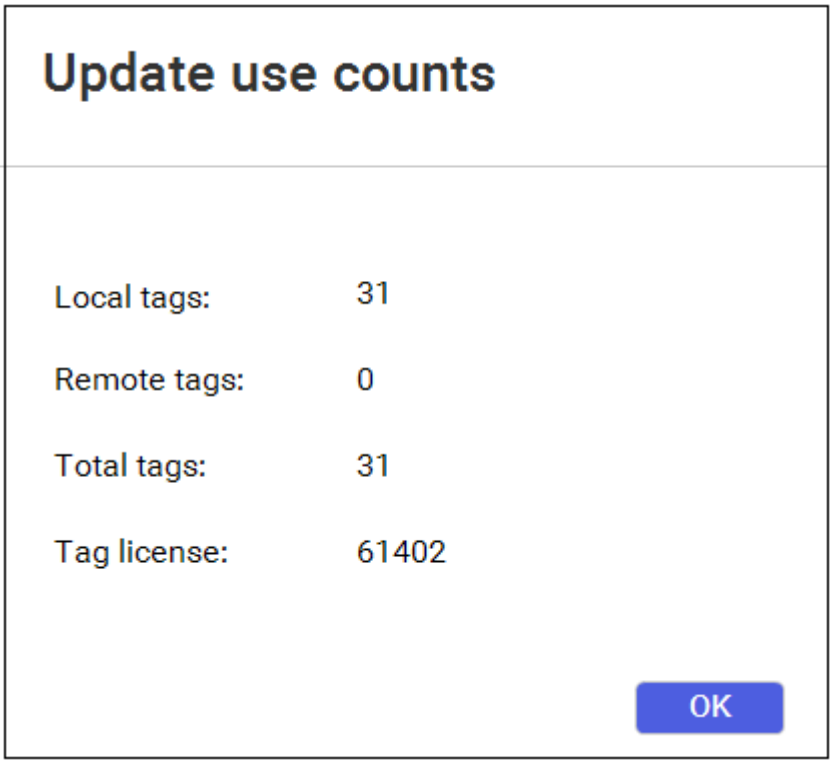
   应用程序的“标记名字典”中所定义的本地标记数显示在 WindowMaker 菜单栏的右侧。

更新应用程序中包括远程引用在内的标记总数

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 关闭所有打开的窗口。
3. 在主页菜单上的标记组中，选择更新使用计数。

   计算完成之后，对话框显示应用程序的本地标记数以及远程标记引用数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



该对话框还包括基于 InTouch 许可证的标记总数和最大标记使用数。

根据许可证确定最大远程标记数

产品许可证决定了您的 InTouch 应用程序中可以使用的标记数。许可证不仅决定本地标记计数，还决定了引用远程节点上标记源的标记数。

如需有关 InTouch 许可证如何决定应用程序中可以使用的远程引用标记数的详细信息，请参阅 InTouch 许可证。

查找标记的使用位置

通过使用“交叉引用实用程序”，可以生成一份在线报告来显示 InTouch 应用程序中的标记使用情况。

“交叉引用实用程序”报告显示以下位置中的本地标记、远程标记以及 SuperTag：

- 动画链接
- 向导
- 所有类型的 InTouch 脚本
- QuickFunction
- ActiveX 控件
- 可选 InTouch 组件，如“SQL 访问管理器”和 Recipe Manager。
- 工业图形引用

使用“交叉引用实用程序”创建报告

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在主页菜单上的标记组中，选择交叉引用。

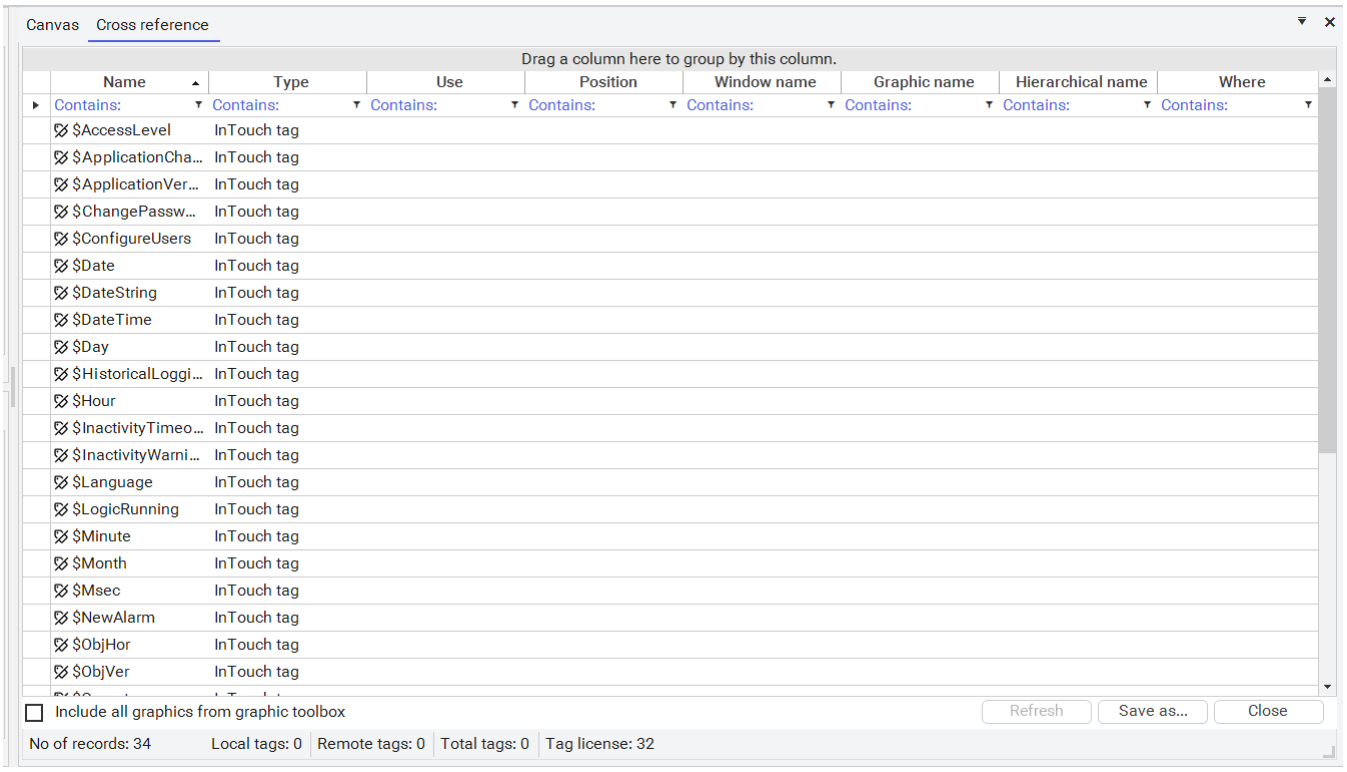
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现交叉引用网格。

“交叉引用”网格中列出了与当前应用程序的“标记名字典”中定义的本地以及远程标记相关的所有记录。

了解交叉引用实用程序报告

该报告使用网格格式列出引用类型的名称、标记引用的类型、使用对象、在窗口上的位置、窗口名、图形名称、包含该标记引用的工业图形的完整层次结构路径，以及在脚本中使用该标记时的条件类型。



例如：

名称	类型	使用对象	位置	窗口名	图形名称	层次名称	情形
NSelect	InTouch 标记	键脚本					F6 键按下时
PLCSim.SP3	对象属性	工业图形	自：(0,0) 至：(826,455)	Section5	SA_Meters2	SA_Meters.SA_Tank_Vessel.SA_OperatingRange	

报告底部的状态栏显示以下内容：

- **记录数**：与应用程序中的标记相关的记录总数。此值将根据所应用的过滤器动态发生改变。
- **本地标记**：在此节点上创建的标记总数。
- **远程标记**：引用远程标记的标记总数。
- **标记总数**：InTouch 应用程序中的标记总数。
- **标记许可证**：每个许可证所允许的标记总数。

InTouch “交叉引用”实用程序报告中出现一些图标，以显示状态与使用情况。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

图标	描述
	该标记或 SuperTag 定义在应用程序的“标记名字典”中，但未指定给某个对象。
	该标记或 SuperTag 用在动画链接、InTouch QuickScript 或工业图形中。
	工业图形正在引用标记或 SuperTag。
	标记或 SuperTag 指定给某个动画链接。
	该标记或 SuperTag 用在某个“应用程序”脚本中。
	对于选择的所有脚本，都会显示此图标。双击脚本名称可以只读模式查看该脚本。
	该标记或 SuperTag 用于“窗口”脚本。
	该标记或 SuperTag 用于“数据改变”脚本。
	该标记或 SuperTag 用于“条件”脚本。
	该标记或 SuperTag 用于“键”脚本。
	该标记或 SuperTag 用于 QuickFunction。
	该标记或 SuperTag 用于“ActiveX 事件”脚本。
	按窗口进行交叉引用时，在使用所显示的标记或 SuperTag 的窗口的名称前面，会出现此图标。双击图标可以查看该窗口中使用的所有标记。
	显示的标记或 SuperTag 用在 SQL 应用程序中。
	显示的标记或 SuperTag 用在 Recipe Manager 应用程序中。
	此标记用作报警约束。

从图形工具箱中包括图形

- 选择刷新可查看最新的标记信息。如果编辑标记信息或编辑嵌入到窗口上的符号时“交叉引用”窗口是打开的，那么会启用“刷新”按钮。
- 选择包括图形工具箱中的所有图形复选框可以查看未用于任何窗口但存在于图形工具箱的标记。缺省条件下，该复选框处于未选择状态。

  例如：标记目录中配置了 10 个标记，其中 4 个标记在未嵌入到任何窗口的图形工具箱的图形中被引用，4 个标记在窗口上放置的图形中被引用。缺省情况下，将显示 4 个标记（放到窗口上的标记）。如果选择该复选框，则显示 8 个标记（窗口中的 4 个标记 + 图形中的 4 个标记）。

显示的图形受以下条件限制：

- 符号内的远程标记引用，InTouch 窗口中未使用的引用不会显示。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 对象符号，InTouch 窗口中未使用的符号不会显示。
- 如果在交叉引用实用程序中选择该复选框，也将为其它操作（例如“更新使用计数”和“删除未使用标记”）设置该条件。

InTouch 标记的交叉引用标记在 ArchestrA 或 Situational Awareness Library 符号中的使用

InTouch“交叉引用”实用程序的功能现已提升，在应用程序的标准 InTouch 标记交叉引用列表中加入了工业图形。交叉引用在搜索工业图形时可以根据下面显示的条件来执行，此功能将加入到“交叉引用实用程序的 InTouch 标记使用”中。

- 引用了 InTouch 标记的自定义属性
- 直接引用了 InTouch 标记的动画
- 直接引用了 InTouch 标记的客户端脚本
- 嵌入符号对 InTouch 标记的引用

“交叉引用”实用程序还将在工业图形中搜索属性引用。属性引用是“对象属性”的一部分。标记使用报告的例外：

1. 对象属性引用：
   - 交叉引用实用程序标记用法报告中将不支持且不会显示使用相关引用（如 me.I1）的对象属性。
   - 使用 <InTouchViewApp 实例名称>.<标记名> 语法的引用将不会视为 InTouch 标记引用。此引用将被视为“对象属性”。
     示例：InTouchViewApp1.Tag1
   - 不使用“InTouch:<标记>”语法的所有引用都将视为“对象属性”引用。
2. 用作脚本函数参数的客户端脚本引用将不会视为 InTouch 标记或对象属性引用。

   例如：引用为字符串参数
   SetCustomPropertyValue("CP1", "InTouch:Tag1", false);

   示例：引用作为属性参数
   SetBad(InTouch:Tag1);
3. 如果工业图形所引用的 InTouch 标记在当前 InTouch 应用程序中未定义，那么该符号的类型会显示为“未定义”。
   示例：对象符号
   ArchestrA 对象“UD_001”的实例符号有“S1”、“S2”和“S3”。
   实例符号“UD_001.S1” 引用 InTouch 标记“Tag51”。
   将 UD_001.S1 嵌入 UD_001.S2；
   将 UD_001.S2 嵌入 UD_001.S3；
   将 UD_001.S3 嵌入 InTouch window "Win1"。
   并将 UD_001.S2 嵌入窗口“Win1”。

实例符号可以在实例或其衍生模板中定义。可以使用绝对或相对引用嵌入实例。

在交叉引用实用程序中显示标记用法

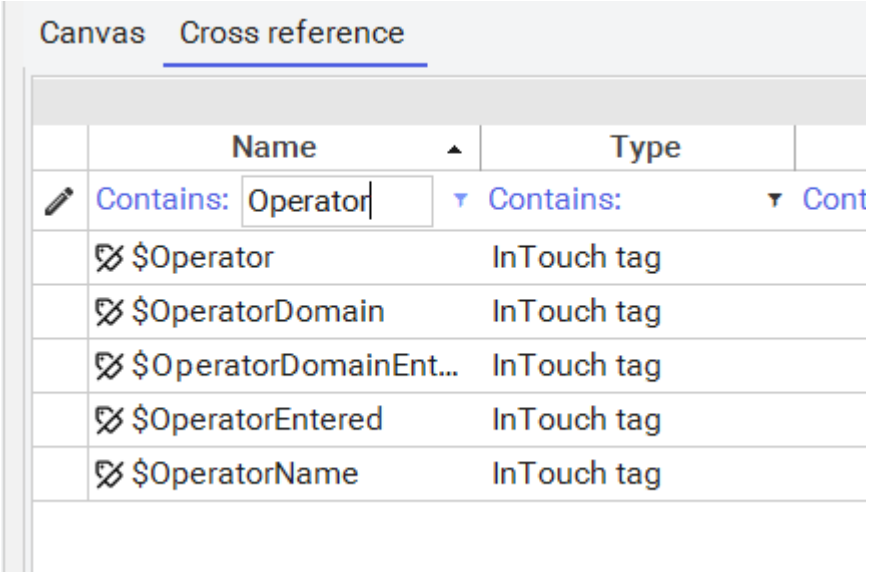
您可以通过过滤、排序或分组选项的方式，来修改交叉引用标记用法报告的显示方式。应用过滤将更新显示的记录数。排序和分组只会修改记录在网格上的显示方式。

使用过滤选项细化标记用法报告

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1. 使用过滤器图标 ( ) 从以下选项列表中进行选择：包含、不包含、开头是、结尾是、等于、不等于、为空、不为空以及自定义。
2. 根据所选择的过滤项输入搜索项。

   网格中会显示符合过滤选项和搜索项的所有记录。



使用自定义过滤选项

使用自定义过滤选项，可以添加多个标记，并创建复杂的搜索表达式。

排序标记用法报告

标记用法报告中的每个列都可以按照升序或降序的顺序进行排序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1. 选择列标题。

   网络中的所有记录都将根据该列进行排序。箭头方向决定排序顺序。

2. 再次选择该列标题可更改排序顺序。

分组标记用法报告

以列组合模式组织标记用法报告。

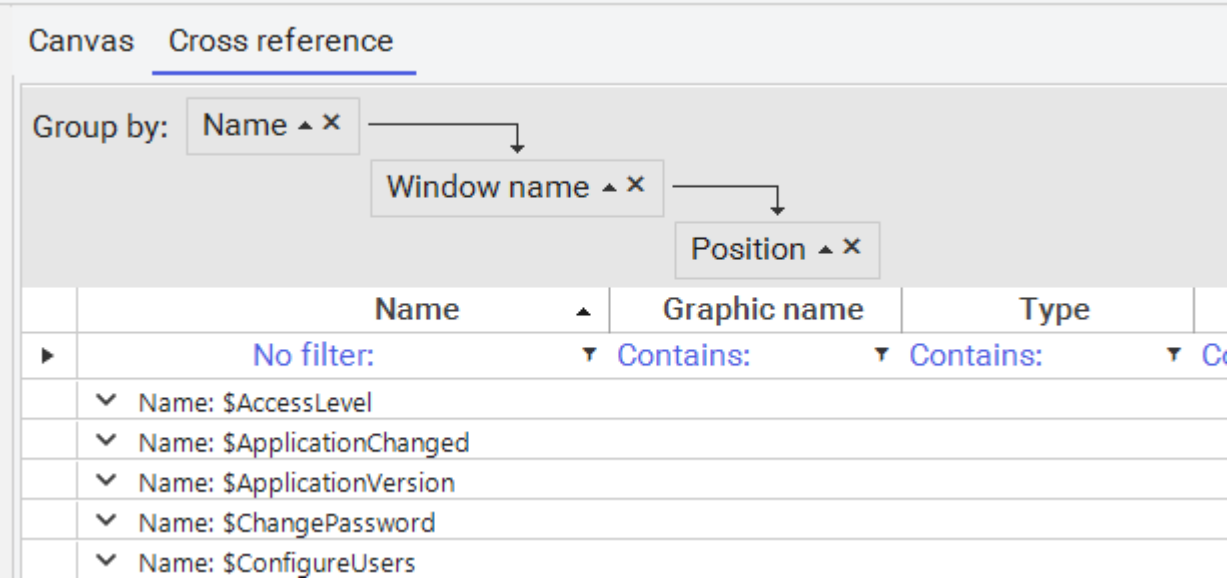
1. 选择一个列名称，并将其放置到列名称上面的空白区域。

   此时会出现一个包含该列名称的方框。

2. 再选择一个列名称，将其放置到第一个方框的上面或者下面。

   实用程序将在这两个列之间自动创建一种层次关系。

3. 要使两个列处于同一级别，请将第二个列名称放置在第一个方框顶端。
4. 选择该模式中的列名称方框，以升序或降序排序各个列。



保存与打印标记交叉引用列表

您可以将交叉引用标记列表保存到文件，然后使用任何支持 .csv 文件格式的应用程序查看该文件。

交叉引用标记列表文件会按照名称、在应用程序中的使用方式以及所在窗口的名称列出所有标记。您可以使用 Excel 或支持 .csv 文件的其它任何程序打开交叉引用标记列表文件。

您还可以打印“标记名字典”的内容。通过打印“标记名字典”的内容，可以显示应用程序中使用的数据库项目。您可以指定所要查看的详细信息级别。

您可以将此报告发送到打印机，或将它保存到文件。

要保存交叉引用文件

1. 在交叉引用实用程序对话框中，选择另存为。此时会出现另存为对话框。
   指定文件的名称与位置。
2. 选择保存。此时标记使用情况文件会保存到指定的文件夹中。

保存到 .csv 文件的标记使用报告将取决于以下条件：

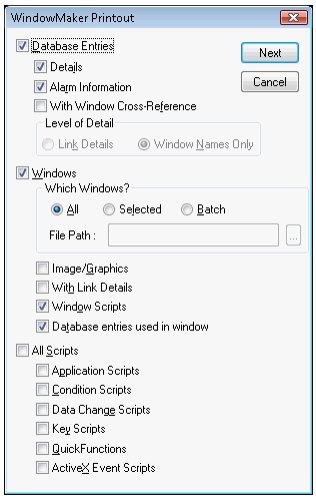
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 如果未选择搜索条件，那么所有记录都将保存到 .csv 文件。例如：如果交叉引用网络中的记录总数是 1000 并进行精细搜索，那么记录数会降至 320。如果使用“另存为...”，那么只会将所选的 320 条记录保存到 .csv 文件。
- 如果已对记录应用分组样式，那么保存.csv 时会保留该分组样式。

要打印“标记名字典”的内容

1. 在 WindowMaker 中打开 InTouch 应用程序。
2. 在快速访问工具栏上，选择打印。

   此时出现 **WindowMaker** 打印件对话框。



3. 如果希望打印“标记名字典”中的标记信息，请选择数据库项目。

   如果选择数据库项目，以下选项将变为活动状态：

   - 选择详细信息，以在报告中包含标记的详细信息。
   - 选择报警信息，以在报告中包含标记报警信息。
   - 选择包含窗口交叉引用，以打印包含窗口交叉引用的所有“标记名字典”项目。如果选择此选项，请指定要打印的详细信息级别。

     链接详细信息会打印标记所在位置与动画链接的详细信息。

     仅窗口名仅打印交叉引用的窗口的名称。
4. 选择下一步。此时出现**选择输出目标**对话框。
5. 选择相应选项以打印“标记名字典”的内容，或是将输出发送到文本或 .html 文件。
6. 选择打印。

您可以将交叉引用标记列表保存到文件，然后使用任何支持 .csv 文件格式的应用程序查看该文件。

删除未使用标记

在更新使用计数之后，可以从 InTouch 应用程序中删除未使用的标记。您可以从“标记名字典”中一次删除一个标记，也可以一次删除多个标记。

从应用程序中删除包含标记的窗口或在链接脚本中更改标记时，标记计数不会自动更新。InTouch 将这些标记视为正在使用，因此会阻止删除它们。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：删除“标记名字典”中的单一标记：如果针对工业图形引用中的未使用标记已禁用删除按钮，请运行更新使用计数来启用“删除”按钮。

必须先更新标记计数以便从总数中删除未使用的标记，然后才能删除这些标记。如需有关更新标记计数的详细信息，请参阅确定标记计数。

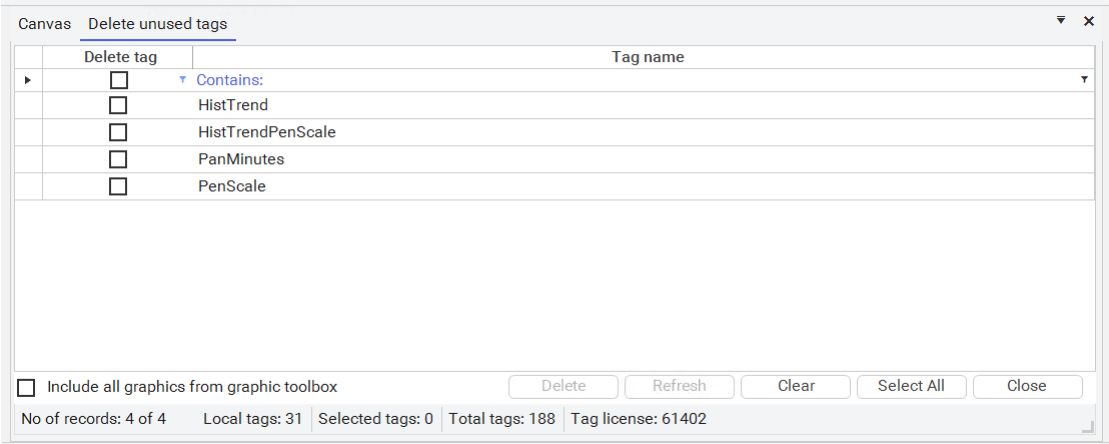
注意：仅用于报警的标记没有使用计数，可能会不小心被删除。要确保使用计数中包含仅用于报警的标记，需要在窗口或 QuickScript 中使用它们。

删除多个未使用的标记

1. 如果 WindowViewer 正在运行，请关闭它。
2. 在 WindowMaker 中打开 InTouch 应用程序。
3. 在主页菜单上的标记组中，选择删除未使用标记。

   此时出现删除未使用标记对话框，其中列出未使用的标记。

   该对话框底部的状态栏中会列出记录数、本地标记数、所选标记数、标记总数以及标记许可证计数。



4. 使用过滤器 ( ) 选项可细化未使用标记的列表。选择该过滤器选项并输入搜索项。

   此时会显示符合该条件的标记。
5. 选择该列标题来排序各列。
6. 选择要删除的标记。要选择当前过滤条件的所有标记，请选择全选。

   注意：状态栏显示已选择的标记数（例如：记录数：1 个 / 共 14 个）。即使列条件改变，仍然会保持该标记选择。“所选标记”值将反应用户已选择的标记总数，与过滤条件无关。要查看所选标记的完整列表，请清除所有过滤条件。
7. 选择清除，清除当前过滤所选择的标记。
8. 选中包括图形工具箱中的所有图形复选框可以只查看未用于任何图形或窗口的标记。这些标记可以安全删除，因为它们在图形或包含图形的窗口中未被引用。

   例如：标记目录中配置了 10 个标记，其中 4 个标记在未嵌入到任何窗口的图形工具箱的图形中被引用，4 个标记在窗口上放置的图形中被引用。缺省条件下，会显示 6 个标记（2 个标记未使用 + 4 个标记来自图形工具箱中的图形，这些图形未嵌入到任何窗口）。如果选中该复选框，则只显示 2 个未使用的标记。
9. 选择删除，删除并非根据该过滤条件选择的所有标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时会出现一条确认消息。该消息将指定要删除的标记数。

10. 选择确定进行确认。

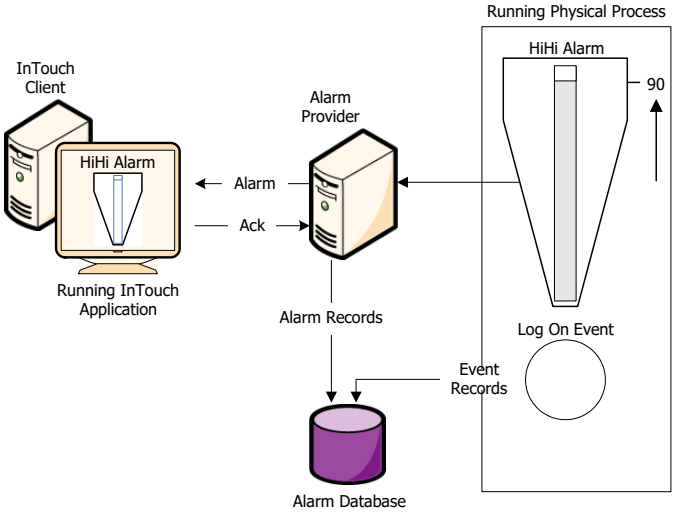
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 24 报警

通过创建可生成报警与事件的 InTouch 应用程序，可以通知操作员有关生产过程活动的状态。

- 报警向运行时操作员警告可能导致潜在问题的过程条件。通常，您设置一个在过程值超过定义的极限时触发的报警。操作员通常必须确认该报警。
- 事件代表正常的系统状态消息。通常事件在发生某种系统条件时触发，如操作员登录到 InTouch 应用程序。操作员不必确认事件。

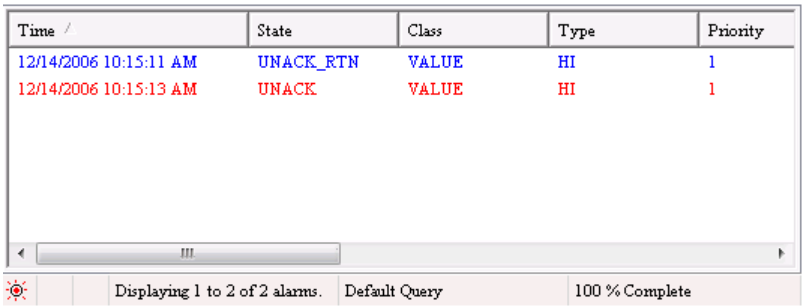
下图显示 InTouch HMI 在应用程序运行期间如何处理报警与事件。报警与事件数据保存到报警数据库。



您可以配置任何标记来监视事件。每次标记值改变时，都有一个事件消息记录到报警系统。事件消息包含：值是如何改变的，是操作员、I/O、脚本还是系统促使了这个改变。

查看当前报警

使用 InTouch Alarm Viewer ActiveX 控件可以查看报警。Alarm Viewer 控件有滚动条、可调整大小的列、多个报警选项、更新状态栏、动态显示类型，并且可以根据报警类型显示不同的颜色。

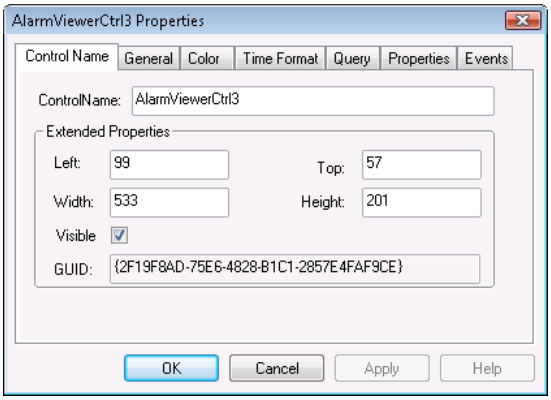


我们建议使用 Alarm Viewer 控件来查看 InTouch 报警。不过，您可以继续使用“分布式报警”对象从 InTouch 7.1 以前版本所创建的应用程序中查看报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置 Alarm Viewer 控件

从 WindowMaker 中，您可以设置 Alarm Viewer 控件选项，也可以设置用户可以在 Alarm Viewer 控件的运行过程中修改的选项。您在 **AlarmViewerCtrl** 属性对话框中设置这些选项。

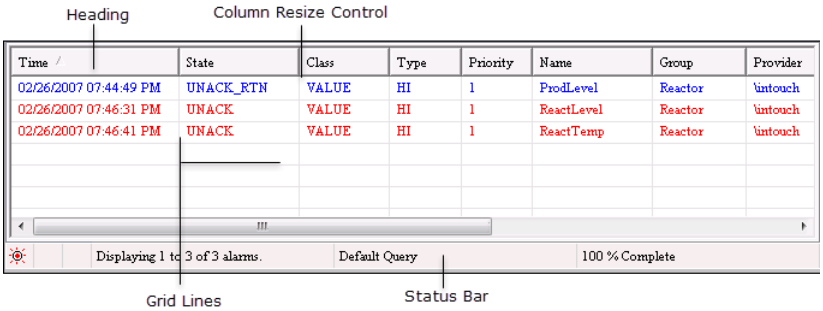


配置网格外观

配置 Alarm Viewer 控件的视觉外观时，可以：

- 包含状态栏。
- 包含列标题。
- 包含显示行与列的水平与垂直网格线。
- 包含允许用户调整列宽的运行时选项。
- 设置可视化元素的颜色。

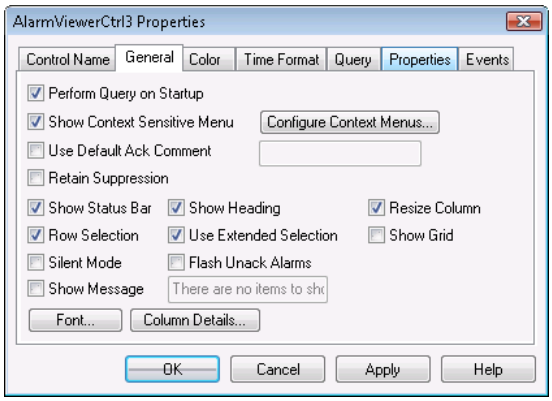
下图显示所有可视化属性都处于活动状态时 Alarm Viewer 控件的外观。



配置视觉外观

1. 使用鼠标右键单击 Alarm Viewer 控件，然后选择属性。此时出现 **AlarmViewerCtrl** 属性对话框。
2. 选择常规选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



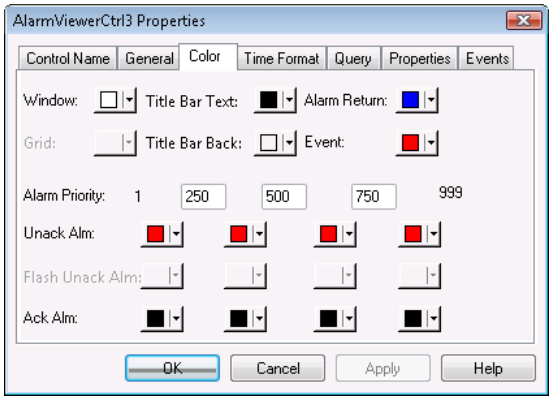
3. 设置视觉外观。配置以下选项之一：

选项	描述
在启动时执行查询	使用缺省查询属性自动开始更新控件。如果查询未在应用程序启动时运行， 则需要运行使用 Requery() 函数的脚本来更新网格。在运行时，网格的快捷菜单上也提供重新查询选项。
显示上下文相关菜单	启用运行时的鼠标右键快捷菜单。
使用缺省确认注释	控制操作员确认报警时是否出现缺省注释。如果选中此框并且输入了一个字符串，则该字符串用作运行时的缺省注释。 如果未选择此框，则操作员确认报警时，会出现一个对话框，要求输入可选注释。该对话框可以填写，也可以留为空白。
保持抑制	更改报警查询时，在报警查询之间保持报警抑制。
显示状态栏	显示或隐藏 Alarm Viewer 控件底部的状态栏。
行选择	使用户可在运行时选择单独的行。每行代表一条报警记录。用户可以选择多个报警。
无提示模式	如果选择无提示模式，则 Alarm Viewer 控件在运行时不显示错误消息。如果未选择它，则“报警显示”显示弹出式错误消息。 在任一种情况下，错误消息总是都发送到 Log Viewer。
显示消息	显示文本框中输入的消息。这是没有报警时显示的消息。
显示标题	显示或隐藏 Alarm Viewer 控件顶部的标题栏。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

选项	描述
使用扩展选择	允许用户在按住 **CTRL** 或 **SHIFT** 键的情况下，结合鼠标按钮来同时选择多个报警。仅当选中“行选择”复选框时才可用。
闪烁未确认报警	使未确认的报警每秒闪烁一次，直至它们得到确认为止。 在 WindowViewer 中冻结报警显示时，不会使未确认的报警停止闪烁。
更改列尺寸	如果选择“更改列尺寸”，则用户可以在运行时调整列的宽度。否则，列宽度是静态的，仅可以从 WindowMaker 中设置。
显示网格	如果选择“显示网格”，则 Alarm Viewer 控件显示分隔报警显示行与列的水平与垂直线。如果未选择它，则网格不可见。

4. 选择应用。
5. 选择颜色选项卡。



6. 选择调色板按钮，以便给 Alarm Viewer 控件的可视化元素指定颜色。
7. 选择应用以保存选择的颜色。
8. 选择确定。

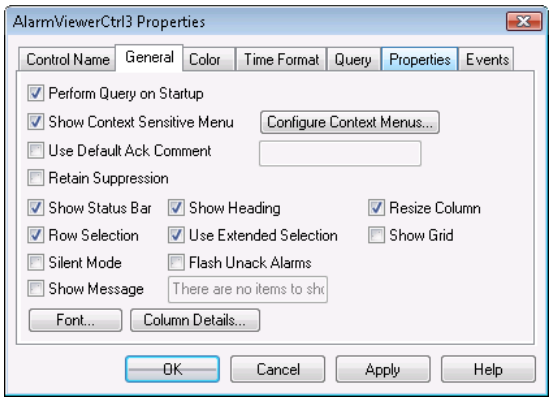
配置字体显示

您可以为 Alarm Viewer 控件配置文本外观。

配置字体属性

1. 使用鼠标右键单击 Alarm Viewer 控件，然后选择属性。此时出现 **AlarmViewerCtrl** 属性对话框。
2. 选择常规选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 选择字体。此时出现标准的 Windows 字体对话框。配置字体，然后单击确定。
4. 选择确定。

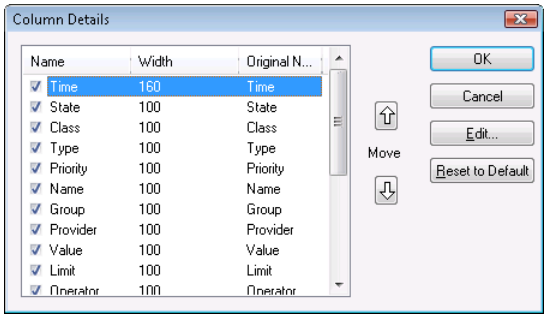
配置显示列明细

对于 Alarm Viewer 控件，您可以：

- 选择各个列并进行排序。
- 以像素为单位设置列宽。
- 重命名列。

要配置显示列明细

1. 使用鼠标右键单击 Alarm Viewer 控件，然后选择“属性”。此时出现 **AlarmViewerCtrl** 属性对话框。
2. 选择常规选项卡。
3. 选择列明细。此时出现“列明细”对话框。



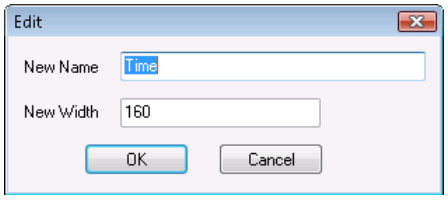
4. 在名称列中，选中要显示的列的名称旁边的复选框。您至少必须从列表中选择一个列。

列	显示
时间	在“时间格式”选项卡中指定的时间。
状态	报警的状态。
类	报警类别。
类型	报警类型。
优先级	报警优先级。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

姓名	标记名。
组	报警组名。
供应器	报警供应器的名称。
值	发生报警时标记的值。列宽应该足够大，以提供所需级别的精度。
限制	标记的报警限值。列宽应该足够大，以提供所需级别的精度。
操作员	与报警条件关联的已登录操作员的 ID。
操作员全名	已登录操作员的全名。
操作员节点	与报警条件关联的已登录操作员的节点。 在“终端服务”环境中，这是操作员建立“终端服务”会话的客户端计算机的名称。如果无法检索节点名，则使用节点的 IP 地址代替。
操作员域	已登录且与报警条件关联的操作员的域。
标记注释	标记的注释。
报警注释	与标记的报警关联的注释。此注释是定义标记的报警时在报警注释框中输入的。为报警引入某个确认注释时，新的注释更新到这个注释列中。
用户 1	报警的 AlarmUserDefNum1 属性的数值。
用户 2	报警的 AlarmUserDefNum2 属性的数值。
User 3	报警的 AlarmUserDefStr 属性的字符串值。

5. 通过选择列名并使用上、下箭头，重新排列各个列。列明细对话框顶部显示的列名是报警控件最左侧的列。
6. 要更改某个列的名称或其宽度，请选择该列并选择编辑。此时出现“编辑”对话框。

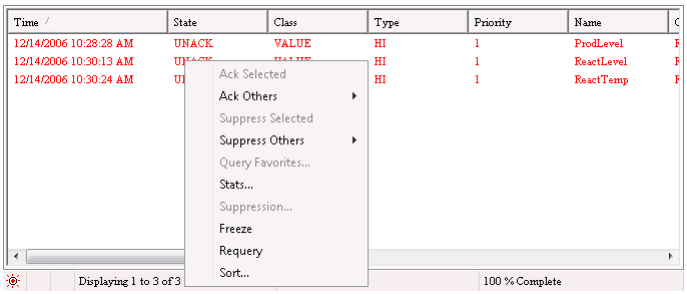


   a. 在新名框中，输入新的列名。
   b. 在新宽度框中，输入列宽。列宽可以在从 1 到 999 个像素的范围内。
   c. 选择确定。
7. 选择“列明细”对话框中的确定。
8. 选择应用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

控制功能在运行时的访问权

如果在运行时使用鼠标右键单击 Alarm Viewer 控件，则出现一个上下文相关（快捷）菜单。



您可以控制在快捷菜单上显示哪些菜单命令。

配置快捷菜单

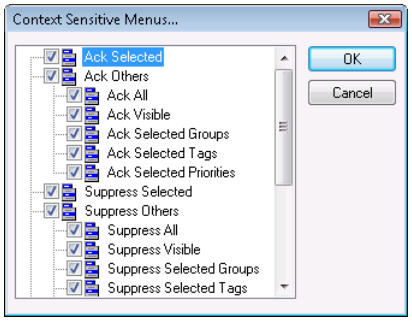
1. 使用鼠标右键单击 **Alarm Viewer** 控件，然后选择属性。

   此时出现 **AlarmViewerCtrl** 属性对话框。

2. 在常规选项卡中，选中显示上下文相关菜单复选框。

3. 选择配置上下文菜单。

   此时出现上下文相关菜单对话框。



   此对话框显示一个层次结构化的命令列表，其中的命令都可以出现在 Alarm Viewer 控件的快捷菜单上。

4. 配置快捷菜单选项。您至少必须选择一个快捷菜单命令。

此命令	允许运行时用户
确认已选项	确认所选报警。如果未选择确认已选项与确认其它菜单项，则会禁用使用缺省确认注释复选框以及文本框。
确认其它	通过其它方法确认报警。用户可以选择确认哪些报警。如果选择确认其它，则至少必须选择一个子菜单项。
确认全部	确认所有活动的报警。
确认可见项	确认可见的报警。
确认已选组	确认与所选的组具有相同的组名并且具有相同供应器名的所有报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

确认已选标记	确认与所选标记具有相同的组名并且具有相同供应器、组以及优先级的所有报警。
确认已选优先级	确认优先级与所选的一个或多个优先级相同并且具有相同供应器与组的所有报警。
抑制已选项	抑制所选报警。
抑制其它	通过快捷菜单中显示的其它方法抑制报警。
抑制全部	抑制所有报警。
抑制可见项	抑制所有可见报警。
抑制已选组	抑制与所选组具有相同组名的所有报警。
抑制已选标记	抑制与所选标记具有相同标记名的所有报警。
抑制已选优先级	抑制优先级与所选的一个或多个优先级相同的所有报警。
全部撤销抑制	撤销抑制所有已抑制的报警。
查询收藏夹	打开“报警查询”对话框。
统计	打开“报警统计”对话框。
抑制	打开“报警抑制”对话框。
冻结	在 Alarm Viewer 控件的冻结/撤销冻结模式之间切换。
重新查询	重新运行报警查询。
排序	打开“排序”对话框。

5. 选择确定。
6. 选择行选择，以允许用户在运行时从 Alarm Viewer 控件中选择某个行。
7. 选择使用扩展选择，以允许用户使用 **SHIFT** 或 **CTRL** 键从 Alarm Viewer 控件中同时选择多条报警记录。
8. 选择应用。

选择要显示的报警

Alarm Viewer 控件可以显示活动报警的摘要或历史报警的列表。

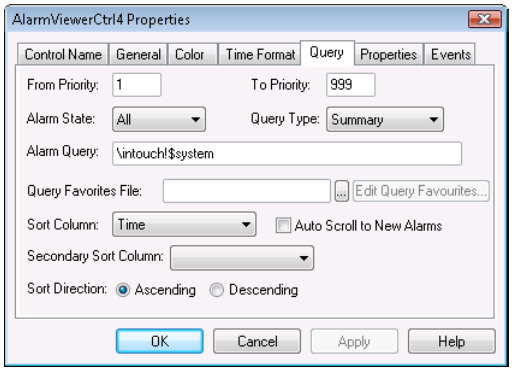
设置常规报警查询属性

1. 使用鼠标右键单击 Alarm Tree Viewer 控件，然后选择属性。此时出现 **AlarmViewerCtrl** 属性对话框。
2. 选择常规选项卡。
3. 选中在启动时执行查询复选框，以在应用程序启动时使用缺省查询属性自动更新 Alarm Viewer 控件。
4. 选中显示消息复选框，以在没有报警时显示缺省消息。在文本框中，输入要显示的消息。
5. 选择应用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置查询缺省值

1. 使用鼠标右键单击 Alarm Tree Viewer 控件，然后选择属性。此时出现 **AlarmViewerCtrl** 属性对话框。
2. 选择查询选项卡。



3. 在从优先级框中，输入最小报警优先级值（1 到 999）。
4. 在到优先级框中，输入最大报警优先级值（1 到 999）。
5. 选择查询类型箭头，然后选择历史或摘要作为缺省的运行时报警显示。

   通过运行包含查询函数的 QuickScript，可以在运行时更改显示对象的缺省类型。例如，如果脚本包含 Type 参数设置为“Summary”的 ApplyQuery() 方法，则网格显示当前报警的摘要。相反地，如果相同的网格运行了 Type 参数设置为“Historical”的 ApplyQuery() 方法，则显示历史报警。QueryType 属性反映报警显示的当前状态。

6. 在报警查询框中，输入一个有效的报警查询。例如，输入 \InTouch!$System 以查询属于缺省的 $System 报警组的所有报警。
7. 选择确定。

使用查询收藏夹创建自定义的已保存查询

您可以配置一个查询收藏夹列表，供操作员从快捷菜单中进行选择。

请确保将查询收藏夹文件放到 Windows Vista 或较新版本标准用户可以访问的文件夹中。如果始终使用相同的用户帐户来运行 WindowViewer，则可以使用：

C:\Users\<用户名>\AppData\Local\

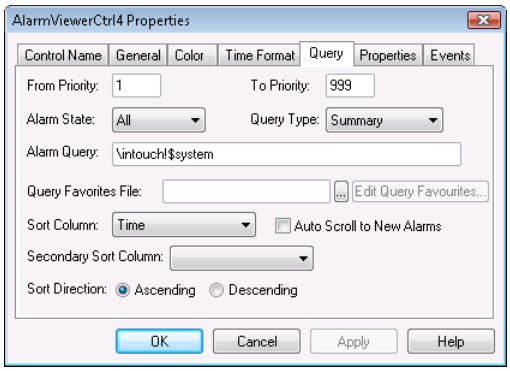
如果有其他用户要运行 WindowViewer，则可以使用：

C:\Users\Public\Documents\

配置查询收藏夹文件

1. 使用鼠标右键单击 Alarm Tree Viewer 控件，然后选择属性。此时出现 **AlarmViewerCtrl** 属性对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



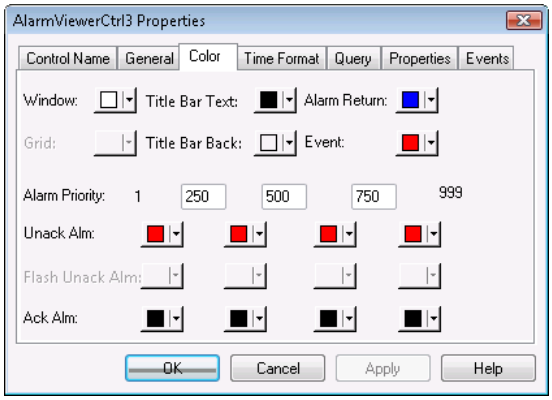
2. 选择**查询**选项卡。
3. 配置**查询**收藏夹文件。
   a. 在**查询收藏夹文件**框中，输入网络路径与文件名，或选择省略号按钮以浏览文件。
   b. 要编辑过滤器收藏夹文件，请选择**编辑收藏夹文件**按钮。此时打开**报警查询**窗口，可供您在收藏夹文件中添加、修改或删除过滤器。完成时选择**确定**，以保存更改并关闭窗口。
4. 选择**确定**。

为不同类型的报警记录使用不同的颜色

您可以为 Alarm Viewer 控件中出现的不同报警状态设置颜色选项。

配置报警显示颜色

1. 使用鼠标右键单击 Alarm Tree Viewer 控件，然后选择**属性**。此时出现 **AlarmViewerCtrl 属性**对话框。
2. 选择**颜色**选项卡。



3. 选择每个颜色框以打开调色板。在调色板中选择您要为以下每项所使用的颜色：

属性	描述
窗口	设置显示背景色。
标题栏文本	设置标题栏文本颜色（仅在选择“显示标题”选项时可用）。
报警返回	设置返回的报警（未经确认即返回到正常状态的报警）的颜色。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

网格	设置网格的颜色。缺省条件下，不会显示网格。缺省的网格颜色是淡灰色。报警对象中网格的颜色自动设置为所选窗口颜色的对比色。
标题栏背景	设置标题栏背景色（仅在选择显示标题选项时可用）。
事件	设置事件的颜色。

4. 在“报警优先级”框中，输入报警优先级数字，这些数字充当不同颜色的断点，这些颜色分别用于标识未确认的报警、确认的报警以及未确认的闪烁报警。
5. 选择**未确认报警**和**确认报警**颜色框以打开调色板。在调色板中选择您要使用的颜色。
6. 要将报警查询配置成闪烁未确认的报警，请选择**常规**选项卡，选中**闪烁未确认报警**复选框，然后单击**颜色**选项卡并选择**闪烁未确认报警**颜色框。选择要用于每个报警优先级范围的颜色。

   注意：Alarm Viewer 控件无法显示少于一秒的时间内的变化。如果报警状态在一秒内变化了两次，则 Alarm Viewer 控件无法识别这种变化。

7. 选择**应用**。

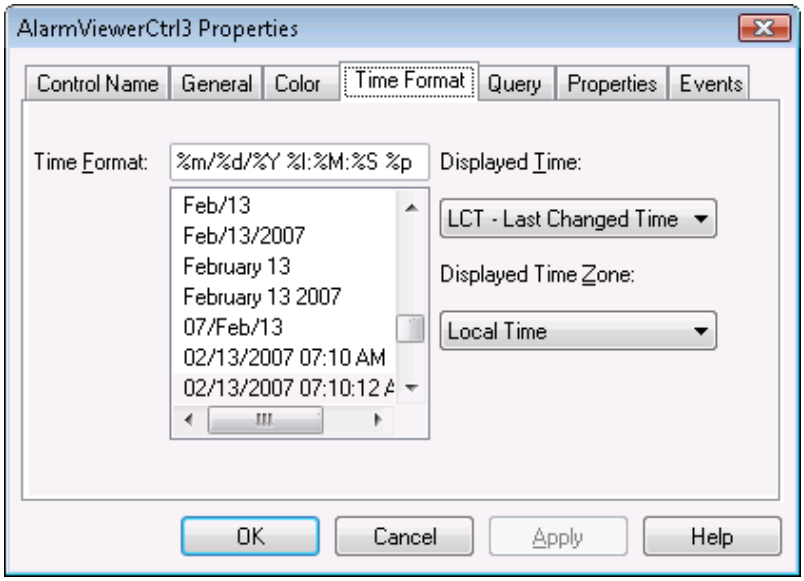
配置显示的报警记录时间格式

您可以为显示的报警记录配置时间格式。

对于 Alarm Viewer 控件，原始报警时间是报警发生时的日期/时间标签。如果标记是 I/O 标记，且 I/O 服务器能够传递时间标签，那么该标记就是来自该服务器的时间标签。

配置时间格式

1. 使用鼠标右键单击 **Alarm Viewer** 控件，然后选择属性。此时出现 **AlarmViewerCtrl** 属性对话框。
2. 选择**时间格式**选项卡。



3. 在**时间格式**列表中，单击所需的时间格式。**时间格式**框显示所选格式对应的字符组成的一组字符串，由 % 符号分隔。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串字符	描述
d	两位数字日期值。
b	三个英文字母的月份缩写。
是	四位数字年份值。
m	两位数字月份值。
y	两位数字年份值。
#x	完整的日期与星期。例如：2007 年 8 月 10 日星期五
B	完整的月份名。
H	以 24 小时时间格式表示的小时值。
M	分钟值。
p	AM 或 PM（用于 12 小时时间格式）。
S	秒值。
s	秒值的小数部分。
I	以 12 小时时间格式表示的小时值。

4. 在**显示时间**列表中，**选择显示的时间**：

OAT	原始报警时间，指报警发生时的日期/时间标签。
LCT	上次更改时间，指最近一次报警实例状态变化的日期/时间标签：报警发生、子状态改变、返回到正常，或者是确认。
LCT 但确认后为 OAT	上次更改时间，但是确认时为原始报警时间。报警未确认时使用上次更改时间，确认报警后使用原始报警时间。

5. 在**显示时区**列表中，**选择时区**：

GMT	“格林尼治标准时间”，也称作“通用协调时间”、UTC 或 Zulu。
本地时间	已调整为本地时区的报警时间。
原始时间	已根据报警源的时区进行调整的报警时间。

6. **选择应用**。

配置报警记录的排序顺序

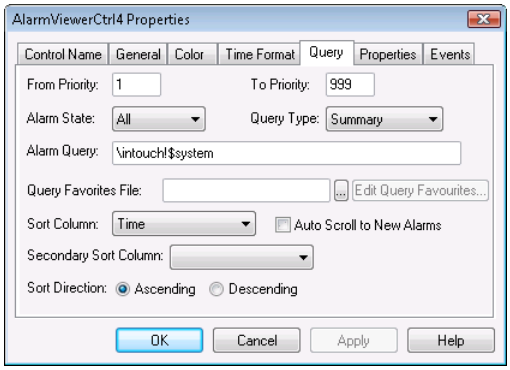
您可以对列表中的报警记录进行排序。缺省条件下，Alarm Viewer 控件按时间的升序列出报警记录。

您可以按主排序列和可选的第二位排序列的升序或降序对报警记录进行排序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置报警记录的排序顺序

1. 使用鼠标右键单击 Alarm Viewer 控件，然后选择属性。此时出现 **AlarmViewerCtrl** 属性对话框。
2. 选择**查询**选项卡。



3. 通过完成以下操作选择排序选项：
   a. 从**排序列**列表中选择主排序列。只有可见的列才出现在**排序列**列表中。如果未看到所需的列，请转到“常规”选项卡，然后从“列明细”中选择该列。
   b. 从**第二位排序列**列表中选择第二位排序列。
   c. 如果已选择“时间”作为主排序列，则**自动滚动到新报警**复选框变为可用状态。如果希望自动滚动并且在发生新报警时显示它们，请选择此选项。
   d. 选择**升序**或**降序**作为排序方向。
4. 选择**应用**。

在运行时使用 Alarm Viewer 控件

Alarm Viewer 控件包含一个快捷菜单，供操作员快速访问可以应用于以下对象的命令：一个或多个所选报警、报警组、标记以及优先级的显示对象。

下表显示 Alarm Viewer 控件快捷菜单上提供的命令：

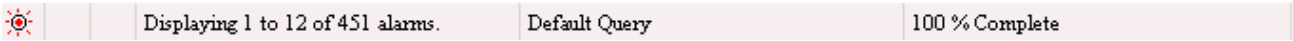
- **确认已选项** - 确认所选报警。
- **确认其它** - 出现一个列出其它确认命令的子菜单.
  - 确认全部 - 确认当前报警查询中的所有报警。由于报警网格的显示区域有限，因此“确认全部”命令可能会确认网格中不可见的报警。
  - 确认可见项 - 仅确认当前报警网格中可见的那些报警。
  - 确认已选组 - 确认所有的特定报警，它们与一个或多个所选的报警来自相同的供应器，并且具有相同的组名。
  - 确认已选标记 - 确认所有特定的报警，它们具有来自相同供应器与组名的相同标记，并且与一个或多个所选报警具有相同的优先级。
  - 确认已选优先级 - 确认所有特定的报警，它们与一个或多个所选的报警来自相同的供应器与组名，并且具有相同的优先级。
- **抑制已选项** - 抑制所选的一个或多个报警。
- **抑制其它项** - 打开一个包含抑制命令的子菜单。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

  - 抑制全部 - 抑制显示当前的所有报警以及将来要发生的这些报警。
  - 抑制可见项 - 抑制显示当前的任何可见报警以及将来要发生的这些报警。
  - 抑制已选组 - 抑制显示当前以及将来要发生的任何特定报警：它们与一个或多个所选的报警属于相同的组，并且具有相同的“供应器名”。
  - 抑制已选标记 - 抑制显示当前以及将来要发生的任何特定报警：它们与一个或多个所选报警属于相同的标记名，并且具有相同的“供应器名”、“组名”及“优先级范围”。
  - 抑制已选优先级 - 抑制显示当前以及将来要发生的任何特定报警：它们与一个或多个所选报警属于相同的优先级，并且具有相同的“供应器名”与“组名”。
  - 全部撤销抑制 - 清除抑制设置。
- **查询收藏夹** - 显示“报警查询”对话框以选择先前保存的报警查询。您也可以添加、修改及删除报警查询。
- **统计** - 显示**报警统计**对话框。
- **抑制** - 显示**报警抑制**对话框。
- **冻结** - 冻结当前显示。
- **重新查询** - 再次查询报警供应器。
- **排序** - 显示**辅助排序**对话框。

查看状态栏信息

如果从“常规”属性页中选择“显示状态栏”选项，则运行时期间会在 Alarm Viewer 控件底部显示状态栏。



状态栏包含三个指示器：状态消息、当前报警查询以及进度条。这些指示器提供显示查询当前状态的概况，并提供 Alarm Viewer 控件中可用抑制的有关详细信息。冻结控件时状态栏的右侧窗格为红色，抑制一个或多个报警时状态栏的左侧窗格为红色。抑制生效时左侧窗格中显示“抑制”字样。

在运行时使用查询收藏夹

通过使用 Alarm Viewer 控件快捷菜单上的“查询收藏夹”命令，可以从先前定义的报警查询列表中快速选择报警查询。您也可以创建新的命名查询、编辑或删除现有的查询。

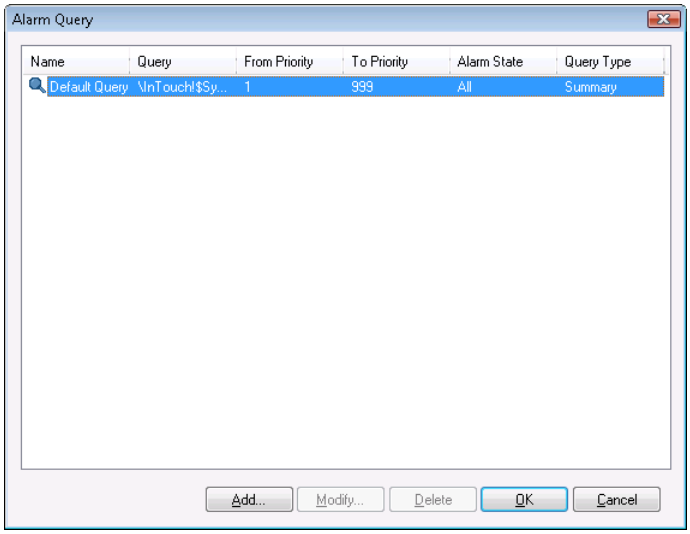
对报警查询所作的更改不自动应用于使用相同查询的其它 Alarm Viewer 控件。删除某个报警查询时，不会从使用相同查询的其它 Alarm Viewer 控件中自动删除该查询。

注意：对于 Alarm Viewer 控件中出现的多行报警查询，换行会出现“乱码”字符。但这不影响功能。

在运行时选择报警查询

1. 使用鼠标右键单击 Alarm Viewer 控件，然后选择**查询收藏夹**。此时出现**报警查询**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



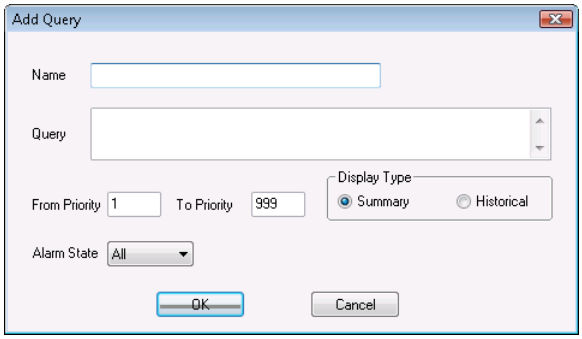
2. 从当前定义的查询列表中，选择要显示的命名查询。
3. 选择确定。现在 Alarm Viewer 控件显示查询检索的报警信息。

在运行时添加新的命名查询

1. 使用鼠标右键单击 **Alarm Viewer** 控件，然后选择**查询收藏夹**。

   此时出现**报警查询**对话框。
2. 选择添加。

   此时出现添加**查询**对话框。



3. 配置查询。执行以下操作：
   a. 在名称框中，输入要用于标识查询的名称。
   b. 在**查询**框中，输入要执行的一组 InTouch 报警查询。您可以指定一个或多个“报警供应器”和组。
   c. 在从**优先级**框中，输入最小报警优先级值（1 到 999）。
   d. 在到**优先级**框中，输入最大报警优先级值（1 到 999）。
   e. 选择**报警状态**箭头，然后选择要在报警查询中使用的报警状态（全部、确认、未确认）。
   f. 在**显示类型**区域中，选择摘要或历史作为要查询的记录类型。
4. 选择确定，关闭添加**查询**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5. 选择**报警查询**对话框中的**确定**，以将**查询**添加到收藏夹。

在运行时修改现有的命名查询

1. 使用鼠标右键单击 **Alarm Viewer** 控件，然后选择**查询收藏夹**。

   此时出现**报警查询**对话框。
2. 在当前定义的**查询**列表中，**选择**要修改的命名**查询**。
3. 选择 **Modify**。

   此时出现**修改查询**对话框。
4. 进行所需的修改，然后选择**确定**。
5. 选择**报警查询**对话框中的**确定**。

在运行时删除现有的命名查询

1. 使用鼠标右键单击 **Alarm Viewer** 控件，然后选择**查询收藏夹**。

   此时出现**报警查询**对话框。
2. 从当前定义的**查询**列表中，**选择**要删除的命名**查询**。
3. 选择**删除**。出现消息时，**选择是**。
4. 选择“报警查询”对话框中的**确定**。

使用 Alarm Viewer 控件 ActiveX 属性

您可以使用脚本直接设置 Alarm Viewer 控件属性的值，或是将它指定给 InTouch 标记或 I/O 引用。如需有关设置属性的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序开发指南*》中的编写 ActiveX 控件脚本。

下表列出了 Alarm Viewer 控件属性。

属性	类型	用途
AckAllMenu	离散	启用/禁用确认全部菜单项。
AckAlmColorRange1	整型	设置用于显示优先级范围从 1 到 ColorPriorityRange1 的已确认的报警的颜色。缺省优先级范围是 1 到 250。
AckAlmColorRange2	整型	设置用于显示优先级范围从 ColorPriorityRange1 到 ColorPriorityRange2 的已确认的报警的颜色。缺省优先级范围是 250 到 500。
AckAlmColorRange3	整型	设置用于显示优先级范围从 ColorPriorityRange2 到 ColorPriorityRange3 的已确认的报警的颜色。缺省优先级为 500 到 750。
AckAlmColorRange4	整型	设置用于显示优先级范围从 ColorPriorityRange3 到 999 的已确认的报警的颜色。缺省优先级为 750 到 999。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	类型	用途
AckOthersMenu	离散	启用/禁用确认其它菜单项。
AckSelectedGroupsMenu	离散	启用/禁用确认已选组菜单项。
AckSelectedMenu	离散	启用/禁用确认已选项菜单项。
AckSelectedPrioritiesMenu	离散	启用/禁用确认已选优先级菜单项。
AckSelectedTagsMenu	离散	启用/禁用确认已选标记菜单项。
AckVisibleMenu	离散	启用/禁用确认可见项菜单项。
AlarmQuery	消息	设置初始报警查询。此域仅接受文本；它不接受标记。 下例使用报警组的完整路径： \\节点\InTouch!Group 本例使用本地报警组的完整路径： \InTouch!Group 本例使用另一个“组列表”： GroupList
AlarmState	消息	要查询的缺省报警状态（“全部”、“未确认”、“确认”）。
AlmRtnColor	整型	设置已返回到正常状态且未确认的报警的颜色。此颜色还用于从已确认状态返回到正常状态但尚未观察到确认状态转换的报警。
AutoScroll	离散	如果用户从开头滚动列表，这会自动跳到新报警。新报警定义为当前未在显示对象中显示的那些报警。
ColorPriorityRange1	整型	设置要显示的报警的优先级范围边界。此属性的值必须大于一且小于 ColorPriorityRange2 的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	类型	用途
ColorPriorityRange2	整型	设置要显示的报警的优先级范围边界。此属性的值必须大于 ColorPriorityRange1 的值且小于 ColorPriorityRange3 的值。
ColorPriorityRange3	整型	设置要显示的报警的优先级范围边界。此属性的值必须大于 ColorPriorityRange2 的值且小于 999。
ColumnResize	离散	返回或设置一个值，确定是否可以在运行时调整列的大小。
CustomMessage	消息	没有报警时显示的缺省消息。
DefaultAckComment	消息	报警已确认且 "UseDefaultAckComment" 为“真”时用作注释。否则将提示用户输入注释。
DisplayedTime	消息	显示报警消息时间。值只能是“OAT”、“LCT”或“LCT 但确认为 OAT”。
DisplayedTimeZone	消息	获取或设置当前时区字符串。 值只能是“GMT”、“原始时间”或“本地时间”。
EventColor	整型	设置事件的颜色。
ExtendedSelection	离散	允许您通过在按住 Ctrl 或 Shift 键的同时结合使用鼠标按钮来选择多个报警。 缺省条件下，选择报警即可切换报警的选择（仅当选中“行选择”复选框时才可用）。
FlashUnAckAlarms	离散	启用或禁用未确认报警的闪烁。它接受离散输入值 1 或 0。如果此属性设置为 1，未确认的报警每秒闪烁一次。如果此属性设置为 0，则未确认的报警不闪烁。 此属性对应于 Alarm Viewer 控件常规选项卡上的闪烁未确认报警复选框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	类型	用途
FlashUnackAlmColorRange1	整型	为属于“报警优先级范围 1”的未确认报警设置闪烁颜色。
FlashUnackAlmColorRange2	整型	为属于“报警优先级范围 2”的未确认报警设置闪烁颜色。
FlashUnackAlmColorRange3	整型	为属于“报警优先级范围 3”的未确认报警设置闪烁颜色。
FlashUnackAlmColorRange4	整型	为属于“报警优先级范围 4”的未确认报警设置闪烁颜色。
字体	无	为控件中的记录与标题设置字体。
FreezeMenu	离散	启用/禁用冻结菜单项。
FromPriority	整型	设置缺省查询的最低优先级值。
GridColor	整型	设置背景网格的颜色。
NewAlarmEventMode	整型	控制 NewAlarm 事件的触发。 0 = NewAlarm 事件无法触发。（缺省值）。 1 = NewAlarm 事件是活动的。 2 = NewAlarm 事件是活动的，并且在会至少一个新的未确认报警到达时继续触发。
QueryFavoritesFile	消息	返回或设置查询收藏夹文件名。
QueryFavoritesMenu	离散	启用/禁用“查询收藏夹”菜单项。
QueryName	字符串型	返回当前查询名。
QueryStartup	离散	如果设置此属性，则使用缺省查询属性自动开始更新网格。 否则，您必须在脚本中运行 ApplyDefaultQuery 或 ApplyQuery 方法才能更新网格。
QueryType	消息	将显示类型设置为“摘要”或“历史”。
RequeryMenu	离散	启用/禁用“重新查询”快捷菜单项。
RetainSuppression	离散	更改报警查询时在报警查询之间保持报警抑制。
RowSelection	离散	允许用户在运行时选择报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	类型	用途
SecondarySortColumn	消息	返回或设置当前第二位排序列。
SelectedCount	整型	返回所选报警的总数。
ShowContextMenu	离散	启用快捷菜单功能。
ShowGrid	离散	返回或设置一个值，确定是否在控件中显示网格线。
ShowHeading	离散	显示控件的标题栏。
ShowMessage	离散	返回或设置一个值，确定是否为控件显示错误消息。
ShowStatusBar	离散	获取或设置一个值，确定是否显示状态栏。
SilentMode	离散	获取或设置一个值，确定控件是否处于“无提示”模式。
SortColumn	消息	获取或设置当前排序列。
SortMenu	离散	启用/禁用排序菜单项。
SortOrder	离散	获取或设置排序方向。可能值有“升序”与“降序”，分别用 0 与 1 表示。
StatsMenu	离散	启用/禁用统计菜单项。
SuppressAllMenu	离散	启用/禁用抑制全部菜单项。
SuppressedAlarms	整型	获取抑制的报警的总数。
SuppressionMenu	离散	启用/禁用抑制菜单项。
SuppressOthersMenu	离散	启用/禁用抑制其它菜单项。
SuppressSelectedGroupsMenu	离散	启用/禁用抑制已选组菜单项。
SuppressSelectedMenu	离散	启用/禁用抑制已选项菜单项。
SuppressSelectedPrioritiesMenu	离散	启用/禁用抑制已选优先级菜单项。
SuppressSelectedTagsMenu	离散	启用/禁用抑制已选标记菜单项。
SuppressVisibleMenu	离散	启用/禁用抑制可见项菜单项。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	类型	用途
TimeFormat	消息	设置报警时间标签的格式。
标题背景色	整型	设置标题栏背景色。
标题前景色	整型	设置标题栏前景色。
ToPriority	整型	设置报警查询的最高优先级。
TotalAlarms	整型	获取报警数。
UnackAlarms	整型	获取未确认的报警总数。
UnAckAlmColorRange1	整型	设置用于显示优先级范围从 1 到 ColorPriorityRange1 的未确认报警的颜色。
UnAckAlmColorRange2	整型	设置用于显示优先级范围从 ColorPriorityRange1 到 ColorPriorityRange2 的未确认报警的颜色。
UnAckAlmColorRange3	整型	设置用于显示优先级范围从 ColorPriorityRange2 到 ColorPriorityRange3 的未确认报警的颜色。
UnAckAlmColorRange4	整型	设置用于显示优先级范围从 ColorPriorityRange3 到 999 的未确认报警的颜色。
UnsuppressAllMenu	离散	启用/禁用“全部撤销抑制”菜单项。
UseDefaultAckComment	离散	如果设置为“真”，则确认报警时使用缺省确认注释。否则提示操作员输入注释。
窗口颜色	整型	设置网格背景色。

配置 ActiveX 控件的颜色

您可以将颜色指定为整数值。报警 ActiveX 控件使用 ABGR 模型将颜色指定为 32 位整数，其中：

A = 透明

B = 蓝色

G = 绿色

R = 红色

报警 ActiveX 控件不支持透明度。所有高 8 位的值都会被忽略。

例如，要将颜色设置为“蓝色”，使用以下 ABGR 值：

A = 0

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

B = 255

G = 0

R = 0

此颜色的十六进制值为 0x00FF0000。十进制值为 16711680。

下表显示可以使用的颜色示例：

颜色	十六进制值	十进制值
白色	0x00FFFFFF	16777215
黑色	0x00000000	0
蓝色	0x00FF0000	16711680
红色	0x000000E1	225
绿色	0x0000FF00	65280

使用 Alarm Viewer 控件 ActiveX 方法

您可以在脚本中使用 Alarm Viewer 控件 ActiveX 方法来：

- 确认报警。
- 抑制报警。
- 获取报警的相关信息。
- 运行报警查询。
- 移动与冻结显示。
- 给报警记录排序。
- 选择特定的报警。
- 显示快捷菜单、关于对话框以及报警统计对话框。

如需有关调用方法的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序开发指南*》中的编写 ActiveX 控件脚本。

在运行时确认报警

使用以下方法在运行时确认报警。

- AckSelected() 方法
- AckAll() 方法
- AckVisible() 方法
- AckSelectedGroup() 方法
- AckSelectedTag() 方法
- AckSelectedPriority() 方法
- AckGroup() 方法

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- AckPriority() 方法
- AckTag() 方法

AckSelected() 方法

在运行时确认 Alarm Viewer 控件中所选的报警。

语法

```
Object.AckSelected (Comment)
```

参数

Comment
报警确认注释。

示例

Tag1 定义为消息标记，控件名为 AlarmViewerCtrl1。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckSelected (Tag1);
```

AckAll() 方法

确认当前报警查询中的所有报警。由于 Alarm Viewer 控件中的显示区域有限, AckAll() 方法可能也会确认未在显示对象中显示的报警。

语法

```
Object.AckAll (Comment)
```

参数

Comment
报警确认注释。

示例

Tag1 定义为消息标记，控件名为 AlarmViewerCtrl1。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckAll (Tag1);
```

AckVisible() 方法

仅确认 Alarm Viewer 控件中当前可见的那些报警。

语法

```
Object.AckVisible (Comment)
```

参数

Comment
报警确认注释。

示例

Tag1 定义为消息标记，控件名为 AlarmViewerCtrl1。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckVisible (Tag1);
```

AckSelectedGroup() 方法

确认与一个或多个所选报警具有相同组名的所有报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
Object.AckSelectedGroup (Comment)
```

参数

Comment
报警确认注释。

示例

Tag1 定义为消息标记，控件名为 AlarmViewerCtrl1。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckSelectedGroup (Tag1);
```

AckSelectedTag() 方法

确认与一个或多个所选报警具有相同标记、组名以及优先级的所有报警。

语法

```
Object.AckSelectedTag (Comment)
```

参数

Comment
报警确认注释。

示例

Tag1 定义为消息标记，控件名为 AlarmViewerCtrl1。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckSelectedTag (Tag1);
```

AckSelectedPriority() 方法

确认与一个或多个所选报警具有相同优先级范围的所有报警。

语法

```
Object.AckSelectedPriority (Comment)
```

参数

Comment
报警确认注释。

示例

Tag1 定义为消息标记，控件名为 AlarmViewerCtrl1。

```
Tag1 = "Alarm Comment";
#AlarmViewerCtrl1.AckSelectedPriority (Tag1);
```

AckGroup() 方法

确认给定组名与供应器的所有报警。

语法

```
Object.AckGroup(ApplicationName, GroupName, Comment)
```

参数

ApplicationName
应用程序的名称，例如 \\node1\Intouch

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

GroupName
组名。例如，Turbine。

Comment
报警确认注释。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.AckGroup ("\Intouch", "Turbine", "Turbine acknowledgement Comment");
```

AckPriority() 方法

确认指定优先级范围内具有相同供应器名与组名的所有报警。

语法

```
Object.AckPriority(ApplicationName, GroupName, FromPriority, ToPriority, Comment)
```

参数

ApplicationName
应用程序的名称。例如，\\node1\Intouch

GroupName
组名。例如，Turbine。

FromPriority
报警的开始优先级。例如，100。

ToPriority
报警的结束优先级。例如，900。

Comment
报警确认注释。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.AckPriority ("\Intouch", "Turbine", 100, 900, "Turbine acknowledgement
Comment");
```

AckTag() 方法

确认给定标记名的报警，这些报警在给定的优先级范围内并且具有相同的供应器名与组名。

语法

```
Object.AckTag(ApplicationName, GroupName, tag, FromPriority, ToPriority, Comment)
```

参数

ApplicationName
应用程序的名称，例如 \\node1\Intouch

GroupName
组名。例如，Turbine。

tag
报警标记的名称。例如，Valve1。

FromPriority
报警的开始优先级。例如，100。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ToPriority
报警的结束优先级。例如，900。

Comment
报警确认注释。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.AckTag ("\Intouch", "Turbine", "Valve1", 100, 900, "Turbine
acknowledgement Comment");
```

在运行时抑制报警

使用以下方法在运行时抑制报警：

- ShowSuppression() 方法
- SuppressSelected() 方法
- SuppressAll() 方法
- SuppressVisible() 方法
- SuppressSelectedGroup() 方法
- SuppressSelectedTag() 方法
- SuppressSelectedPriority() 方法
- UnSuppressAll() 方法
- SuppressGroup() 方法
- SuppressPriority() 方法
- SuppressTag() 方法

ShowSuppression() 方法

显示抑制对话框，其中包含所有抑制的报警。

语法

```
Object.ShowSuppression()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.ShowSuppression();
```

SuppressSelected() 方法

抑制显示当前所选的报警以及将来要发生的这些报警。

语法

```
Object.SuppressSelected()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressSelected();
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

SuppressAll() 方法

抑制显示当前所有活动的报警以及将来要发生的这些报警。

语法

```
Object.SuppressAll()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressAll();
```

SuppressVisible() 方法

抑制显示当前和将来发生的任何可见报警。

语法

```
Object.SuppressVisible()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressVisible();
```

SuppressSelectedGroup() 方法

抑制显示当前以及将来要发生的任何特定报警：它们与一个或多个所选的报警属于相同的“组”与“供应器”。

语法

```
Object.SuppressSelectedGroup()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressSelectedGroup();
```

SuppressSelectedTag() 方法

抑制显示当前以及将来要发生的任何特定报警：它们与一个或多个所选报警属于相同的标记名，并且具有相同的“组名”、“供应器名”、及“优先级范围”。

语法

```
Object.SuppressSelectedTag()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressSelectedTag();
```

SuppressSelectedPriority() 方法

抑制显示当前以及将来要发生的任何特定报警：它们与一个或多个所选报警具有相同的优先级范围。

语法

```
Object.SuppressSelectedPriority()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressSelectedPriority();
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

UnSuppressAll() 方法

清除报警抑制。

语法

```
Object.UnSuppressAll()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.UnSuppressAll();
```

SuppressGroup() 方法

抑制显示当前以及将来要发生的任何特定报警：它们属于给定的“组名”。

语法

```
Object.SuppressGroup(ApplicationName, GroupName)
```

参数

ApplicationName
应用程序的名称，例如 \\node1\Intouch

GroupName
组名。例如，Turbine。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressGroup ("\Intouch", "Turbine");
```

SuppressPriority() 方法

抑制显示当前以及将来要发生的任何报警：它们具有指定的优先级范围，并且具有相同的“供应器名”与“组名”。

语法

```
Object.SuppressPriority(ApplicationName, GroupName, FromPriority, ToPriority)
```

参数

ApplicationName
应用程序的名称，例如 \\node1\Intouch

GroupName
组名。例如，Turbine。

FromPriority
报警的开始优先级。例如，100。

ToPriority
报警的结束优先级。例如，900。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressPriority ("\Intouch", "Turbine", 100, 900);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

SuppressTag() 方法

抑制显示当前以及将来要发生的任何特定报警：它们由给定的标记名或组名发出，并且在指定的优先级范围内。

语法

```
Object.SuppressTag(ApplicationName, GroupName, tag, FromPriority, ToPriority)
```

参数

ApplicationName
应用程序的名称，例如 \\node1\Intouch。

GroupName
组名。例如，Turbine。

tag
报警标记的名称。例如，valve 1。

FromPriority
报警的开始优先级。例如，100。

ToPriority
报警的结束优先级。例如，900。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SuppressTag ("\Intouch", "Turbine", "Valve1", 100, 900);
```

检索有关特殊报警的信息

使用 GetItem() 函数检索有关报警的信息。

GetItem() 方法

将指定的行与列的数据作为字符串来返回。

语法

```
Object.GetItem(Integer, Message)
```

参数

整型
整型表达式，为控件中特定的行赋值。

消息
字符串表达式，为控件中的列名赋值。

示例

控件名为 AlmDbView1，tag 定义为消息标记。

```
tag = #AlmDbView1.GetItem(1, "Group");
```

运行报警查询

使用以下方法运行查询。

- ShowQueryFavorites() 方法
- Requery() 方法

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- ApplyQuery() 方法
- ApplyDefaultQuery() 方法
- SetQueryByName() 方法, SetQueryByName() 方法

ShowQueryFavorites() 方法

如果 QueryFavoritesFile 属性包含有效查询收藏夹文件（采用 .xml 格式）的名称，则显示报警查询对话框。

语法

```
Object.ShowQueryFavorites()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.ShowQueryFavorites();
```

Requery() 方法

再次查询报警供应器。

语法

```
Object.Requery()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.Requery();
```

ApplyQuery() 方法

按照指定的“报警查询”、“从优先级”、“到优先级”、要查询报警的“状态”以及要检索报警的“类型”等参数来执行查询。

语法

```
Object.ApplyQuery(AlarmQuery, FromPriority, ToPriority, State, Type)
```

参数

AlarmQuery
报警查询。例如：\InTouch!$System

FromPriority
报警的开始优先级。例如，100。

ToPriority
报警的结束优先级。例如，900。

State
指定要显示报警的类型。例如，“未确认”或消息标记。有效状态包括“全部”、“未确认”或“确认”。

Type
指定查询的类型，例如 Historical（历史报警）或 Summary（摘要报警）。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.ApplyQuery ("\InTouch!$System",100,900,"All", "Historical");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ApplyDefaultQuery() 方法

按照设计时指定的 FromPriority、ToPriority、AlarmState、QueryType 以及 AlarmQuery 等属性来执行查询。缺省属性仅能在开发时更改，不能由其它报警查询来改写。

语法

```
Object.ApplyDefaultQuery()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.ApplyDefaultQuery();
```

SetQueryByName() 方法

将当前查询设置为传递的查询名所指定的查询。查询必须在查询收藏夹文件中。

语法

```
Object.SetQueryByName(QueryName)
```

参数

QueryName
使用查询收藏夹创建的查询的名称。例如，Turbine Queries。

示例

控件名为 AlarmTreeViewerCtrl1。

```
#AlarmTreeViewerCtrl1.SetQueryByName("Turbine Queries");
```

移动与冻结显示

使用以下函数移动或冻结显示：

- MoveWindow() 方法
- FreezeDisplay() 方法

MoveWindow() 方法

在控件中按指定的方法滚动报警。

语法

```
Object.MoveWindow(Option, Repeat)
```

参数

Option
要执行的操作的类型。

Type	描述
LineDn	向下滚行。Repeat 参数控制要滚动的行数。
LineUp	向上滚行。Repeat 参数控制要滚动的行数。
PageDn	向下翻页。Repeat 参数控制要滚动的页数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Type	描述
PageUp	向上翻页。Repeat 参数控制要滚动的页数。
顶端	滚动到控件顶部。
底端	滚动到控件底部。
PageRt	向右翻页。Repeat 参数控制要滚动的页数。
PageLf	向左翻页。Repeat 参数控制要滚动的页数。
向右	向右滚动。Repeat 参数控制要滚动的列数。
向左	向左滚动。Repeat 参数控制要滚动的列数。
主页	滚动到控件顶行最左一列。

Repeat
应重复此操作的次数。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.MoveWindow ("Bottom", 0);
#AlarmViewerCtrl1.MoveWindow ("LineUp", 3);
#AlarmViewerCtrl1.MoveWindow ("PageLf", 7);
```

FreezeDisplay() 方法

冻结显示。

语法

```
Object.FreezeDisplay(Freeze)
```

参数

Freeze
True = 冻结显示。
False = 撤销冻结显示。

示例

Tag1 定义为“内存离散”标记，控件名为 AlarmViewerCtrl1。

```
Tag1 = 1;
#AlarmViewerCtrl1.FreezeDisplay(Tag1);
```

给报警记录排序

使用以下函数给报警记录排序：

- ShowSort() 方法, ShowSort() 方法
- SetSort() 方法

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ShowSort() 方法

如果启用 SortMenu 属性，则显示第二位排序对话框。

语法

```
Object.ShowSort()
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.ShowSort();
```

SetSort() 方法

按照指定的 SortColumn 与 SortOrder 属性来设置排序准则。

语法

```
Object.SetSort()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SetSort();
```

显示其它信息

使用以下函数显示关于对话框与报警统计对话框：

- AboutBox() 方法, AboutBox() 方法, AboutBox() 方法
- ShowStatistics() 方法

AboutBox() 方法

显示关于对话框。

语法

```
Object.AboutBox()
```

示例

控件名为 AlarmPareto1。

```
#AlarmPareto1.AboutBox();
```

ShowStatistics() 方法

显示报警统计对话框。

语法

```
Object.ShowStatistics()
```

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.ShowStatistics();
```

选择特定的报警

使用以下函数选择特定的报警：

- SelectGroup() 方法
- SelectPriority() 方法

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- SelectTag() 方法
- SelectAll() 方法
- SelectItem() 方法
- UnSelectAll() 方法

SelectGroup() 方法

选择包含相同报警组名与供应器名的所有报警。

语法

```
Object.SelectGroup(ApplicationName, GroupName)
```

参数

ApplicationName
应用程序的名称，例如 \\node1\Intouch

GroupName
组名。例如，Turbine。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SelectGroup ("\Intouch", "Turbine");
```

SelectPriority() 方法

选择指定的优先级范围内具有相同供应器名与组名的所有报警。

语法

```
Object.SelectPriority(ApplicationName, GroupName, FromPriority, ToPriority)
```

参数

ApplicationName
应用程序的名称，例如 \\node1\Intouch

GroupName
组名。例如，Turbine。

FromPriority
报警的开始优先级。例如，100。

ToPriority
报警的结束优先级。例如，900。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SelectPriority ("\Intouch", "Turbine", 100, 900);
```

SelectTag() 方法

选择来自特定“供应器/组/标记”的所有报警。您也可以指定“优先级范围”，或使用 1-999。

语法

```
Object.SelectTag(ApplicationName, GroupName, tag, FromPriority, ToPriority)
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ApplicationName
应用程序的名称，例如 \\node1\Intouch

GroupName
组名。例如，Turbine。

tag
报警标记的名称。例如，valve 1。

FromPriority
报警的开始优先级。例如，100。

ToPriority
报警的结束优先级。例如，900。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.SelectTag ("\Intouch", "Turbine", "Valve1",100, 900);
```

SelectAll() 方法

切换对显示对象中所有报警的选择。由于报警显示对象的显示区域有限，因此 SelectAll() 函数可能会选择到显示对象中不可见的报警。

语法

```
Object.SelectAll()
```

示例

```
#AlarmViewerCtrl1.SelectAll();
```

SelectItem() 方法

切换对给定行上某个报警记录的选择。

语法

```
Object.SelectItem(RowNumber)
```

参数

RowNumber
整数值，指定要选择的报警记录的行号。控件中的第一行是 0。

示例

控件名为 AlarmViewerCtrl1，Tag1 定义为内存整型。这会切换对 Alarm Viewer 控件中第十条报警记录的选择。

```
Tag1 = 9;
#AlarmViewerCtrl1.SelectItem (Tag1);
```

UnSelectAll() 方法

取消选择所选的全部记录。

语法

```
Object.UnSelectAll()
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

控件名为 AlarmViewerCtrl1。

```
#AlarmViewerCtrl1.UnSelectAll();
```

在运行时显示上下文菜单

使用 ShowContext() 方法在运行时显示快捷菜单。

ShowContext() 方法

如果启用 RefreshMenu、ResetMenu 或 SortMenu 属性中的任何一个，则显示快捷菜单。

语法

```
Object.ShowContext()
```

示例

控件名为 AlmDbView1。

```
#AlmDbView1.ShowContext();
```

使用方法与属性时处理错误

您可以使用 SilentMode 属性隐藏运行时的错误。如果 SilentMode 属性设置为 1，则 Alarm Viewer 控件在运行时不显示错误消息。如果设置为 0，则 Alarm Viewer 控件显示错误消息。错误消息总是发送到 Operations Control Log viewer 中。

使用 ActiveX 事件触发脚本

您可以将 QuickScript 指定给 Alarm Viewer 控件事件（如鼠标单击或双击）。该事件发生时，此 QuickScript 便会运行。

Alarm Viewer 控件支持以下事件：

- 选择
- Double-click
- NewAlarm
- ShutDown
- StartUp

Select 事件有一个参数 ClicknRow，它可以确定运行时所选择的行。

Double-click 事件有一个参数 DoubleclicknRow，它可以确定运行时所双击的行。

Select 和 Double-click 事件都是零基的。向用户发布 Select 与/或 DoubleClick 事件时，显示中的行计数从 0 开始。

注意：Alarm Viewer 控件从 StartUp 事件调用方法时会忽略用户界面方法，原因在于控件尚不可见。这些包括：ShowSort()、ShowContext()、GetSelectedItem()、GetNext()、GetPrevious() 以及 AboutBox()。

如需有关编写 ActiveX 事件脚本的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序开发指南*》中的编写 ActiveX 控件脚本。

检测到新报警时运行脚本

您可以将 Alarm Viewer 控件配置成在检测到新的未确认报警（也就是，由正常状态转到未确认状态，并且满足控件的查询与优先级过滤准则的任何报警）时运行 ActiveX 事件脚本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

NewAlarmEventMode 属性控制 NewAlarm 事件的触发。

- 如果将 NewAlarmEventMode 属性设置为 0，则不触发 NewAlarm 事件。这是缺省值。
- 如果将 NewAlarmEventMode 属性设置为 1，则在以下情况中会发出新报警：
  - 触发了事件。
  - 运行与 NewAlarm 事件关联的 ActiveX 事件脚本。
  - NewAlarmEventMode 属性设置为 0。

  您必须将 NewAlarmEventMode 属性更改回 1 才能处理后续事件。

  如果直到纠正或确认条件之后应用程序才应该再次执行某个操作，请使用此设置。例如，触发事件时，ActiveX 脚本可以发出报警声，直至报警得到确认为止。随后在下次收到新报警时，可以再次发出报警声。
- 如果将 NewAlarmEventMode 属性设置为 2，则 NewAlarm 事件是活动的，并且在至少一个新的未确认报警到达时继续触发。您不需要更改这个值便可以处理后续事件。新事件每秒最多触发一次，不论同一秒内是否还有其它新的报警到达。

实时确认报警

当标记数据从正常状态转换为报警状态时，生成一个新的报警实例。InTouch“ 分布式报警系统”通过以下状态跟踪每个报警实例：

- 标记第一次进入报警状态时
- 报警发生子状态转换（如果该报警是一个多状态报警）时
- 报警返回到正常时
- 报警是否正在等待确认
- 确认报警时

与报警关联的标记值返回正常的非报警状态时，报警实例的生命周期便会结束。进入报警状态的后续转换会生成新的报警实例。

理解报警确认模型

InTouch HMI 支持三种报警确认模型。

- 对于面向条件的报警，在进行确认之前，确认过程会统计进入报警状态的所有项目。
- 对于扩展的摘要报警，确认只针对特定的转换，无论是进入报警状态、子状态，还是恢复到正常状态。进入不同报警子状态的所有转换都必须得到确认，之后整个报警才会视为已确认。
- 对于面向事件的报警（例如在 OPC 中的情况），仅当确认指最新的激活事件时，它才会被接受。

条件确认报警模型

对于面向条件的报警，在进行确认之前，确认过程会统计进入报警状态的所有项目。

确认针对的是报警的实例。报警实例在第一次进入报警状态时，便开始等待确认。如果报警得到确认，并随后转入新的报警子状态（例如从 "Hi" 转入 "HiHi"），则它开始等待下一次确认。只要得到确认，它都会被接受并应用于目前为止已发生的所有报警状态转换。

最新的实例得到确认时，报警会视为已确认。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

扩展的摘要报警模型

对于扩展的摘要报警，确认只针对特定的转换，无论是进入报警状态、子状态，还是恢复到正常状态。进入不同报警子状态的所有转换都必须得到确认，之后整个报警才会视为已确认。

首次进入报警状态必须得到确认，返回到正常也必须单独确认。

进入新报警子状态的任何转换都视为必须确认的新实例，其返回到正常状态的转换也必须得到确认。子状态转换视为属于“返回正常组”，自项目从先前的正常状态首次进入报警状态开始。

如果该项目返回到正常状态，但后来又进入报警状态，那么会创建一个新的返回到正常组。

每次转换都必须明确得到确认；仅当该项目已经返回到正常状态且处于暂停状态的所有转换都返回到正常组且已得到确认后，才会将该报警视为已确认。

注意：“摘要”一词表示“等待确认”。此模型中的报警也称为“回铃”报警。

扩展的摘要报警记录

对于扩展的摘要报警，发生报警时，在报警显示对象中生成一条记录，显示已发生报警条件。此记录显示报警的日期与时间标签。此记录保持可见，直到操作员确认报警并发生返回到正常状态为止。如果在确认报警之前发生返回到正常状态，则报警对象中显示两条记录。

例如，锅炉温度超出 high 限状态并触发报警。随后，在操作员确认报警之前锅炉返回到正常温度范围。报警系统生成一条记录。指出发生了 high 限报警；同时还生成另一条记录，指出报警尚未确认。

使用扩展的摘要报警

定义标记并选择扩展的摘要作为确认模型时，操作员必须确认发生的报警，即便触发报警的状态已返回到正常。确认报警会更改报警项的颜色，但不会更改时间标签。仅当报警得到确认并已返回到正常状态时，报警才从显示中清除。

注意：使用扩展的摘要确认模式定义标记时，“报警属性”对话框中的返回意味着确认选项不应用于该标记。

基于事件的报警模型

对于面向事件的报警（例如在 OPC 中的情况），仅当确认指最新的激活事件时，它才会被接受。

报警实例第一次进入报警状态时，便开始等待确认。如果实例得到确认，并随后转入新的报警子状态，则它开始等待下一次确认。每个后续转换都会被指定一个序号，确认必须在其后附加与之对应的转换的序号。

确认只有在针对最近一次转换时才会被接受。如果被接受，它将应用于该报警目前为止发生的所有状态转换。最新的实例得到确认时，报警会视为已确认。

被拒绝的确认可能会记录下来以用于诊断目的，但系统中不会以其它方式进行跟踪。

面向事件的模型确保报警在不同的状态之间变化时，确认可以对应到当前的信息。在延迟时间较短的系统中，这可能看上去像是面向条件的报警。在其它环境（如 Internet）中，此模型的功能可能会变得非常重要。

在运行时检查标记的确认模型

使用 .AlarmAckModel 点域监视与标记关联的确认模型。

.AlarmAckModel 点域

监视与标记关联的确认模型，具体如下：

0 = 条件（缺省值）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1 = 面向事件

2 = 扩展的摘要

类别

报警

用法

```
Tagname.AlarmAckModel
```

参数

Tagname
任何离散、整型、实型、间接离散以及模拟标记。

附注

此点域的缺省值为 0（条件确认模型）。

数据类型

模拟型（可读写）

有效值

0、1 或 2

示例

如果 PumpStation 标记与事件报警关联，则会处理此 IF-THEN 语句的主体。

```
IF (PumpStation.AlarmAckModel == 1) THEN
   MyAlarmMessage="PumpStation is an Event alarm";
ENDIF;
```

另请参阅

.Alarm、.Ack、.UnAck、.AckDev、.AckDSC、.AckROC

使用点域确认报警

您可以创建使用点域的脚本来确认所有的当前报警、所选报警或仅限特定类型的报警。

确认报警或报警组

您可以创建使用以下点域的脚本来确认指定的本地标记的报警，或指定的报警组中的报警。

- .Ack 点域
- .UnAck 点域

您无法使用这些点域来确认应用程序服务器产生的报警。

要确认正在运行的 InTouch 应用程序的所有本地报警，请将 $System 报警组与适当的 .Ack 点域结合起来使用。

.Ack 点域

监视或控制所有类型的本地报警的报警确认状态。

类别

报警

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用法

```
TagName.Ack=1;
```

参数

TagName
任何离散、整型、实型、间接离散与模拟标记，或报警组。

附注

将此点域设置为值 1 以确认与指定的标记或报警组关联的任何未确认报警。指定的标记是报警组时，与指定的组中的标记关联的所有未确认报警都会得到确认。指定的标记是报警组之外的任何其它类型时，则只有与该标记关联的未确认报警才会得到确认。将 .Ack 点域设置为 1 以外的值没有任何意义。

数据类型

离散型（可读写）

有效值

1

示例

以下语句确认与 Tag1 标记关联的报警：

```
Tag1.Ack=1;
```

下例用于确认 PumpStation 报警组内所有未确认的报警：

```
PumpStation.Ack=1;
```

注意：.ACK 点域有一个逆向点域 .UnAck。出现未确认的报警时，.UnAck 设置为 1。.UnAck 点域可以用在动画链接或条件脚本中，以触发任何未确认报警的触发器。

另请参阅

Alarm、UnAck、AckDev、AckROC、AckDSC、AckValue、AlarmAckModel

.UnAck 点域

监视或控制本地报警的报警确认状态。

类别

报警

用法

```
TagName.UnAck=0;
```

参数

TagName
任何离散、整型、实型、间接离散与模拟标记，或报警组。

附注

将此点域设置为值 0 以确认与指定的标记或报警组关联的任何未确认报警。指定的标记是报警组时，与指定的组中的标记关联的所有未确认报警都会得到确认。指定的标记是任何其它类型时，只有与该标记关联的未确认报警才会得到确认。将此点域设置为 0 以外的值没有任何意义。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

仅限离散（只读/复位）

有效值

0

示例

以下语句确认与 Tag1 标记关联的任何报警。

```
Tag1.UnAck=0;
```

此语句确认报警组 PumpStation 内所有未确认的报警。

```
PumpStation.UnAck=0;
```

.UnAck 有一个逆向点域 .Ack。已确认报警之后，.Ack 点域的值设置为 1。

另请参阅

.Ack、Ack()、.Alarm、.AlarmAckModel

确认值报警

您可以创建使用 .AckValue 点域的脚本来确认指定的本地标记的所有值报警，或指定的报警组中的所有值报警。

.AckValue 点域

监视或控制本地值报警的确认。

用法

```
TagName.AckValue=1;
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

将 **.AckValue** 点域设置为 1 以确认与指定的标记/组关联的任何未确认的值报警。指定的标记是报警组时，与指定的组中的标记关联的所有未确认的值报警都会得到确认。指定的标记是任何其它类型时，只有与该标记关联的未确认的值报警才会得到确认。

将此点域设置为 1 以外的值没有任何意义。

数据类型

离散型（可读写）

有效值

1

示例

以下语句确认与 Tag1 标记关联的值报警：

```
Tag1.AckValue=1;
```

本例确认报警组 PumpStation 内所有未确认的值报警：

```
PumpStation.AckValue=1;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

间接报警组（使用“组变量”）可以用来确认值报警。例如，使用以下赋值语句：

```
StationAlarms.Name = "PumpStation";
```

其中，StationAlarms 定义为报警组类型的标记，然后与 PumpStation 关联。因此，下面的语句与上例类似，只是它用于确认 PumpStation 报警组中任何未确认的值报警，该报警组当前与 StationAlarms 报警组标记关联。

```
StationAlarms.AckValue=1;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.UnAck、.AckDev、.AckDSC、.AckROC、.AlarmAckModel

确认离散报警

您可以创建使用 .AckDsc 点域的脚本来确认指定的标记的离散报警，或指定的报警组的所有离散报警。

.AckDsc 点域

确认指定的标记的离散报警，或指定的报警组的所有离散报警。

用法

```
TagName.AckDsc=1;
```

参数

TagName
指定给离散标记的名称，或报警组的名称。

附注

设置为 1 以确认与指定的标记或报警组关联的活动离散报警。指定的标记是报警组时，与指定的组中的标记关联的所有未确认的离散报警都会得到确认。指定的标记是报警组之外的任何类型时，只有与该标记关联的未确认的离散报警才会得到确认。将此点域设置为 1 以外的值没有任何意义。

数据类型

离散型（可读写）

有效值

0 或 1

示例

以下语句验证 Tag1 是否有关联的活动离散报警：

```
IF (Tag1.AlarmDsc == 1) THEN
   MyAlarmMessage="The pumping station currently has an ALARM!";
ENDIF;
```

此点域不链接到 .Ack 或 .UnAck 点域。因此，即使已经确认某个活动的报警，此点域仍等于 1。

另请参阅

.Alarm、.AlarmDSC、.Ack、.UnAck、.AckDev、.AckDSC、.AckROC、.AckValue、.AlarmAckModel

确认偏差报警

您可以创建使用 .AckDev 点域的脚本来确认指定的本地标记主、副偏差报警，或指定的报警组的偏差报警。

.AckDev 点域

确认指定的本地标记的主、副偏差报警，或指定的报警组的所有偏差报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

用法

```
TagName.AckDev=1;
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

将此点域设置为值 1 以确认与指定的标记或报警组关联的任何未确认的偏差报警。指定的标记是报警组时，与指定的组中的标记关联的所有未确认的偏差报警都会得到确认。将此点域设置为 1 以外的值没有任何意义。

数据类型

离散型（可读写）

有效值

1

示例

以下语句确认与 Tag1 标记关联的偏差报警：

```
Tag1.AckDev=1;
```

接下来的这个示例用于确认 PumpStation 报警组中所有未确认的偏差报警：

```
PumpStation.AckDev=1;
```

另请参阅

.Alarm、.AlarmDev、.Ack、.UnAck、.AckDSC、.AckROC、.AckValue、.AlarmAckModel

确认变化率报警

您可以创建使用 .AckROC 点域的脚本来确认指定的本地标记的变化率报警，或指定的报警组的变化率报警。

.AckROC 点域

确认指定的本地标记的变化率报警，或指定的报警组的所有变化率报警。

用法

```
TagName.AckROC=1;
```

参数

TagName
指定给整型、实型、间接模拟标记的名称，或报警组的名称。

附注

将此点域设置为值 1 以确认与指定的标记或报警组关联的任何未确认的变化率报警。指定的标记是报警组时，与指定的组中的标记关联的所有未确认的变化率报警都会得到确认。指定的标记是报警组之外的任何其它类型时，只有与该标记关联的未确认的变化率报警才会得到确认。将此点域设置为 1 以外的值没有任何意义。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

离散型（可读写）

有效值

1

示例

以下语句确认与 Tag1 标记关联的变化率报警。

```
Tag1.AckROC=1;
```

接下来的这个示例确认 PumpStation 报警组内所有未确认的变化率报警：

```
PumpStation.AckROC=1;
```

另请参阅

.Alarm、.AlarmROC、.Ack、.UnAck、.AckDev、.AckDSC、.AckValue、.AlarmAckModel

使用脚本函数确认报警

您可以使用 Ack() 脚本函数确认标记或组的所有报警。

如果使用 Alarm Viewer 控件，则可以使用方法来确认报警。如需有关详细信息，请参阅确认报警。

如果使用“分布式报警显示”对象，则可以使用脚本函数来确认报警。如需有关详细信息，请参阅确认报警。

Ack() 函数

确认任何未确认的 InTouch 报警。

类别

报警

语法

```
Ack TagName;
```

参数

TagName
任何 InTouch 标记、报警组或组变量。

附注

此函数可应用于标记或报警组。

示例

以下语句可以用在按钮上，以确认任何未确认的报警：

```
Ack $System; {All alarms}
Ack Tagname;
Ack GroupName;
```

另请参阅

**almAckAll()、almAckGroup()、almAckTag()、almAckDisplay()、almAckRecent()、almAckPriority().
almAckSelect()、almAckSelectedGroup()、almAckSelectedPriority()、almAckSelectedTag()**

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

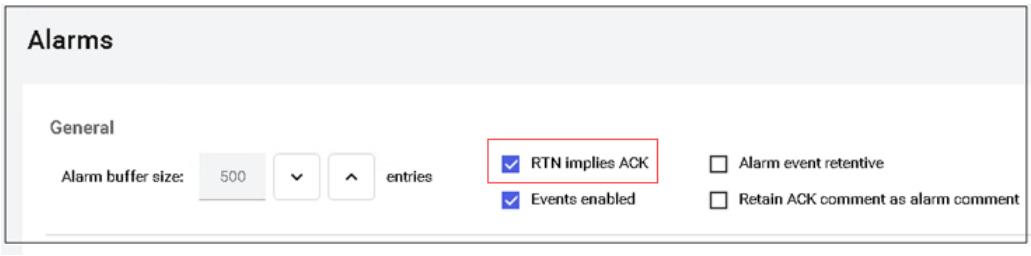
标记值恢复到正常时使用自动确认

InTouch HMI 可以在报警标记的值返回到正常时自动确认报警。此选项不应用于扩展的摘要报警。

配置返回时的报警确认

1. 在 WindowMaker 中打开一个应用程序。
2. 在文件菜单上，指向配置，然后选择报警。

   此时出现报警配置屏幕。
3. 在“常规”区域中，选择返回意味着确认，以使 InTouch HMI 自动确认其值返回到正常状态 (RTN) 的报警。

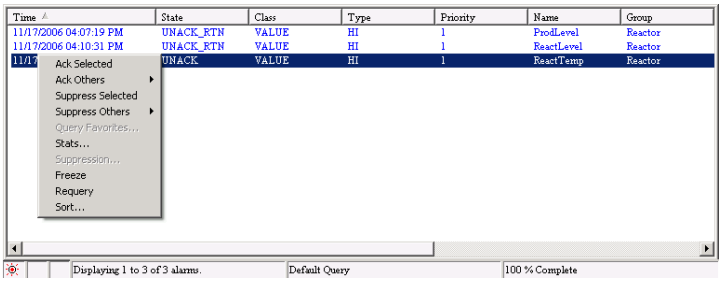


1. 选中返回意味着确认复选框。
2. 选择确定。

使用报警客户端确认报警

操作员可以从 WindowViewer 中确认 Alarm Viewer 控件中出现的报警。

在显示对象中，状态列显示所选报警记录的当前确认状态。此外，记录的文本也采用特定的颜色，以指出其确认状态。



此时确认的报警从显示对象中删除。

确认所有报警

1. 在显示对象中单击鼠标右键，指向确认其它，然后单击适当的命令：
   - 选择确认全部以确认所有当前报警。
   - 选择确认可见项以确认显示对象中所有的可见报警。

   此时出现确认注释对话框。
2. 输入可选确认注释，然后选择确定。

确认所选报警

1. 选择一个或多个报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 单击鼠标右键，然后选择确认已选项。此时出现确认注释对话框。
3. 输入可选确认注释，然后选择确定。

按组、标记或优先级确认报警

1. 选择报警。
2. 在显示对象中单击鼠标右键，指向确认其它，然后单击适当的命令：
   - 选择确认已选组，以确认属于所选报警组的所有报警。
   - 选择确认已选标记，以确认其所有标记的名称与所选报警的标记名相同的报警。
   - 选择确认已选优先级，以确认与所选报警具有相同的一个或多个优先级的所有报警。

   此时出现确认注释对话框。
3. 输入可选确认注释，然后选择确定。

使用报警与确认注释

注释有两种类型：报警注释与确认注释。

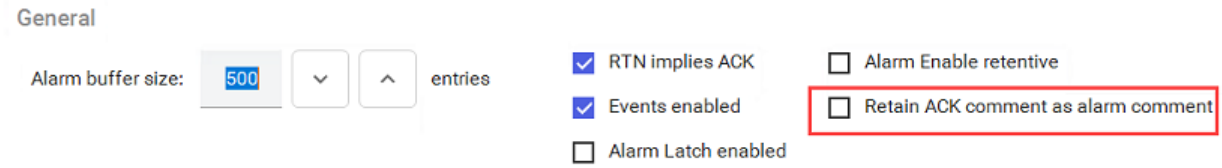
- 报警注释在发生新的报警实例时设置。.AlarmComment 点域用作报警注释，并可以在 InTouch 脚本中进行设置或读取。您可以在“标记名字典”的标记定义中指定此注释的缺省值。报警注释最多可以包含 131 个字符。
- 确认注释由操作员在确认报警时提供。

您可以使用确认注释来更新标记数据库中的报警注释。

允许报警确认注释更新 .AlarmComment 点域

1. 在 WindowMaker 中打开一个应用程序。
2. 在文件菜单上，指向配置，然后选择报警。

   此时出现报警屏幕。



3. 选中保留确认注释作为报警注释复选框，以使用随报警确认输入的注释来更新标记的 .AlarmComment 点域与“标记名字典”。

   如果未选中此复选框，确认注释会随同确认的报警一起显示（在数据库、打印件以及显示对象中），但 .AlarmComment 不会改变。
4. 选择确定。

在运行时解除报警

当报警发生时，运行时操作员（或系统）可确认该警报以表明他们注意到该报警。如果为 Galaxy 启用了锁存功能，那么将继续显示返回到正常值的已确认报警。在启用锁存功能后，会将返回到正常状态的已确认报警置于 LATCHED 报警状态。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

可以在当前报警模式下显示 LATCHED 报警，以表明这些报警确实发生过。在以下情况下，报警将进入 LATCHED 状态：

- 确认 UNACK_RTN 状态的报警。

  或

- 报警从 ACK 状态返回到正常状态。

要查看 LATCHED 状态，您必须在全局设置中启用 LATCHED 状态。要了解更多信息，请参阅 *AVEVA OMI* 帮助中的“启用锁存状态”。

您可以解除 LATCHED 报警，以从报警客户端控件网格的当前模式中删除 LATCHED 报警。解除的 LATCHED 报警将在报警客户端控件的最近模式中可见。

要解除所选报警

1. 选择一个或多个处于 **LATCHED** 状态的报警。
2. 使用鼠标右键单击网格上的任意位置，然后选择**关闭已选项**。

这会从网格中删除所选报警。

要解除其它报警

1. 选择一个或多个处于 **LATCHED** 状态的报警。
2. 使用鼠标右键单击网格上的任意位置，指向**关闭其他**，然后选择以下其中一个选项：
   - 全部**关闭**，以解除处于报警状态的所有报警
   - **关闭可见项**，以解除所有可见报警
   - **关闭已选组**，以解除与一个或多个所选活动报警具有相同供应器名和组名的报警
   - **关闭已选标记**，以解除与一个或多个所选活动报警具有相同供应器名、组名和标记名的报警

这会从网格中删除相关报警。

使用点域解除报警

您可以创建脚本以使用点域来解除所有处于 LATCHED 状态的报警。

类别

报警

用法

```
TagName.Dismiss=1;
```

参数

TagName
任何离散、整型、实型、间接离散与模拟标记，或报警组。

附注

将此点域设置为值 1 以解除与指定标记或报警组关联的任何未确认报警。当指定的标记是报警组时，将会解除与指定组中的标记关联的所有 LATCHED 报警。当指定的标记是报警组以外的任何其它类型时，只会解除与该标记关联的 LATCHED 报警。将 .Dismiss 点域设置为 1 以外的值没有任何意义。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

离散型（可读写）

有效值

1

示例

以下语句会解除与 Tag1 标记关联的报警：

```
Tag1.Dismiss=1;
```

接下来的这个示例用于解除 PumpStation 报警组内的所有 LATCHED 报警：

```
PumpStation.Dismiss=1;
```

在运行时控制标记与组的报警属性

您可以使用报警点域来动态管理报警条件。许多这样的点域都可以通过使用 I/O、表达式及脚本进行访问。通过 I/O 访问，您可以使用其它 Windows 应用程序（如 Excel 或远程节点上运行的 WindowViewer）来监视并控制特定标记的报警信息。

要访问与标记关联的点域，请使用以下语法：

tag.dotfield

例如，如果希望允许在运行时更改名为 Analog_tag 的标记的 HiHi 报警限，可为一个按钮创建“模拟 - 用户输入”触动链接，并在该链接的对话框中输入 Analog_tag.HiHiLimit 作为表达式。在运行期间，操作员只需选择该按钮，并为指定给 Analog_tag 的 HiHi 报警限输入新的值即可。

下表简要介绍每个报警点域。

点域	描述
.Ack	监视/控制标记与报警组的报警确认状态。 .Ack 有一个逆向标记点域 .UnAck。出现未确认的报警时，.UnAck 设置为 1。.UnAck 点域可以用在动画链接或条件脚本中，以触发任何未确认报警的触发器。
.AckDev	监视/控制模拟标记或报警组上活动的偏差报警的报警确认状态。
.AckDsc	监视/控制离散标记的当前确认状态。
.AckROC	监视/控制标记上活动的变化率报警的报警确认状态。
.AckValue	监视标记上活动的值报警的报警确认状态。
.Alarm	发出存在报警条件的信号。
.AlarmAckModel	监视与标记关联的确认模型，具体如下： 0 = 条件（缺省值） 1 = 事件

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	描述
	2 = 扩展 应用于发生报警的离散或模拟标记。只读，但可以在 WindowMaker 中配置。
.AlarmDev	发出存在偏差报警的信号。
.AlarmDevCount	统计给定的标记或报警组上活动的偏差报警总数。
.AlarmDevDeadband	监视/控制主、副偏差报警的偏差死区百分比。
.AlarmDevUnAckCount	统计给定的标记或报警组上未确认的活动偏差报警总数。
.AlarmDisabled	禁用/启用事件与报警。应用于发生报警的离散与模拟标记，或是应用于报警组。
.AlarmDsc	指出离散报警条件当前处于活动状态。
.AlarmDscCount	统计给定的标记或报警组上活动的离散报警总数。
.AlarmDscDisabled	指出标记是否可以生成离散报警。 注意：对于离散标记，此点域与 .AlarmDisabled 点域相同。
.AlarmDscEnabled	指出标记是否可以生成离散报警。 注意：对于离散标记，此点域与 .AlarmEnabled 点域相同。
.AlarmDscInhibitor	返回指定给此标记的离散报警（如果有）的约束标记名。
.AlarmDscUnAckCount	统计给定的标记或报警组上未确认的活动离散报警总数。
.AlarmEnabled	禁用/启用事件与报警。
.AlarmHiDisabled	禁用/启用发生报警的模拟标记的 High 报警限。
.AlarmHiHiDisabled	禁用/启用发生报警的模拟标记的 HiHi 报警限。
.AlarmHiHiEnabled	禁用/启用发生报警的模拟标记的 HiHi 报警限。
.AlarmHiHiInhibitor	返回 HiHi 报警限的约束标记引用。应用于发生报警的模拟标记。只读，但可以在 WindowMaker 中配置。
。**AlarmHiInhibitor**	返回 High 报警限的约束标记引用。应用于发生报警的模拟标记。 只读，但可以在 WindowMaker 中配置。
.AlarmLoDisabled	禁用/启用发生报警的模拟标记的 Low 报警限。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	描述
.AlarmLoEnabled	禁用/启用发生报警的模拟标记的 Low 报警限。
。**AlarmLoInhibitor**	返回 Low 报警限的约束标记引用。应用于发生报警的模拟标记。只读，但可以在 WindowMaker 中配置。
.AlarmLoLoDisabled	禁用/启用发生报警的模拟标记的 LoLo 报警限。
.AlarmLoLoEnabled	禁用/启用发生报警的模拟标记的 LoLo 报警限。
。**AlarmLoLoInhibitor**	返回 LoLo 报警限的约束标记引用。应用于发生报警的模拟标记。只读，但可以在 WindowMaker 中配置。
.AlarmMajDevDisabled	禁用/启用发生报警的模拟标记的“主偏差”报警限。
.AlarmMajDevEnabled	禁用/启用发生报警的模拟标记的“主偏差”报警限。
。**AlarmMajDevInhibitor**	返回“主偏差”报警限的约束标记引用。应用于发生报警的模拟标记。只读，但可以在 WindowMaker 中配置。
.AlarmMinDevDisabled	禁用/启用发生报警的模拟标记的“副偏差”报警限。
.AlarmMinDevEnabled	禁用/启用发生报警的模拟标记的“副偏差”报警限。
。**AlarmMinDevInhibitor**	返回“副偏差”报警限的约束标记引用。应用于发生报警的模拟标记。只读，但可以在 WindowMaker 中配置。
.AlarmROC	发出存在变化率报警的信号。
.AlarmROCCount	统计给定的标记或报警组上活动的变化率报警总数。
.AlarmROCDisabled	禁用/启用发生报警的模拟标记的变化率报警限。
.AlarmROCEnabled	禁用/启用发生报警的模拟标记的变化率报警限。
。**AlarmROCInhibitor**	返回变化率报警限的约束标记引用。应用于发生报警的模拟标记。只读，但可以在 WindowMaker 中配置。
.AlarmROCUnAckCount	统计给定的标记或报警组上未确认的变化率报警总数。
.AlarmTotalCount	统计给定的标记或报警组上活动的报警总数。
.AlarmUnAckCount	统计给定的标记或报警组上未确认的活动报警总数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	描述
.AlarmUserDefNum1Set	可读写实型（浮点）值，缺省值为 0，且值未设置。应用于发生报警的离散标记、发生报警的模拟标记，或应用于报警组。 **注意：**此点域的值将附加到报警上，但“仅限于”已设置值（例如通过脚本或 POKE）的情况。
.AlarmUserDefNum2	可读写实型（浮点）值，缺省值为 0，且值未设置。应用于发生报警的离散标记、发生报警的模拟标记，或应用于报警组。 **注意：**此点域的值将附加到报警上，但“仅限于”已设置值（例如通过脚本或 POKE）的情况。
.AlarmUserDefNum2Set	可读写离散值。如果脚本为相应的标记定义了 .AlarmUserDefNum2，则为 TRUE。要取消与该标记的值 .AlarmUserDefNum2 的关联，请将此点域设置为 FALSE。缺省值为 FALSE。
.AlarmUserDefStr	可读写文本字符串，缺省值为 ""，且值未设置。 应用于发生报警的离散标记、发生报警的模拟标记，或应用于报警组。 **注意：**此点域的值将附加到报警上，但“仅限于”已设置值（例如通过脚本或 POKE）的情况。
.AlarmUserDefStrSet	可读写离散值。如果脚本为相应的标记定义了 .AlarmUserDefStr，则为 TRUE。要取消与该标记的值 .AlarmUserDefStr 的关联，请将此点域设置为 FALSE。缺省值为 FALSE。
.AlarmValDeadband	监视/控制报警值死区的值。
.AlarmValueCount	统计给定的标记名或报警组上活动的值报警总数。
.AlarmValueUnAckCount	统计给定的标记名或报警组上未确认的活动值报警总数。
.DevTarget	监视/控制主、副偏差报警的目标。
.HiLimit、.HiHiLimit、.LoLimit、.LoLoLimit	可读写模拟标记名点域，用于监视/控制值报警检查的极限。这些点域只对整型与实型标记有效。
.HiStatus、.HiHiStatus、.LoStatus、.LoLoStatus	只读离散点域，确定是否存在指定类型的报警。这些点域只对整型与实型标记有效。
.MajorDevPct	可读写整型点域，用于监视或控制报警检查的主偏差百分比。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

点域	描述
.MajorDevStatus	只读离散点域，用于确定指定的标记名是否存在主偏差报警。
.MinorDevPct	可读写整型点域，用于监视与/或控制报警检查的副偏差百分比。
.MinorDevStatus	只读离散点域，用于确定指定的标记名是否存在副偏差报警。
.Name	可读写消息点域，用于显示标记名的实际名称。例如，它可用于确定“组变量”所指向的“报警组”的名称，或 TagID 标记的名称。它也可以写入，以更改“组变量”所指向的“报警组”。
.Normal	只读离散点域，指定的标记名不存在报警时它等于 1。此点域对于“报警组”、“组变量”及常规标记而言是有效的。
.ROCPct	可读写点域，用于监视与/或控制报警检查的变化率。
.ROCStatus	只读离散点域，用于确定指定的标记是否存在变化率报警。

确定标记或报警组是否处在报警条件中

使用以下点域与系统标记确定正在运行的应用程序中的报警状态。您可以找出是否发生了新报警，标记或报警组处在报警还是正常状态。您也可以找出离散、偏差及变化率报警的状态。

- $NewAlarm 系统标记
- $System 系统标记
- .Alarm 点域
- .Normal 点域
- .AlarmDsc 点域
- .AlarmDev 点域
- .AlarmROC 点域
- .LoStatus 点域
- .LoLoStatus 点域
- .HiStatus 点域
- .HiHiStatus 点域
- .MinorDevStatus 点域
- .MajorDevStatus 点域
- .ROCStatus 点域

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

$NewAlarm 系统标记

如果正在运行的应用程序中发生新的报警，则设置为 1。**$NewAlarm** 系统标记不能针对远程报警进行设置。

语法

```
$NewAlarm=value;
```

数据类型

离散型（可读写）

有效值

0 或 1

附注

将 **$NewAlarm** 系统标记关联到应用程序窗口中的动画对象。例如，将该标记关联到某个确认按钮（操作员选择时可以将该标记的值重置为 0 并确认报警）。您也可以将 **$NewAlarm** 系统标记链接到 **PlaySound** 逻辑函数，以便在发生报警时发出有声警告。

示例

将某个按钮添加到报警确认窗口，并给它附加在操作员选择按钮时运行的以下动作脚本。

```
Ack $System;
$NewAlarm=0;
HideSelf;
```

操作员选择该按钮时，会确认所有报警，**$NewAlarm** 系统标记重置为 0，报警确认窗口会隐藏起来。

$System 系统标记

缺省报警组。

类别

报警

用法

```
$System
```

附注

缺省情况下，系统将标记指定给这个根报警组。定义的所有报警组都由 $System 衍生而来。

数据类型

系统报警组

示例

```
$System.Ack = 1; {确认所有报警}
```

.Alarm 点域

指定的标记或报警组当前不处于报警状态时返回 0。出现警报时，**.Alarm** 点域返回 1。在报警条件消失之前，它保持为 1。**.Alarm** 点域有一个逆向点域，名为 **.Normal**。

如果指定的标记是报警组的名称，则只要属于该组的任何标记处于报警状态，**.Alarm** 点域都返回 1。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

用法

```
TagName.Alarm
```

参数

TagName
任何离散、整型、实型、间接离散与模拟标记，或报警组标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

以下语句验证 **Tag1** 是否有关联的活动报警：

```
IF (Tag1.Alarm == 1) THEN
```

如果 **PumpStation** 报警组内存在活动的报警，则处理此 IF-THEN 语句的主体。

```
IF (PumpStation.Alarm == 1) THEN
   MyAlarmMessage="The pumping station currently has an ALARM!";
ENDIF;
```

此点域不链接到 **.Ack** 或 **.UnAck** 点域。因此，即使已经确认活动报警，**.Alarm** 仍等于 1。

.Normal 点域

指定的标记不处于报警状态时返回 1。出现警报时，**.Normal 点域** 返回 0。**.Normal** 点域有一个逆向点域，名为 **.Alarm**。

类别

报警

语法

```
TagName.Normal
```

参数

TagName
任何离散、整型、实型、间接离散与模拟标记，或报警组标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

没有任何报警与 **Tag1** 标记关联时，运行以下 IF-THEN 语句。**Tag1** 有一个或更多活动报警时，运行 "ELSE" 主体。

```
IF (Tag1.Normal==1) THEN
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
   MyOperatorMessage="Tag1 is OK - No alarms associated with it";
ELSE
   MyOperatorMessage="Tag1 has one or more alarms active!";
ENDIF;
```

.AlarmDsc 点域

指出指定的离散标记或报警组是否存在报警条件。缺省值为 0。当指定标记的离散报警条件存在时，它设置为值 1。在报警条件消失之前，值保持为 1。

如果指定的标记是报警组的名称，则只要该组中的任何标记处于活动离散报警状态，.**AlarmDsc** 点域都会设置为 1。

类别

报警

用法

```
TagName.AlarmDsc
```

参数

TagName
任何离散标记、间接离散标记，或报警组。

数据类型

离散型（只读）

有效值

0 或 1

示例

以下语句验证 Tag1 是否有关联的活动离散报警：

```
IF (Tag1.AlarmDsc == 1) THEN
   MyAlarmMessage="The pumping station currently has an ALARM!";
ENDIF;
```

此点域不链接到 **.Ack** 或 **.UnAck** 点域。因此，即使已经确认活动报警，**.AlarmDsc** 点域仍等于 1。

另请参阅

.Ack、.UnAck、.Alarm、.AlarmDsc、.AckDsc

.AlarmDev 点域

指出指定的标记或报警组的偏差报警何时变为活动状态。缺省值为 0。当指定标记的偏差报警条件存在时，它设置为值 1。在报警条件消失之前，值保持为 1。

如果指定的标记是报警组的名称，则只要该组中的任何标记处于活动报警状态，**.AlarmDev** 点域都会设置为 1。

类别

报警

用法

```
TagName.AlarmDev
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

数据类型

离散型（只读）

有效值

0 或 1

示例

以下语句验证 Tag1 是否有关联的活动偏差报警：

```
IF (Tag1.AlarmDev == 1) THEN
```

如果 PumpStation 报警组内存在活动的偏差报警，则处理此 IF-THEN 语句的主体。

```
IF (PumpStation.AlarmDev == 1) THEN
   MyAlarmMessage="The pumping station currently has an ALARM!";
ENDIF;
```

此点域不链接到 .Ack 或 .UnAck 点域。因此，即使已经确认某个活动的报警，此点域仍等于 1。

另请参阅

.Ack、.UnAck、.Alarm、.AckDev

.AlarmROC 点域

指出指定的标记或报警组的变化率报警条件何时变为活动状态。缺省值为 0。当指定标记的变化率报警条件存在时，它设置为值 1。在变化率报警条件消失之前，值保持为 1。

如果指定的标记是报警组的名称，则只要该组中的任何标记处于变化率报警状态，**.AlarmROC** 点域都会设置为 1。

类别

报警

用法

```
TagName.AlarmROC
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

数据类型

离散型（只读）

有效值

0 或 1

示例

以下语句验证 Tag1 是否有关联的活动变化率报警：

```
IF (Tag1.AlarmROC == 1) THEN
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果 PumpStation 报警组内存在活动的变化率报警，则处理此 IF-THEN 语句的主体。

```
IF (PumpStation.AlarmROC == 1) THEN
   MyAlarmMessage="The pumping station currently has an ALARM!";
ENDIF;
```

此点域不链接到 .Ack 或 .UnAck 点域。因此，即使已经确认活动的变化率报警，此点域仍等于 1。

另请参阅

.Ack、.AckROC、.Alarm、.AlarmROCEnabled、.AlarmROCDisabled

.LoStatus 点域

指出指定的标记或报警组的 Low 报警条件何时变为活动状态。缺省值为 0。当指定标记的 Low 报警条件存在时，它设置为值 1。在 Low 报警条件消失之前，值保持为 1。

此点域常常同 **.Alarm** 与 **.Ack** 点域结合使用，以确定特定标记的特定报警状态。

类别

报警

用法

```
TagName.LoStatus
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

MyTag 标记的 .LoStatus（Low 报警条件）等于 1 时，运行以下 IF-THEN 语句。

```
IF (MyTag.LoStatus == 1) THEN
   OperatorMessage="MyTag has gone into Low Alarm";

ENDIF;
```

另请参阅

.Alarm、.AlarmValue、.Ack, .LoLimit、.LoSet、.AlarmDisabled、.AlarmEnabled、.AlarmLoDisabled、.AlarmLoEnabled、.AlarmLoInhibitor

.LoLoStatus 点域

指出指定的标记或报警组的 LoLo 报警条件何时变为活动状态。缺省值为 0。当指定标记的 LoLo 报警条件存在时，它设置为值 1。在 LoLo 报警条件消失之前，值保持为 1。

此点域常常同 **.Alarm** 与 **.Ack** 点域结合使用，以确定系统内特定标记的报警状态的确切性质。

类别

报警

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用法

```
TagName.LoLoStatus
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

MyTag 标记的 .LoLoStatus（LoLo 报警限）等于 1 时，运行以下 IF-THEN 语句。

```
IF (MyTag.LoLoStatus == 1) THEN
   OperatorMessage="MyTag has gone into LoLo Alarm";
ENDIF;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.LoLoLimit、.LoLoSet、.AlarmDisabled、.AlarmEnabled、.AlarmLoLoDisabled、.AlarmLoLoEnabled、.AlarmLoLoInhibitor

.HiStatus 点域

指出指定的标记或报警组的 High 报警条件何时变为活动状态。缺省值为 0。当指定标记的 High 报警条件存在时，它设置为值 1。在 High 报警条件消失之前，值保持为 1。

此点域常常同 **.Alarm** 与 **.Ack** 点域结合使用，以确定标记的特定报警状态。

类别

报警

用法

```
TagName.HiStatus
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

如果 MotorAmps 标记进入 High 报警限报警状态，则此脚本会调用另一个脚本，以停止泵浦马达输出。

```
IF (MotorAmps.HiStatus == 1) THEN
   CALL PumpShutdown( );
ENDIF;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

.Alarm、.AlarmValue、.Ack、.HiLimit、.HiSet、.AlarmDisabled、.AlarmEnabled、.AlarmHiDisabled、.AlarmHiEnabled、.AlarmHiInhibitor

.HiHiStatus 点域

指出指定的标记或报警组的 HiHi 报警条件何时变为活动状态。缺省值为 0。当指定标记的 HiHi 报警条件存在时，它设置为值 1。在 HiHi 报警条件消失之前，值保持为 1。

此点域常常同 **.Alarm** 与 **.Ack** 点域结合使用，以确定标记的特定报警状态。

类别

报警

用法

```
TagName.HiHiStatus
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

MyTag 标记的 .HiHiStatus（HiHi 报警）为 1 时，运行以下 IF-THEN 语句。

```
IF (MyTag.HiHiStatus == 1) THEN
   OperatorMessage="MyTag has gone into HiHi Alarm";
ENDIF;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.HiHiLimit、.HiHiSet、.AlarmDisabled、.AlarmEnabled、.AlarmHiHiDisabled、.AlarmHiHiEnabled、.AlarmHiHiInhibitor

.MinorDevStatus 点域

指出指定的标记或报警组的副偏差报警何时变为活动状态。缺省值为 0。当指定标记的副偏差报警条件存在时，它设置为值 1。在副偏差报警条件消失之前，值保持为 1。

此点域常常同 **.Alarm** 与 **.Ack** 点域结合使用，以确定标记的特定报警状态。

类别

报警

用法

```
TagName.MinorDevStatus
```

参数

TagName
任何整型、实型或间接模拟标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

离散型（只读）

有效值

0 或 1

示例

MyTag 标记的 .MinorDevStatus（副偏差报警）等于 1 时，运行以下 IF-THEN 语句。

```
IF (MyTag.MinorDevStatus == 1) THEN
   OperatorMessage="MyTag has gone into a Minor Deviation Alarm";
ENDIF;
```

另请参阅

.AckDev、**.AlarmDev**、**.AlarmMinDevDisabled**、**.AlarmMinDevEnabled**、**.AlarmMinDevInhibitor**、**.MinorDevPct**、**.MajorDevStatus**

.MajorDevStatus 点域

指出指定的标记或报警组的主偏差报警何时变为活动状态。缺省值为 0。当主偏差报警条件存在时，指定的点域设置为值 1。在主偏差报警条件消失之前，值保持为 1。

此点域常常同 **.Alarm** 与 **.Ack** 点域结合使用，以确定标记的特定报警状态。

类别

报警

用法

```
TagName.MajorDevStatus
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

MyTag 标记的 .MajorDevStatus（主偏差报警）等于 1 时，运行以下 IF-THEN 语句。

```
IF (MyTag.MajorDevStatus == 1) THEN
   OperatorMessage="MyTag has gone into a Major Deviation Alarm";
ENDIF;
```

另请参阅

.AckDev、.AlarmDev、.AlarmMajDevDisabled、.AlarmMajDevEnabled、.AlarmMajDevInhibitor、.MajorDevPct、.MajorDevSet、.MinorDevStatus

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.ROCStatus 点域

指出指定的标记或报警组的变化率报警何时变为活动状态。缺省值为 0。当指定标记的变化率报警条件存在时，它设置为值 1。在变化率报警条件消失之前，值保持为 1。

此点域常常同 **.Alarm** 与 **.Ack** 点域结合使用，以确定标记的特定报警状态。

类别

报警

用法

```
TagName.ROCStatus
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

MyTag 标记的 .ROCStatus（变化率报警）等于 1 时，运行以下 IF-THEN 语句。

```
IF (MyTag.ROCStatus == 1) THEN
   OperatorMessage="MyTag has gone into a Rate-Of-Change alarm";
ENDIF;
```

另请参阅

.ROCPct、.ROCSet

将报警状态处理转换为 InTouch 7.1 行为

对于 InTouch 7.11 以及更新的版本，发生 HiHi、LoLo、MinDev 或 MajDev 报警时的缺省行为是，将 In Alarm、ACK 或 $NewAlarm 状态分别设置为 Hi、Lo、MajDev 或 MinDev。对于 InTouch 7.1 以及先前的版本，HiHi、LoLo、MinDev 或 MajDev 报警不会影响这些报警状态。

要支持传统应用程序使用 Hi、Lo、MajDev 或 MinDev 报警来维护其 In Alarm、ACK 和 $NewAlarm 状态，而不再接受其他报警，可以在 InTouch.ini 文件中加入 IT71StatusFlag 参数。

要强制报警处理依照 InTouch 7.1 以及先前版本中的方式进行操作

- 在 InTouch.ini 文件中输入以下行：
  ```
  IT71StatusFlags=1
  ```

要将报警处理更改回 InTouch 7.11 或更高版本的缺省操作

- 在 InTouch.ini 文件中输入以下行：
  ```
  IT71StatusFlags=0
  ```

如需有关如何编辑 InTouch.ini 文件以配置此参数以及其它 InTouch 操作参数的详细信息，请参阅《*AVEVA™ InTouch HMI 应用程序开发指南*》。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

确定是否为标记设置了报警限

以下点域指出在应用程序运行期间是否给标记设置了报警限。

- .LoLoSet 点域
- .LoSet 点域
- .HiSet 点域
- .HiHiSet 点域
- .MinorDevSet 点域
- .MajorDevSet 点域
- .ROCSet 点域

.LoLoSet 点域

指出是否已经给整型或实型标记设置了 LoLo 报警限。

类别

报警

用法

```
TagName.LoLoSet
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

如果给 MyTag 标记设置了 LoLo 报警限，则执行 THEN 代码块：

```
IF (MyTag.LoLoSet== 1) THEN
   MsgTag="LoLo alarm limit has been set for MyTag";
ENDIF;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.LoLoStatus、.LoLoLimit、.AlarmDisabled、.AlarmEnabled、.AlarmLoLoDisabled、.AlarmLoLoEnabled、.AlarmLoLoInhibitor

.LoSet 点域

指出是否已经给整型或实型标记设置了 Low 报警限。

类别

报警

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用法

```
TagName.LoSet
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

如果给 MyTag 标记设置了 Low 报警限，则执行 THEN 代码块：

```
IF (MyTag.LoSet== 1) THEN
   MsgTag="Low alarm limit has been set for MyTag";
ENDIF;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.LoStatus、.LoLimit、.AlarmDisabled、.AlarmEnabled、.AlarmLoDisabled、.AlarmLoEnabled、.AlarmLoInhibitor

.HiSet 点域

指出是否已经给整型或实型标记设置了 High 报警限。

类别

报警

用法

```
TagName.HiSet
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

如果给 MyTag 标记设置了 High 报警限，则执行 THEN 代码块：

```
IF (MyTag.HiSet== 1) THEN
   MsgTag="High alarm limit has been set for MyTag";
ENDIF;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

.Alarm、.AlarmValue、.Ack、.HiHiStatus、.HiHiLimit、.AlarmDisabled、.AlarmEnabled、.AlarmHiHiDisabled、.AlarmHiHiEnabled、.AlarmHiHiInhibitor

.HiHiSet 点域

指出是否已经给整型或实型标记设置了 HiHi 报警限。

类别

报警

用法

```
TagName.HiHiSet
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

如果给 MyTag 标记设置了 HiHi 报警限，则执行 THEN 代码块：

```
IF (MyTag.HiHiSet== 1) THEN
   MsgTag="HiHi alarm limit has been set for MyTag";
ENDIF;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.HiHiStatus、.HiHiLimit、.AlarmDisabled、.AlarmEnabled、.AlarmHiHiDisabled、.AlarmHiHiEnabled、.AlarmHiHiInhibitor

.MinorDevSet 点域

指出是否已经给整型或实型标记设置了副偏差报警限。

类别

报警

用法

```
TagName.MinorDevSet
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

有效值

0 或 1

示例

如果给 MyTag 标记设置了副偏差百分比报警限，则执行 THEN 代码块：

```
IF (MyTag.MinorDevSet== 1) THEN
   MsgTag="Minor deviation alarm limit has been set for MyTag";
ENDIF;
```

另请参阅

.AckDev、.AlarmDev、.AlarmMinDevDisabled、.AlarmMinDevEnabled、.AlarmMinDevInhibitor、.MinorDevPct、.MinorDevStatus

.MajorDevSet 点域

指出是否已经给整型或实型标记设置了主偏差报警限。

类别

报警

用法

```
TagName.MajorDevSet
```

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

如果给 MyTag 标记设置了主偏差百分比报警限，则执行 THEN 代码块：

```
IF (MyTag.MajorDevSet== 1) THEN
   MsgTag="Major deviation alarm limit has been set for MyTag";
ENDIF;
```

另请参阅

.AckDev、.AlarmDev、.AlarmMajDevDisabled、.AlarmMajDevEnabled、.AlarmMajDevInhibitor、.MajorDevPct、.MajorDevStatus

.ROCSet 点域

指出是否已经给整型或实型标记设置了变化率报警限。

类别

报警

用法

```
TagName.ROCSet
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

TagName
任何整型、实型或间接模拟标记。

数据类型

离散型（只读）

有效值

0 或 1

示例

如果给 MyTag 标记设置了变化率报警限，则执行 THEN 代码块：

```
IF (MyTag.ROCSet == 1) THEN
   MsgTag="Rate-of-change alarm limit has been set for MyTag";
ENDIF;
```

另请参阅

.Alarm、.Ack, .LoLimit、.LoLoLimit、.HiHiLimit、.HiLimit、.HiSet、.LoSet、.LoLoSet、.HiStatus、.HiHiStatus、.ROCPct、.ROCStatus

启用与禁用标记或报警组的报警

InTouch HMI 提供一组点域，可以在应用程序运行期间启用或禁用给标记设置的报警。

启用/禁用所有报警

.AlarmEnabled 与 **.AlarmDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的报警。与这两个点域关联的报警状态互为反向。对于 **.AlarmEnabled**，值为 1 时给标记或报警组启用报警。**.AlarmDisabled** 的值为 1 时给标记或报警组禁用报警。

其中任一个点域给报警组启用报警时，都会启用属于该组的所有标记的报警。其中任一个点域禁用报警时，会忽略所有的事件与报警。这些报警不存储在报警内存中，也不写入磁盘。

.AlarmEnabled 点域

启用或禁用标记或报警组的报警。

类别

报警

用法

```
TagName.AlarmEnabled
```

参数

TagName
任何离散、整型、实型、间接离散、间接模拟标记，或报警组。

附注

.AlarmEnabled 设置为 0 时，会忽略所有的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

指定的标记是一个报警组时，指定的报警组中与标记关联的所有报警都会启用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

离散型（可读写）

有效值

0 = 禁用报警

1 = 启用报警（缺省）

示例

下例禁用 Tag1 标记的报警：

```
Tag1.AlarmEnabled=0;
```

另请参阅

.AlarmDisabled

.AlarmDisabled 点域

启用或禁用标记或报警组的报警。

类别

报警

用法

```
TagName.AlarmDisabled
```

参数

TagName
任何离散、整型、实型、间接离散、间接模拟标记，或报警组。

附注

.AlarmDisabled 设置为 1 时，会忽略所有的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

指定的标记是一个报警组时，指定的报警组中与标记关联的所有报警都会禁用。

这与 .AlarmEnabled 点域正好相反。

示例

下例启用 Tag1 标记的报警。

```
Tag1.AlarmDisabled=0;
```

另请参阅

.AlarmEnabled

启用/禁用 LoLo 报警

.AlarmLoLoEnabled 与 **.AlarmLoLoDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的 LoLo 报警。与这两个点域关联的 LoLo 报警状态互为反向。对于 **.AlarmLoLoEnabled**，值为 1 时启用标记或报警组的 LoLo 报警。**.AlarmLoLoDisabled** 的值为 1 时禁用标记或报警组的 LoLo 报警。

其中任一个点域启用报警组的 LoLo 报警时，都会启用属于该组的所有标记的 LoLo 报警。其中任一个点域禁用 LoLo 报警时，会忽略所有的 LoLo 报警。这些报警不存储在报警内存中，也不写入磁盘。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.AlarmLoLoEnabled 点域

启用或禁用 LoLo 条件的事件与报警。

类别

报警

用法

```
TagName.AlarmLoLoEnabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmLoLoEnabled 设置为 0 时，会忽略所有 LoLo 条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

数据类型

离散型（可读写）

有效值

0 = 禁用报警

1 = 启用报警（缺省）

示例

下例禁用 Tag1 标记的 LoLo 报警：

```
Tag1.AlarmLoLoEnabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmLoLoDisabled

.AlarmLoLoDisabled 点域

启用或禁用 LoLo 条件的事件与报警。

类别

报警

用法

```
TagName.AlarmLoLoDisabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmLoLoDisabled 设置为 1 时，会忽略所有 LoLo 条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

离散型（可读写）

有效值

1= 禁用报警

0 = 启用报警（缺省）

示例

下例启用 Tag2 标记的 LoLo 报警：

```
Tag2.AlarmLoLoDisabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmLoLoEnabled

启用/禁用 Low 报警

.AlarmLoEnabled 与 **.AlarmLoDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的 Low 报警。与这两个点域关联的 Low 报警状态互为反向。对于 **.AlarmLoEnabled**，值为 1 时启用标记或报警组的 Low 报警。**.AlarmLoDisabled** 的值为 1 时禁用标记或报警组的 Low 报警。

其中任一个点域启用报警组的 Low 报警时，都会启用属于该组的所有标记的 Low 报警。其中任一个点域禁用 Low 报警时，会忽略所有的 Low 报警。这些报警不存储在报警内存中，也不写入磁盘。

.AlarmLoEnabled 点域

启用或禁用 Low 条件的事件与报警。

类别

报警

用法

```
TagName.AlarmLoEnabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmLoEnabled 设置为 0 时，会忽略所有 Low 条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

数据类型

离散型（可读写）

有效值

0 = 禁用报警

1 = 启用报警（缺省）

示例

下例禁用 Tag1 标记的 Low 报警：

```
Tag1.AlarmLoEnabled=0;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmLoDisabled

.AlarmLoDisabled 点域

启用或禁用 Low 条件的事件与报警。

类别

报警

用法

```
TagName.AlarmLoDisabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmLoDisabled 设置为 1 时，会忽略所有 Low 条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

数据类型

离散型（可读写）

有效值

1= 禁用报警

0 = 启用报警（缺省）

示例

下例启用 Tag2 标记的 Low 报警：

```
Tag2.AlarmLoDisabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmLoEnabled

启用/禁用 High 报警

.AlarmHiEnabled 与 **.AlarmHiDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的 High 报警。与这两个点域关联的 High 报警状态互为反向。对于 **.AlarmHiEnabled**，值为 1 时启用标记或报警组的 High 报警。**.AlarmHiDisabled** 的值为 1 时禁用标记或报警组的 High 报警。

其中任一个点域启用报警组的 High 报警时，都会启用属于该组的所有标记的 High 报警。其中任一个点域禁用 High 报警时，会忽略所有的 High 报警。High 报警不存储在报警内存中，也不写入磁盘。

.AlarmHiEnabled 点域

启用或禁用 High 条件的事件与报警。

类别

报警

用法

```
TagName.AlarmHiEnabled
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmHiEnabled 设置为 0 时，会忽略所有 High 条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmHiDisabled 点域正好相反。

数据类型

离散型（可读写）

有效值

0 = 禁用报警

1 = 启用报警（缺省）

示例

下例禁用标记 Tag1 的 High 报警：

```
Tag1.AlarmHiEnabled=0;
```

另请参阅

.AlarmHiDisabled、.AlarmEnabled

.AlarmHiDisabled 点域

启用或禁用 High 条件的事件与报警。

类别

报警

用法

```
TagName.AlarmHiDisabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmHiDisabled 设置为 1 时，会忽略所有 High 条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmHiEnabled 点域正好相反。

数据类型

离散型（可读写）

有效值

1= 禁用报警

0 = 启用报警（缺省）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

下例启用 Tag2 标记的 High 报警：

```
Tag2.AlarmHiDisabled=0;
```

另请参阅

.AlarmHiEnabled、.AlarmDisabled

启用/禁用 HiHi 报警

.AlarmHiHiEnabled 与 **.AlarmHiHiDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的 HiHi 报警。与这两个点域关联的 HiHi 报警状态互为反向。对于 **.AlarmHiHiEnabled**, 值为 1 时启用标记或报警组的 HiHi 报警。**.AlarmHiHiDisabled** 的值为 1 时禁用标记或报警组的 HiHi 报警。

其中任一个点域启用报警组的 HiHi 报警时, 都会启用属于该组的所有标记的 HiHi 报警。其中任一个点域禁用 HiHi 报警时，会忽略所有的 HiHi 报警。这些 HiHi 报警不存储在报警内存中，也不写入磁盘。

.AlarmHiHiEnabled 点域

启用与/或禁用 HiHi 条件的事件与报警。

类别

报警

用法

```
TagName.AlarmHiHiEnabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmHiHiEnabled 设置为 0 时，会忽略所有 HiHi 条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

数据类型

离散型（可读写）

有效值

0 = 禁用报警

1= 启用报警（缺省）

示例

下例禁用 Tag1 标记的 HiHi 报警：

```
Tag1.AlarmHiHiEnabled=0;
```

另请参阅

.AlarmHiHiDisabled、.AlarmEnabled

.AlarmHiHiDisabled 点域

启用或禁用 HiHi 条件的事件与报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

用法

```
TagName.AlarmHiHiDisabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmHiHiDisabled 设置为 1 时，会忽略所有 HiHi 条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmHiHiEnabled 点域正好相反。

数据类型

离散型（可读写）

有效值

1= 禁用报警

0 = 启用报警（缺省）

示例

下例启用 Tag2 标记的 HiHi 报警：

```
Tag2.AlarmHiHiDisabled=0;
```

另请参阅

.AlarmHiHiEnabled、.AlarmDisabled

启用/禁用离散报警

.AlarmDscEnabled 与 **.AlarmDscDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的离散报警。与这两个点域关联的离散报警状态互为反向。对于 **.AlarmDscEnabled**，值为 1 时启用标记或报警组的离散报警。**.AlarmDscDisabled** 的值为 1 时禁用标记或报警组的离散报警。

其中任一个点域启用报警组的离散报警时，都会启用属于该组的所有标记的离散报警。其中任一个点域禁用离散报警时，会忽略所有的离散报警。这些离散报警不存储在报警内存中，也不写入磁盘。

.AlarmDscEnabled 点域

指出标记是否可以生成离散报警。

类别

报警

用法

```
TagName.AlarmDscEnabled
```

参数

TagName
任何离散、间接离散标记，或报警组。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

.AlarmDscEnabled 设置为 0 时，会忽略所有离散条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmDscDisabled 点域正好相反。

数据类型

离散型（可读写）

有效值

0 = 禁用报警

1= 启用报警（缺省）

示例

下例禁用 Tag1 标记的离散报警：

```
Tag1.AlarmDscEnabled=0;
```

另请参阅

.AlarmDscDisabled

.AlarmDscDisabled 点域

指出标记是否可以生成离散报警。

类别

报警

用法

```
TagName.AlarmDscDisabled
```

参数

TagName
任何离散、间接离散标记，或报警组。

附注

.AlarmDscDisabled 设置为 1 时，会忽略所有离散条件的事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmDscEnabled 点域正好相反。

数据类型

离散型（可读写）

有效值

1= 禁用报警

0 = 启用报警（缺省）

示例

下例启用 Tag2 标记的离散报警：

```
Tag2.AlarmDscDisabled=0;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

启用/禁用副偏差报警

.AlarmMinDevEnabled 与 **.AlarmMinDevDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的副偏差报警。与这两个点域关联的副偏差报警状态互为反向。对于 **.AlarmMinDevEnabled**，值为 1 时启用标记或报警组的副偏差报警。**.AlarmMinDevDisabled** 的值为 1 时禁用标记或报警组的副偏差报警。

其中任一个点域启用报警组的副偏差报警时，都会启用属于该组的所有标记的副偏差报警。其中任一个点域禁用副偏差报警时，会忽略所有的副偏差报警。这些副偏差报警不存储在报警内存中，也不写入磁盘。

.AlarmMinDevEnabled 点域

启用或禁用副偏差事件与报警。

类别

报警

用法

```
TagName.AlarmMinDevEnabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmMinDevEnabled 设置为 0 时，会忽略所有的副偏差事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

数据类型

离散型（可读写）

有效值

0 = 禁用报警

1= 启用报警（缺省）

示例

下例禁用 Tag1 标记的副偏差报警：

```
Tag1.AlarmMinDevEnabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmMinDevDisabled

.AlarmMinDevDisabled 点域

启用或禁用副偏差事件与报警。

类别

报警

用法

```
TagName.AlarmMinDevDisabled
```

参数

TagName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmMinDevDisabled 设置为 1 时，会忽略所有的副偏差事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 **.AlarmMinDevEnabled** 点域正好相反。

数据类型

离散型（可读写）

有效值

1= 禁用报警

0 = 启用报警（缺省）

示例

下例启用 Tag2 标记的副偏差报警：

```
Tag2.AlarmMinDevDisabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmMinDevEnabled

启用/禁用主偏差报警

.AlarmMajDevEnabled 与 **.AlarmMajDevDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的主偏差报警。与这两个点域关联的主偏差报警状态互为反向。对于 **.AlarmMajDevEnabled**，值为 1 时启用标记或报警组的主偏差报警。**.AlarmMajDevDisabled** 的值为 1 时禁用标记或报警组的主偏差报警。

其中任一个点域启用报警组的主偏差报警时，都会启用属于该组的所有标记的主偏差报警。其中任一个点域禁用主偏差报警时，会忽略所有的主偏差报警。这些主偏差报警不存储在报警内存中，也不写入磁盘。

.AlarmMajDevEnabled 点域

启用或禁用主偏差事件与报警。

类别

报警

用法

```
TagName.AlarmMajDevEnabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmMajDevEnabled 设置为 0 时，会忽略所有的主偏差事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmMajDevDisabled 点域正好相反。

数据类型

离散型（可读写）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

有效值

0 = 禁用报警

1= 启用报警（缺省）

示例

下例禁用 Tag1 标记的主偏差报警：

```
Tag1.AlarmMajDevEnabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmMajDevDisabled

.AlarmMajDevDisabled 点域

启用或禁用主偏差事件与报警。

类别

报警

用法

```
TagName.AlarmMajDevDisabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmMajDevDisabled 设置为 1 时，会忽略所有的主偏差事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmMajDevEnabled 点域正好相反。

数据类型

离散型（可读写）

有效值

1= 禁用报警

0 = 启用报警（缺省）

示例

下例启用 Tag2 标记的主偏差报警：

```
Tag2.AlarmMajDevDisabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmMajDevEnabled

启用/禁用变化率报警

.AlarmROCEnabled 与 **.AlarmROCDisabled** 点域可以根据标记或报警组各自的值来启用或禁用它们的变化率报警。与这两个点域关联的变化率报警状态互为反向。对于 **.AlarmROCEnabled**，值为 1 时启用标记或报警组的变化率报警。**.AlarmROCDisabled** 的值为 1 时禁用标记或报警组的变化率报警。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

其中的任一个点域启用报警组的变化率报警时，都会启用属于该组的所有标记的变化率报警。其中任一个点域禁用变化率报警时，会忽略所有的变化率报警。这些变化率报警不存储在报警内存中，也不写入磁盘。

.AlarmROCEnabled 点域

启用或禁用变化率事件与报警。

类别

报警

用法

```
TagName.AlarmROCEnabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

.AlarmROCEnabled 设置为 0 时，会忽略所有的变化率条件事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmROCDisabled 点域正好相反。

数据类型

离散型（可读写）

有效值

0 = 禁用报警

1 = 启用报警（缺省）

示例

下例禁用 Tag1 标记的变化率报警：

```
Tag1.AlarmROCEnabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmROCDisabled

.AlarmROCDisabled 点域

禁用或启用变化率事件与报警。

类别

报警

用法

```
TagName.AlarmROCDisabled
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

.AlarmROCDisabled 设置为 1 时，会忽略所有的变化率事件与报警。它们不存储在报警内存中，也不写入磁盘。因此应尽可能重新启用事件/报警，以免丢失数据，这点非常重要。

这与 .AlarmROCEnabled 点域正好相反。

数据类型

离散型（可读写）

有效值

1= 禁用报警

0= 启用报警（缺省）

示例

下例启用 Tag2 标记的变化率报警：

```
Tag2.AlarmROCDisabled=0;
```

另请参阅

.AlarmDisabled、.AlarmEnabled、.AlarmROCEnabled

更改标记的报警限

使用以下点域在应用程序运行期间更改标记的报警限。您可以更改 LoLo、Low、High 及 HiHi 报警的极限，更改主、副偏差的百分比与目标，以及变化率偏差。

- .LoLoLimit 点域
- .LoLimit 点域
- .HiLimit 点域
- .HiHiLimit 点域
- .MinorDevPct 点域
- .MajorDevPct 点域
- .DevTarget 点域
- .ROCPct 点域

.LoLoLimit 点域

更改标记的 LoLo 报警限。

类别

报警

用法

```
TagName.LoLoLimit
```

参数

TagName
任何整型、实型或间接模拟标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

数据类型

模拟（可读写）

有效值

必须在为指定的标记配置的值范围内。

示例

此语句按 10 递减 MyTag1 标记的 LoLo 报警限：

```
MyTag1.LoLoLimit=MyTag1.LoLoLimit - 10;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.LoLoStatus、.LoLoSet、.AlarmDisabled、.AlarmEnabled、.AlarmLoLoDisabled、.AlarmLoLoEnabled、.AlarmLoLoInhibitor

.LoLimit 点域

更改标记的 Low 报警限。

类别

报警

用法

```
Tagname.LoLimit
```

参数

Tagname
任何整型、实型或间接模拟标记。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

数据类型

模拟（可读写）

有效值

必须在为指定的标记配置的值范围内。

示例

此语句按 10 递减 MyTag 标记的 Low 报警限：

```
MyTag.LoLimit=MyTag.LoLimit - 10;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.LoStatus、.LoSet、.AlarmDisabled、.AlarmEnabled、.AlarmLoDisabled、.AlarmLoEnabled、.AlarmLoInhibitor

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

.HiLimit 点域

更改标记的 High 报警限。

类别

报警

用法

```
TagName.HiLimit
```

参数

TagName
任何整型、实型或间接模拟标记。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

数据类型

模拟（可读写）

有效值

必须在为指定的标记配置的值范围内。

示例

此语句将 **PumpTemp** 标记的 High 报警限设置为 212：

```
PumpTemp.HiLimit = 212;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.HiHiStatus、.HiHiSet、.AlarmDisabled、.AlarmEnabled、.AlarmHiHiDisabled、.AlarmHiHiEnabled、.AlarmHiHiInhibitor

.HiHiLimit 点域

更改标记的 HiHi 报警限。

类别

报警

用法

```
TagName.HiHiLimit
```

参数

TagName
任何整型、实型或间接模拟标记。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

模拟（可读写）

有效值

必须在为指定的标记配置的值范围内。

示例

以下语句按 5 递增 MyTag 标记的 HiHi 报警限：

```
MyTag.HiHiLimit=MyTag.HiHiLimit + 5;
```

另请参阅

.Alarm、.AlarmValue、.Ack、.HiHiStatus、.HiHiSet、.AlarmDisabled、.AlarmEnabled、.AlarmHiHiDisabled、.AlarmHiHiEnabled、.AlarmHiHiInhibitor

.MinorDevPct 点域

更改标记的副偏差报警限。

类别

报警

用法

```
TagName.MinorDevPct
```

参数

TagName
任何整型、实型或间接模拟标记。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

数据类型

实型（可读写）

有效值

0 到 100

示例

以下语句将 MyTag 标记的副偏差极限属性设置为 25％：

```
MyTag.MinorDevPct=25;
```

另请参阅

.AckDev、.AlarmDev、.AlarmMinDevDisabled、.AlarmMinDevEnabled、.AlarmMinDevInhibitor、.MinorDevSet、.MinorDevStatus

.MajorDevPct 点域

更改标记的主偏差报警限。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

用法

```
TagName.MajorDevPct
```

参数

TagName
任何整型、实型或间接模拟标记。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

数据类型

实型（可读写）

有效值

0 到 100

示例

以下语句将 MyTag 标记的主偏差极限属性设置为 25％：

```
MyTag.MajorDevPct=25;
```

另请参阅

.AckDev、.AlarmDev、.AlarmMajDevDisabled、.AlarmMajDevEnabled、.AlarmMajDevInhibitor、.MajorDevSet、.MajorDevStatus

.DevTarget 点域

更改标记的主、副偏差报警的目标。

类别

报警

用法

```
TagName.DevTarget
```

参数

TagName
任何整型、实型或间接模拟标记。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

数据类型

实型（可读写）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

有效值

必须在给标记指定的值范围内。

示例

以下语句将 MyTag 标记的偏差目标设置为 500：

```
MyTag.DevTarget=500;
```

另请参阅

.AckDev、.AlarmDev、.AlarmMajDevDisabled、.AlarmMajDevEnabled、.AlarmMajDevInhibitor、.MajorDevSet、.MajorDevStatus

.ROCPct 点域

更改标记的变化率报警限。

类别

报警

用法

```
TagName.ROCPct
```

参数

TagName
任何整型、实型或间接模拟标记。

附注

此点域直接对应于“标记名字典”报警部分所配置的相同字段。

数据类型

整型（读写）

有效值

0 到 100

示例

以下语句将 MyTag 标记的变化率报警限设置为 25%：

```
MyTag.ROCPct=25;
```

另请参阅

.ROCStatus、.ROCSet

更改标记的报警死区

使用以下点域在应用程序运行期间更改标记的报警死区范围：

- .AlarmValDeadband 点域
- .AlarmDevDeadband 点域

.AlarmValDeadband 点域

在 InTouch 应用程序运行期间更改标记的死区值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

用法

```
TagName.AlarmValDeadband
```

参数

TagName
任何整型、实型或间接模拟标记。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

数据类型

模拟（可读写）

有效值

必须在给标记指定的值范围内。

示例

以下语句将 Tag1 标记的死区更改为 25：

```
Tag1.AlarmValDeadband=25;
```

另请参阅

.AlarmDevDeadband

.AlarmDevDeadband 点域

更改主、副偏差报警的偏差死区百分比。

类别

报警

用法

```
TagName.AlarmDevDeadband
```

参数

TagName
任何整型、实型或间接模拟标记。

附注

如果希望在有意或无意关闭 WindowViewer 之后，仍可以继续使用此点域的运行时值，请在“标记名字典”中选择保留参数选项。

数据类型

整型（读写）

有效值

0 到 100

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

以下语句将偏差死区百分比更改为 25%：

```
tag.AlarmDevDeadband=25;
```

另请参阅

.AlarmValDeadband、.AlarmDev

更改与标记关联的报警注释

.AlarmComment 点域返回与标记或报警组的报警关联的注释文本字符串。

.AlarmComment 点域

返回与标记或报警组的报警关联的注释文本字符串。缺省情况下，在新应用程序中它是一个空字符串。

将用户自定义信息关联到报警实例

您可以给报警记录附加三个项目：两个数字和一个字符串。使用以下点域将信息添加到报警记录中。

- .AlarmUserDefNumX 点域
- .AlarmUserDefStr 点域

.AlarmUserDefNumX 点域

要简化用户值的设置，可以在报警组及特定的标记上设置这些点域。例如，InBatch 可以在 $System 报警组的 .AlarmUserDefNum1 中设置批号，从而让所有的报警都附加上该批号。

.AlarmUserDefNum1 与 .AlarmUserDefNum2 点域分别对应于 Alarm Viewer 控件中的 User1 与 User2 列。

如果在某个报警组上设置 .AlarmUserDefNum1，则它会应用于该组及其任何子组中的所有报警。您也可以在标记上专门设置 .AlarmUserDefNum1 的值。在这种情况下，它只应用于该标记，并且会改写标记的报警组中 .AlarmUserDefNum1 的任何设置。

类别

报警

用法

```
TagName.AlarmUserDefNum1
TagName.AlarmUserDefNum2
```

参数

TagName
任何离散、整型、实型、间接离散、间接模拟标记，或报警组。

附注

这个用户自定义的点域可用于各种标记，尤其是离散标记、模拟标记以及报警组（无论是否给它们定义了报警）。针对任何单独的标记、组或父组，可以不设置这些项目、设置所有这些项目，或者只设置其中的一部分。

此点域的值将附加到报警上，但“仅限于”已设置值（例如通过脚本或 POKE）的情况。

数据类型

模拟（可读写）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

有效值

任何实数值或不设置（缺省）

示例

以下示例使用常数值。不过，您可以使用 InTouch QuickScript 将另一个标记的值复制到这些用户自定义点域中的任何一个。您也可以使用 PtAcc 来设置或检测它们，或者将 InTouch 用作“I/O 服务器”来获取或设置这些值。

```
$System.AlarmUserDefNum1 = 4;
GroupA.AlarmUserDefNum1 = 27649;
```

从概念上讲，将报警通知发送给“分布式报警系统”时，使用最低级别的设置。这就是说，如果标记的 .AlarmUserDefNum1 设置为某个值，则应该使用该设置来填写报警记录。不过，如果标记没有这样的值，则 WindowViewer 会检查该标记的报警组是否有值，如此逐层升级，直至到达 $System 根组。如果在任何级别都没有发现任何设置，则报警记录中的输入项将保持空白（如果是数字，则为零；如果是字符串，则为空字符串）。

注意：这种层次结构化的搜索是根据每个项目分别进行处理的。因此，如果标记已设置 .AlarmUserDefNum2，而未设置 AlarmUserDefNum1，但其父组已设置 .AlarmUserDefNum1，则标记将从它的父组继承 .AlarmUserDefNum1 的设置。

另请参阅

.AlarmUserDefStr

.AlarmUserDefStr 点域

.AlarmUserDefStr 点域会附加到报警数据库中由 Alarm DB Logger 记录的每个报警的信息中。.AlarmUserDefStr 点域对应于数据库字段 User3。您可以在 SELECT 语句中使用“用户自定义”列来选择用于数据库操作的特定报警集合。例如，如果将 $System.AlarmUserDefStr 设置为“批次字符串”，并且它每次都会随着“批次”的更改而变化，则包含数据库字段 User3 的选择可用于选择特定批次的报警。

类别

报警

用法

```
Tagname.AlarmUserDefStr
```

参数

Tagname
任何离散、整型、实型、间接离散、间接模拟标记，或报警组。

附注

这个用户自定义的点域可用于各种标记，尤其是离散标记、模拟标记以及报警组（无论是否给它们定义了报警）。针对任何单独的标记、组或父组，可以不设置这些项目、设置所有这些项目，或者只设置其中的一部分。

此点域的值附加到报警上，但“仅”限于值已设置（例如通过脚本或 POKE）的情况。

数据类型

消息（可读写）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

有效值

NULL 与任何有效的字符串

示例

本例使用常数值。不过，您可以使用 InTouch QuickScript 将另一个标记的值复制到这些用户自定义点域中的任何一个。您也可以使用 PtAcc 来设置或检测它们，或者将 InTouch 用作“I/O 服务器”来获取或设置这些值。

```
Tag04.AlarmUserDefStr = "Joe";
```

从概念上讲，将报警通知发送给“分布式报警系统”时，使用最低级别的设置。这就是说，如果将标记的 .AlarmUserDefStr 设置为某个值，则应该使用该设置来填写报警记录。不过，如果标记没有这样的值，则 WindowViewer 将检查该标记的报警组是否有值，如此逐层升级，直至到达 $System 根组。如果在任何级别都没有发现任何设置，则报警记录中的输入项将保持空白（如果是数字，则为零；如果是字符串，则为空字符串）。

此外请注意，这种层次结构化的搜索是根据每个项目分别进行处理的。因此，如果某个标记已设置为 .AlarmUserDefNum1，而未设置为 .AlarmUserDefStr，但其父组已设置 .AlarmUserDefStr，则该设置会用在报警记录中。

另请参阅

.AlarmUserDefNumX

确定标记或报警组的约束标记

使用以下点域确定各种报警类型的约束标记。

- .AlarmDscInhibitor 点域
- .AlarmLoLoInhibitor 点域
- .AlarmLoInhibitor 点域
- .AlarmHiInhibitor 点域
- .AlarmHiHiInhibitor 点域
- .AlarmMinDevInhibitor 点域
- .AlarmMajDevInhibitor 点域
- .AlarmROCInhibitor 点域

.AlarmDscInhibitor 点域

返回指定给离散报警的约束标记的名称。

类别

报警

用法

```
TagName.AlarmDscInhibitor
```

参数

TagName
任何离散标记或报警组。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

在 WindowMaker 中配置。不能在运行时更改。

数据类型

消息（只读）

示例

.AlarmDSCInhibitor 的使用方式如下：将 .Name 设置为等于 .AlarmDscInhibitor 标记值的某个间接标记，然后操纵该间接标记的值。

以下语句返回离散报警的报警约束标记的名称（假设 SomeIndirectTag 是一个模拟间接标记）：

```
SomeIndirectTag.Name = AlarmedTag.AlarmDscInhibitor;
```

报警标记的约束状态可以通过设置间接标记的值来控制，具体如下：

```
SomeIndirectTag = 1;
```

打开约束。禁用 AlarmedTag 的离散报警

```
SomeIndirectTag = 0;
```

关闭约束
可以为 AlarmedTag 生成离散报警

.AlarmLoLoInhibitor 点域

返回指定给 LoLo 报警的约束标记的名称。

类别

报警

用法

```
TagName.AlarmLoLoInhibitor
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组标记。

附注

在 WindowMaker 中配置。不能在运行时更改。

数据类型

消息（只读）

示例

.AlarmLoLoInhibitor 点域的使用方式如下：将 .Name 设置为等于 .AlarmLoLoInhibitor 标记值的间接标记，然后操纵该间接标记的值。

以下语句返回 LoLo 报警限的报警约束标记的名称（假设 SomeIndirectTag 是一个模拟间接标记）：

```
SomeIndirectTag.Name = AlarmedTag.AlarmLoLoInhibitor;
```

报警标记的约束状态可以通过设置间接标记的值来控制，具体如下：

```
SomeIndirectTag = 1;
```

打开约束
禁用 AlarmedTag 的 LoLo 报警

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
SomeIndirectTag = 0;
```

关闭约束
可以为 AlarmedTag 生成 LoLo 报警

另请参阅

.AlarmHiInhibitor、.AlarmHiHiInhibitor、.AlarmLoInhibitor

.AlarmLoInhibitor 点域

返回指定给 Low 报警的约束标记的名称。

类别

报警

用法

```
TagName.AlarmLoInhibitor
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组标记。

附注

在 WindowMaker 中配置。不能在运行时更改。

数据类型

消息（只读）

示例

.AlarmLoInhibitor 点域的使用方式如下：将 .Name 设置为等于 .AlarmLoInhibitor 标记值的间接标记，然后操纵该间接标记的值。

以下语句返回 Low 报警限的报警约束标记的名称（假设 SomeIndirectTag 是一个模拟间接标记）：

```
SomeIndirectTag.Name = AlarmedTag.AlarmLoInhibitor;
```

报警标记的约束状态可以通过设置间接标记的值来控制，具体如下：

```
SomeIndirectTag = 1;
```

打开约束
禁用 AlarmedTag 的 Low 报警

```
SomeIndirectTag = 0;
```

关闭约束
可以为 AlarmedTag 生成 Low 报警

另请参阅

.AlarmHiInhibitor、.AlarmHiHiInhibitor、.AlarmLoLoInhibitor

.AlarmHiInhibitor 点域

返回指定给 High 报警的约束标记的名称。

类别

报警

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用法

```
TagName.AlarmHiInhibitor
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组标记。

附注

在 WindowMaker 中配置。不能在运行时更改。

数据类型

消息（只读）

示例

.AlarmHiInhibitor 点域的使用方式如下：将 .Name 设置为等于 .AlarmHiInhibitor 标记值的间接标记，然后操纵该间接标记的值。

以下语句返回 High 报警限的报警约束标记的名称（假设 SomeIndirectTag 是一个模拟间接标记）：

```
SomeIndirectTag.Name = AlarmedTag.AlarmHiInhibitor;
```

报警标记的约束状态可以通过设置间接标记的值来控制，具体如下：

```
SomeIndirectTag = 1;
```

打开约束
禁用 AlarmedTag 的 High 报警

```
SomeIndirectTag = 0;
```

关闭约束
可以为 AlarmedTag 生成 High 报警

另请参阅

.AlarmHiHiInhibitor、.AlarmLoInhibitor、.AlarmLoLoInhibitor

.AlarmHiHiInhibitor 点域

返回指定给 HiHi 报警条件的约束标记的名称。

类别

报警

用法

```
TagName.AlarmHiHiInhibitor
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组标记。

附注

在 WindowMaker 中配置。不能在运行时更改。

数据类型

消息（只读）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

.AlarmHiHiInhibitor 点域的使用方式如下：将 .Name 设置为等于 .AlarmHiHiInhibitor 标记值的间接标记，然后操纵该间接标记的值。以下语句返回 HiHi 报警限的报警约束标记的名称（假设 SomeIndirectTag 是一个模拟间接标记）：

```
SomeIndirectTag.Name = AlarmedTag.AlarmHiHiInhibitor;
```

报警标记的约束状态可以通过设置间接标记的值来控制，具体如下：

```
SomeIndirectTag = 1;
```

打开约束
禁用 AlarmedTag 的 HiHi 报警

```
SomeIndirectTag = 0;
```

关闭约束
可以为 AlarmedTag 生成 HiHi 报警

另请参阅

.AlarmHiInhibitor、.AlarmLoInhibitor、.AlarmLoLoInhibitor

.AlarmMinDevInhibitor 点域

返回与副偏差报警条件关联的报警约束标记的名称。

类别

报警

用法

```
TagName.AlarmMinDevInhibitor
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组标记。

附注

在 WindowMaker 中配置。不能在运行时更改。

数据类型

消息（只读）

示例

.AlarmMinDevInhibitor 点域的使用方式如下：将 .Name 设置为等于 .AlarmMinDevInhibitor 标记值的间接标记，然后操纵该间接标记的值。以下语句返回副偏差报警限的报警约束标记的名称（假设 SomeIndirectTag 是一个模拟间接标记）：

```
SomeIndirectTag.Name = AlarmedTag.AlarmMinDevInhibitor;
```

报警标记的约束状态可以通过设置间接标记的值来控制，具体如下：

```
SomeIndirectTag = 1;
```

打开约束
禁用 AlarmedTag 的副偏差报警

```
SomeIndirectTag = 0;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

关闭约束
可以为 AlarmedTag 生成副偏差报警

另请参阅

.AlarmMajDevInhibitor

.AlarmMajDevInhibitor 点域

返回与主偏差报警条件关联的报警约束标记的名称。

类别

报警

用法

```
TagName.AlarmMajDevInhibitor
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组标记。

数据类型

消息（只读）

示例

.AlarmMajDevInhibitor 点域的使用方式如下：将 .Name 设置为等于 .AlarmMajDevInhibitor 标记值的间接标记，然后操纵该间接标记的值。以下语句返回主偏差报警限的报警约束标记的名称（假设 SomeIndirectTag 是一个模拟间接标记）：

```
SomeIndirectTag.Name = AlarmedTag.AlarmMajDevInhibitor;
```

报警标记的约束状态可以通过设置间接标记的值来控制，具体如下：

```
SomeIndirectTag = 1;
```

打开约束
禁用 AlarmedTag 的主偏差报警

```
SomeIndirectTag = 0;
```

关闭约束
可以为 AlarmedTag 生成主偏差报警

另请参阅

.AlarmMinDevInhibitor

.AlarmROCInhibitor 点域

返回与变化率报警条件关联的报警约束标记的名称。

类别

报警

用法

```
TagName.AlarmROCInhibitor
```

参数

TagName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

任何整型、实型、间接模拟标记，或报警组标记。

数据类型

消息（只读）

示例

.AlarmROCInhibitor 点域的使用方式如下：将 .Name 设置为等于 .AlarmROCInhibitor 标记值的间接标记，然后操纵该间接标记的值。以下语句返回变化率报警限的报警约束标记的名称（假设 SomeIndirectTag 是一个模拟间接标记）：

```
SomeIndirectTag.Name = AlarmedTag.AlarmROCInhibitor;
```

报警标记的约束状态可以通过设置间接标记的值来控制，具体如下：

```
SomeIndirectTag = 1;
```

打开约束
禁用 AlarmedTag 的变化率报警

```
SomeIndirectTag = 0;
```

关闭约束
可以为 AlarmedTag 生成变化率报警

统计活动的或未确认的报警数

使用以下点域找出应用程序运行期间活动的或未确认的报警的数量。

点域	描述
.AlarmTotalCount 点域	统计与标记或报警组关联的报警数。
.AlarmUnAckCount 点域	统计与标记或报警组关联的未确认报警数。
.AlarmValueCount 点域	统计与标记关联的值报警数。
.AlarmValueUnAckCount 点域	统计与标记关联的未确认的值报警数。
.AlarmDscCount 点域	统计离散报警数。
.AlarmDscUnAckCount 点域	统计未确认的离散报警数。
.AlarmDevCount 点域	统计偏差报警数。
.AlarmDevUnAckCount 点域	统计未确认的偏差报警数。
.AlarmROCCount 点域	统计变化率报警数。
.AlarmROCUnAckCount 点域	统计未确认的变化率报警数。

.AlarmTotalCount 点域

统计指定的标记或报警组的活动报警总数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

报警

用法

```
TagName.AlarmTotalCount
```

参数

TagName
任何类型的标记或报警组。

附注

该计数包括值、偏差、变化率及离散报警。它包括已确认与未确认的报警数。

数据类型

整型（只读）

有效值

0 或任何正整数

示例

Tag1 是给报警配置的模拟标记。ATC 也是一个模拟标记，它获取 Tag1 中存在的所有活动报警（同时包含“未确认”与“确认”的）的总数。

```
ATC = Tag1.AlarmTotalCount;
```

另请参阅

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDSCCount、.AlarmDSCUnAckCount、.AlarmValueCount、.AlarmUnAckCount、.AlarmValueUnAckCount、.AlarmROCCount、.AlarmROCUnACkCount

.AlarmUnAckCount 点域

统计指定的标记或报警组的未确认报警总数。

类别

报警

用法

```
TagName.AlarmUnAckCount
```

参数

TagName
任何类型的标记或报警组。

附注

该计数包括未确认的值、偏差、变化率及离散报警。

数据类型

整型（只读）

有效值

0 或任何正整数

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

Tag1 是给报警配置的模拟或离散标记。AUC 是一个模拟标记，它获取 Tag1 中存在的未确认报警总数。

```
AUC = Tag1.AlarmUnAckCount;
```

另请参阅

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmValueCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCCount、.AlarmROCUnACkCount

.AlarmValueCount 点域

统计指定的标记或报警组的活动值报警总数。

类别

报警

用法

```
TagName.AlarmValueCount
```

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

附注

这包括 HiHi、High、Low 及 LoLo 报警的计数。它包括已确认与未确认的报警数。对于非扩展的摘要报警标记，此计数不超过 1。不过，此计数可能会因报警组的不同而不同。

数据类型

整型（只读）

有效值

0 或任何正整数

示例

Tag1 是给值报警配置的模拟标记。AVC 也是一个模拟标记，它获取 Tag1 中存在的所有报警值的总数。

```
AVC = Tag1.AlarmValueCount;
```

另请参阅

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmROCCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmValueUnAckCount 点域

统计指定的标记或报警组的未确认值报警总数。这包括 HiHi、High、Low 及 LoLo 报警的计数。

类别

报警

用法

```
TagName.AlarmValueUnAckCount
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

TagName
任何整型、实型、间接模拟标记，或报警组。

数据类型

整型（只读）

有效值

0 或任何正整数

示例

Tag1 是给值报警配置的模拟标记。AVUC 也是一个模拟标记，它获取 Tag1 中存在的所有未确认的值报警总数。

```
AVUC = Tag1.AlarmValueUnAckCount;
```

另请参阅

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmROCCount、.AlarmTotalCount、.AlarmValueCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmDscCount 点域

统计指定的标记或报警组的活动离散报警总数。

类别

报警

用法

```
TagName.AlarmDscCount;
```

参数

TagName
任何离散标记、间接离散标记，或报警组。

附注

指定给 **.AlarmDscCount** 点域的计数同时包括已确认和未确认的报警。对于非扩展的摘要报警标记，此计数始终等于 1。不过，此计数可能会因报警组的不同而不同。

数据类型

整型（只读）

有效值

0 或任何正整数

示例

Tag1 是给离散报警配置的离散标记。ADC 是模拟标记，它获取 Tag1 中存在的活动离散报警（同时包括已确认和未确认的）的总数。

```
ADC = Tag1.AlarmDSCCount;
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmValueCount、.AlarmROCUnAckCount、.AlarmTotalCount、.AlarmDscUnAckCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmDscUnAckCount 点域

统计指定的标记或报警组的未确认离散报警总数。

类别

报警

用法

```
TagName.AlarmDscUnAckCount
```

参数

TagName
任何离散标记、间接离散标记，或报警组。

数据类型

整型（只读）

有效值

0 或任何正整数

示例

Tag1 是给离散报警配置的离散标记。ADUC 是模拟标记，它获取 Tag1 中存在的未确认离散报警的总数。

```
ADUC = Tag1.AlarmDscUnAckCount;
```

另请参阅

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmValueCount、.AlarmROCCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmDevCount 点域

统计指定的标记或报警组的活动偏差报警总数。

类别

报警

用法

```
TagName.AlarmDevCount
```

参数

TagName
任何实型、整型、间接模拟标记，或报警组。

附注

这包括主、副偏差报警的计数。它包括已确认与未确认的报警数。对于非扩展的摘要报警标记，此计数始终等于 1。不过，此计数可能会因报警组的不同而不同。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

模拟型（可读写）

有效值

0 或任何正整数

示例

Tag1 是给“偏差”报警配置的模拟标记。ADC 也是一个模拟标记，它获取 Tag1 中存在的活动偏差（同时包含已确认和未确认的）报警的总数。

```
ADC=Tag1.AlarmDevCount;
```

另请参阅

.AlarmDSCCount、.AlarmValueCount、.AlarmROCUnAckCount、.AlarmTotalCount、.AlarmDSCUnAckCount、.AlarmValueUnAckCount、.AlarmDevUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmDevUnAckCount 点域

统计指定的标记或报警组的未确认偏差报警的总数。这包括主、副偏差报警的计数。

类别

报警

用法

```
TagName.AlarmDevUnAckCount
```

参数

TagName
任何实型、整型、间接模拟标记，或报警组。

数据类型

模拟型（可读写）

有效值

0 或任何正整数

示例

Tag1 是给“偏差”报警配置的模拟标记。ADUC 也是一个模拟标记，它获取 Tag1 中存在的未确认偏差报警的总数。

```
ADUC = Tag1.AlarmDevUnAckCount;
```

另请参阅

.AlarmDevCount、.AlarmDSCCount、.AlarmValueCount、.AlarmROCUnAckCount、.AlarmTotalCount、.AlarmDSCUnAckCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmROCCount 点域

统计指定的标记或报警组的活动变化率报警总数。它包括已确认与未确认的报警数。对于非扩展的摘要报警标记，此计数将始终等于 1。不过，此计数可能会因报警组的不同而不同。

类别

报警

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用法

```
TagName.AlarmROCCount
```

参数

TagName
任何实型、整型、间接模拟标记，或报警组。

数据类型

整型（只读）

有效值

0 或任何正整数

示例

Tag1 是给变化率报警配置的模拟标记。ARC 也是一个模拟标记，它获取 Tag1 中存在的活动变化率报警（同时包含已确认和未确认的）的总数。

```
ARC = Tag1.AlarmROCCount;
```

另请参阅

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmValueCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCUnAckCount、.AlarmUnAckCount

.AlarmROCUnAckCount 点域

统计指定的模拟标记或报警组的未确认变化率报警总数。

类别

报警

用法

```
TagName.AlarmROCUnAckCount
```

参数

TagName
任何实型、整型、间接模拟标记，或报警组。

数据类型

整型（只读）

有效值

0 或任何正整数

示例

Tag1 是给变化率报警配置的模拟标记。ARUC 也是一个模拟标记，它获取 Tag1 中存在的未确认变化率报警的总数。

```
ARUC = Tag1.AlarmROCUnAckCount;
```

另请参阅

.AlarmDevCount、.AlarmDevUnAckCount、.AlarmDscCount、.AlarmDscUnAckCount、.AlarmValueCount、.AlarmTotalCount、.AlarmValueUnAckCount、.AlarmROCCount、.AlarmUnAckCount

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Maintain

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 25 迁移和升级应用程序

本节介绍如何迁移和升级不同的 InTouch 应用程序。

从传统应用程序移至新的独立应用程序

在 System Platform 2020 之前，InTouch HMI 用户可创建以下类型的应用程序：

- 独立
- 新型
- 托管的
- 发布的

独立应用程序使用传统符号和控件构建。新型应用程序除了支持使用传统符号，还支持使用工业图形（以前称为 ArchestrA 图形/符号）。托管的应用程序使用 IDE 和 Galaxy 对象构建。独立的应用程序可以发布到软件包中，然后分发到其它节点，从而成为发布的应用程序。

在 InTouch HMI 2020 中，新型应用程序已重新设计为更全面的独立应用程序。相比传统的独立应用程序，新的独立应用程序具有许多改进。

- 分发简便 - 复制应用程序文件夹并粘贴到其它节点。无需导入或导出操作。
- 使用工业图形 - 新的独立应用程序将较早的传统应用程序的易用性与新型工业图形相结合。
- 云就绪 - 现在可以在兼容 HTML5 的浏览器上查看应用程序创建的内部部署节点。
- 轻量 - 应用程序文件为轻量级，可以实现更好的性能和使用。

托管和发布的应用程序的行为没有变化。

迁移和升级旧版应用程序

要支持在较早版本的 InTouch HMI 中创建的应用程序，可以使用两个工作流程以转换到新的独立应用程序。

- 就地迁移旧版新型应用程序：如果节点包含旧版新型应用程序并且节点上的产品版本已升级，则使用“应用程序管理器”迁移应用程序。
- 导入从较早版本的 InTouch HMI 导出的新型应用程序的 .aapkg 文件。

将较早版本的 InTouch 应用程序迁移到当前版本

您可以将旧版 InTouch HMI 开发的应用程序迁移到当前版本。试图使用 WindowMaker 或 WindowViewer 打开旧版应用程序时，会显示“应用程序迁移”对话框：在此处，您可以：

- 选择转换应用程序分辨率。
- 在将旧版应用程序迁移到当前版本的 InTouch HMI 之前，创建一个备份副本。

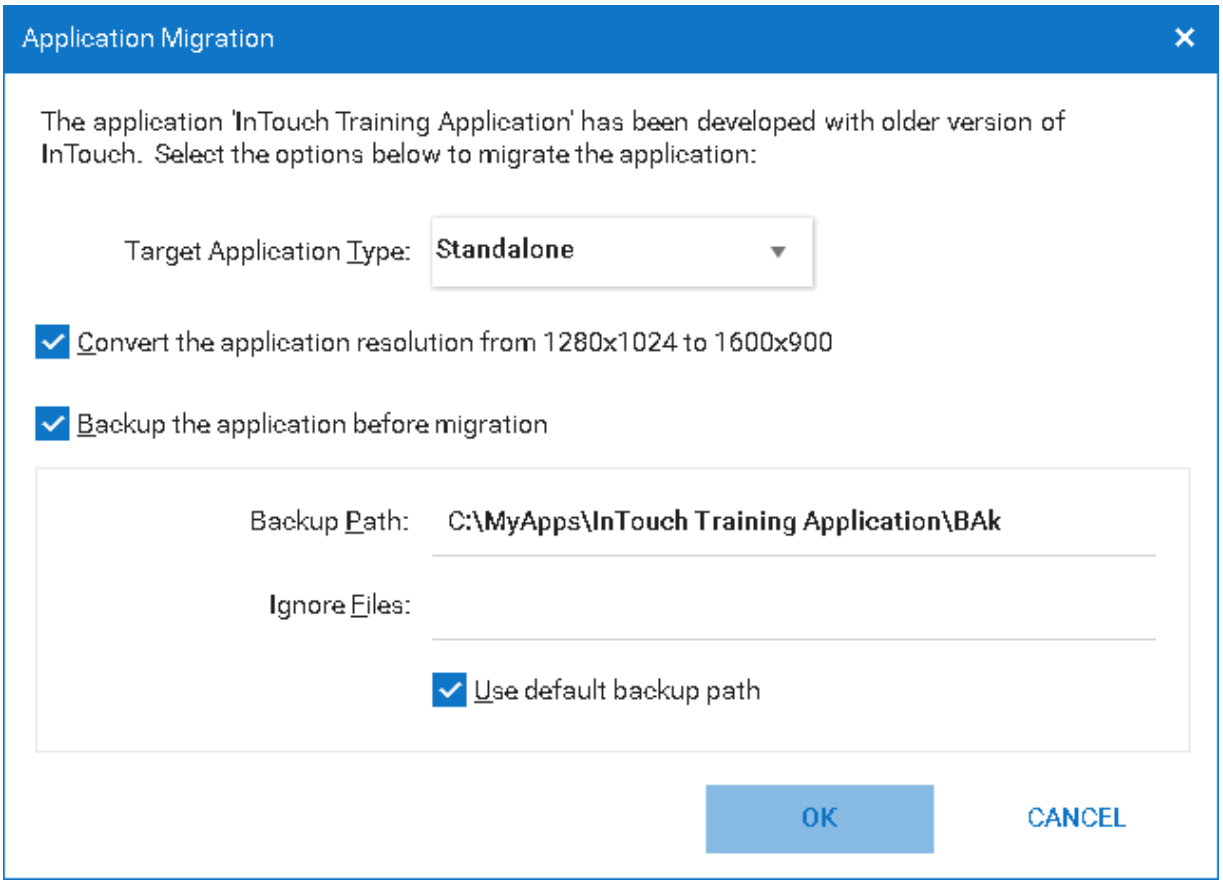
您可以将现有的独立、新型或发布的 InTouch 应用程序迁移到当前 InTouch 版本。您必须指定要用于创建备份副本的文件夹，以及是否要从备份中排除任何文件。

1. 在应用程序列表中双击应用程序。

   此时出现应用程序迁移对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 要将应用程序分辨率从原始分辨率转换为当前分辨率，请选中将应用程序分辨率从 <现有分辨率> 转换为 <新分辨率> 复选框。
3. 要更改缺省备份路径（<应用程序目录>\Bak），请清除迁移前备份应用程序旁边的复选框。然后在备份路径框中，输入要保存备份的文件夹的路径。如果该文件夹不存在，则必须创建它，然后再创建备份。
4. 在忽略文件框中，可以指定要从备份中排除的任何文件。缺省条件下，应用程序目录中的所有文件都会备份。输入要排除的文件名。用分号 (;) 分隔。或者使用标准通配符（'*' 与 '?'），以根据名称中的公共字符来排除一组文件。
5. 配置必要选项后，选择确定。



转换旧的报警显示

在 WindowViewer 中打开使用 InTouch 7.11 之前的版本构建的应用程序时，会出现一个对话框，提示您运行 WindowMaker 以转换该应用程序。如果继续转换，则所有的“标准报警对象”都会转换为包含缺省值的“分布式报警对象”。颜色、字体、表达式及报警查询等设置不会保留下来。

管理应用程序设置

InTouch 应用程序设置（如应用程序路径）存储在 Win.ini 文件中。Win.ini 文件位于以下目录中：

C:\Users\<用户名>\AppData\Local\Wonderware

WindowMaker 作为管理员用户运行，而 WindowViewer 可以作为管理员或标准用户运行。标准用户无法访问管理员用户配置文件的 Win.ini 目录。因此，作为应用程序开发人员，您需要在开发该应用程序时将公

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

共的 Win.ini 属性复制到标准用户的 Win.ini 配置文件中。这可以确保在标准用户启动 WindowViewer 之后，以管理员用户身份设置的所有属性也都可用。每次更改公共的 Win.ini 属性时，都必须复制这些属性。

导入 InTouch 应用程序

您可以使用“应用程序管理器”导入现有新型应用程序，这些应用程序将转换为独立应用程序。独立应用程序可以从一个节点复制到另一个节点，并使用“查找应用程序”选项进行查找，无需将其导入或导出。

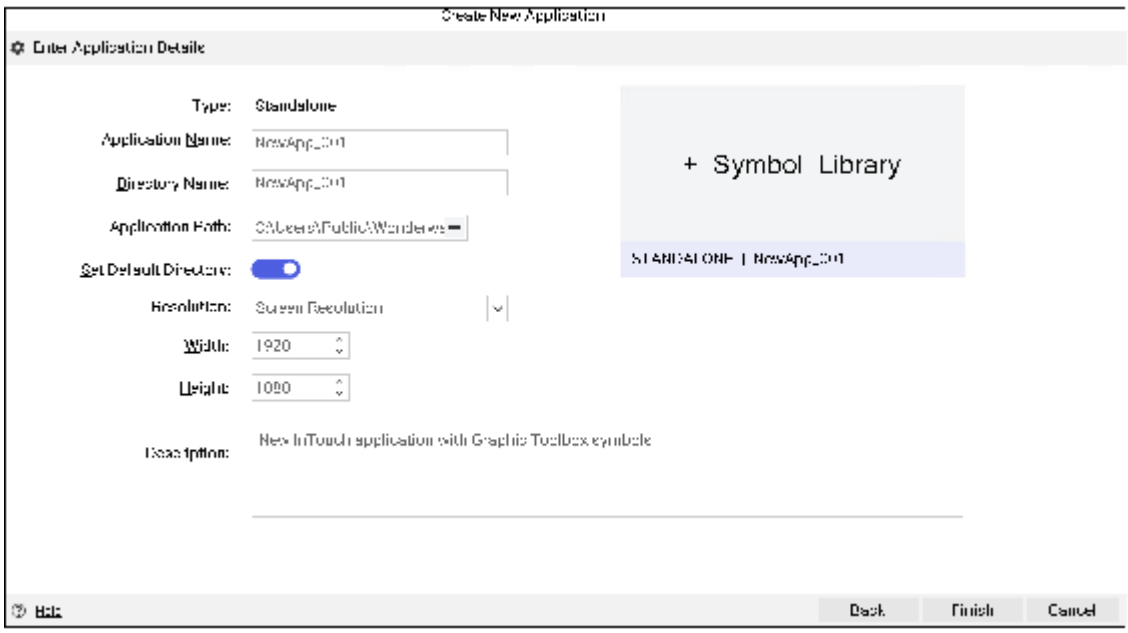
导入现有新型应用程序：

1. 在文件菜单上的导入组中，选择导入。

   此时出现“新建应用程序：选择要导入的应用程序”屏幕。

2. 使用查找应用程序部分搜索要导入的应用程序。搜索要导入的文件夹或文件。
3. 选择应用程序，然后选择下一步。

   此时出现“输入应用程序详细信息”屏幕。



4. 对设置进行任何更改。

   选择完成。

   将创建一个新的应用程序，并且该应用程序将显示在“应用程序管理器”中。

注意：所有新型应用程序都作为独立应用程序导入。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

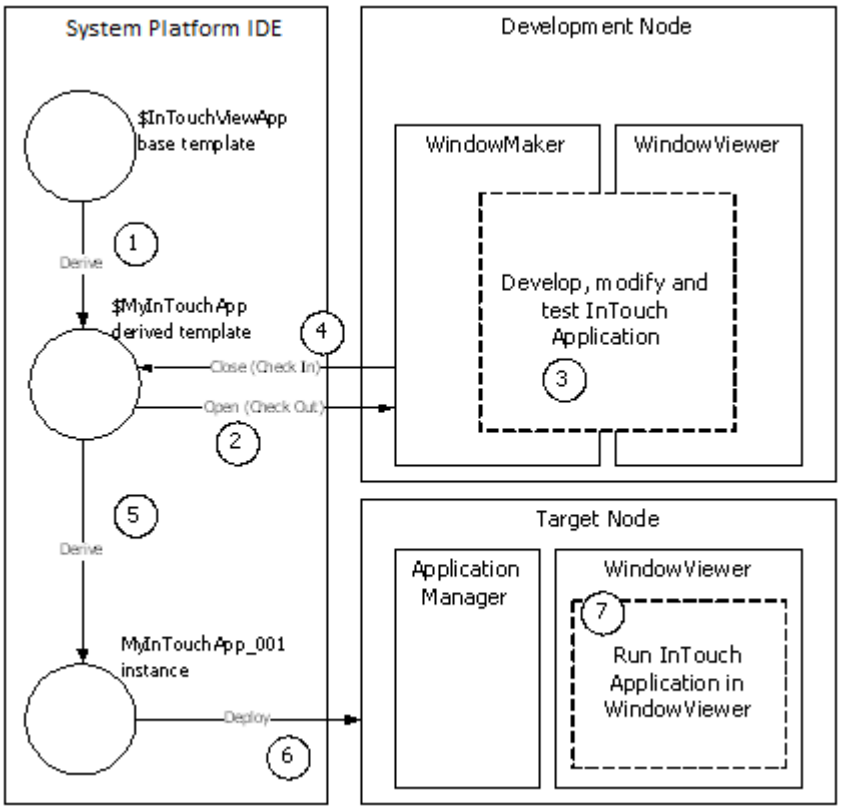
章 26 使用 System Platform IDE 管理 InTouch 应用程序

您可以使用 IDE 管理 InTouch 应用程序。以下操作程序说明了在一般情况下如何完成此操作。如需有关具体信息，请参阅使用 IDE 管理 InTouch 应用程序。

System Platform IDE 中的 InTouch 功能通过两个自动化对象来实现：

- InTouchViewApp 对象代表设计时与运行时的 InTouch 应用程序。
- ViewEngine 对象控制 InTouch 应用程序如何在 Galaxy 中的目标节点上运行。

下图显示如何使用 System Platform IDE 来管理 InTouch 应用程序：



使用 IDE 管理 InTouch 应用程序

1. 在 System Platform IDE 中，创建一个托管的 InTouch 应用程序。
2. 在 WindowMaker 中将其打开。
3. 在 WindowMaker 中配置 InTouch 应用程序。您可以切换到 WindowViewer 来测试应用程序。
4. 保存 InTouch 应用程序并关闭 WindowMaker 与 WindowViewer。
5. 确定将 InTouch 应用程序部署到哪些节点。
6. 将 InTouch 应用程序部署到 Galaxy 中的目标节点。
7. 在目标节点上的 WindowViewer 中运行 InTouch 应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouchViewApp 对象

Application Server 使用一个名称为 InTouchViewApp 对象的特定类型的 Application Server 对象来管理 InTouch 应用程序。

InTouchViewApp 模板在设计时引用托管的特定 InTouch 应用程序，在运行时无法执行。

您必须创建一个 InTouchViewApp 模板的实例。此实例可以部署到目标节点。目标节点是在 WindowViewer 中运行托管的 InTouch 应用程序的节点。

要分发 InTouch 应用程序，您可以创建相同模板的多个实例，并将它们部署到多个节点。

作为可选项，您可以：

- 导出与导入 InTouchViewApp 对象，以便在不同的 Galaxy 之间交换托管的 InTouch 应用程序。
- 按照 .csv 文件的形式导出与导入标记字典数据。
- 在不同类型的 InTouch 应用程序之间导出与导入窗口。
- 发布托管的 InTouch 应用程序。发布的 InTouch 应用程序像独立的 InTouch 应用程序那样运行，但它可以包含工业图形。
- 使用部署的 InTouchViewApp 对象的属性对包含 ArchestrA 属性的 InTouch 标记进行读取和写入。

使用 InTouchViewApp 对象

1. 从 $InTouchViewApp 基本模板中衍生 InTouchViewApp 模板。
2. 通过创建新的 InTouch 应用程序或导入独立的 InTouch 应用程序，将衍生的模板与 InTouch 应用程序关联起来。
3. 在 WindowMaker 中打开应用程序。
4. 在 WindowMaker 中配置应用程序并在 WindowViewer 中测试它。
5. 保存并关闭 WindowMaker。InTouchViewApp 模板已签入。
6. 从 InTouchViewApp 模板中衍生实例。
7. 将这些实例部署到 Galaxy 中所选的目标节点上。
8. 在目标节点上运行“应用程序管理器”，在 WindowViewer 中运行托管的 InTouch 应用程序。

关联 InTouchViewApp 模板与 InTouch 应用程序

创建新的 InTouchViewApp 模板之后，可以按以下方式关联 InTouchViewApp 模板与 InTouch 应用程序：

- 创建新的 InTouch 应用程序。
- 导入独立的 InTouch 应用程序。

InTouchViewApp 模板不包含 InTouch 应用程序数据本身，如标记配置与值，但会简单地引用应用程序。

编辑托管的 InTouch 应用程序

您可以像对待独立的 InTouch 应用程序那样使用 WindowMaker 编辑托管的 InTouch 应用程序；只是需要打开 InTouchViewApp 模板的编辑器，以便在 WindowMaker 中启动关联的 InTouch 应用程序。

更改 InTouch 应用程序之后关闭 WindowMaker 时，InTouchViewApp 对象会自动签入。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

测试托管的 InTouch 应用程序

您可以像对待独立的 InTouch 应用程序那样使用 WindowViewer 测试托管的 InTouch 应用程序。

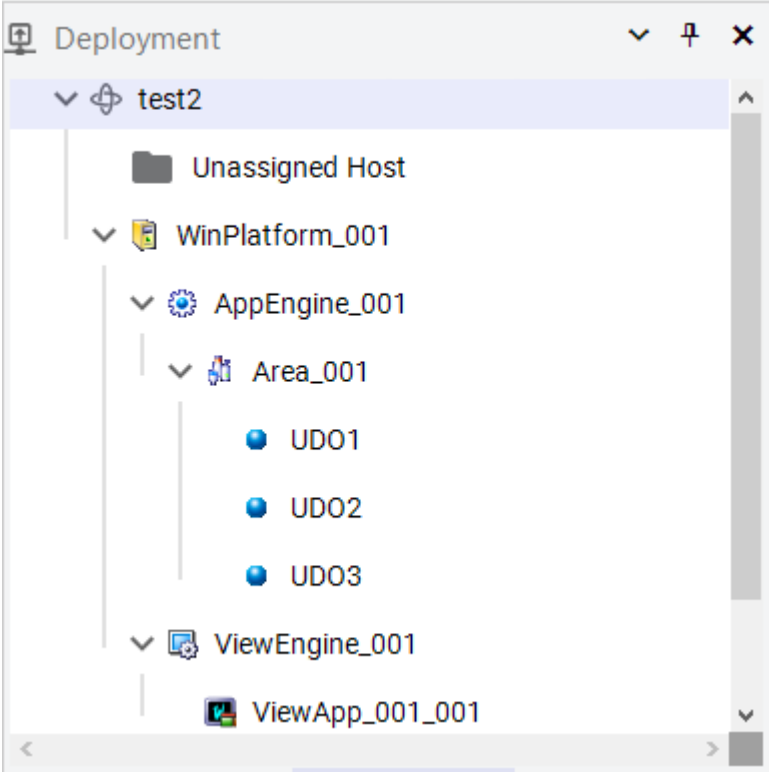
如果从 System Platform IDE 打开 WindowMaker，则可以在 WindowMaker 和 WindowViewer 之间快速切换来测试托管的应用程序。

如果托管的 InTouch 应用程序包含对 Application Server 数据的引用，如 `galaxy:UDA`，则需要将 WinPlatform 对象部署到正在编辑 InTouch 应用程序的节点上。否则数据会显示空值。

部署 InTouchViewApp 对象

衍生 InTouchViewApp 模板的实例之后， 可以将它指定到目标平台上 ViewEngine 对象的下面。

您无法在一个 ViewEngine 下指定多个具有相同父对象的 InTouchViewApp 实例。相反， 您可以创建第二个 ViewEngine 实例来存放具有相同父对象的其它 InTouchViewApp 实例。



部署 InTouchViewApp 对象之后，您可以在目标节点上打开“InTouch 应用程序管理器”。关联的托管 InTouch 应用程序出现在列表中，并在修改日期列中显示最近一次部署的时间标签。

将 InTouchViewApp 实例部署到目标节点时，InTouch 应用程序包含于：

- 开发节点上的文件夹。这包含 InTouchViewApp 模板的源文件。
- 运行 InTouch 应用程序的目标节点上的文件夹。这包含 InTouch 应用程序的一个实例副本。

导出与导入 InTouchViewApp 对象

您可以导出 InTouchViewApp 对象。例如，您可以执行此操作将托管的 InTouch 应用程序同其它 Galaxy 中存放它的 InTouchViewApp 对象配合使用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

导出对象时会创建一个数据包文件 (.aaPKG)，此文件包含该对象、关联的托管 InTouch 应用程序以及该应用程序使用的任何工业图形的有关信息。

导入 InTouchViewApp 对象时，System Platform IDE 也导入托管的 InTouch 应用程序。

InTouchViewApp 对象的属性

您可以使用 InTouchViewApp 对象的 Application Server 属性来访问关联的 InTouch 应用程序标记的运行时数据。对于在 Galaxy 域名空间中直接读取和写入 InTouch 数据，这很有用，并且提供与 InTouchProxy 对象相同的功能。

在本例中，部署的托管的 InTouch 应用程序使用 TankLevel 实型标记来报告贮料罐的填充级别，使用 ValveOut 离散标记来控制阀门的状态。

从 InTouchViewApp 对象实例中读取和写入 InTouch 标记

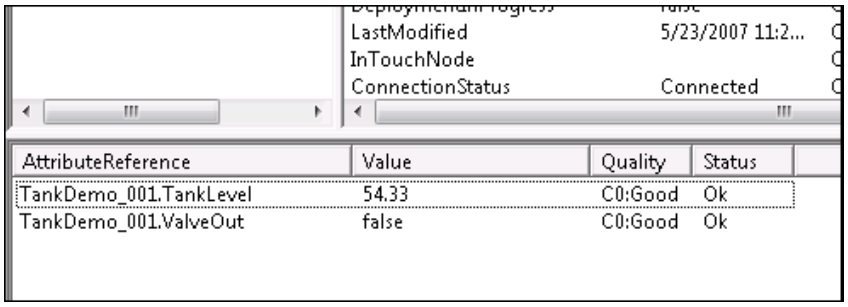
1. 使用鼠标右键单击部署的 InTouchViewApp 对象，然后选择**监视**。

   此时出现 **Object Viewer** 屏幕。

2. 使用鼠标右键单击 Watch（观察）区域，然后选择 **Add Attribute Reference**（添加属性引用）。

   此时出现添加属性引用对话框。

3. 在属性引用框中，输入 InTouchViewApp 对象的名称，后面跟一个点及希望读取或写入的 InTouch 标记的名称。例如，`TankDemo_001.TankLevel`。
4. 选择确定。此时该属性添加到 Watch（观察）区域。
5. 对于希望读取或写入的任何其它 InTouch 标记，重复步骤 2 至 4。
6. 您现在可以查看 InTouch 标记值。



7. 执行以下操作以写入 InTouch 标记值，
   a. 双击它。此时出现修改值对话框。
   b. 输入新值，然后选择确定。此时该值写回到正在运行的 InTouch 应用程序的标记。

InTouchViewApp 对象与其它 AutomationObject 的区别

InTouchViewApp 对象与其它自动化对象不同。您无法执行通常可以对其它自动化对象执行的一些操作。

- 如果试图配置 InTouchViewApp 实例，则此时会出现一条消息，询问是否要打开其父模板。您无法直接配置实例，只能配置父模板。
- 如果尝试一次在一个节点上打开多个 InTouchViewApp 模板进行配置，IDE 会阻止您这样操作。关闭 WindowMaker、WindowViewer 及“应用程序管理器”并重试。此外，您也可以使用 InTouch WindowMaker 在不同的节点上编辑 InTouchViewApp 对象。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 如果在使用 WindowMaker 编辑 InTouch 应用程序时关闭 IDE, 则 WindowMaker 会提示您保存任何更改。然后它会关闭，并签入 InTouchViewApp 模板。
- 如果在使用 WindowViewer 测试 InTouch 应用程序时关闭 IDE，则 WindowViewer 会关闭。

如果希望：

- 更改 InTouchViewApp 与 InTouch 应用程序之间的关联，创建新的 InTouchViewApp 衍生模板。
- 在 InTouch 中使用 ArchestrA 安全性（Galaxy 安全性），将 WinPlatform 实例部署到正在运行部署的托管 InTouch 应用程序的节点上。

ViewEngine 对象

ViewEngine 是一个存放与运行部署的 InTouchViewApp 对象的 Application Server 对象。

要将 InTouchViewApp 实例部署到目标平台，需要先将它指定给 ViewEngine 对象。然后 ViewEngine 对象会指定给目标 WinPlatform 对象。

ViewEngine 可以对 InTouchViewApp 实例执行相同的功能，就像 AppEngine 实例处理“应用程序对象”那样。ViewEngine 可以：

- 在 InTouchViewApp 对象最初部署与启动时，对它们进行设置与初始化，使它们可与以 Galaxy 中的其它对象通讯。
- 对可以监视、报警、及作为历史写入的属性执行诊断。
- 将历史数据写入 Historian。

您可以使用不同的 ViewEngine 对象：

- 将历史数据写入不同的 Historian。
- 以不同的扫描速率与部署的 InTouch 应用程序进行交互。这会设置 InTouch 标记数据可以按什么频率与 Galaxy 域名空间进行交互。

平台可以存放多个 ViewEngine 对象。每个 InTouchViewApp 都必须指定给 ViewEngine。

您无法创建相同 InTouchViewApp 模板的多个实例在相同的 ViewEngine 对象下运行。但是可以在不同的 ViewEngine 对象下运行相同模板的多个实例。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 27 导出与导入 InTouch 组件

通过导入或导出现有应用程序中的部分或所有组件，可以更快地构建 InTouch 应用程序。您可以从现有的应用程序中将标记定义、窗口、脚本、应用程序样式库、工业图形、客户端控件、本地化字符串、HTML5 小组件和脚本函数库导入到新的应用程序。标记定义从“应用程序管理器”导入和导出，其它组件通过 WindowMaker 导入和导出。

导出与导入同托管的 InTouch 应用程序关联的标记数据

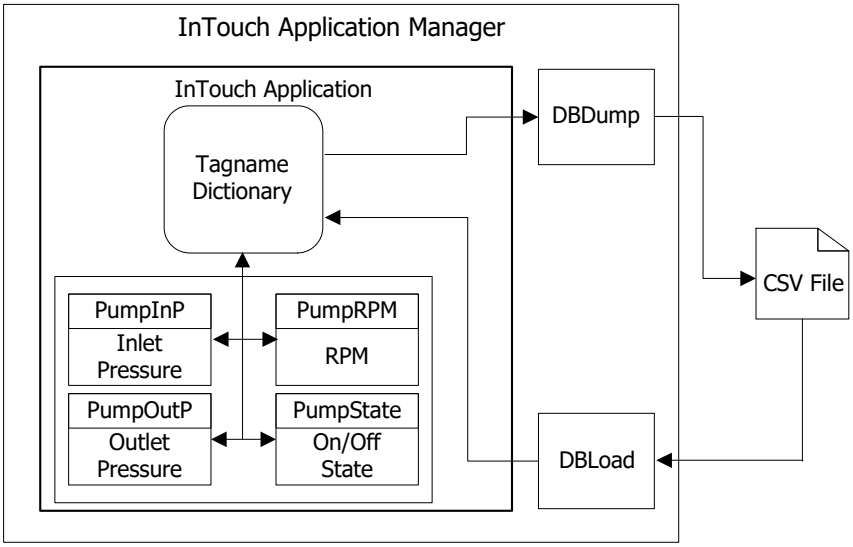
您可以将与托管的 InTouch 应用程序关联的标记数据导出到 .csv 文件。这等同于“InTouch 应用程序管理器”的 DB Dump 功能。

您可以像使用 DB Load 功能那样，将导出的标记数据从 .csv 文件中导回到托管的 InTouch 应用程序中。

从托管的 InTouch 应用程序与独立的 InTouch 应用程序中导出的那些 .csv 文件完全可以互换。

导出标记定义

下图显示在中间导出文件与应用程序的“标记名字典”之间导出与导入标记定义的步骤。



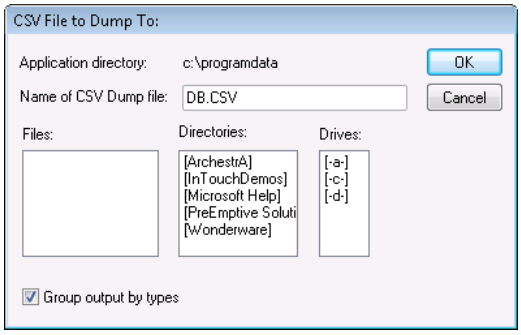
您使用“应用程序管理器”中的 DBDump 实用程序将“标记名字典”的内容导出到“逗号分隔值”（Comma Separated Value，简称 CSV）文件。您可以使用 Microsoft 的“记事本”或 Microsoft Excel 来查看与编辑导出的文件。在进行编辑之后，使用同样是“应用程序管理器”实用程序的 DBLoad 实用程序，将标记定义导入 InTouch 应用程序。

要能够导出标记定义，必须先将应用程序转换为 InTouch HMI 软件的当前版本。

导出标记定义

1. 关闭 WindowMaker 和 WindowViewer。
2. 启动“应用程序管理器”。此时**应用程序管理器对话**框显示一个 InTouch 应用程序列表。
3. 从列表中选择应用程序。
4. 在文件菜单上的数据组中，选择 **DBDump**。此时出现**要转储到的 CSV 文件**：对话框。

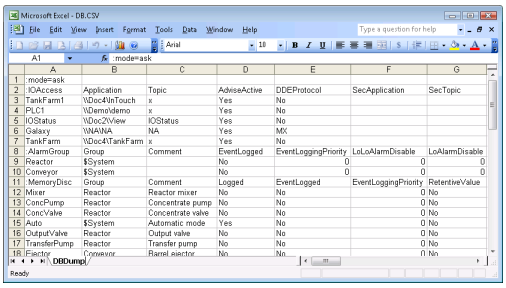
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 在 **CSV 转储文件名**框中，输入带 .csv 文件扩展名的文件名。
6. 选择导出文件中数据组的类型。
   - 选择按类型的组输出复选框，以便在导出文件中按标记类型对数据进行分组。这是缺省值。
   - 清除按类型的组输出以便按标记名的字母顺序将输出内容保存到导出文件。
7. 选择确定，以便将“标记名字典”的内容保存到所选的文件。此时出现一条消息，指出内容已成功保存到文件。

查看导出的标记定义

如果使用 Microsoft Excel 来查看使用 DBDump 实用程序创建的导出文件，则每条数据记录都出现在单独的电子表格单元格中。



	A	B	C	D	E	F	G
1	:mode=ask						
2	IOAccess	Application	Topic	AdviseActive	DDEProtocol	SecApplication	SecTopic
3	TankFarm1	\\Doc4\InTouch	x	Yes	No		
4	PLC1	\\Demo\demo	x	Yes	No		
5	IOStatus	\\Doc2\View	IOStatus	Yes	No		
6	Galaxy	\\NA\NA	NA	Yes	MX		
7	TankFarm	\\Doc4\TankFarm	x	Yes	No		
8	AlarmGroup	Group	Comment	EventLogged	EventLoggingPriority	LoLoAlarmDisable	LoAlarmDisable
9	Reactor	$System		No	0	0	0
10	Conveyor	$System		No	0	0	0
11	MemoryDisc	Group	Comment	Logged	EventLogged	EventLoggingPriority	RetentiveValue
12	Mixer	Reactor	Reactor mixer	No	No	0	No
13	ConcPump	Reactor	Concentrate pump	No	No	0	No
14	ConcValve	Reactor	Concentrate valve	No	No	0	No
15	Auto	$System	Automatic mode	Yes	No	0	No
16	OutputValve	Reactor	Output valve	No	No	0	No
17	TransferPump	Reactor	Transfer pump	No	No	0	No
18	Ejector	Conveyor	Barrel ejector	No	No	0	No

此文件由关键字及其属性以及“标记名字典”中的数据组成，这些数据在关键字属性下方按列顺序进行排列。

请注意示例 Excel 电子表格中的 :MemoryDisc 关键字。此关键字标识从“标记名字典”中导出的内存离散标记。内存离散标记的属性出现在电子表格相同的行、单独的列中。例如，Logged 属性列显示是否记录内存离散标记的数据。

关键字与属性行的正下方是导出的标记及其关联的属性。在 Excel 电子表格示例中，OutputValve 是不记录数据的内存离散标记。

您可以使用支持 .csv 文件格式的任何程序来查看或编辑 DBDump 创建的导出文件。通常使用 Excel, 这是因为它的分栏电子表格格式使得标记数据更便于组织。但是，如果您希望以本机逗号分隔的字符串格式查看或编辑文件的内容，也可以使用 Microsoft“记事本”。

导入标记定义

您可以使用“应用程序管理器”中的 DBLoad 实用程序，将 .csv 标记定义文件导入应用程序的“标记名字典”。您可以导入原来使用 DBDump 实用程序创建的定义文件。您也可以创建自己的导入文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您还可以使用 DBLoad 实用程序来创建 SuperTag 实例。如需详细信息，请参阅创建 SuperTag 实例。

标记名字典导入文件格式

您可以使用支持 .csv 文件格式的任何程序来手工创建 DBLoad 导入文件。如果使用 Excel 创建导入文件，则每一项都放置在单独的电子表格单元格中。这使得它更便于阅读，并且可以减少出错的可能。

如需有关创建导入文件的详细信息，请参阅创建导入文件模板。

DBLoad 导入文件包含一组关键字，它们将“访问名”、报警组以及标记数据在文件中组织起来。

- 所有的关键字前面都有一个冒号 (:)。
- 如果要续行，请在行尾输入一个反斜杠 (\)。
- 要输入注释，请在前面加上一个分号 (;)。

下表列出 DBLoad 导入文件中的关键字。表格中按使用 DBDump 创建文件时指定关键字的顺序列出这些关键字。但是您可以在文件中按任何顺序指定关键字。

关键字	描述
:mode	指定将 DBLoad 文件的内容导入应用程序的“标记名字典”时如何处理重复的标记记录。
:IOAccess	为 InTouch 应用程序定义的访问名。
.AlarmGroup	为 InTouch 应用程序定义的报警组。
:MemoryDisc	内存离散标记。
:IODisc	I/O 离散标记。
:MemoryInt	内存整型标记。
:IOInt	I/O 整型标记。
:MemoryReal	内存实型标记。
:IOReal	I/O 实型标记。
:MemoryMsg	内存消息标记。
:IOMsg	I/O 消息标记。
:GroupVar	“组变量”标记。
:HistoryTrend	“历史趋势”标记。
:TagID	“标记 ID”标记
:IndirectDisc	间接离散标记。
:IndirectAnalog	间接模拟标记。
:IndirectMsg	间接消息标记。

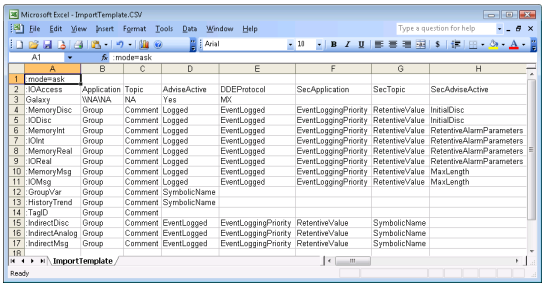
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

每个关键字都包含指定“访问名”、报警组以及标记的属性的一组关联属性。例如，:IOAccess 关键字包含一组属性，用于指定应用程序、主题以及通讯协议，这些都是每个 InTouch“访问名”的属性。

创建导入文件模板

您可以使用任何支持 .csv 文件格式的应用程序来手动创建“标记名字典”导入文件。但是创建整个导入文件可能既费时又容易出错。将现有的 .csv 文件用作模板则既快捷又可靠。

下图显示 DBDump 创建的模板导入文件。图中显示从某个既没有窗口也没有标记的 InTouch 应用程序中创建的文件。产生的文件只包含所需的关键字和没有标记数据的属性。



在创建模板之后，您可以随后在确定标记类型的关键字下方手动添加标记数据。在与标记类型关键字关联的相应属性列中，插入标记的属性。

创建模板导入文件

1. 打开“应用程序管理器”。
2. 创建新的 InTouch 应用程序。

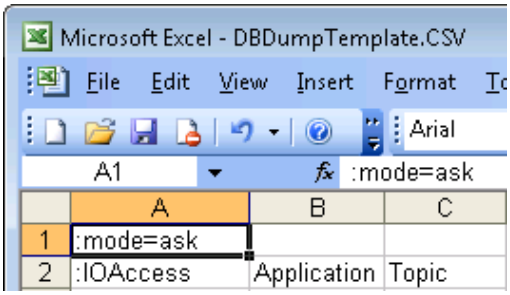
   如需有关创建应用程序的步骤的详细信息，请参阅创建 InTouch 应用程序。
3. 从“应用程序管理器”显示的列表中，选择新的应用程序。
4. 使用 DBDump 实用程序导出应用程序“标记名字典”的内容。

   如需有关导出标记的详细信息，请参阅导出标记定义。
5. 编辑文件以插入要导入的标记数据。

设置字典导入文件的操作模式

您必须指定从导入文件将数据加载到应用程序“标记名字典”时，DBLoad 如何处理重复的标记记录。

如果使用通过 DBDump 创建的导入文件模板，则文件的第一行包含 **:mode** 关键字。例如，您可以将值 ask 指定给 Excel 应用程序 A1 单元格中的 :mode 关键字。



您可以将以下这些值指定给 :mode 关键字：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
:MODE=替换
:MODE=更新
:MODE=询问
:MODE=忽略
:MODE=终止
:MODE=测试
```

:MODE=REPLACE

如果遇到重复的标记，则 DBLoad 实用程序删除“标记名字典”中现有的标记，并使用导入文件中同名的标记来替换它。

:MODE=UPDATE

如果遇到重复的标记，则 DBLoad 实用程序仅在导入文件中明确指定数据的情况下才会覆盖“标记名字典”中现有的标记定义。“标记名字典”中与该标记关联的其它所有数据都保持不变。

如果字段存在于记录中，并且已输入内容或已通过 ":KEYWORD=value" 机制进行设置，则这些此字段被视为明确定义的。如果字段未在记录中指定，并且使用 ":KEYWORD=" 命令重置过关键字，则当前字段值不更新。

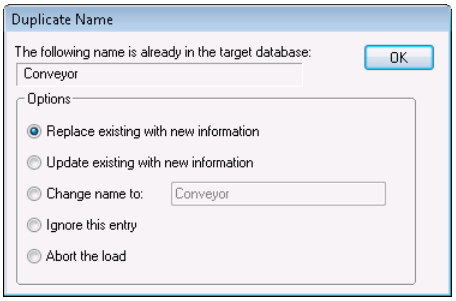
下例显示采用更新模式将导入文件加载/合并到“标记名字典”时产生的结果：

```
:Mode=update
:Group=Group1
:IODisc,Group,DConversion
Tagname1,Group2,
; Tagname1's Group updated to Group2 only
Tagname2,,
; Tagname2's Group updated to Group1 and the DConversion left as is
Tagname3,,Reverse
; Tagname3's Group updated to Group1 and the DConversion to "Reverse"
; the following line "resets" the Group field to its default value
:Group=
;  Data field "Group" is reset to its default value
Tagname4,,
; Tagname4 will be left alone
```

如果要更改标记类型且标记正在使用，则标记类型必须兼容。例如，如果应用程序正在使用标记，则现有的历史趋势标记无法更改为“I/O 整型”。同样，如果应用程序中的输入链接正在使用标记，则该标记无法更改为 ReadOnly=yes。由于存在这些限制，运行 DBLoad 实用程序之前，请先更新目标应用程序的使用计数。

:MODE=询问

在加载“标记名字典”期间遇到重复标记时，DBLoad 会停止。此时出现重复名称对话框，显示一个列表，列出处理重复标记的各个选项。这是缺省导入模式。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

用于处理重复项的选项包括：

- 选择用新信息替换现有信息，以便将现有的标记记录替换为导入文件中的记录。
- 选择用新信息更新现有信息，以便仅在导入文件中明确定义字段的情况下才覆盖现有的标记记录。
- 选择更改名称，以便将导入标记的名称替换为重复名称对话框的方框中所输入的名称。
- 选择忽略此项，以忽略标记并继续导入文件的内容。
- 选择放弃加载，以取消导入过程。

:MODE=IGNORE

DBLoad 导入实用程序忽略重复标记，并继续处理导入文件的剩余记录。

:MODE=TERMINATE

遇到重复标记时，DBLoad 导入操作停止。

:MODE=TEST

DBLoad 在导入文件中扫描错误，而不尝试将标记定义加载到“标记名字典”。DBLoad 生成一份报告，使用导入文件中的行号与位置指出任何格式错误。

使用 :**mode=test** 运行 DBLoad，以确定导入文件中的任何错误。纠正所有错误之后，在运行 DBLoad 之前，将 **mode** 关键字的值更改为 :**mode=replace** 或 :**mode=update**。

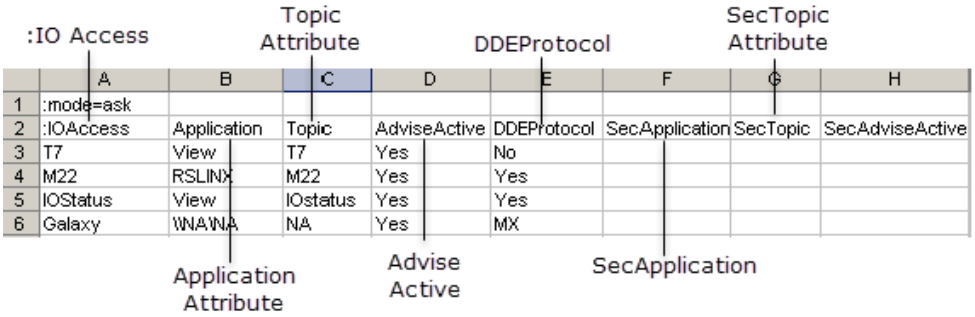
设置访问名与报警组

DBLoad 导入文件包含一些关键字，它们指定 InTouch 应用程序中定义的“访问名”与报警组。

:IOAccess 关键字属性

:IOAccess 关键字标识为 InTouch 应用程序定义的“访问名”。**:IOAccess** 关键字包含一组属性，用于描述定义的 InTouch“访问名”的特征。

下图显示如何使用 :IOAccess 关键字在 Excel 电子表格中定义“访问名”。各个属性在单独的电子表格列中从左到右进行指定。



下表显示与 :IOAccess 关键字关联的属性的列表。此表格按以下顺序列出各个属性：使用以 DBDump 实用程序创建的模板导入文件时指定的顺序。

字符串位置	属性	可接受的值	缺省值
1	应用程序	为“访问名”定义的应用程序名	无

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性	可接受的值	缺省值
2	主题	为“访问名”定义的主题名	无
3	AdviseActive	要从服务器轮询哪些信息 No = 提示所有项 Yes = 只提示激活项	是
4	DDEProtocol	为“访问名”定义的通讯协议 No = Suitelink Yes = DDE MX = 消息交换	否
5	SecApplication	为“访问名”的辅助数据源定义的应用程序名	无
6	SecTopic	为“访问名”的辅助数据源定义的主题名。	无
7	SecAdviseActive	何时轮询辅助服务器上存储的信息 NO = 提示所有项 YES = 只提示激活项	无
8	SecDDEProtocol	为“访问名”的辅助数据源定义的通讯协议 NO = Suitelink YES = DDE MX = 消息交换	无
9	FailoverExpression	值为“真”时将访问名切换到辅助数据源的故障转移表达式	无
10	FailoverDeadband	开始向辅助数据源进行故障转移之前的秒数的整数部分，由“访问名”定义。	无

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性	可接受的值	缺省值
11	DFOFlag	禁用故障转移标帜 Yes =“禁用故障转移标帜”已设置 No =“禁用故障转移标帜”未设置	无
12	FBDFlag	切换回“主数据源”标帜 YES = 故障转移条件消失之后切换回“主数据源” NO = 故障转移条件消失之后不切换回“主数据源”	无
13	FailbackDeadband	故障转移条件消失之后切换回主“访问名”之前的秒数的整数部分	无

:AlarmGroup 关键字属性

DBLoad 导入文件包含某个关键字，它用于标识为 InTouch 应用程序定义的报警组。**:AlarmGroup** 关键字包含一组属性，用于描述 InTouch 应用程序的报警组的特征。

下表显示与 :AccessGroup 关键字关联的属性的列表。此表格按以下顺序列出各个属性 : 使用以 DBDump 实用程序创建的模板导入文件时指定的顺序。

字符串位置	属性	可接受的值	缺省值
1	组	报警组的名称	$System
2	注释	指定给报警组的注释 任何文本字符串	无
3	EventLogged	启用或禁用事件记录 Yes 或 On = 启用事件记录 No 或 Off = 禁用事件记录	否
4	EventLoggingPriority	指定给事件的优先级 1 到 999，0 = 不记录	0
5	LoLoAlarmDisable	禁用或启用 LoLo 报警 0 = 启用 LoLo 报警 1 = 禁用 LoLo 报警	0

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性	可接受的值	缺省值
6	LoAlarmDisable	禁用或启用 Low 报警 0 = 启用 Low 报警 1 = 禁用 Low 报警	0
7	HiAlarmDisable	禁用或启用 High 报警 0 = 启用 High 报警 1 = 禁用 High 报警	0
8	HiHiAlarmDisable	禁用或启用 HiHi 报警 0 = 启用 HiHi 报警 1 = 禁用 HiHi 报警	0
9	MinDevAlarmDisable	禁用或启用“副偏差”报警 0 = 启用“副偏差”报警 1 = 禁用“副偏差”报警	0
10	MajDevAlarmDisable	禁用或启用“主偏差”报警 0 = 启用“主偏差”报警 1 = 禁用“主偏差”报警	0
11	RocAlarmDisable	禁用或启用“变化率”报警 0 = 启用 ROC 报警 1 = 禁用 ROC 报警	0
12	DSCAlarmDisable	禁用或启用离散报警 0 = 启用离散报警 1 = 禁用离散报警	0
13	LoLoAlarmInhibititor	用于抑制 LoLo 报警的标记的名称 标记引用：任何离散或模拟标记	无
14	LoAlarmInhibitor	用于抑制 Low 报警的标记的名称 标记引用：任何离散或模拟标记	无

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性	可接受的值	缺省值
15	HiAlarmInhibitor	用于抑制 High 报警的标记的名称 标记引用：任何离散或模拟标记	无
16	HiHiAlarmInhibitor	用于抑制 HiHi 报警的标记的名称 标记引用：任何离散或模拟标记	无
17	MinDevAlarmInhibitor	用于抑制“副偏差”报警的标记的名称 标记引用：任何离散或模拟标记	无
18	MajDevAlarmInhibitor	用于抑制“主偏差”报警的标记的名称 标记引用：任何离散或模拟标记	无
19	RocAlarmInhibitor	用于抑制“变化率”报警的标记的名称 标记引用：任何离散或模拟标记	无
20	DSCAlarmInhibitor	用于抑制离散报警的标记的名称 标记引用：任何离散或模拟标记	无

定义标记类型关键字与属性

标记记录从标识标记类型的关键字行开始。每个标记关键字都包含一组唯一的属性，用于指定同标记类型关联的数据的特征。

在下例中，:**IODisc** 关键字标识 I/O 离散标记类型。关键字行中其余的值确定与 I/O 离散标记关联的数据的属性。本例使用“记事本”按原始的逗号分隔字符串格式显示该文件的内容。

```
:IODisc,Group,Comment,Logged,EventLogged, EventLoggingPriority,RetentiveValue,InitialDis,
OffMsg,OnMsg,AlarmState,AlarmPri,DConversion, AccessName,ItemUseTagname,ItemName,ReadOnly,
AlarmComment,AlarmAckModel,DSCAlarmDisable, DSCAlarmInhibitor,SymbolicName
```

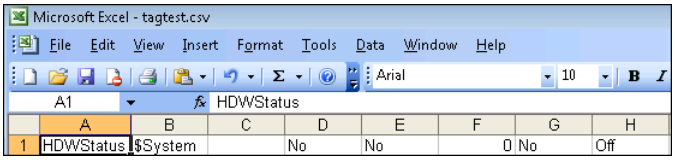
标记类型关键字行下面有一些单独的行，这些行使用一组属性值指定该类型的标记。在下例中，导入文件中的 **HDWStatus** 标记属于“I/O 离散”标记类型。

```
"HDWStatus","$System","",No,No,0,No,Off,"","",,1,Direct,"HistdataViewstr",No,"Status",No,""
,0,0,"",""
```

记录使用引号确定空字符串。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

下图在 Excel 电子表格中显示相同的导入文件数据。由于导入文件中没有指定标记注释，因此 **Comment** 单元格为空白。



标记关键字属性

下表列出与 InTouch 标记关键字关联的所有属性。表格包含一些列，描述与每个标记属性关联的数据类型及其缺省值。

只要跟随的标记数据与相应的属性相匹配，这些标记属性可以在 DBLoad 导入文件中按任何顺序指定。例如，如果在 Excel 导入文件中插入 :IODisc 关键字，则所有的 I/O 离散标记的工程单位都必须放在与 EngUnits 属性相同的 Excel 列中。

属性	可接受的值	缺省值
AccessName	指定给标记的 InTouch“访问名”	无
AlarmAckModel	报警确认模型 整型 0 = 条件 1 = 面向事件 2 = 扩展的摘要	0
AlarmComment	指定给标记的报警注释 文本字符串	无
AlarmDevDeadband	标记偏差报警死区 实型	无
AlarmPri	指定给标记的报警优先级 1 到 999	1
AlarmState	标记报警状态 On、Off 或无	无
AlarmValueDeadband	标记报警死区 实型	0
注释	指定给标记的注释 文本字符串	无
换算	标记值转换 “线性”或“平方根”	线性

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	可接受的值	缺省值
死区	指定给标记的值死区 实型	0
DevTarget	标记偏差目标值 实型	0
DSCAlarmDisable	禁用或启用离散报警 0 = 启用离散报警 1 = 禁用离散报警	0
DSCAlarmInhibitor	用于抑制离散报警的标记的名称	无
EngUnits	指定给标记的工程单位 文本字符串	无
EventLogged	启用或禁用事件记录 Yes 或 On = 启用事件记录 No 或 Off = 禁用事件记录	否
EventLogging	启用或禁用标记事件记录 No 或 Off = 禁用记录功能 Yes 或 On = 启用记录功能	否
EventLoggingPriority	指定给事件的优先级 1 到 999，0 = 不记录	0
组	标记所属报警组的名称	$System
HiAlarmDisable	禁用或启用 High 报警 0 = 启用 High 报警 1 = 禁用 High 报警	0
HiAlarmInhibitor	用于抑制 High 报警的标记的名称 任何离散或模拟标记	无
HiAlarmPri	指定给 High 报警的优先级 1 到 999	1

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	可接受的值	缺省值
HiAlarmState	启用或禁用 High 报警 No 或 Off = 禁用 Yes 或 On = 启用	否
HiAlarmValue	指定给标记的 High 报警点 实型	0
HiHiAlarmDisable	禁用或启用 HiHi 报警 0 = 启用 HiHi 报警 1 = 禁用 HiHi 报警	0
HiHiAlarmInhibitor	用于抑制 HiHi 报警的标记的名称 任何离散或模拟标记	无
HiHiAlarmPri	指定给 HiHi 报警的优先级 1 到 999	1
HiHiAlarmState	启用或禁用 HiHi 报警 No 或 Off = 禁用 Yes 或 On = 启用	否
HiHiAlarmValue	指定给标记的 HiHi 报警点 实型	0
InitialDisc	指定给离散标记的初始值 0、Off、False 或 No = 关闭 1、On、True 或 Yes = 打开	0
InitialMessage	初始标记消息 文本字符串	无
InitialValue	指定给标记的初始值 实型	0
ItemName	指定给标记的项目的名称 文本字符串	无
ItemUseTagname	启用或禁用将标记名用作项目名选项 No 或 False = 禁用 Yes 或 True = 启用	否

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	可接受的值	缺省值
LoAlarmDisable	禁用或启用 Low 报警 0 = 启用 Low 报警 1 = 禁用 Low 报警	0
LoAlarmInhibitor	用于抑制 Low 报警的标记的名称 任何离散或模拟标记	无
LoAlarmPri	指定给 Low 报警的优先级 1 到 999	1
LoAlarmState	启用或禁用 Low 报警 No 或 Off = 禁用 Yes 或 On = 启用	否
LoAlarmValue	指定给标记的 Low 报警点 实型	0
LogDeadband	指定给标记的记录死区 实型	0
已记录的	启用或禁用标记值记录 No 或 Off = 禁用记录功能 Yes 或 On = 启用记录功能	否
LoLoAlarmDisable	禁用或启用 LoLo 报警 0 = 启用 LoLo 报警 1 = 禁用 LoLo 报警	0
LoLoAlarmInhibitor	用于抑制 LoLo 报警的标记的名称 任何离散或模拟标记	无
LoLoAlarmPri	指定给 LoLo 报警的优先级 1 到 999	1
LoLoAlarmState	启用或禁用 LoLo 报警 No 或 Off = 禁用 Yes 或 On = 启用	否
LoLoAlarmValue	指定给标记的 LoLo 报警点 实型	0

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	可接受的值	缺省值
MajDevAlarmDisable	禁用或启用“主偏差”报警 0 = 启用“主偏差”报警 1 = 禁用“主偏差”报警	0
MajDevAlarmInhibitor	用于抑制“主偏差”报警的标记的名称 任何离散或模拟标记	无
MajorDevAlarmPri	指定给“主偏差”报警的优先级 1 到 999	1
MajorDevAlarmState	启用或禁用“主偏差”报警 No 或 Off = 禁用 Yes 或 On = 启用	否
MajorDevAlarmValue	指定给标记的主偏差报警百分比 实型	0
MaxEU	指定给标记的最大工程单位值 实型	32767
MaxLength	最大消息长度 实型	131
MaxRaw	指定给标记的最大原始值 实型	32767
MaxValue	指定给标记的最大值 实型	32767
MinDevAlarmDisable	禁用或启用“副偏差”报警 0 = 启用“副偏差”报警 1 = 禁用“副偏差”报警	0
MinDevAlarmInhibitor	用于抑制“副偏差”报警的标记的名称 任何离散或模拟标记	无

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	可接受的值	缺省值
MinEU	指定给标记的最小工程单位值 实型	-32768
MinorDevAlarmPri	指定给“副偏差”报警的优先级 1 到 999	1
MinorDevAlarmState	启用或禁用副偏差报警 No 或 Off = 禁用 Yes 或 On = 启用	否
MinorDevAlarmValue	指定给标记的副偏差报警百分比 实型	0
MinRaw	指定给标记的最小原始值 实型	-32768
MinValue	指定给标记的最小值 实型	-32768
OffMsg	离散标记“关闭”消息 文本字符串	无
OnMsg	离散标记“打开”消息 文本字符串	无
ReadOnly	标记值只读或可读写 Yes = 只读 No = 可读写	否
RetentiveAlarmParameters	启用或禁用标记“保留”参数 No 或 Off = 禁用 Yes 或 On = 启用	否
RetentiveValue	启用或禁用标记“保留值” 0、Off、False 或 No = 禁用 1、On、True 或 Yes = 启用	否

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	可接受的值	缺省值
RocAlarmDisable	禁用或启用“变化率”报警 0 = 启用 ROC 报警 1 = 禁用 ROC 报警	0
RocAlarmInhibitor	用于抑制“变化率”报警的标记的名称 任何离散或模拟标记	无
ROCAlarmPri	指定给“变化率”报警的优先级 1 到 999	1
ROCAlarmState	启用或禁用“变化率”报警 No 或 Off = 禁用 Yes 或 On = 启用	否
ROCAlarmValue	以百分比表示的标记值变化 实型	0
ROCTimeBase	计算变化率的测量周期 Sec、Min 或 Hr	最小
SymbolicName	S7 Tag Creator 产品赋给输入数据块的符号名称。符号名称列在 S7 Tag Creator 符号表中。	无

:MemoryDisc 关键字属性

DBLoad 导入文件包含 :MemoryDisc 关键字，用于定义可导入到“标记名字典”的内存离散标记。下表列出与内存离散标记属性关联的 :MemoryDisc 关键字的属性。

此表格按使用 DBDump 创建导入文件时指定的 :MemoryDisc 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	已记录的
4	EventLogged
5	EventLoggingPriority

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
6	RetentiveValue
7	InitialDisc
8	OffMsg
9	OnMsg
10	AlarmState
11	AlarmPri
12	AlarmComment
13	AlarmAckModel
14	DSCAlarmDisable
15	DSCAlarmInhibitor
16	SymbolicName

:IODisc 关键字属性

DBLoad 导入文件包含 :IODisc 关键字，用于定义可导入到“标记名字典”的 I/O 离散标记。下表列出与 I/O 离散标记属性关联的 :IODisc 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IODisc 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	已记录的
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	InitialDisc
8	OffMsg
9	OnMsg
10	AlarmState
11	AlarmPri
12	换算

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
13	AccessName
14	ItemUseTagname
15	ItemName
16	ReadOnly
17	AlarmComment
18	AlarmAckModel
19	DSCAlarmDisable
20	DSCAlarmInhibitor
21	SymbolicName

:MemoryInt 关键字属性

DBLoad 导入文件包含 :MemoryInt 关键字，用于定义可导入到“标记名字典”的内存整型标记。下表列出与内存整型标记属性关联的 :MemoryInt 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :MemoryInt 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	已记录的
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinValue
13	MaxValue
14	死区

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHIAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
40	AlarmAckModel
41	LoLoAlarmDisable
42	LoAlarmDisable
43	HiAlarmDisable

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
44	HiHiAlarmDisable
45	MinDevAlarmDisable
46	MajDevAlarmDisable
47	RocAlarmDisable
48	LoLoAlarmInhibitor
49	LoAlarmInhibitor
50	HiAlarmInhibitor
51	HiHiAlarmInhibitor
52	MinDevAlarmInhibitor
53	MajDevAlarmInhibitor
54	RocAlarmInhibitor
55	SymbolicName

:IOInt 关键字属性

DBLoad 导入文件包含 :IOInt 关键字，用于定义可导入到“标记名字典”的 I/O 整型标记。下表列出与 I/O 整型标记属性关联的 :IOInt 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IOInt 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	已记录的
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
12	MinEU
13	MaxEU
14	死区
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHIAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
39	MinRaw

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
40	MaxRaw
41	换算
42	AccessName
43	ItemUseTagname
44	ItemName
45	ReadOnly
46	AlarmComment
47	AlarmAckModel
48	LoLoAlarmDisable
49	LoAlarmDisable
50	HiAlarmDisable
51	HiHiAlarmDisable
52	MinDevAlarmDisable
53	MajDevAlarmDisable
54	RocAlarmDisable
55	LoLoAlarmInhibitor
56	LoAlarmInhibitor
57	HiAlarmInhibitor
58	HiHiAlarmInhibitor
59	MinDevAlarmInhibitor
60	MajDevAlarmInhibitor
61	RocAlarmInhibitor
62	SymbolicName

:MemoryReal 关键字属性

DBLoad 导入文件包含 :MemoryReal 关键字，用于定义可导入到“标记名字典”的内存实型标记。下表列出与内存整型标记属性关联的 :MemoryReal 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :MemoryReal 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
1	组
2	注释
3	已记录的
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinValue
13	MaxValue
14	死区
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState
29	MinorDevAlarmValue

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	AlarmComment
40	AlarmAckModel
41	LoLoAlarmDisable
42	LoAlarmDisable
43	HiAlarmDisable
44	HiHiAlarmDisable
45	MinDevAlarmDisable
46	MajDevAlarmDisable
47	RocAlarmDisable
48	LoLoAlarmInhibitor
49	LoAlarmInhibitor
50	HiAlarmInhibitor
51	HiHiAlarmInhibitor
52	MinDevAlarmInhibitor
53	MajDevAlarmInhibitor
54	RocAlarmInhibitor
55	SymbolicName

:IOReal 关键字属性

DBLoad 导入文件包含 :IOReal 关键字，用于定义可导入到“标记名字典”的 I/O 实型标记。下表列出与 I/O 实型标记属性关联的 :IOReal 关键字属性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此表格按使用 DBDump 创建导入文件时指定的 :IOReal 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	已记录的
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	RetentiveAlarmParameters
8	AlarmValueDeadband
9	AlarmDevDeadband
10	EngUnits
11	InitialValue
12	MinEU
13	MaxEU
14	死区
15	LogDeadband
16	LoLoAlarmState
17	LoLoAlarmValue
18	LoLoAlarmPri
19	LoAlarmState
20	LoAlarmValue
21	LoAlarmPri
22	HiAlarmState
23	HiAlarmValue
24	HiAlarmPri
25	HiHiAlarmState
26	HiHiAlarmValue
27	HiHiAlarmPri
28	MinorDevAlarmState

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
29	MinorDevAlarmValue
30	MinorDevAlarmPri
31	MajorDevAlarmState
32	MajorDevAlarmValue
33	MajorDevAlarmPri
34	DevTarget
35	ROCAlarmState
36	ROCAlarmValue
37	ROCAlarmPri
38	ROCTimeBase
39	MinRaw
40	MaxRaw
41	换算
42	AccessName
43	ItemUseTagname
44	ItemName
45	ReadOnly
46	AlarmComment
47	AlarmAckModel
48	LoLoAlarmDisable
49	LoAlarmDisable
50	HiAlarmDisable
51	HiHiAlarmDisable
52	MinDevAlarmDisable
53	MajDevAlarmDisable
54	RocAlarmDisable
55	LoLoAlarmInhibitor
56	LoAlarmInhibitor
57	HiAlarmInhibitor

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
58	HiHiAlarmInhibitor
59	MinDevAlarmInhibitor
60	MajDevAlarmInhibitor
61	RocAlarmInhibitor
62	SymbolicName

:MemoryMsg 关键字属性

DBLoad 导入文件包含 :MemoryMsg 关键字，用于定义可导入到“标记名字典”的内存消息标记。下表列出与内存消息标记属性关联的 :MemoryMsg 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :MemoryMsg 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	已记录的
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	MaxLength
8	InitialMessage
9	AlarmComment
10	SymbolicName

:IOMsg 关键字属性

DBLoad 导入文件包含 :IOMsg 关键字，用于定义可导入到“标记名字典”的 I/O 消息标记。下表列出与 I/O 消息标记属性关联的 :IOMsg 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IOMsg 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	已记录的

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

字符串位置	属性
4	EventLogged
5	EventLoggingPriority
6	RetentiveValue
7	MaxLength
8	InitialMessage
9	AccessName
10	ItemUseTagname
11	ItemName
12	ReadOnly
13	AlarmComment
14	SymbolicName

:GroupVar 关键字属性

DBLoad 导入文件包含 :GroupVar 关键字，用于定义可导入到“标记名字典”的“组变量”标记。下表列出与“组变量”标记属性关联的 :GroupVar 关键字属性。

注意：InTouch“组变量”标记已废弃。包含 :GroupVar 关键字仅仅是为了支持旧有的应用程序。

此表格按使用 DBDump 创建导入文件时指定的 :GroupVar 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	SymbolicName

:HistoryTrend 关键字属性

DBLoad 导入文件包含 :HistoryTrend 关键字，用于定义可导入到“标记名字典”的 HistTrend 标记。下表列出与 HistTrend 标记属性关联的 :HistoryTrend 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :HistoryTrend 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	SymbolicName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

:TagID 关键字属性

DBLoad 导入文件包含 :TagID 关键字，用于定义可导入到“标记名字典”的“标记 ID”标记。下表列出与“标记 ID”标记属性关联的 :TagID 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :TagID 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释

:IndirectDisc 关键字属性

DBLoad 导入文件包含 :IndirectDisc 关键字，用于定义可导入到“标记名字典”的间接离散标记。下表列出与间接离散标记属性关联的 :IndirectDisc 关键字属性。

此表按使用 DBDump 创建导入文件时指定 :IndirectDisc 关键字属性的顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue
6	SymbolicName

:IndirectAnalog 关键字属性

DBLoad 导入文件包含 :IndirectAnalog 关键字，用于定义可导入到“标记名字典”的间接模拟标记。下表列出与间接模拟标记属性关联的 :IndirectAnalog 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IndirectAnalog 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue
6	SymbolicName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

:IndirectMsg 关键字属性

DBLoad 导入文件包含 :IndirectMsg 关键字，用于定义可导入到“标记名字典”的间接消息标记。下表列出与间接消息标记属性关联的 :IndirectMsg 关键字属性。

此表格按使用 DBDump 创建导入文件时指定的 :IndirectMsg 关键字属性顺序列出这些属性。如需与这些属性及其缺省值关联的数据，请参阅标记关键字属性。

字符串位置	属性
1	组
2	注释
3	EventLogging
4	EventLoggingPriority
5	RetentiveValue
6	SymbolicName

在导入文件中使用空字符串

对于字典导入文件而言，包含空字符串的字段与没有数据的字段之间是有区别的。可以指定空字符串的关键字属性包括：

注释	工程单位	OffMsg
InitialMessage	OnMsg	应用程序
ItemName	主题	

在下例中，空字符串使用英文双引号 (" ") 表示：

```
:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,"",$System
```

其中：

Tagname1 的 Comment 字段的值为 Hi，Tagname2 的 Comment 字段的值为空注释。

Microsoft Excel 在保存文件时忽略表示空字符串的英文双引号，这会产生以下结果：

```
:Comment="HI"
:MemoryDisc,Comment,Group
Tagname1,,$System
Tagname2,,$System
```

要确保空字符串在 Excel 中能起作用，请在单元格中输入一个空格作为属性值。

使用字段的缺省值

您可以使用关键字给某条记录的特定字段设置缺省值。缺省值是该标记类型的原始 InTouch 值。例如，某个内存离散标记使用 Group=$System、EventLogging=Off、InitialValue=Off 作为缺省值。

例如：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
:KEYWORD=value
```

这就为所有后续的数据记录设置了所引用字段的缺省值。使用此功能可以为许多记录中保持不变的字段设置缺省值。如果某个字段定义了缺省值，则在记录中没有该值的数据时会使用缺省值。

例如，如果设置 :GROUP=Reactor_Site，则 GROUP 列中包含空白项的所有标记都会指定给 Reactor_Site“报警组”。如果标记的 GROUP 列中输入了 $System，则此标记仍指定给 $System“报警组”。

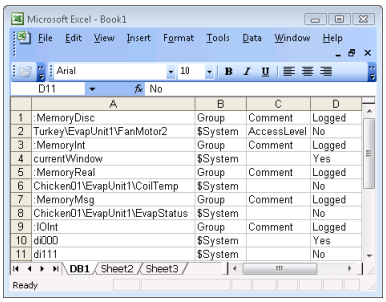
您可以通过忽略方程中的值将某个单独的关键字重置为原始缺省值。例如，:GROUP=。

要重置所有关键字，请使用 :RESET 命令。此命令不含参数，并且会影响使用此命令之后处理的文件中的所有项目。

创建 SuperTag 实例

您可以使用“应用程序管理器”中的 DBLoad 实用程序来创建 SuperTag。但所创建的 SuperTag 实例不会反映在 SuperTag 模板定义中。

您必须使用有效的 SuperTag 格式，并且 SuperTag 实例数据记录必须以该标记类型的有效关键字开头。例如：



下面是有效语法的示例：

```
ParentInstance\ChildMember
ParentInstance\ChildMember\Submember
```

以下是无效语法的示例：

```
ParentInstance\
ParentInstance\ChildMember\
```

如果使用无效格式，则会出现一条错误消息。

导入包含 SuperTag 实例的 CSV 文件时，这些实例自动添加到“标记名字典”中，并且可以立即在动画链接与 InTouch QuickScript 中使用。

使用 DBLoad 导入标记定义

使用 DBLoad 导入文件的内容时，所有标记定义都导入到所选 InTouch 应用程序的“标记名字典”中。

如果导入失败，则出现一条消息，说明失败的原因。错误消息写入 Logger 中。

将标记定义导入 InTouch 应用程序

1. 关闭 WindowMaker 和 WindowViewer。
2. 对于要从导入文件中加载标记定义的“标记名字典”，备份其所属的应用程序。
3. 启动应用程序管理器。
4. 从列表中选择其“标记名字典”将接收导入的标记定义的应用程序。

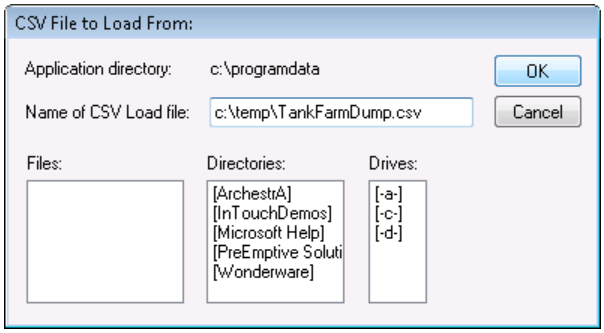
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5. 在文件菜单上的数据组中，选择 **DBLoad**。

   此时会出现一条消息，要求确认是否已备份 InTouch 应用程序。

6. 选择是以确认应用程序已备份。

   此时出现要从中加载的 **CSV** 文件对话框。



7. 在 **CSV** 加载文件名框中，找到并选择要导入的文件。
8. 选择确定。

   根据 DBLoad 是将新的还是现有的标记定义导入“标记名字典”中，下一个步骤可能有所不同。

   - 如果导入新的标记定义，则将新标记数据加载到应用程序的“标记名字典”。此时出现一条消息，确认已成功加载并合并数据。
   - 如果导入现有的标记定义，:mode 关键字设置为 :mode=ask 并且导入文件包含重复标记，则导入将停止。此时显示用于处理重复标记的选项，您也可以取消导入。如需有关关键字选项的详细信息，请参阅设置字典导入文件的操作模式。

使用 DBLoad 实用程序导入标记定义的已知限制

导入文件大小超过 2 GB 的大量标记计数 newtag.tag 文件时，DBLoad 实用程序可能无法导入文件或显示性能问题。

限制：

- 如果 newtags.tag 文件大小超过 2 GB 限制，则 DBLoad 可能无法导入大量标记计数文件。
- DBLoad 在导入大量标记计数 CSV 文件时会遇到性能问题。

变通方法：

使用以下注册表项可避免将标记导入到 newtags.tag 文件，而是导入到 Tagname.x 文件。

1. 打开“注册表编辑器”。
2. 导航到 **HKEY_LOCAL_MACHINE\SOFTWARE\Wow6432Node\Wonderware\InTouch\Installation**。
3. 使用鼠标右键单击导航窗格中的安装文件夹或主区域的任意位置，选择新建，然后选择 **DWORD 值**。
4. 将新创建的文件重命名为 `DisableWriteToNewTag`。
5. 将“DisableWriteToNewTag”键设置为 1。
6. 现在使用 DBLoad 从 CSV 文件导入标记，导入会成功。

DisableWriteToNewTag 注册表项的不同值对导入功能的影响如下所示：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

DisableWriteToNewTag 的 DWORD 值	对导入的影响
DisableWriteToNewTag 注册表项存在，并且 DWORD 值设置为 1	DBLoad 实用程序不会将标记导入到 newtags.tag 文件
DisableWriteToNewTag 注册表项存在，并且 DWORD 值设置为 0	DBLoad 实用程序会将标记导入到 newtags.tag 文件

注意：从 InTouch HMI 2023 版起，缺省条件下，注册表项 DisableWriteToNewTag 不存在，DBLoad 实用程序不会将标记导入到 newtags.tag 文件。

在 InTouch 应用程序之间导出与导入 InTouch 窗口

您可以从 WindowMaker 导出所有三种 InTouch 应用程序类型的窗口，但是在导入所导出的窗口或从 InTouch 应用程序中直接导入窗口时，存在一些限制。

- 对于独立的 InTouch 应用程序，无法从包含工业图形的发布的与托管的 InTouch 应用程序中导入任何窗口。此时出现警告消息，并将关于哪些窗口未导入的信息写入 Logger。

  如果从包含工业图形的托管的或发布的 InTouch 应用程序中导入窗口，则窗口会导入，但工业图形不起作用并显示为“未找到”。

- 对于托管的 InTouch 应用程序，您可以从发布的、独立的及其它托管的 InTouch 应用程序中导入任何窗口。内嵌的工业图形不会导入。
- 对于发布的 InTouch 应用程序，您可以从独立的 InTouch 应用程序中导入任何窗口。内嵌的工业图形不会导入。

导入窗口

从现有的 InTouch 应用程序将窗口导入当前应用程序时，由于可以复用先前创建的窗口、对象以及窗口脚本，因此可以缩短开发时间。

要能够导入窗口，必须先将应用程序转换为 InTouch HMI 软件的当前版本。

缺省条件下，将为与导入窗口关联的标记创建占位符。在导入之后，您可以将占位符转换为本地标记或远程标记引用。如需详细信息，请参阅导入的窗口与脚本中的标记占位符。如果目标应用程序中已经存在关联的标记，则可以在导入期间选择使用这些标记。

导入包含 SmartSymbol 的窗口并且选择使用现有的标记时，InTouch HMI 仍然为恢复的符号保留占位符，即便标记存在于目标应用程序中时也是如此。

从包含 SuperTag 的应用程序中导入窗口时，只有窗口中实际使用的 SuperTag 实例才会导入新应用程序。整个 SuperTag 模板结构不会导入。例如，如果应用程序中定义了数百个 SuperTag 成员标记，其中只有 50 个用在导入的窗口中，则仅导入这 50 个标记。

注意事项：如果使用除导入或导出之外的方法移动 InTouch 窗口文件，则可能破坏应用程序“标记名字典”的内容。

导入窗口

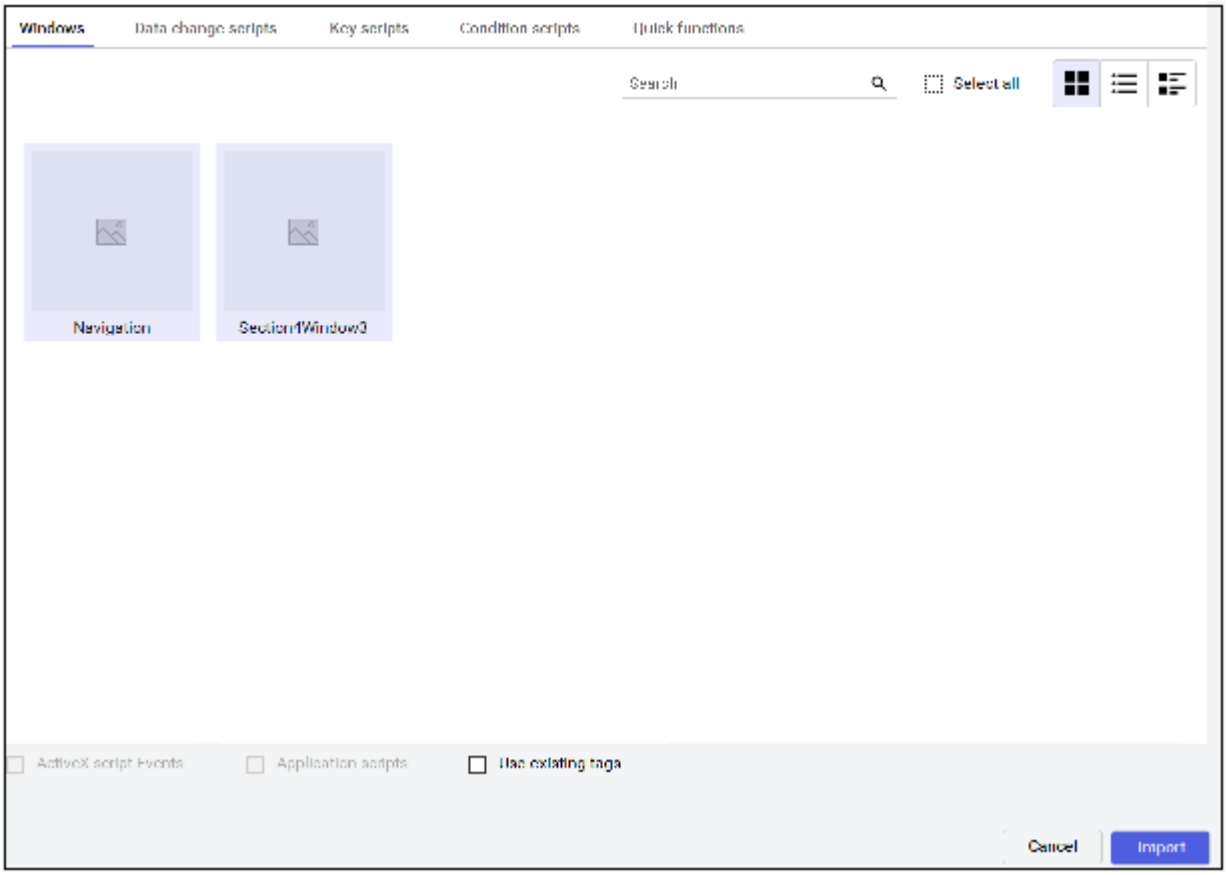
1. 关闭当前应用程序中的所有窗口。
2. 在文件菜单上，选择导入，选择可视化，然后选择窗口与脚本。

   此时出现打开文件夹对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 选择包含要导入窗口的应用程序所在的文件夹。
4. 选择确定。

   此时出现应用程序数据导入选项对话框。



5. 在窗口选项卡中选择要导入的各个单独窗口。
6. 如果应用程序中已存在与导入窗口关联的标记，并且您希望使用这些标记而不是占位符，请选中使用已有的标记复选框。
7. 选择导入。
8. 将占位符标记转换为本地标记或远程标记引用。如需详细信息，请参阅转换导入窗口的占位符标记。
9. 如果导入的窗口包含一个或多个向导，双击每个向导以打开其属性面板。如果导入的窗口包含一个或多个 SmartSymbol，编辑每个 SmartSymbol 并创建新的实例。

转换导入窗口的占位符标记

从当前应用程序导入或导出窗口或 QuickScript 时，与该窗口或 QuickScript 关联的所有标记都会随窗口一起传输。但是，这些标记并不添加到新应用程序的“标记名字典”。相反，除非在导入时选择了节省占位符选项，否则标记自动标记为占位符标记。您必须转换这些占位符标记，如果需要，请在新应用程序的“标记名字典”中定义它们。

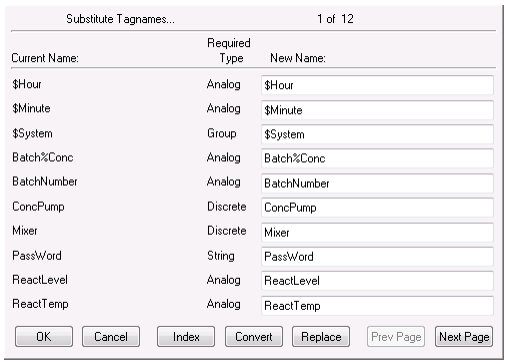
要转换窗口的标记

1. 在 WindowMaker 中打开窗口。

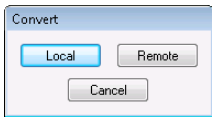
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 按 F2 以**选择**窗口中的所有**对**象。
3. 在**动画**菜**单**上的**替换组**中，**选择标记**。

   此时出现**替换标记**名对话框。

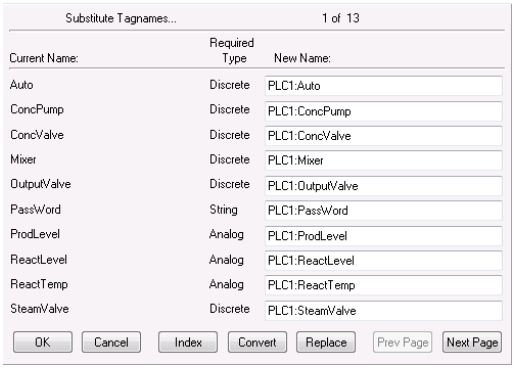


4. 选择**转换**。此时出现**转换**对话框。



5. 转换这些标记。
   - **选择本地**，以便将占位符**标记转换为**本地**标记**。程序会提示您在“**标记**名字典”中定**义每**个**标记**。
   - **选择远程**，以便将占位符**标记转换为远**程**标记**引用。此**时**出**现访问名对话**框。**选择**“**访问**名”，然后**选择关闭**。

   转换之后，**替换标记**名对话框中显示新的标记。



6. 选择确定。

导出窗口

您可以**导**出**应**用程序窗口，以便

- **创**建或**维护**一个包含所有窗口的**库应**用程序。
- 在另一个**应**用程序中**创**建**远**程**标记**引用。

要能**够导**出窗口，必**须**先将**应**用程序**转换为** InTouch HMI **软**件的当前版本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

导出窗口时，与该窗口关联的所有对象与动画链接都将导出。与窗口中的对象关联的标记转换为占位符标记，以防止覆盖目标应用程序中现有的标记。如需有关转换占位符标记的详细信息，请参阅转换导入窗口的占位符标记。

重要事项：如果使用除导入或导出之外的方法移动 InTouch 窗口文件，则可能破坏应用程序“标记名字典”的内容。

导出窗口

1. 关闭当前应用程序中的所有窗口。
2. 在文件菜单上，选择导出，选择可视化，然后选择窗口。
3. 选择要导出的窗口，然后选择导出。

   此时出现打开文件夹对话框。
4. 选择要将窗口导出到其中的应用程序所在的文件夹。
5. 选择确定。
6. 如果出现问题，此时出现导出操作出现问题对话框。选择要执行动作的选项，然后选择确定。

导入脚本

通过将现有 QuickScript 从 InTouch 应用程序导入当前应用程序，可以节省开发时间。

要能够导入脚本，必须先将应用程序转换为 InTouch HMI 软件的当前版本。

缺省条件下，将为与导入的 QuickScript 关联的标记创建占位符。在导入之后，您可以将占位符转换为本地标记或远程标记引用。如需详细信息，请参阅导入的窗口与脚本中的标记占位符。如果目标应用程序中已经存在关联的标记，则可以在导入期间选择使用这些标记。

要导入窗口脚本，必须导入整个窗口。

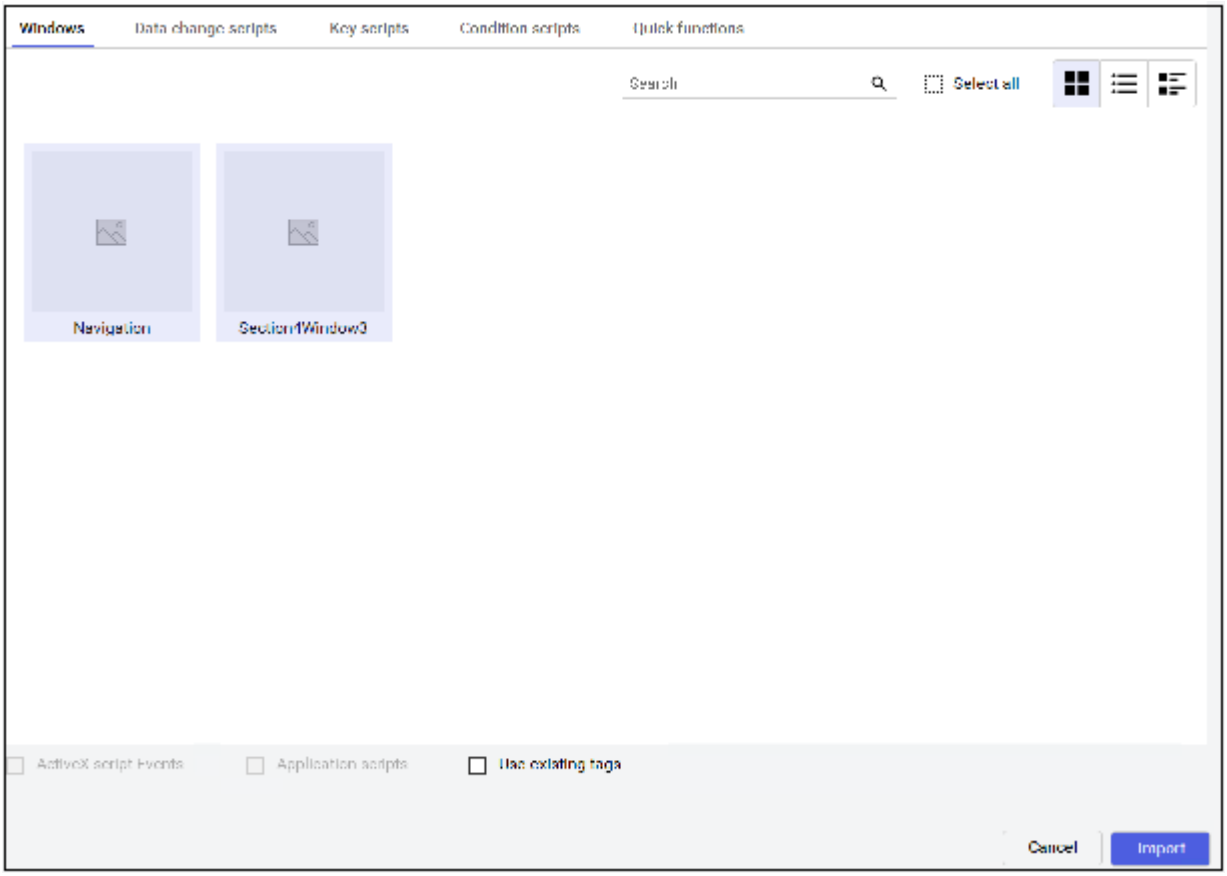
要让导入的“ActiveX 事件”脚本在目标应用程序中正确运行，必须还要在目标应用程序中使用原先为其创建脚本的相同的 ActiveX 控件与事件,并且必须将它加载到内存中。如果关闭包含 ActiveX 控件的窗口，则与该窗口关联的任何脚本（“ActiveX 事件”脚本或 QuickScript）都无法正确运行。

导入 QuickScript

1. 关闭当前应用程序中的所有窗口。
2. 在文件菜单上，选择导入，选择可视化，然后选择窗口与脚本。

   此时出现打开文件夹对话框。
3. 选择包含要导入脚本的应用程序所在的文件夹。
4. 选择确定。此时出现应用程序数据导入选项对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



5. 选中要导入的 QuickFunction 类型旁边的复选框，然后选择**选择**以选择要导入的各个单独脚本。

   注意：要导入窗口脚本，必须导入整个窗口。如需有关详细信息，请参阅导入窗口。

6. 如果应用程序中已存在与导入脚本关联的标记，并且您希望使用这些标记而不是占位符，请选中**使用已有的标记**复选框。
7. **选择导入**。如果应用程序中包含使用相同名称的脚本，则程序会提示您选择覆盖、忽略或重命名。
8. 将占位符标记转换为本地标记或远程标记引用。如需有关详细信息，请参阅转换导入的脚本中的占位符标记。

转换导入的脚本中的占位符标记

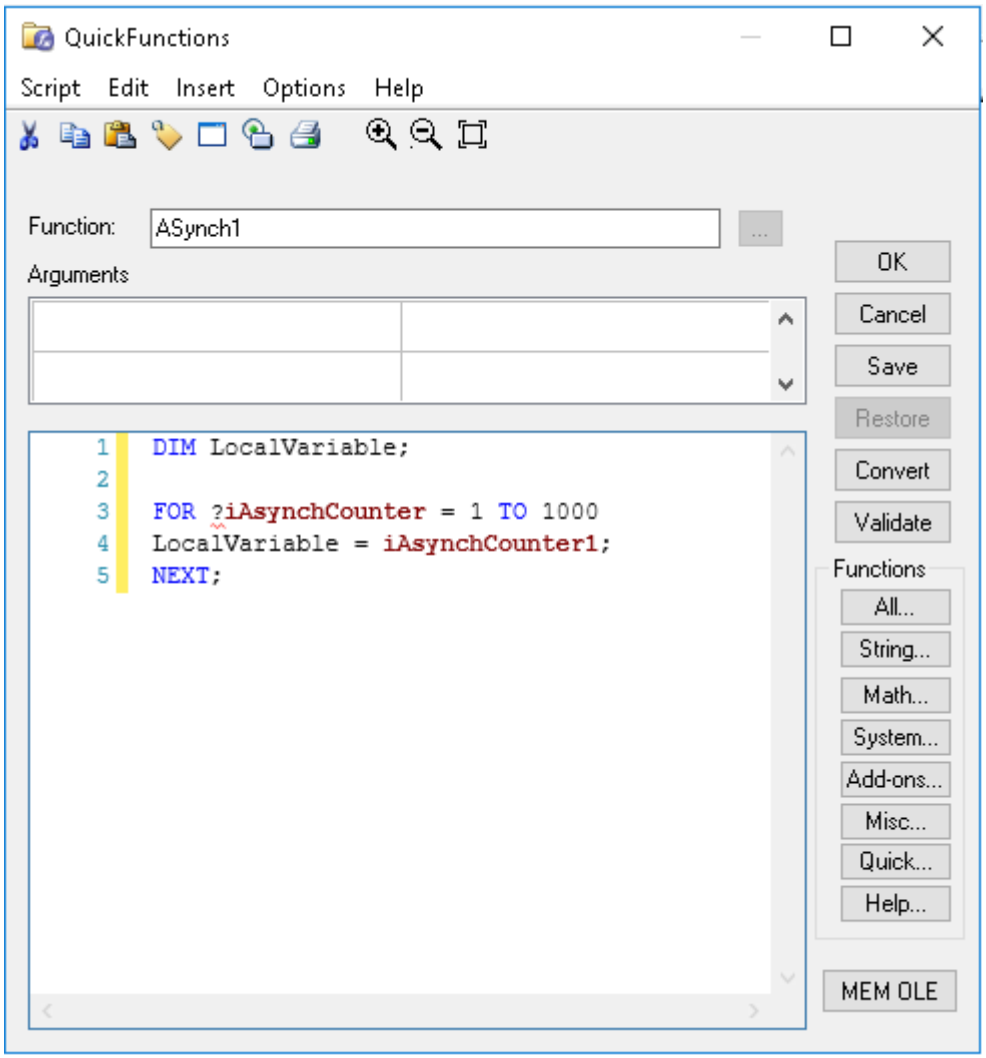
在当前应用程序中导入或导出 QuickFunction 时，与该 QuickFunction 关联的所有标记都会一起传输。但是，这些标记并不添加到新应用程序的“标记名字典”。相反，它们会自动标为占位符标记。您必须转换这些占位符标记，如果需要，请在新应用程序的“标记名字典”中定义它们。

转换导入的脚本中的占位符标记

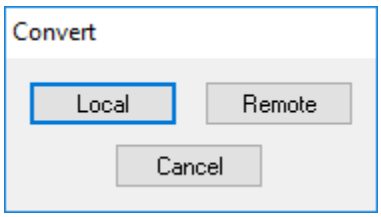
1. 在**脚本**窗格上，选择已导入的 QuickFunction 的类型。

   此时出现 QuickFunction 脚本编辑器，显示文件中与所选的脚本类型对应的第一个 QuickFunction。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 选择**转换**。此时出现**转换**对话框。



1. 转换这些标记。
   - 选择**本地**，以便将占位符标记转换为本地标记。程序会提示您在“标记名字典”中定义每个标记。
   - 选择**远程**，以便将占位符标记转换为远程标记引用。此时出现**访问名对话**框。突出显示“访问名”，然后选择**关闭**。
2. 转换标记之后，选择 QuickScript 编辑器中的确定。

导入的窗口与脚本中的标记占位符

导入窗口或 QuickScript 时，您可以配置如何处理关联的标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- **使用占位符标记。**

  缺省条件下，导入的标记会转换为“占位符”（或“索引”）标记。最多允许使用 4096 个占位符。

  占位符标记包含一个三字符的前缀。例如，如果原始标记为 WaterHeater，则占位符标记为 ?d:WaterHeater。

  如果导入的标记包含 30、31 或 32 个字符，占位符前缀仍会添加到标记的开头，并且不截短现有标记的长度。例如，仅对占位符标记而言，32 个字符的标记增加到 35 个字符长。对于标准标记，不支持这种标记长度的增加。

  要在应用程序中使用占位符标记，必须执行以下操作之一：

  将此标记转换为常规（本地）标记并在“标记名字典”中定义它。

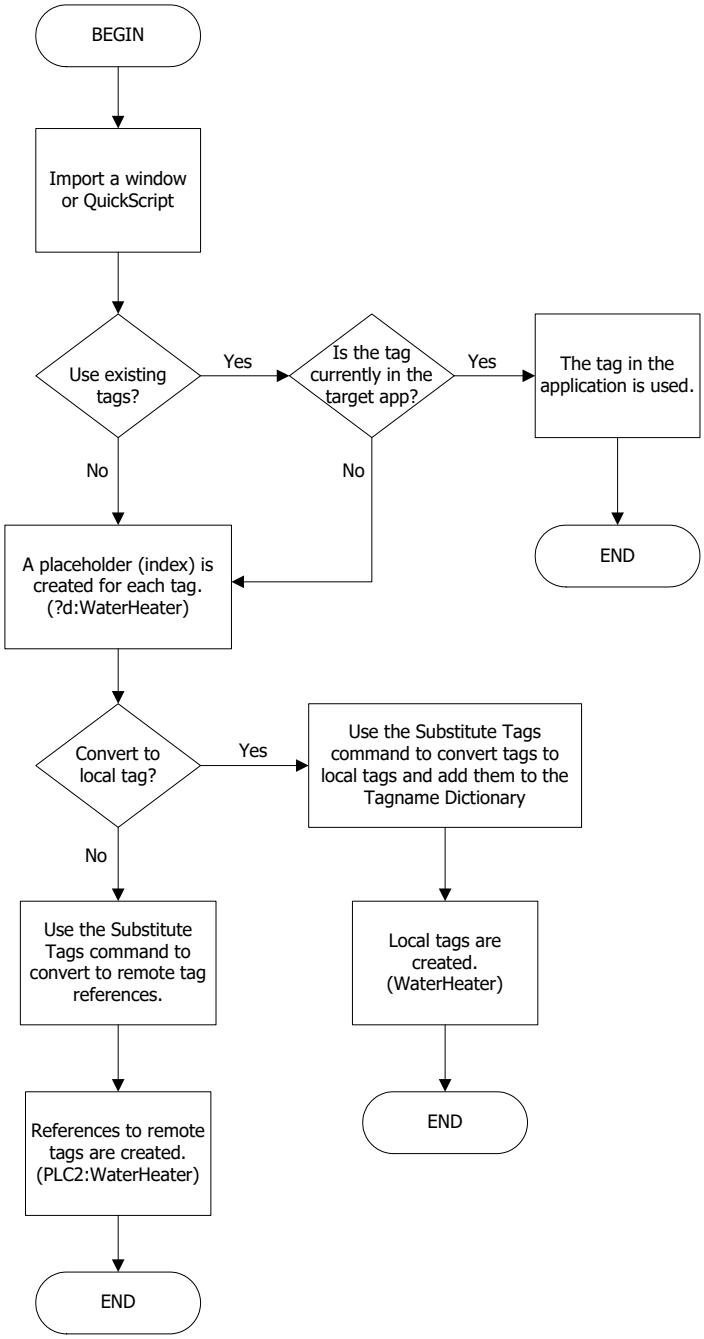
  将它转换为远程标记引用。远程标记的示例有 PLC2:WaterHeater。远程标记引用使得应用程序可以立刻接收来自远程标记服务器的数据，而不必在本地“标记名字典中”定义单个标记。

- **使用现有的标记。**

  在导入期间，如果选择使用现有的标记，则 InTouch HMI 确认“标记名字典”中是否已经存在导入的标记。如果标记已经存在，则该标记作为全限定标记导入。通过使用此选项，可以减少占位符的总数，以便您导入标记数据库较大的应用程序。

下图介绍如何处理导入的窗口与 QuickScript 中的标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



从应用程序中导出工业图形

您可以将应用程序中的所有工业图形导出到 aaPKG 文件。然后可将图形从该文件导入到相同或不同计算机上的其它应用程序中。

您不能单独选择要从应用程序中导出的工业图形。所有工业图形都将从应用程序中导出。

从应用程序中导出工业图形

1. 在 WindowMaker 中打开包含您要导出的工业图形的应用程序。
2. 在文件菜单上，选择导出，然后从可视化组中，选择所有工业图形。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现导出工业图形对话框，该对话框用于指定导出文件的目标文件夹和名称。

3. 选择要将 aaPKG 文件导出到的目标文件夹。
4. 如果需要，在文件名字段中输入导出文件的名称。

   缺省的导出文件名为 IndustrialGraphics.aaPKG。
5. 选择保存。

   一个水平条形显示工业图形加载到导出文件的进度。
6. 完成导出过程后，在 Windows 资源管理器中导航到目标文件夹，并验证是否已创建导出文件。

将工业图形导入到应用程序

您可以将其它应用程序中创建的工业图形导入到 WindowMaker 中正在运行的活动应用程序。

只有工业图形会从 aaPKG 文件中导入。导入的图形将覆盖在 WindowMaker 中打开以进行编辑的应用程序中的任何图形。如果 aaPKG 文件包括不支持的组件，导入将失败并显示一个错误对话框。

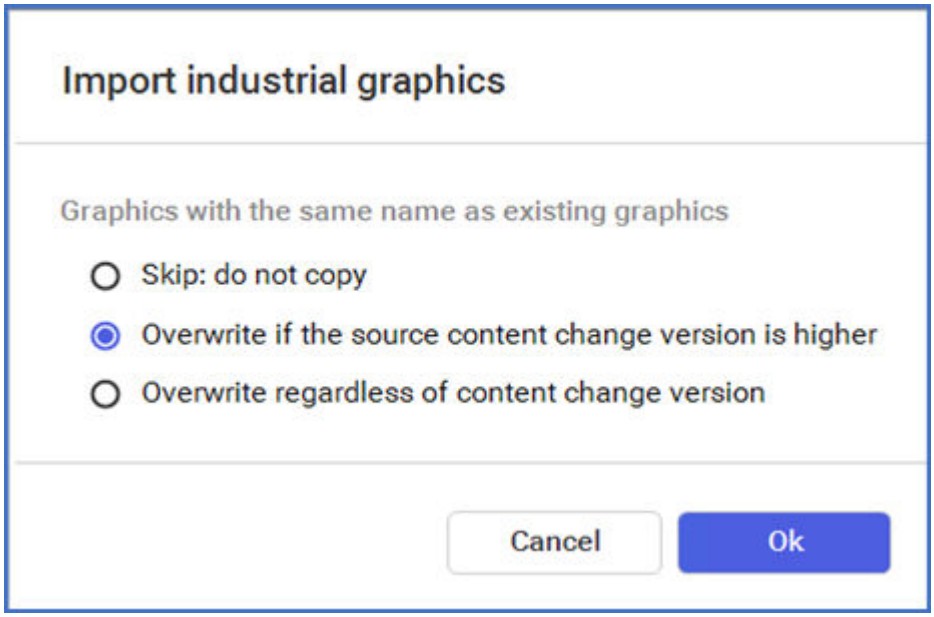
将工业图形导入到应用程序

1. 在 WindowMaker 中打开要导入工业图形的应用程序。
2. 在文件菜单上，选择导入，然后从可视化组中，选择工业图形。

   此时出现导入工业图形对话框，该对话框用于指定包含工业图形导出文件的文件夹。
3. 使用 Windows 资源管理器，转到包含导出的工业图形的 aaPKG 文件的文件夹。
4. 选择要导入的 aaPKG 文件。

   文件名字段显示所选文件的名称。
5. 选择打开。

   此时出现“导入工业图形”对话框，其中包含以下覆盖图形的选项。



- **跳过：不复制** - 不会导入图形。

- **如果源内容更改版本较高,则覆盖** - 只有导入的文件版本高于已安装的版本时，才会导入图形。
- **无论内容更改版本如何都覆盖** - 将导入图形。

6. 选择确定。

   一个水平条形显示工业图形导入到活动的应用程序的进度。完成后，进度指示器将消失。

从工业图形工具箱导出所选符号

您可以将所选工业图形从应用程序的“工业图形工具箱”导出到 aaPKG 文件。然后可将这些图形从该文件导入到相同或不同计算机上的其它应用程序中。

注意：此过程说明了如何导出所选工业图形。如需有关导出所有工业图形的操作说明，请参阅从应用程序中导出工业图形。

从应用程序中导出所选工业图形

1. 在 WindowMaker 中打开包含您要选择导出的工业图形的应用程序。
2. 在“工业图形工具箱”中选择要导出的符号。
3. 使用鼠标右键单击所选符号以显示快捷菜单。
4. 从快捷菜单中选择导出，然后选择符号...。

   此时出现导出工业图形对话框，该对话框用于指定导出文件的目标文件夹和名称。
5. 选择要将 aaPKG 文件导出到的目标文件夹。
6. 如果需要，在文件名字段中输入导出文件的名称。

   缺省的导出文件名是“工业图形工具箱”中第一个所选符号的名称。
7. 选择保存。

   一个水平条形显示工业图形加载到导出文件的进度。

导入与嵌入自定义客户端控件

您可以创建自定义的 Windows 客户端控件并将其嵌入应用程序的工业图形中。首先，您必须将该客户端控件导入到 WindowMaker 的“工业图形工具箱”。本节分别介绍导入后再嵌入自定义客户端控件的步骤。

导入自定义客户端控件

1. 为应用程序创建自定义客户端控件。
2. 将该客户端控件放入安装 InTouch WindowMaker 的电脑可访问的文件夹。
3. 在文件菜单上，选择导入，然后从可视化组中，选择客户端控件。

重要事项：只有独立应用程序可导入自定义客户端控件。自定义客户端控件不能导入到传统或已发布的 InTouch HMI 应用程序。

此时出现导入客户端控件对话框，其中的字段可输入您所创建的自定义客户端控件名称。

1. 使用 Windows 资源管理器，转到放置客户端控件 .dll 文件的文件夹。
2. 选择该客户端控件 .dll 文件，然后选择打开。

   WindowMaker 会更新并显示您已导入“工业图形工具箱”的自定义客户端控件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

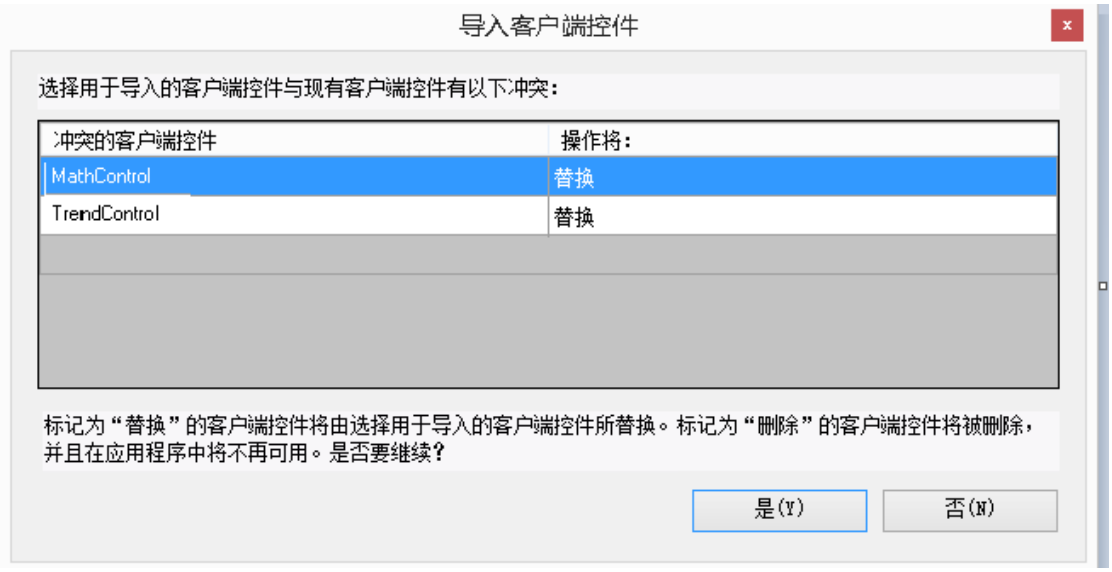
您也可以从“工业图形工具箱”删除已导入的客户端控件。首先，在“工业图形工具箱”中选择该客户端控件。然后，单击右键显示快捷菜单并选择删除。

解决当导入重复客户端控件时产生的冲突

您可以导入不同版本的客户端控件并覆盖现有控件。存放现有控件的 .dll 将被导入的库所替换。在导入新的客户端控件 .dll 时，会检测冲突的客户端控件。

注意：冲突检测仅基于控件名称进行。库文件名或版本对冲突检测没有任何影响。

例如，如果导入的客户端控件 .dll 中包含两个控件 **MathControl** 和 **TrendControl**，而当前库中包含名称相同的控件，那么将显示**导入客户端控件对话**框：



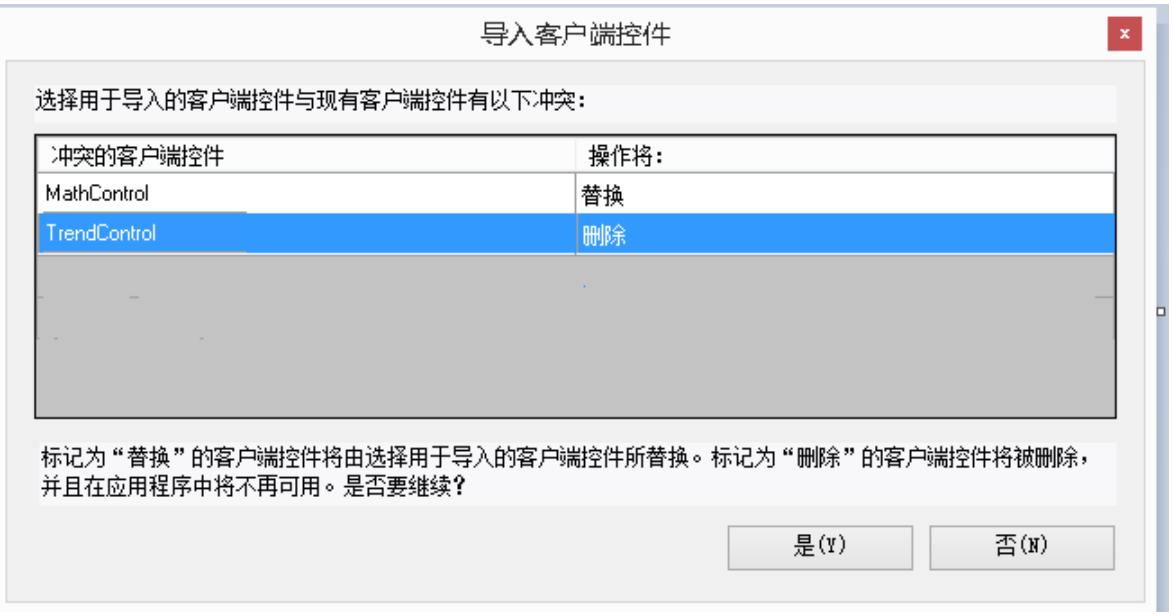
现有的客户端控件 .dll 将被替换，新控件现在将在库中可用。

如果您在操作将列中看到“删除”字样，则表示当前库中存在导入库中没有的控件。因为存放 .dll 必须被替换以解决控件冲突的问题，所以当前 .dll 中有但导入 .dll 中没有的所有控件都会在导入过程中被删除。

例如，导入的客户端控件 .dll 包含控件 MathControl 和 DatabaseControl，而当前库包含 MathControl 和 TrendControl，那么在导入时将从库中删除 TrendControl。

导入客户端控件对话框将提示您确认删除：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



导入完成时，库将被替换，TrendControl 将被删除。

重新启动 WindowMaker 以更新图形工具箱中的控件。

注意：如果导入了已嵌入符号的较新版本的客户端控件，那么重新启动 WindowMaker 并刷新图形缩略图时不会更新控件的内容。必须编辑并保存新客户端控件的该符号，才能使其反映在缩略图中。

在工业图形中嵌入客户端控件

客户端控件是从“工业图形工具箱”中嵌入的。图形工具箱中已包含几个客户端控件。您可以将这些现有控件嵌入到工业图形中，也可以导入自定义控件并嵌入这些控件。

将客户端控件嵌入到工业图形中

1. 在 WindowMaker 中打开要嵌入自定义客户端控件的应用程序。
2. 打开包含中要嵌入自定义客户端控件的工业图形的窗口。
3. 选择工业图形。
4. 从菜单栏选择嵌入工业图形图标。

   重要事项：您不能将自定义客户端控件从“工业图形工具箱”拖放到工业图形上。只能嵌入自定义客户端控件。

5. 根据需要为应用程序配置您的自定义客户端控件。

导入 HTML5 小组件

您可以导入独立和托管的应用程序的小组件。文件格式为自定义小组件软件包 (.cwp)，其中包括 HTML5、CSS 和 Javascript 文件。

导入 HTML 小组件

1. 启动 WindowMaker。
2. 在文件菜单上，选择导入，突出显示可视化，然后选择 **HTML5 小组件**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此时出现导入 HTML5 小组件对话框。

3. 选择包含要导入窗口的应用程序所在的文件夹。
4. 选择确定。

   小组件将出现在工具箱中。

导入小组件后

1. 创建图形。
2. 编辑图形并嵌入小组件。
3. 设置“小组件属性”部分下的属性。每个小组件都有自己的一组属性。
4. 将小组件插入到窗口。

现在可以在 WindowViewer 和任意 Web 浏览器上查看小组件。根据在设计时设置的属性，您可以在运行时操纵小组件。不支持那些使用“小组件属性”下的“自定义属性”来修改小组件的脚本。

将脚本函数库导入 InTouch 应用程序

您可以将脚本函数库导入 InTouch 应用程序。可以导入不同类型的脚本函数库，包括.NET（*.dll 和其它 .NET 文件扩展名）、脚本库文件（*.aaSLIB）和 InTouch 脚本扩展文件（*.wdf）。

在导出某个应用程序以创建其他应用程序时，将自动包括已导入到该应用程序的脚本函数库。该脚本函数库在发布导入它的应用程序时也可用。

将脚本函数库导入应用程序

1. 打开要导入脚本函数库的 InTouch 应用程序。
2. 选择 WindowMaker 主菜单上的文件，然后选择导入，再选择脚本。

   此时会打开导入脚本函数库对话框。
3. 浏览到要导入的函数库。
4. 选择要导入的文件并选择打开以开始导入脚本函数库。

注意：导入期间不会显示进度条或进度信息窗口。导入成功完成后会打开信息窗口。

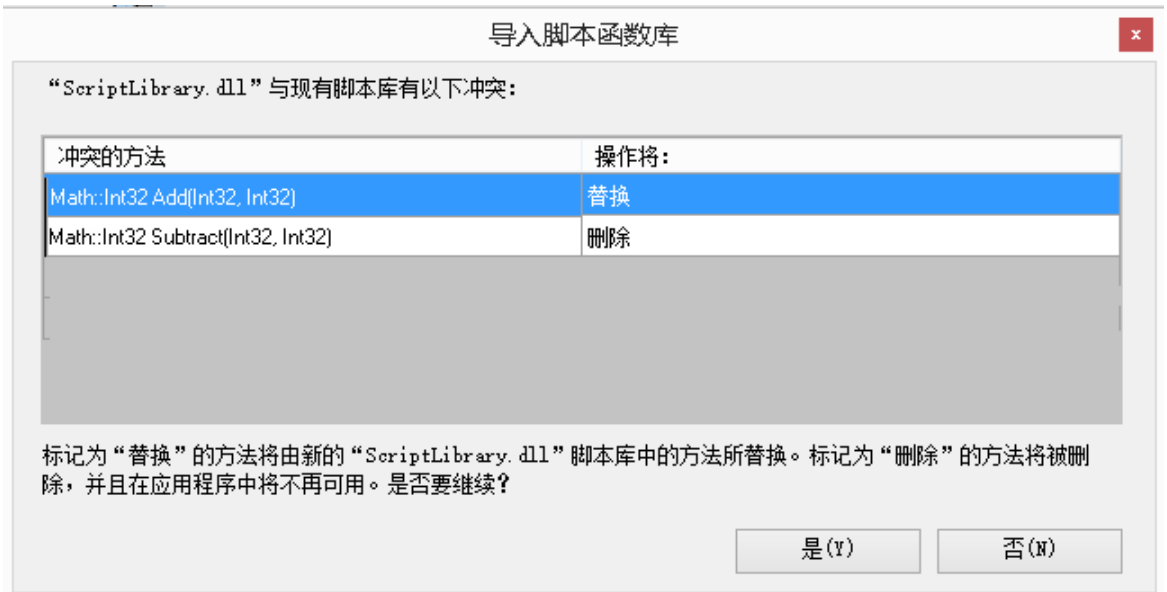
解决 .NET 脚本库中冲突方法的导入问题

将 .NET 类脚本库导入应用程序时，现有脚本库将被导入的库所替换。同时还会检测冲突的脚本方法。冲突检测是基于名称空间、分类名称、方法名称和参数声明进行的。

注意：版本、文件名或任何 .dll 文件均不会影响方法冲突检测。

导入时，冲突的方法会显示在导入脚本函数库对话框中：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



在此示例中，`Math::Int32 Add(Int32, Int32)` 存在于当前库中，并且包含的分类、方法名称和参数与导入库中的方法相同。在“操作将”列中会为其标记“替换”。如果继续导入，就会将应用程序中的整个脚本库替换为导入库。

`Math::132 Subtract(Int32, Int32)` 会标记“删除”，因为导入库中不包含 `subtract` 方法。解决脚本方法冲突需要替换整个脚本库，所以如果导入库中没有此方法，也就会导致删除此方法。

如以上示例中所示，即使导入会删除库中的某个现有方法，也无法单独取消导入某个方法。只能继续导入所有冲突的方法，或者取消整个导入。

重要事项：导入时，仅 .NET 类库文件能被检测为重复项。.aaSLIB 库和 .wdf 脚本扩展文件如果与现有库中的方法冲突，就无法导入。在此情况下，不会发出冲突通知。

配置应用程序的应用程序样式库

您可以为 InTouch 应用程序中的图形配置样式库。您可以配置质量和状态应用程序样式、元素样式以及数字格式样式。配置更改将保存到应用程序的储备库中。

- “质量和状态”指示器是一组图形图标，表示应用程序数据的当前质量以及应用程序符号所显示的设备状态。
- “元素样式”定义了一组视觉属性，用于确定工业图形中显示的文本、线条、图形轮廓以及内部填充的外观。
- “格式样式”提供的选项可用于为工业图形中使用的常见数字类型分别配置应用程序范围的样式。

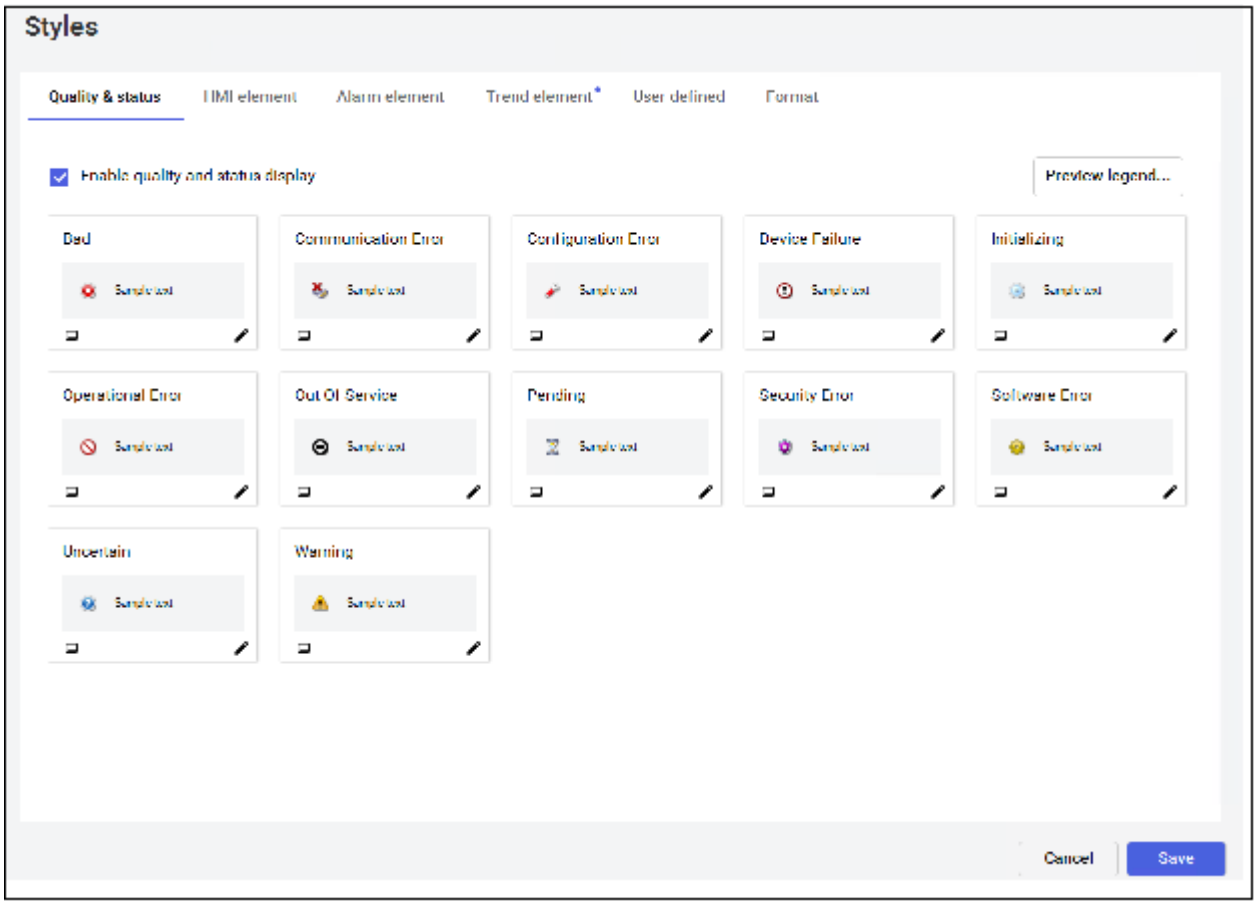
重要事项：本节介绍 WindowMaker 内用于访问应用程序样式库的工作流程。如需有关编辑应用程序样式的详细信息，请参阅 WindowMaker 联机帮助或《*工业图形编辑器用户指南*》文件。

配置应用程序样式库

1. 在 WindowMaker 中打开一个应用程序。
2. 在文件菜单上，选择配置，然后选择样式。

   此时出现样式配置屏幕。使用其中的选项卡，可以配置质量和状态指示器、图形元素样式以及数字格式样式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 选择要编辑的应用程序样式的选项卡。

注意：WindowViewer 一次只能运行一个应用程序。如果在本地节点上部署平台，则 Galaxy 的已配置样式将优先于任何其它独立或托管应用程序中的任何已配置样式。

导出和导入应用程序样式库

您可以从应用程序导出应用程序样式库，然后将其导入其它应用程序。质量设置、元素样式和数字格式将导出到 XML 文件。

从应用程序导出应用程序样式库

1. 打开 WindowMaker。
2. 在文件菜单上，选择导出，然后选择样式。

   此时出现导出应用程序样式库文件浏览器屏幕，其中的字段用于指定文件名。
3. 选择要放置导出的 XML 文件的文件夹以及文件的名称。
4. 选择保存。

   一个对话框确认应用程序样式库已成功导出。

将应用程序样式库导入应用程序

1. 打开 WindowMaker。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 在文件菜单上，选择导入，然后选择样式。

   此时出现导入应用程序样式库文件浏览器屏幕，其中的字段用于指定文件名。

3. 导航到 C:\Program Files (x86)\ArchestrA\Framework\Bin\AdditionalElementStyles。选择导出的 XML 文件所在的文件夹，然后选中文件以在文件名字段中显示导出文件的名称。

4. 选择打开。

   一个对话框确认应用程序样式库已成功导入。

5. 单击确定。

   在您重新启动 WindowMaker 之前，WindowMaker 中的图形仍将被禁用。

配置应用程序的报警优先级映射

您可以配置 InTouch 应用程序的报警优先级映射，以设置每个报警严重程度级别的优先级范围。

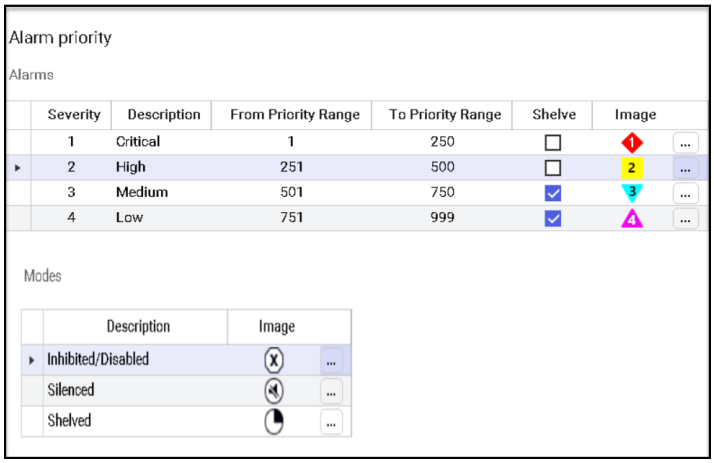
重要事项：本节介绍 WindowMaker 内用于将报警优先级范围映射到报警严重程度的工作流程。尽管 InTouch 并不像 Application Server 一样有内置的“报警严重程度”管理，但用户仍然可以使用 InTouch 标记来实现“报警边框”动画。在这种情况下，对话框中的严重程度映射的优先级仅用作可视化辅助，以关联报警边框颜色和报警指示器图标的优先级。如需有关配置报警优先级映射和报警搁置的详细信息，请参阅 WindowMaker 联机帮助或*《工业图形编辑器用户文档》*。

配置应用程序的报警优先级映射

1. 在 WindowMaker 中打开一个应用程序。

2. 在文件菜单上，选择配置，然后选择报警。

   此时出现报警优先级部分。使用其中的字段，可以将优先级范围映射到各个报警严重程度。此屏幕还包含用于根据报警严重程度启用报警搁置的字段。



3. 在优先级起始范围和优先级终止范围字段中，选择并输入介于 1 到 999 之间的数字，以设置各个报警严重程度的报警优先级范围的上下边界。

   各个优先级范围应该是连续的，优先级范围之间不应重叠。缺省条件下，报警严重程度 1 从优先级 1 开始。

4. 在搁置列中，选中或清除用于针对各个报警严重程度启用报警搁置的复选框。

5. 选择确定以保存更改。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

所做更改保存到应用程序的应用程序文件夹中。

从应用程序中导出工业图形文本字符串

如果您的应用程序支持运行时语言切换，则可以将其工业图形的文本字符串导出到字典文件。然后可以使用文本编辑器、XML 编辑器或者 Microsoft Excel 或语言助理这样的电子表格程序将字典文件内的字符串翻译成其它语言。

当导出图形文本字符串时，必须指定字典文件的输出文件夹。最佳做法是为每个其字符串将被翻译成其它语言的字典文件创建一个单独的文件夹。

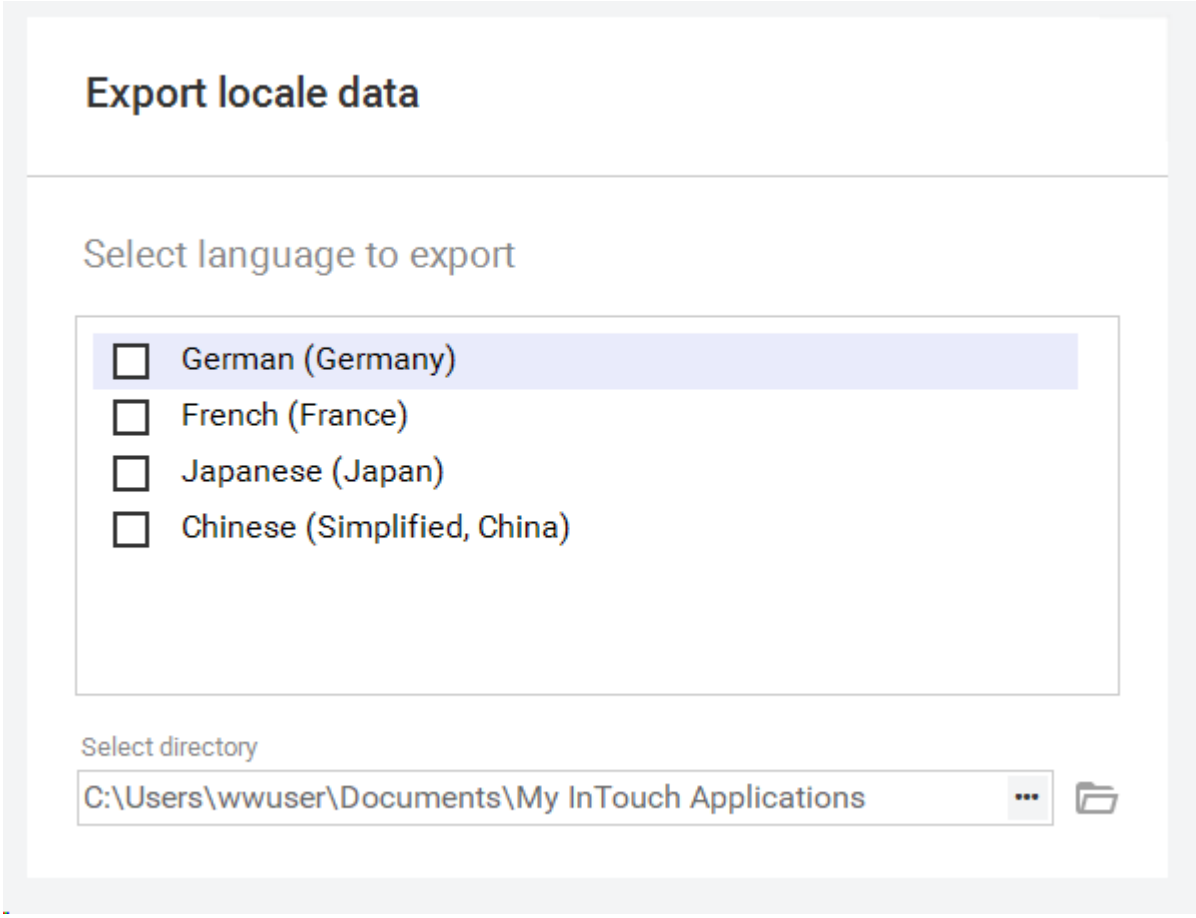
所有导出的字典文件均遵守命名惯例：<AppFolderName>AA_<LanguageID>.xml. 例如，如果应用程序文件夹名为 PumpStation，导出的语言为法语（语言 ID = 1036），则文件名为 PumpStationAA_1036.xml。

如果要在不同的时间导出不同对象的语言字符串，请使用单独的目标文件夹以防止后续的导出覆盖最前面的导出。

导出工业图形文本字符串

1. 在 WindowMaker 中打开应用程序。
2. 在文件菜单上，选择导出，选择本地化，然后选择工业图形翻译。

   此时出现导出语言环境数据屏幕。



3. 配置要导出的符号文本字符串。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 在要导出的语言列表中，选中要导出的语言字典旁边的复选框。缺省语言未列出。
- 在选择目录字段中，输入要将字典文件导出到的文件夹。

  您也可以浏览来选择现有的文件夹或创建一个新文件夹。

4. 选定导出。

将工业图形的文本字符串导入到应用程序

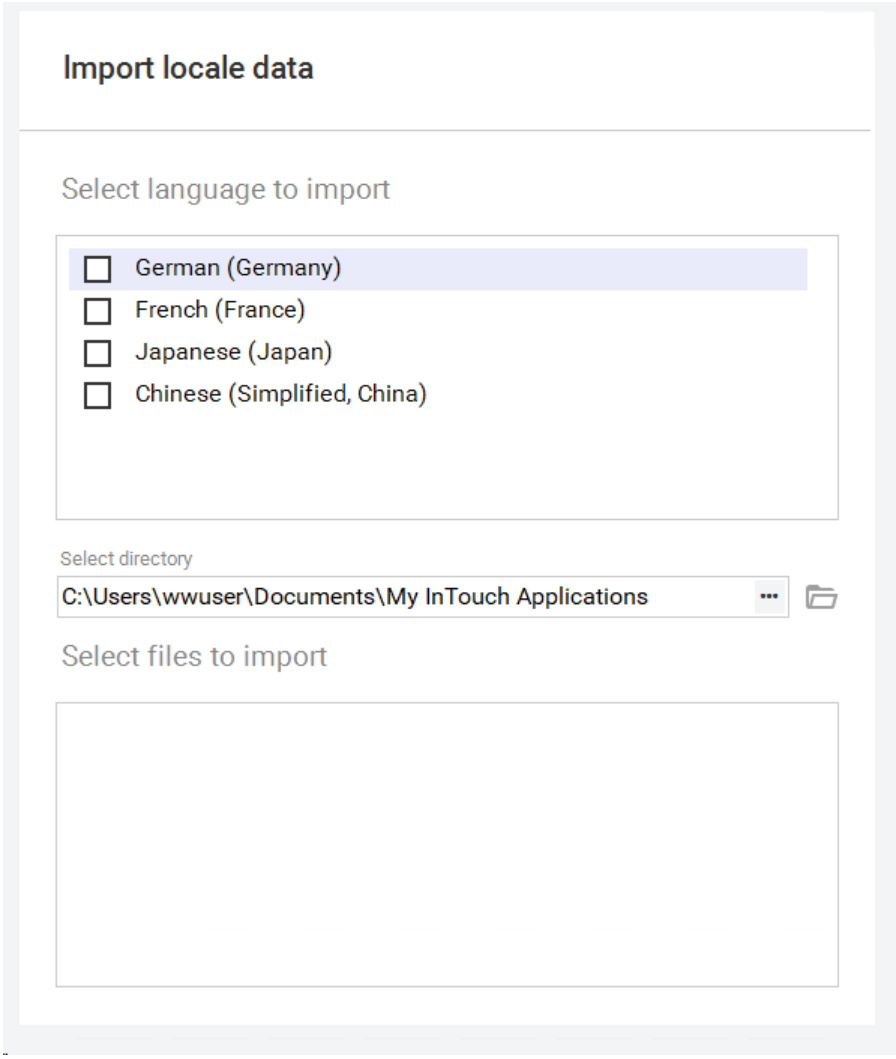
对于符号文本，必须为每种语言导入翻译的字典文件，以便在运行时可以切换这些语言。给定语言的所有字典文件都应该放置在同一文件夹中。

一次只能导入一种语言的文件。导入时，选择所需语言并指定要导入的字典文件。

导入翻译的字典文件

1. 打开要导入工业图形文本的应用程序。
1. 在文件菜单上，选择导入，选择本地化，然后选择工业图形翻译。

   此时出现导入语言环境数据屏幕。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1. 配置导入设置。
   - 在要导入的语言列表中，选中要导入的语言字典旁边的复选框。
   - 在选择目录框中，指定包含要导入的字典文件的文件夹。
   - 在选择要导入的文件框中，选择要导入的 .xml 文件。只有包含当前应用程序文件夹名和所选语言的语言环境 ID 的文件会显示出来。
2. 选择导入。

从符号导出本地化字符串

如果应用程序支持运行时语言切换，则可以导出从“工业图形工具箱”选择的一个或多个符号的文本字符串。然后可以使用文本编辑器、XML 编辑器或 Microsoft Excel 这样的电子表格程序将文件内的导出字符串翻译成其它语言。

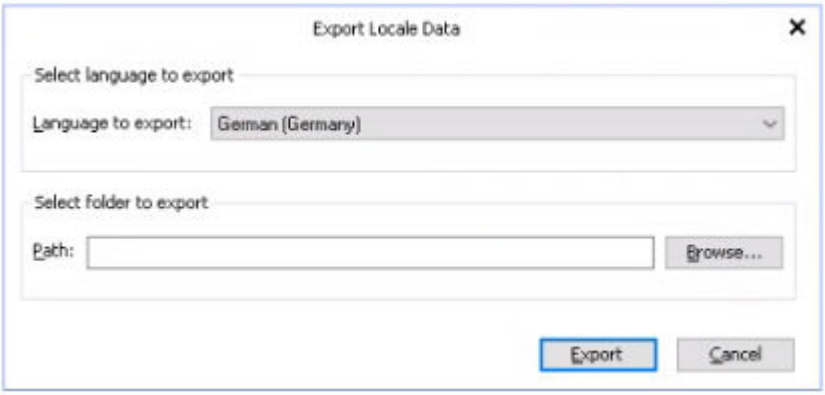
当从符号导出文本字符串时，必须指定输出文件夹。最佳做法是为每个其字符串将被翻译成其它语言的文件创建一个单独的文件夹。

所有导出的本地化文件均遵守命名惯例 : <AppFolderName>AA_<LanguageID>.xml. 例如，如果应用程序文件夹名为 PumpStation1，导出的本地化字符串的语言为墨西哥西班牙语（语言 ID = 2058），则文件名为 PumpStation1AA_2058.xml。

从符号导出本地化字符串

1. 在 WindowMaker 中打开应用程序。
2. 从“工业图形工具箱”中选择要导出其本地化字符串的符号。
   - 选择符号名称可选择单个符号。
   - 按 Ctrl 键并左键单击符号名称可选择两个或更多符号。
   - 左键单击符号名称，然后按 Shift 键并左键单击其它符号名称可选择两个所选符号之间的所有符号。
3. 使用鼠标右键单击所选符号以显示快捷菜单。
4. 选择导出，然后选择本地化，最后选择所选符号。

   此时出现导出语言环境数据对话框。



5. 配置要导出的符号文本字符串。
   - 在要导出的语言列表中，选择要从符号导出的本地化字符串。缺省语言未列出。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 在路径字段中，输入要将本地化字符串导出到的文件夹。选择浏览以选择现有的文件夹或创建一个新文件夹。

6. 选定导出。进度条将显示导出操作的进度。
7. 选择查看详细信息并验证每个所选符号内的本地化字符串均已成功导出。

导入工业图形库

在应用程序开发期间，您可以将工业图形库和 Situational Awareness Library 导入到独立应用程序，如果

- 应用程序使用空模板创建，不包含工业图形库或 Situational Awareness Library
- 旧版独立应用程序或新型应用程序已迁移，但库未导入

要将工业图形库导入到应用程序：

- 在“工业图形工具箱”中，右键单击应用程序名称，然后选择导入工业图形库。

  此时出现“导入工业图形”对话框。工业图形库先导入，随后导入 Situational Awareness Library。

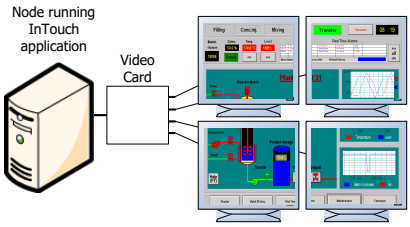
  完成后，工业图形库和 Situational Awareness Library 出现在“工业图形工具箱”中。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 28 设置多监视器系统

多监视器系统在多个监视器上同时显示 InTouch 应用程序。同时，多监视器配置合在一起创建一个复合屏幕，由连接到运行 InTouch 应用程序的计算机的所有监视器组成。每个监视器可以显示屏幕的一部分，或是只显示单个窗口组件（如数字小键盘）。

运行 InTouch 应用程序时，可以在监视器之间移动鼠标，以及将窗口从一个监视器拖到另一个监视器。同样，在某些多监视器配置中，可以在所有的监视器上显示整个 InTouch 应用程序窗口，具体如下图所示。



多监视器配置

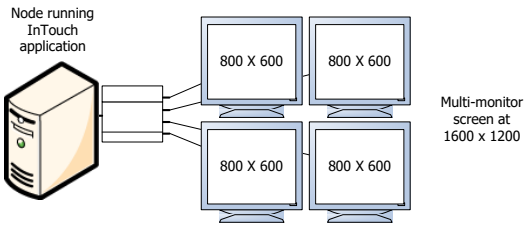
您可以使用两种基本的监视器配置。

- 单显卡
- 多显卡

每种配置都有特有的硬件、软件以及配置要求。同样，每种配置都支持不同的多监视器功能集。

单显卡配置

在单显卡配置中，计算机上安装单个显卡，有多个连接到监视器的输出端口。



复合屏幕分辨率是每个监视器的水平与垂直分辨率之和。例如，一个普通显卡连接四个 17 寸监视器，像立方体一样堆起；两个在底部，两个在顶部。在上图中，每个监视器都按 800 x 600 像素的分辨率来运行。此时复合虚拟屏幕分辨率为 1600 x 1200 像素。

单显卡配置的特征

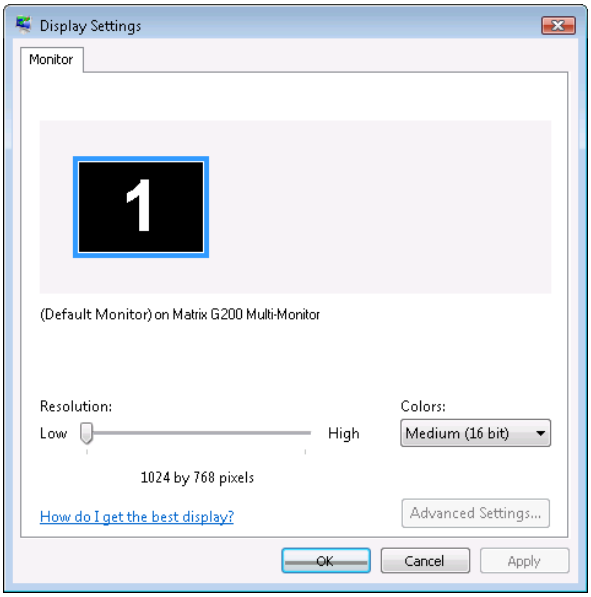
单显卡驱动程序有以下特征：

- 单个显卡同时驱动所有的监视器，以形成一个大屏幕。
- 连接的所有监视器的属性都可以使用一组屏幕值来配置。
- 在配置的尾行，复合屏幕在所有的监视器上显示 Windows 任务栏。
- Windows 应用程序可以最大化，以填满所有的监视器。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

单显卡驱动程序的特征

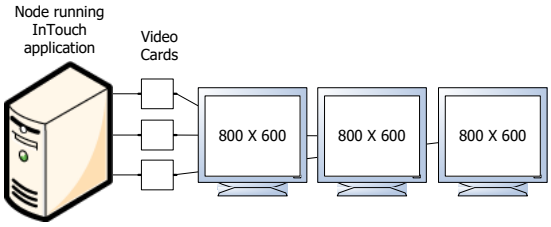
下图显示 Windows 显示设置对话框，为连接到单个显卡（带多输出端口）的所有监视器配置驱动程序。



在此图中，分辨率设置是四个监视器并排放置成一行的分辨率。每个监视器的分辨率都是 1024 x 768。这些分辨率加到一起，复合屏幕分辨率为 4096 x 768。您只需配置单个监视器的分辨率、颜色深度以及刷新速率。分辨率设置应用于连接到单个显卡的所有监视器。

多显卡配置

在多显卡配置中，计算机上安装多个显卡。每个显卡将一个监视器连接到运行 InTouch 应用程序的计算机。



多显卡配置的特征

“动态分辨率转换”(DRC) 同发布的其它功能一起，可以消除屏幕分辨率的限制。在 NAD 架构中，您在一个开发节点上创建并维护 InTouch 应用程序，然后将它复制到多个 View 节点。DRC 使所有 view 节点都可以显示应用程序，即使是以不同的屏幕分辨率运行的节点也是如此。

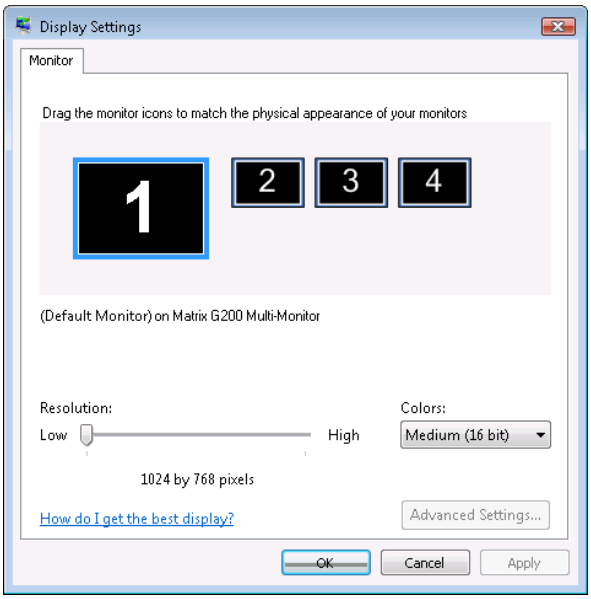
DRC 使每个 View 节点根据多个用户定义的选项（包括自定义分辨率）来缩放应用程序。这种比例调整可以在 WindowViewer 编译应用程序时进行，并不要求使用 WindowViewer。由于每个 View 节点都可以使用不同的 DRC 设置，因此必须配置每个单独的 View 节点。

DRC 可以轻松支持多监视器系统。只需从 DRC 分辨率转换选项中进行选择，便可以在整个复合屏幕或其中一部分上显示 InTouch 应用程序。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

多显卡驱动程序的特征

下图显示 Windows 显示设置对话框，对于连接到计算机（运行 InTouch 应用程序）上安装的各个单独显卡的所有监视器，可以为其配置驱动程序。



选择显示设置对话框中带编号的矩形，以选择要配置的监视器。您可以通过对带编号的矩形进行排列，使之符合监视器的实际位置。屏幕分辨率、颜色深度以及刷新速率仅应用于您选择的监视器。

规划多监视器应用程序

要为应用程序设置多个监视器

- 选择多监视器显卡
- 确定应用程序屏幕分辨率
- 确定用于显示应用程序的监视器数
- 确定应用程序窗口的位置

选择多监视器显卡

“技术支持”可以为您提供一份建议的显卡列表，这些显卡支持多监视器 InTouch 应用程序。

在选择显卡之前，可以从“技术支持”获取详细信息以回答以下问题：

- 显卡支持哪些 InTouch 版本？
- 显卡是支持单显卡还是多显卡配置？
- 显卡的建议驱动程序是什么？
- 显卡的建议配置设置是什么？

确定应用程序屏幕分辨率

通过确定总屏幕分辨率并了解查看区域的准确尺寸，可以简化为多监视器环境创建应用程序的过程。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

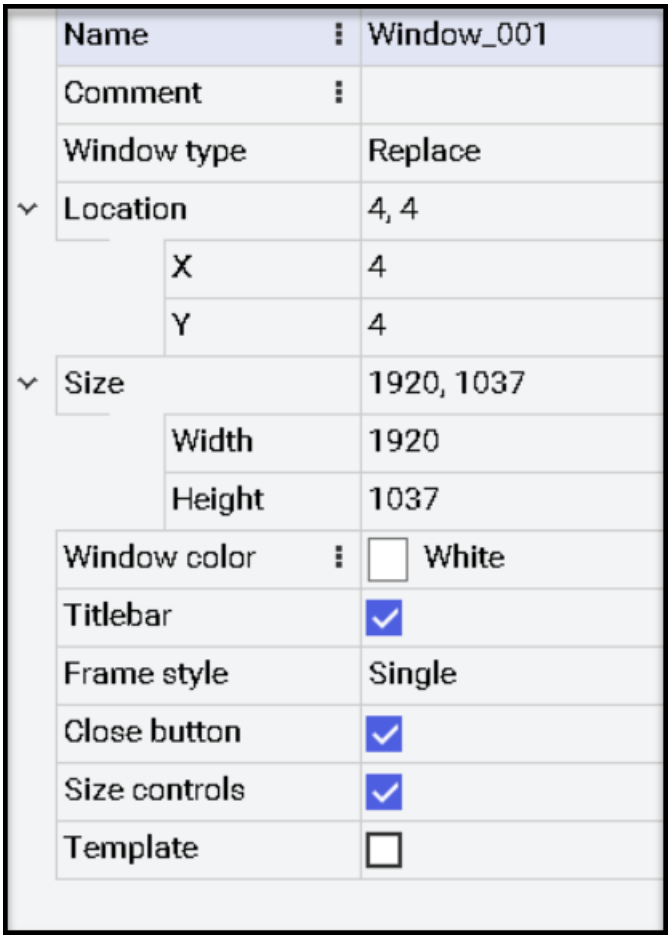
创建一个显示总监视器配置的图形。图形应该显示每个监视器的分辨率，以及所有监视器加到一起的组合分辨率。此图形帮助您以图形化方式显示每个监视器的水平与垂直像素范围。

例如，如果有一个由两台水平监视器组成的复合屏幕，每个监视器的屏幕分辨率为 800 x 600，则第二台监视器的左上角像素位置为 800 x 0 像素。第一台监视器的屏幕像素数是从 0 到 799，第二台监视器的屏幕像素数是从 800 到 1599。通过将该图形用作基准，可以确定应用程序窗口在复合多监视器屏幕上的位置。

确定用于显示应用程序的监视器数

通过使用类似于生产环境的开发环境，可以简化创建多监视器 InTouch 应用程序的工作。使用多监视器开发环境不是在所有情况下都有可能。仅有一个监视器连接到用于开发 InTouch 应用程序的计算机时，通过开发窗口并根据估计的显示需要来配置窗口的尺寸与位置，仍然可以构建多监视器应用程序。

使用 WindowMaker 属性窗格修改窗口的特征。在窗口窗格中，选择要修改的窗口。窗口属性窗格显示在右侧导航窗格中。



X 位置与 **Y 位置值**确定窗口在屏幕左上角的水平与垂直像素位置。水平与垂直像素刻度的原点位于屏幕左上角。

窗口宽度与**窗口高度设**置确定窗口的总体尺寸。例如，您可以使用以下设置配置窗口：

- X 位置 = 1024
- Y 位置 = 0
- 窗口宽度 = 1024

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 窗口高度 = 768

多监视器配置由水平放置成一排的四个监视器组成。每个监视器的分辨率都是 1024 X 768。总的复合屏幕分辨率为 4096 X 768。

通过将窗口的水平原点设置为 1024、垂直原点设置为 0，可以在运行时强制此窗口出现在第二个监视器上。此窗口占据第二个监视器的整个屏幕表面。

确定应用程序窗口的位置

为多监视器环境开发 InTouch 窗口时，可以使用多种不同的配置。

显示在强制位置的窗口

一种方法是，只需开发窗口并强制它显示在指定的位置。确保 WindowViewer 最大化在所有监视器的总查看区域上。这允许在指定的显示器上显示 InTouch 应用程序窗口。

您可以使用 InTouch 安全性功能来拒绝对 Windows 桌面的访问。

可以手工移动窗口

另一个选项是开发一个应用程序，将其中的窗口手工移到所选的监视器，使单个应用程序在不同的监视器配置上运行。这涉及到以下要求：

- 应用程序中的所有窗口必须都是弹出型。
- WindowViewer 主窗口可以小一点，不占据所有的监视器。不过，由于 InTouch 没有最大化，因此不能在此配置中使用 InTouch 安全性来拒绝访问 Windows 桌面。

在此配置中使用的是弹出窗口，可以轻松地移到任何监视器，而不论 WindowViewer 主窗口的位置如何。弹出窗口不必保持在 WindowViewer 父窗口中。您可以缩放主窗口的大小并将它移到监视器的一角，使所有弹出窗口都可以自由移到所选的监视器上。

窗口根据环境自动放置

最后一种方法包含以上方法之外的一个额外步骤。此步骤使应用程序可以根据使用的环境自动放置窗口。这是配置中最复杂的部分，要求进行大量的脚本编写与规划工作。

在此配置中，ShowAt() 与 ShowTopLeftAt() 脚本函数根据一组缺省的坐标与算法来动态地放置窗口。这可以使用许多不同的方法进行配置，具体取决于应用程序要求。

开发多监视器 InTouch 应用程序

您必须给 InTouch.ini 与 Win.ini 文件中所选的参数指定数值，才能支持多监视器。这些参数使您可以在复合屏幕中适当的位置上放置 InTouch 系统对话框与数字小键盘。

配置多监视器参数

要启用多监视器支持，需要将一组 InTouch 参数添加到 Windows 的 Win.ini 文件。这些参数可以为运行 InTouch 应用程序的节点与每个监视器的分辨率启用多监视器支持。

在节点上配置多监视器设置

1. 在运行 InTouch HMI 软件的计算机上，编辑 Windows 文件夹中的 Win.ini 文件。
2. 找到 Win.ini 文件中的 [InTouch] 部分，并添加以下参数：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数	描述
MultiScreen=1	值为 1 时启用多监视器模式。值为 0 时禁用多监视器模式。
MultiScreenWidth=*nnnn*	以像素为单位的单屏幕宽度。
MultiScreenHeight=*nnnn*	以像素为单位的单屏幕高度。

例如，如果希望在两个水平监视器上以 2560 x 1024 的屏幕分辨率显示 InTouch 应用程序，请输入以下内容：

```
[InTouch]
MultiScreen=1
MultiScreenWidth=1280
MultiScreenHeight=1024
```

配置屏幕分辨率转换

您可以指定一个参数值，在运行不同屏幕分辨率的节点之间迁移 InTouch 应用程序窗口时可以维持当前的分辨率。

ScaleForResolution 参数值确定运行 WindowViewer 的计算机上的分辨率发生更改之后，WindowMaker 是否自动缩放应用程序窗口 (*.win)。ScaleForResolution 参数不影响 WindowMaker 对话框的分辨率。

在节点上配置屏幕分辨率转换

1. 编辑运行 InTouch 的计算机上的 .ini 文件。
2. 将 ScaleForResolution 参数添加到文件中。

   ```
   ScaleForResolution=1
   ```

   设置为 0 时，禁用分辨率转换。

   设置为 1 时，启用分辨率转换。

注意：如果 ScaleForResolution 参数没有添加到 InTouch.ini 文件，缺省值是启用 (ScaleForResolution=1)。禁用参数 (ScaleForResolution=0) 时，程序仍然会提示您是否转换分辨率。但并不会发生分辨率转换。

部署应用程序与验证多监视器设置

在单监视器系统开发打算在多监视器系统上运行的应用程序时，ScaleForResolution 参数变得尤其重要。指定给 ScaleForResolution 参数的值确定将应用程序从一个环境移到另一个环境时，是否对其进行缩放。

重要事项：在将应用程序移到不同的环境之前，建议先对它进行备份。

例如，如果应用程序在使用单监视器（分辨率为 1024 x 768）的计算机上开发，并且打算在使用四个并排放置的监视器（总分辨率为 4096 x 768）的系统上运行，这要求进行应用程序转换。

在多监视器系统上部署应用程序时，会出现一条消息，提示您是否转换应用程序。

如果配置了 ScaleForResolution .ini 设置，仍然会看见这条消息，但应用程序不转换，而且随后可以按设计来运行。选择是继续启动。

如果 .ini 设置尚未配置，则 InTouch HMI 会将应用程序中所有图形与窗口转换并缩放为新的分辨率。执行此操作会拉伸并放大所有窗口与图形的显示画面，因此会造成某些不希望的结果。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

重要事项：在运行应用程序之前，确保目标计算机上也配置了多监视器 Win.ini 参数设置。Win.ini 设置不随 InTouch 应用程序自动传输。

在运行时验证多监视器支持

您可以从“技术支持”脚本库中下载一个可选脚本函数，此函数可以验证运行 InTouch 应用程序的本地节点是否提供多监视器支持。

WWMultiMonitorNode() 函数确定节点是否支持多监视器以及连接到该节点的监视器数。

通常，您可以从 QuickScript 中运行 WWMultiMonitorNode() 函数，以确定指定给运行 InTouch 应用程序的节点的监视器数。

下例显示一个 QuickScript 语句示例，它将 WWMultiMonitorNode() 函数的值指定给 InTouch 整型标记。QuickScript 可以设置为在 WindowViewer 中启动应用程序时运行。

```
{MultiMonitors 定义为整型标记}
MultiMonitors = WWMultiMonitorNode();
{执行此函数后，结果 = 4}
```

WWMultiMonitorNode() 读取节点的 Win.ini 文件中指定的 MultiScreen 参数。WWMultiMonitorNode() 函数返回 0 或一个正整数。

- 返回值 0

  如果 MultiScreen=0 或 Win.ini 文件 [InTouch] 部分中 MultiScreenWidth 或 MultiScreenHeight 参数错误地设置为 0，则 WWMultiMonitorNode() 返回 0。

- 正整数返回值

  如果 MultiScreen=1 并且给 MultiScreenWidth 与 MultiScreenHeight 参数指定了正确的屏幕分辨率值，则 WWMultiMonitorNode() 返回多监视器配置中的监视器数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 29 在 Tablet PC 上使用 InTouch

一系列便携式 Tablet PC 上预安装了 Windows XP Tablet PC Edition 和 InTouch。这些牢固的 Tablet PC 防水抗震，能够适应大多数工业环境。Tablet PC 也可以在能运行 InTouch 应用程序的其它计算机制造商处购买。

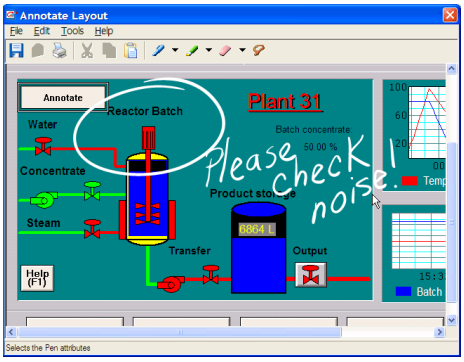
操作员可以携带着 Tablet PC 巡视工厂。Tablet PC 运行代表实际工厂过程的 InTouch 应用程序。通过将笔用作屏幕指针或输入设备，操作员可以在屏幕上选择 InTouch 对象，或用它代替键盘直接在屏幕上书写备注。

操作员可以书写备注，也可以根据对实际工厂过程的直接观察给正在运行的 InTouch 应用程序作出注解。

注解可视化屏幕与将它当作电子邮件消息发送

使用 AnnotateLayout() 脚本函数截取 Tablet PC 上显示的屏幕。仅当 InTouch 运行在使用 Windows XP Tablet PC 操作系统的 Tablet PC 上时，AnnotateLayout() 函数才可以使用。

AnnotateLayout() 函数截取活动 InTouch 窗口可见部分的屏幕截图。捕获的屏幕出现在注解布局对话框中。



注解布局对话框包含工具栏与菜单选项。对话框在其客户端区域显示屏幕截图。您可以使用各种绘图工具作出注解，然后保存、打印，或将屏幕截图作为电子邮件消息发送。

给窗口添加注解

使用以下工具给窗口添加注解

- 笔：绘制和书写注释。

- 荧光笔：使用半透明颜色突出显示窗口中的区域。

- 橡皮擦：删除注解的某些部分。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

每种工具都有特定的选项，如大小、颜色或透明度。

- 要设置这些选项，选择每种工具图标旁的向下箭头，然后选择该选项的命令。
- 要将这些选项恢复成它们的缺省设置，在工具菜单上选择恢复缺省值。

选择、复制及删除窗口注解

您可以选择、复制及删除希望在窗口中作出的注解。

选择注解

1. 选择工具栏中的套索工具图标。
2. 按下笔触按钮期间，在您希望选择的注解周围绘制一个区域。
   您现在可以剪切、复制或删除所选的注解。

剪切、复制及粘贴注解

- 使用 Windows 的标准“剪切”、“复制”及“粘贴”命令。

删除注释

- 执行以下任意操作：
  - 要删除窗口上的所有注解，在编辑菜单上，指向清除，然后选择全部。
  - 要删除使用选择工具选择的注解，请在编辑菜单上，指向清除，然后点击选择。

保存、打印及使用电子邮件发送注解的窗口

给窗口作出注解之后，可以将它保存为图像文件，打印它，或者将它当作电子邮件附件来发送。

您只需要配置电子邮件服务器一次。

保存注解的窗口

1. 在文件菜单上，选择保存。此时出现 Windows 标准的另存为对话框。
2. 输入文件的名称与格式，然后选择确定。

打印注解的窗口

1. 在文件菜单上，选择打印。此时出现 Windows 的标准“打印”对话框。
2. 指定任何打印选项，然后选择确定。

将注解的窗口作为电子邮件附件发送

1. 在编辑菜单上，选择电子邮件配置。此时出现电子邮件配置对话框。
2. 输入要用于发送电子邮件的 SMTP 电子邮件服务器的主机名。如果不确定，请向管理员寻求帮助。选择确定。
3. 在文件菜单上，选择电子邮件。此时出现电子邮件对话框。
4. 输入发送人与接收人的地址并书写消息。注解的窗口的图像文件会自动添加为附件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5. 选择发送以发送电子邮件消息。

AnnotateLayout() 函数

此时出现注解布局对话框, 您可以在调用此脚本函数的地方对当前查看屏幕作出注解。只有在 Windows XP Tablet PC Edition 操作系统上，此函数才受到支持。

类别

系统

语法

```
AnnotateLayout()
```

附注

出现注解布局对话框时，捕获 WindowViewer 的屏幕图像。使用此对话框以：

- 使用笔以及工具栏、菜单项目设置给屏幕截图作出注解。
- 将图像与注解保存为 .gif 或 .jpeg 文件。
- 打印图像与注解（如果配置了打印机）。
- 将图像与注解当作电子邮件消息的附件来发送（如果配置了 SMTP）。

更改屏幕方向

如果 Tablet PC 按平板配置运行，并且 WindowViewer 配置成将应用程序分辨率动态更改为屏幕分辨率，则在横向模式中开发的 InTouch 应用程序会进行缩放以适合纵向模式。

如果 WindowViewer 未配置成动态更改应用程序分辨率，则横向应用程序不会进行缩放。这种情况下，某些 InTouch 窗口在平板电脑上可能会被截断。

缺省情况下，从一种配置切换到另一种时，屏幕分辨率也会切换。例如，如果按笔记本配置运行的 Tablet PC 切换到平板配置，则屏幕方向也会从横向 (1024 x 768) 模式切换到纵向 (768 x 1024) 模式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 30 管理 InTouch 服务

服务是执行无人照管的特定后台系统功能的某种 Windows 进程，它没有用户界面，也不要求用户登录。

以下启动选项可供 Windows 服务使用：

- **自动**。Windows 重新启动时，服务自动启动而无需任何用户干预。
- **手动**。用户或应用程序进程必须明确启动服务。
- **禁用**。禁止服务启动。这对于疑难排解非常有用。

注意：不支持 InTouch WindowViewer 服务中的参数选项。

服务启动时不影响 Windows 安全系统。

InTouch HMI 包含以下 Windows 服务：

- Alarm DB Logger
- Alarm DB Purge-Archive
- NetDDE Helper
- SuiteLink
- WindowViewer

将 WindowViewer 作为服务运行

如果将 WindowViewer 配置成作为服务运行，WindowViewer 会在安装了该应用程序的计算机启动时自动启动。WindowViewer 服务在后台运行。如果 WindowViewer 服务正在运行，则无法启动 WindowViewer 的另一个实例。

将 WindowViewer 作为服务运行具有以下优点：

- 大多数灾难恢复计划都要求在恢复供电之后，立即启动关键的计算机系统。Microsoft Windows 服务器可以在供电恢复之后自动重新启动。WindowViewer 作为服务运行时，工厂自动化系统可以立即开始运行。计算机重新启动时，WindowViewer 中上次打开的 InTouch 应用程序自动启动。
- WindowViewer 继续记录历史数据、搜集报警消息、处理脚本、充当“I/O 服务器”、作为 I/O 客户端写入数值、甚至作为不同的操作员进行登录与注销。

  注意：如果将网络应用程序用来作为服务运行或将网络路径用作历史记录文件夹，则登录的用户必须具有对网络位置的正常访问权限。

如果 WindowViewer 已作为服务运行，并且您尝试使用快捷方式图标或通过选择 Windows **开始**菜单上的“WindowViewer”来启动 WindowViewer，则 Operations Control Log Viewer 中会记录一条消息。此消息描述在 WindowViewer 配置成作为服务运行时，它在启动方面所受到的限制。

如果 WindowViewer 已作为服务运行，并且您尝试启动“应用程序管理器”或 WindowMaker，则 Operations Control Log Viewer 中会记录一条警告消息。此消息说明当 WindowViewer 作为服务运行时，“应用程序管理器”和 WindowMaker 无法打开。

重要事项：将 WindowViewer 作为服务运行时，用户帐户权限已设置为非交互式，以减少使用管理员权限运行服务时可能发生的安全隐患。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将 WindowViewer 配置成作为服务启动

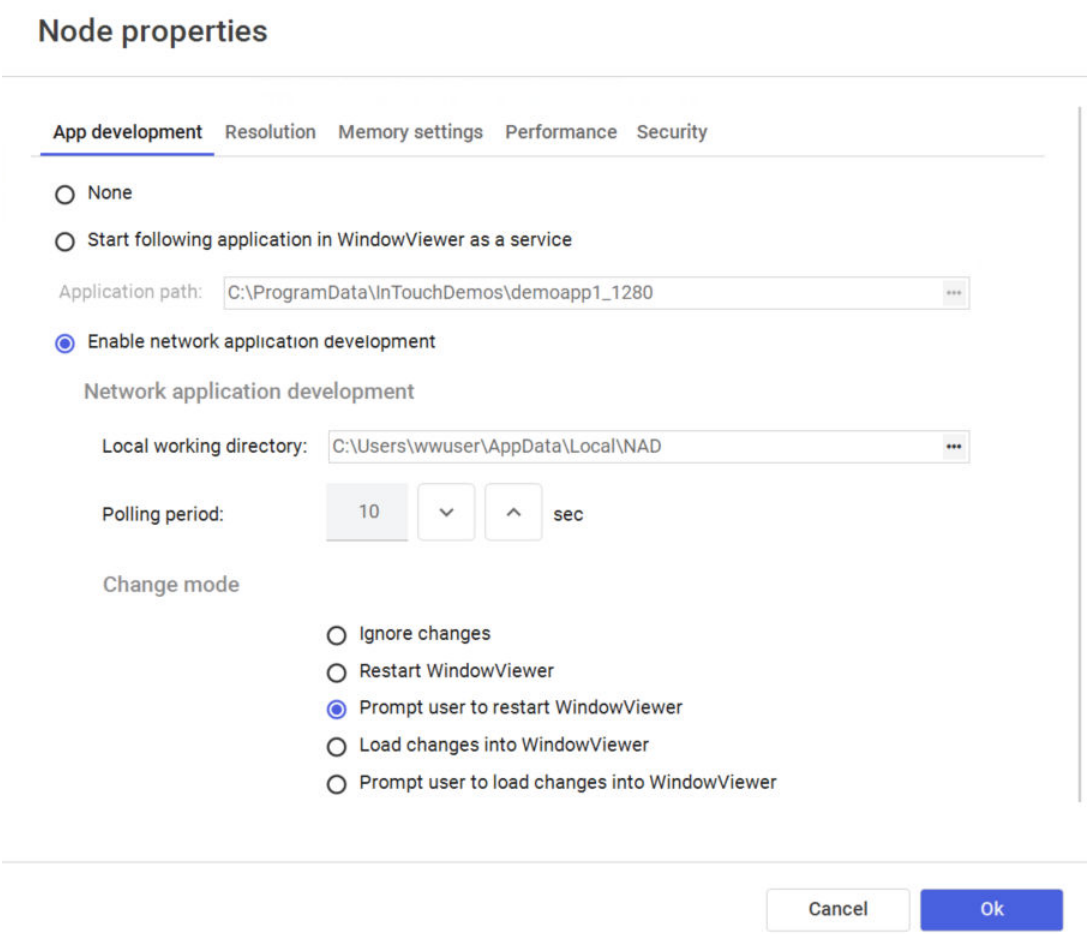
WindowViewer 作为 Windows 服务运行时，可以在操作员注销之后提供连续的操作，以及在系统启动期间自动启动，而无须操作员干预。这样便可以让无人工作站启动 WindowViewer，而不会影响操作系统的安全性。

WindowViewer 配置为作为服务启动时，还必须指定 InTouch 应用程序以在 WindowViewer 中作为服务运行。您可以在节点属性对话框中指定应用程序目录，也可以手动输入到 WIN.INI 文件中。

将 WindowViewer 配置成作为服务启动

1. 启动“InTouch 应用程序管理器”。
2. 在工具菜单上，选择节点属性。

   此时出现节点属性对话框。



3. 选择在 **WindowViewer** 中启动以下应用程序作为服务，以将 WindowViewer 配置为作为服务自动运行。

   应用程序路径字段将变为启用状态。
4. 选择省略号按钮提示文件资源管理器，然后导航到 InTouch 应用程序。

   应用程序目录将填充在组框中。
5. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 选择“应用程序管理器”工具栏中的 WindowViewer 图标。

   WindowViewer 现在对于指定的 InTouch 应用程序将作为服务运行。

注意：如果按上面的步骤所述配置了“节点属性”，还可以从 WindowMaker 快速切换到 WindowViewer，以针对 InTouch 应用程序启动 WindowViewer 服务。这样可以替代从“应用程序管理器”启动 WindowViewer。

编辑 WIN.INI 以在 WindowViewer 中将应用程序作为服务运行

如果在**节点属性**中启用了**在 WindowViewer 中启动以下应用程序作为服务**，则可以手动将应用程序目录输入到 WIN.INI 文件中。如果在“应用程序管理器”中选择应用程序之前更新了 WIN.INI 文件，WindowViewer 将针对所选的应用程序作为服务运行。

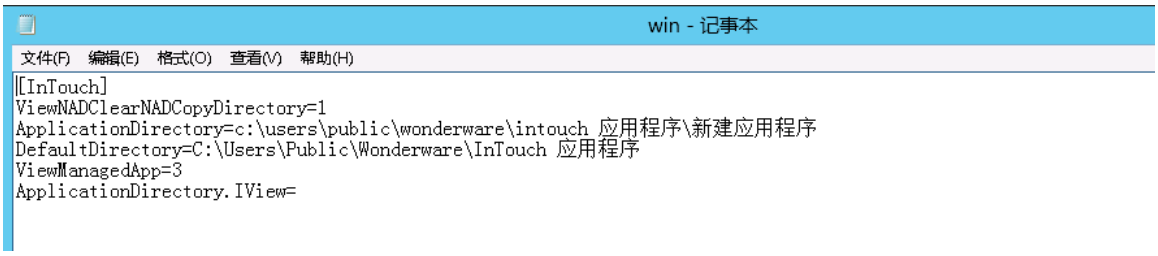
还可以将 WindowMaker 中打开的应用程序更新到 WIN.INI。如果随后快速切换到运行时，WindowViewer 将针对该应用程序作为服务运行。

注意：托管的 InTouch 应用程序不支持上述功能。如果尝试将托管的应用程序从 WindowMaker 快速切换到在 WindowViewer 中作为服务运行，将在 Logger 中记录一条警告消息。

WIN.INI 位于：

```
C:\ProgramData\Wonderware\InTouch\Service\win.ini
```

输入要作为服务运行的应用程序的目录，如下例所示：



手动启动服务

您可以使用 Windows 的“控制面板”手动启动 InTouch WindowViewer 服务。

WindowViewer 不出现在“服务”控制面板中，除非您将它配置成作为服务运行。如需有关详细信息，请参阅将 WindowViewer 配置成作为服务启动。

使用“控制面板”启动 WindowViewer 服务

1. 启动“控制面板”。
2. 双击管理工具，然后双击服务。此时出现服务对话框。
3. 在详细信息窗格中，使用鼠标右键单击 **Wonderware WindowViewer** 服务，然后选择启动。

重要事项：不能使用命令提示符将 WindowViewer 作为服务启动。

停止服务

您可以使用“控制面板”手工停止 WindowViewer 服务。

使用“控制面板”停止 WindowViewer 服务

1. 启动“控制面板”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 双击管理工具，然后双击服务。此时出现服务对话框。
3. 在详细信息面板中，使用鼠标右键单击 WindowViewer，然后选择停止。

配置 InTouch 服务的用户帐户

缺省条件下，Windows 服务使用本地系统帐户运行。InTouch 服务要求使用拥有某些管理权限的用户帐户，本地系统帐户可能没有这些权限。

安装 InTouch HMI 时，如果还没有创建帐户，则可以指定一个所有 AVEVA 服务都在其下运行的管理帐户。此帐户被视为主帐户。InTouch 服务使用该主帐户自动启动。

注意：主帐户也称为模拟帐户。模拟帐户是一个用户或用户组帐户，它提供访问您的站点中或服务器中受限资源“区域”的权限。

如果希望更改主帐户，请使用“更改网络帐户实用程序”。

警示：更改主帐户会影响所有 AVEVA 服务，不仅仅是 InTouch 服务。

更改主帐户

1. 在开始菜单上，指向程序，指向 **AVEVA**，然后选择更改网络帐户。此时出现更改网络帐户对话框。
2. 更改用户帐户。如需有关详细信息，请参阅“更改网络帐户”文档。
3. 选择确定。

InTouch 服务疑难排解

如果某项服务可以启动与否取决于其它一些服务是否已启动，则 Windows 会在启动该服务之前验证这些作为前提条件的服务是否正在运行。

根据运行 WindowViewer 的要求，注意以下相关性：

- 如果计划使用“分布式报警”或“分布式历史”，或者是您打算访问网络 DDE 数据，则 NetDDE Helper 服务必须正在运行。

  NetDDE Helper 服务也取决于 Network DDE 与 Network DDE DSDM 服务安装并配置成“手动”还是“自动”启动。安装期间，NetDDE Helper 服务配置成“手动”启动。WindowViewer 在计算机启动时自动启动此服务。

- 如果需要 WindowViewer 作为 SuiteLink 服务器或客户端运行，则 SuiteLink 服务必须正在运行。

  SuiteLink 服务还要求安装 Microsoft TCP/IP。

- 如果希望存储 WindowViewer 运行时的任何消息或错误，则必须确保安装了 Operations Control Log Viewer 服务。

  SuiteLink 与 Operations Control Log Viewer 服务都应该安装并配置为在自动启动时运行。

查看服务的错误消息

使用 Windows 的“事件查看器”来排解同服务相关的错误消息。例如，您可能会看到“一个或多个服务无法启动…”。Windows 的“事件查看器”列出在启动 Windows 服务时发生的提示性消息、警告或错误。如需有关“事件查看器”的详细信息，请参阅 Microsoft 文档。

您可以看到 InTouch 服务启动失败而导致的任何警告或错误消息。如果“事件查看器”指出 WindowViewer 服务启动失败，最有可能的原因是所依赖的必要服务不在运行。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

服务用户帐户问题的疑难排解

如果无法安装 InTouch 服务，或安装 InTouch HMI 之后无法启动，则运行它们的用户帐户可能有问题。

排解服务用户帐户问题的疑难

1. 打开 Windows 的“用户管理器”窗口，并创建一个新的主用户帐户。

   此用户帐户必须拥有本地计算机上的管理权限, 以便以服务方式启动 InTouch 组件。如果在域名列表中看不到您计算机的节点名，请手工输入节点名。

   如需有关详细信息，请参阅配置 InTouch 服务的用户帐户。

2. 确认计算机的节点名不超过 14 个字符。如果节点名包含下划线字符 (_) 或短划线 (-)，请删除它们。
3. 在安装期间，如果提示输入域名，请输入计算机的节点名而不是域名。然后输入步骤 1 中创建的用户名以及口令。
4. 如果已经安装 InTouch HMI，则仍可以通过运行 ArchestrA Change Network Account 实用程序来指定域名、用户名以及口令。
5. 重新启动计算机。
6. 使用任何有效的用户帐户登录到网络域。即便是域停机, 也不会影响在本地计算机上运行的 InTouch 应用程序。

停用提示的 I/O 项目

启动 Windows 操作系统时，配置成自动启动的服务将在“后台”启动，而不会在桌面上出现可见的用户界面。在这种情况下，服务在系统环境下运行。操作员登录到系统时，对于在系统环境下运行的任何服务，如果有关联的用户界面，则会自动出现在桌面上。在这种情况下，服务在桌面环境下运行。

如果将 WindowViewer 服务配置成自动启动，则在启动操作系统时，服务在系统环境中运行。然后，在用户登录时，WindowViewer 服务继续运行，但是会转换到桌面环境中，WindowViewer 用户界面自动出现。

如果有一些 InTouch“访问名”在定义时打开了只提示激活项选项，并且有一些 I/O 标记只在特定的 InTouch 应用程序窗口中是活动的（这些标记未在应用程序中的任何其它地方使用），则可能“停用”那些标记。例如，如果 WindowViewer 作为服务运行，您使用脚本关闭应用程序窗口，则该窗口会自动从内存中释放，从而终止与那些标记的链接。

InTouch 服务的注册表项

InTouch 服务作为注册表项列在 Windows 注册表中：

SuiteLink:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SLS
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\slssvc
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SuiteLink

NetDDE Helper:

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\WWNetDDE

WindowViewer：

- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\VIEW

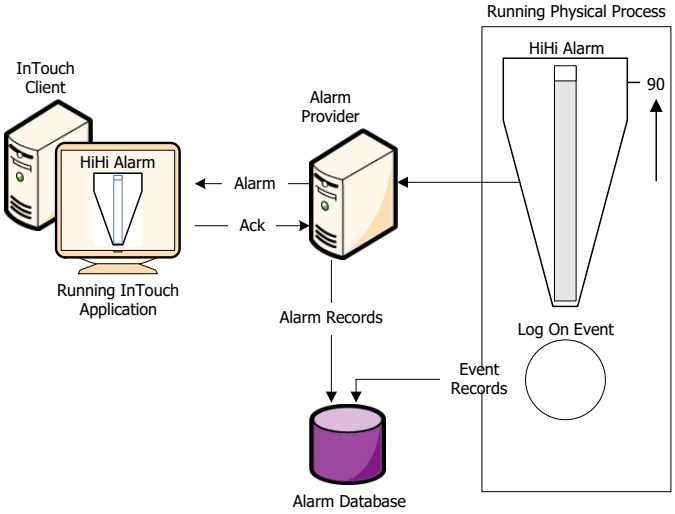
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 31 报警

通过创建可生成报警与事件的 InTouch 应用程序，可以通知操作员有关生产过程活动的状态。

- 报警向运行时操作员警告可能导致潜在问题的过程条件。通常，您设置一个在过程值超过定义的极限时触发的报警。操作员通常必须确认该报警。
- 事件代表正常的系统状态消息。通常事件在发生某种系统条件时触发，如操作员登录到 InTouch 应用程序。操作员不必确认事件。

下图显示 InTouch HMI 在应用程序运行期间如何处理报警与事件。报警与事件数据保存到报警数据库。



您可以配置任何标记来监视事件。每次标记值改变时，都有一个事件消息记录到报警系统。事件消息包含：值是如何改变的，是操作员、I/O、脚本还是系统促使了这个改变。

从旧有的报警系统迁移

您可以迁移使用“标准报警系统”或 AlarmSuite 构建的应用程序。

从标准报警系统迁移到分布式报警系统

从“标准报警系统”迁移到“分布式报警系统”时，主/从应用程序中的所有“标准报警显示”都会迁移为“分布式报警显示”对象。

颜色、字体、表达式及报警查询等设置不迁移。新的“分布式报警显示”对象具有以下缺省查询，其中 nodename 是主节点的名称：

\\节点名\InTouch!$system

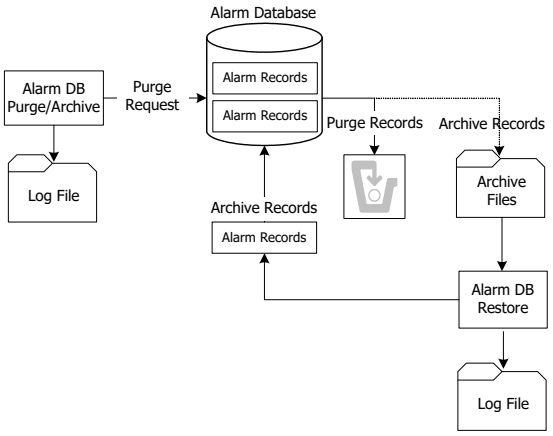
确认与报警状态点域继续像以前一样起作用。根据 I/O 标记是为 NetDDE 还是为 SuiteLink 而配置，有可能需要启用 NetDDE。不过，由于现在可以使用“分布式报警显示”对象来确认报警，因此您可能会决定不再需要使用单独的控件来发出确认。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

维护报警数据库

您可以使用两个 InTouch 实用程序来管理报警数据库。Alarm DB Purge-Archive 实用程序可用于从数据库中永久删除记录或将记录归档到文件中。如果数据库已损坏，则可以使用 Alarm DB Restore 实用程序来恢复归档的记录。

下图显示这两个实用程序如何清除/归档记录，然后再将它们恢复到数据库。

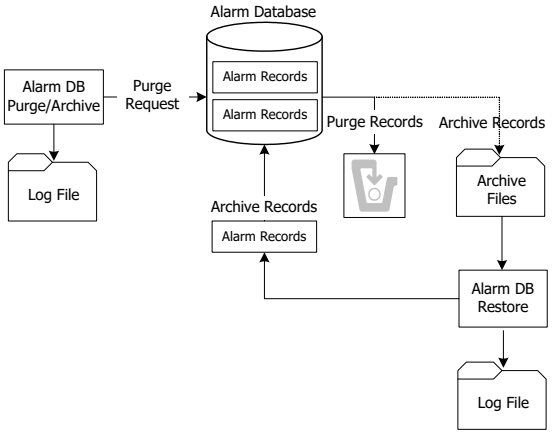


您必须作为管理员登录到计算机才能使用 Alarm DB Purge-Archive 实用程序。

关于维护报警数据库

您可以使用两个 InTouch 实用程序来管理报警数据库。Alarm DB Purge-Archive 实用程序可用于从数据库中永久删除记录或将记录归档到文件中。如果数据库已毁坏，则可以使用 Alarm DB Restore 实用程序来恢复归档的记录。

下图显示这两个实用程序如何清除/归档记录，然后再将它们恢复到数据库。



您必须作为管理员登录到计算机才能使用 Alarm DB Purge-Archive 实用程序。

配置清除或归档设置

使用 Alarm DB Purge-Archive 实用程序可以：

- 选择要从报警数据库中清除的记录的类型。
- 按每天、每星期或每月的日程自动清除记录。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 将清除的数据库记录归档到文件（可选）。
- 将归档或清除操作的状态保存到日志文件中，以便就出现的问题进行疑难排解。
- 显示清除或归档操作的状态。

重要事项：Alarm DB Purge-Archive 实用程序作为一组 Windows 服务运行。为减少以管理员权限运行 Alarm DB Purge-Archive 实用程序时的安全隐患，该用户帐户权限已设置为非交互式。

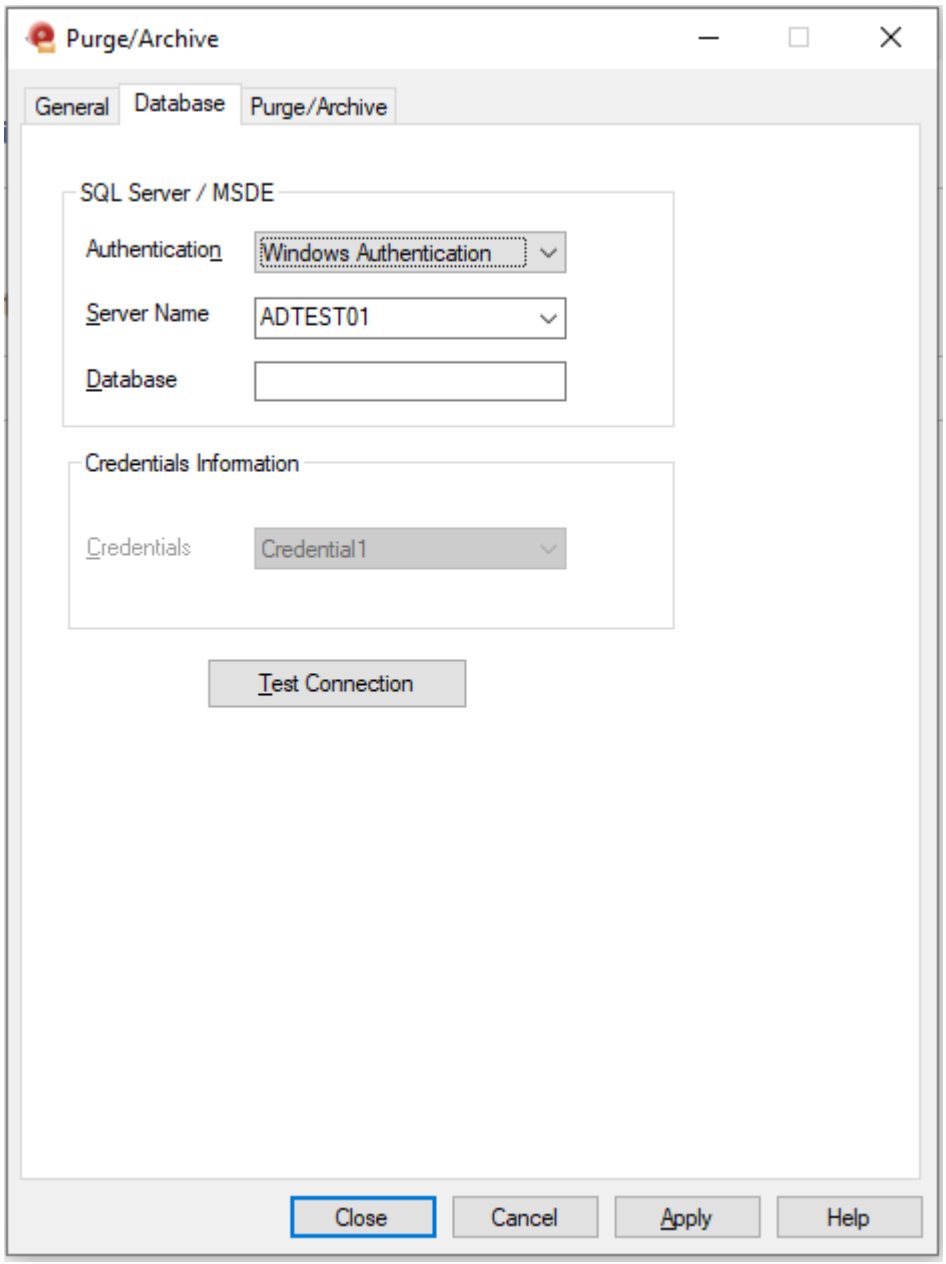
连接报警数据库

必须先连接到报警数据库，才能够使用 Alarm DB Purge-Archive 实用程序。

配置数据库连接

1. 打开 **Alarm DB Purge-Archive** 实用程序。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Purge-Archive**。
2. 选择数据库选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 配置数据**库连**接。**执**行以下操作：

   a. 在**身份验证**列表中，**选择**身份验证方法：**SQL Server 身份验证**或 **Windows 身份验证**（缺省）。

注意：与 SQL 身份**验证**相比，Windows 身份**验证**可提供更好的**应**用程序安全性。因此，如果从 Windows 身份**验证**切**换**到 SQL 身份**验证**，那么将**显**示一个**弹**出**对话**框，建**议**您使用 Windows 身份**验证**。如果您**选择**忽略此警告并**继续**使用 SQL 身份**验证**，**请选择**确定。在 Log Viewer 中将**记录**一条**类**似的消息。

   a. 在**服务器名**列表中，**选择**服务器的**节**点名。

   b. 在**数据库**框中，**输**入**报**警数据**库**的名称。

   c. 在**凭据信息**区域的**凭据**下拉列表中，**选择**用于身份**验证**的凭据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：仅当选择 SQL Server 身份验证类型时，才会启用“凭据”字段。Windows 身份验证方法使用当前登录用户的凭据，并禁用“凭据”字段。对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发》中的使用凭据管理器。

4. 选择**测试连接**以测试与数据库的连接。此时出现一条消息，指出与报警数据库的连接是否成功。
5. 选择确定。
6. 选择应用。

配置要从服务器中清除的数据量

您可以：

- 选择要从报警数据库中清除的报警记录的类型。
- 将从报警数据库中清除的记录归档到文件中（可选）。
- 选择要存储清除日志文件的文件夹位置。

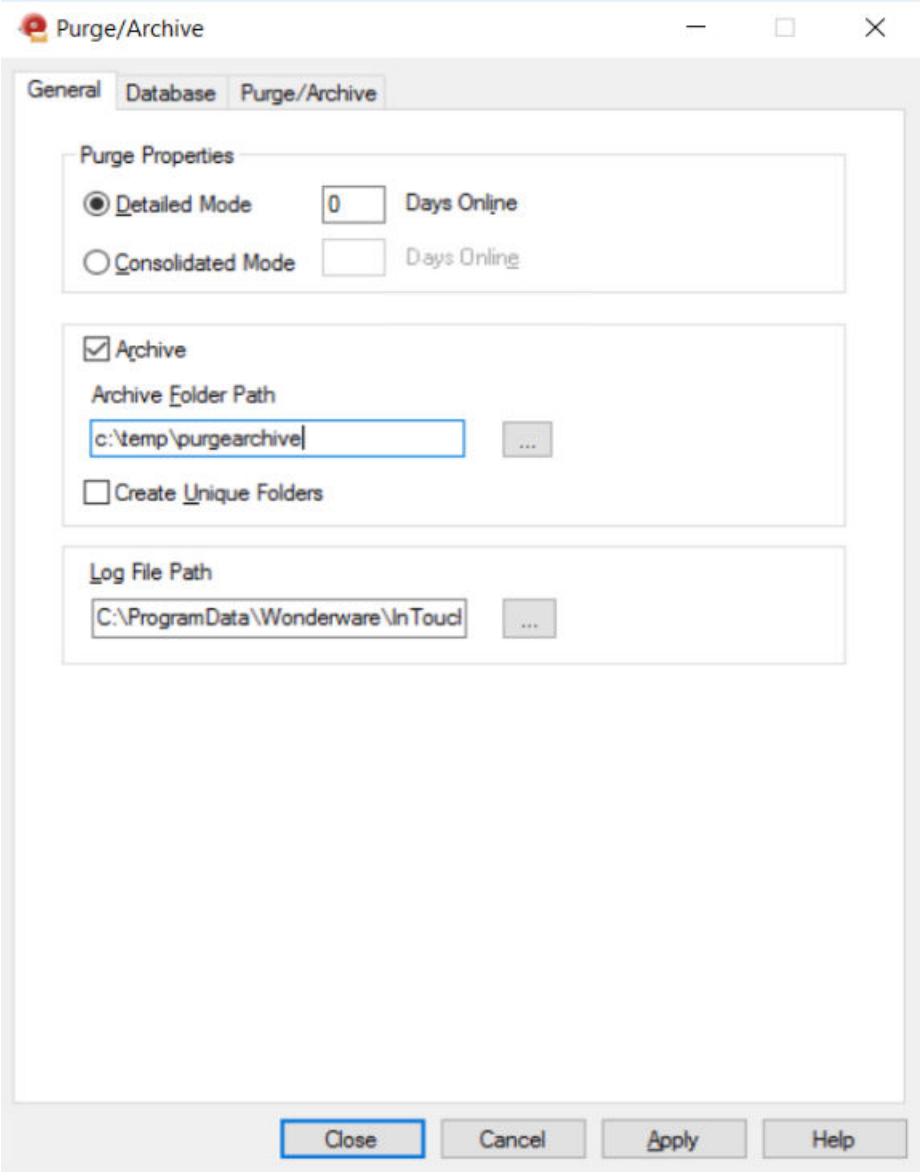
您可以选择需要清除的表的类型，即 AlarmDetail 或 AlarmConsolidated 表。

在指定天数之前的所有数据都会清除掉。有效输入为 0 到 9999。如果选择 0，则会从报警数据库中清除除当天的记录之外的所有记录。

选择要清除的记录

1. 打开 **Alarm DB Purge-Archive** 实用程序。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Purge-Archive**。
2. 选择常规选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**清除属性**区域中，配置要清除的**记录**的**类**型。**执**行以下任一操作：
   - **选择详细模式**，以清除数据**库**中按“**详细**”模式保存的**报警记录**。
   - **选择合并模式**，以清除数据**库**中按“合并”模式保存的**报警记录**。
4. 在**在线天数**框中，**输**入要在**报警**数据**库**中保存**记录**的天数。
5. **选择应用**。

配置清除的数据的归档

归档从**报警**数据**库**中清除的**记录**，然后使用 Alarm DB Restore **实**用程序来恢**复**它**们**。

清除**报警**数据**库时**，Alarm DB Purge-Archive **实**用程序自**动创**建一**组**九个**归**档文件，**这**些文件**对应**于已清除的**报警**数据**库**表。**每**个文件都包含**单张**表中清除的**记录**。

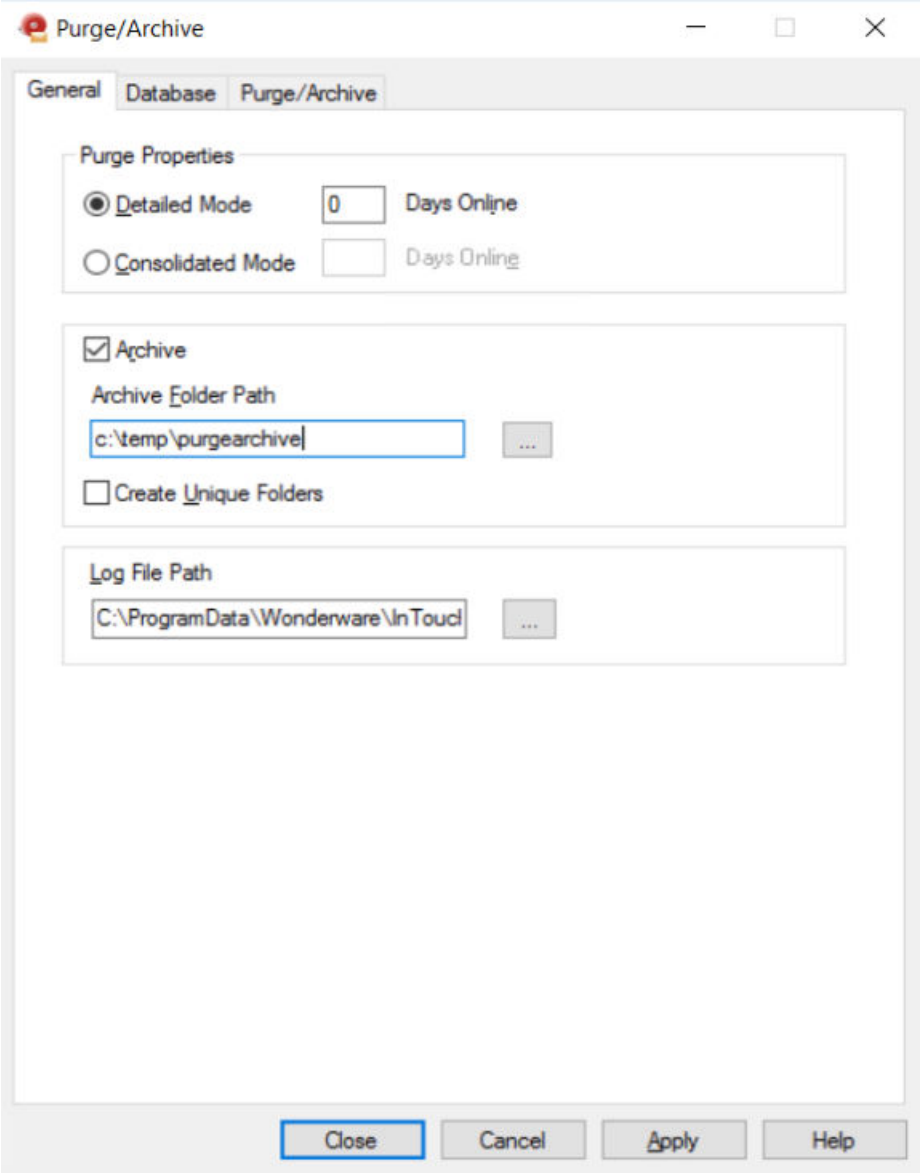
Alarm DB Purge-Archive **实**用程序根据表名、以及清除操作**发**生的日期与**时间给归**档文件指定名称。例如，**对**于 2007 年 6 月 22 日下午 5:30 清除的 AlarmMaster 表，其**归**档文件的名称格式如下：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

AlarmMaster_06222007_1730.txt

配置归档

1. 打开 **Alarm DB Purge-Archive** 实用程序。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Purge-Archive**。
2. 选择常规选项卡。



3. 选择归档复选框。
4. 在归档文件夹路径框中，输入要保存归档文件的文件夹位置，或选择省略号按钮来浏览到该位置。
5. 如果希望将归档文件放置在归档文件所在文件夹下单独的子文件夹中，请选择创建唯一的文件夹复选框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

6. 选择应用。

配置日志文件设置

Alarm DB Purge-Archive 实用程序在清除操作期间生成状态消息。您可以从该实用程序的状态窗口中在线查看这些消息。Alarm DB Purge-Archive 实用程序还将清除消息写入清除日志文件 WWAlmPurge.log 中。

下例显示清除操作成功之后日志文件中存储的消息。

```
清除开始时间 2007 年 6 月 22 日下午 12:16:48
正在开始事务....
正在归档 ProviderSession 表...
正在归档 Query 表...
正在归档 Cause 表...
正在归档 Alarm Core 表...
正在归档 OperatorDetails 表...
正在归档 AlarmDetail 表...
正在归档 Comment 表...
正在归档 Events 表...
正在归档 TagStatus 表...
正在清除数据库中的记录...
正在提交...
清除结束时间 2007 年 6 月 22 日下午 12:16:52
144 条记录（从 AlarmMaster 中）已连同其它表格中的相关记录一起清除。
```

缺省情况下，清除日志文件存储在此文件夹中：C:\Documents and Settings\All Users\Application Data\Wonderware\InTouch。对于运行 Microsoft Windows Vista 操作系统的计算机，缺省应用程序文件夹是：C:\用户\用户名\Documents\我的 InTouch 应用程序。

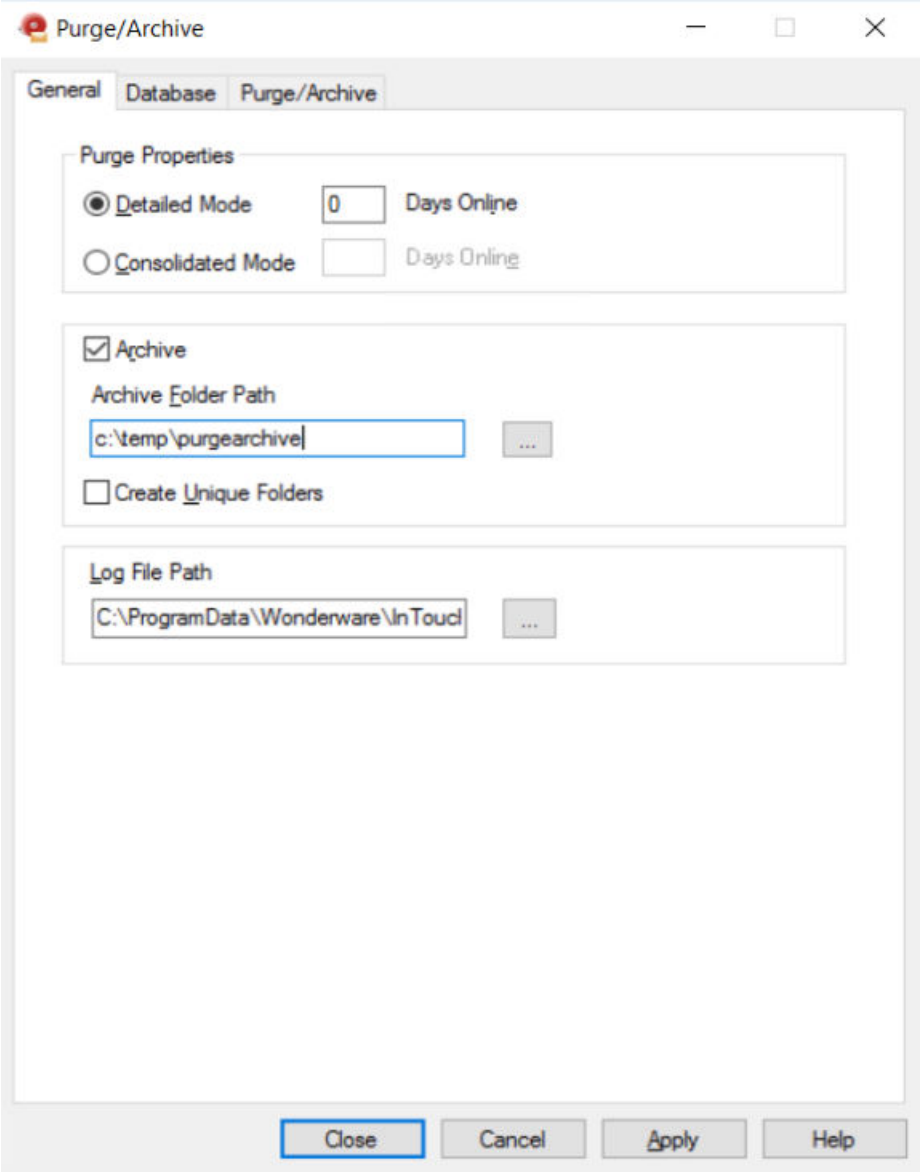
您可以更改清除日志文件的存储位置。

每次发生清除时，Alarm DB Purge-Archive 实用程序都会将新消息追加到日志文件中。

设置归档记录

1. 打开 **Alarm DB Purge-Archive** 实用程序。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Purge-Archive**。
2. 选择常规选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在日志文件路径框中，输入要放置清除日志文件的文件夹位置，或选择省略号按钮来浏览该位置。
4. 选择应用。

手动清除与归档数据库

您可以手动清除与归档报警数据库。此操作会覆盖激活时间并立即开始清除与归档。

清除操作检查是否存在归档文件，并将内容追加到相同的文件中。如果归档文件不存在，则按照命名惯例创建一个文件，然后将它用于归档。

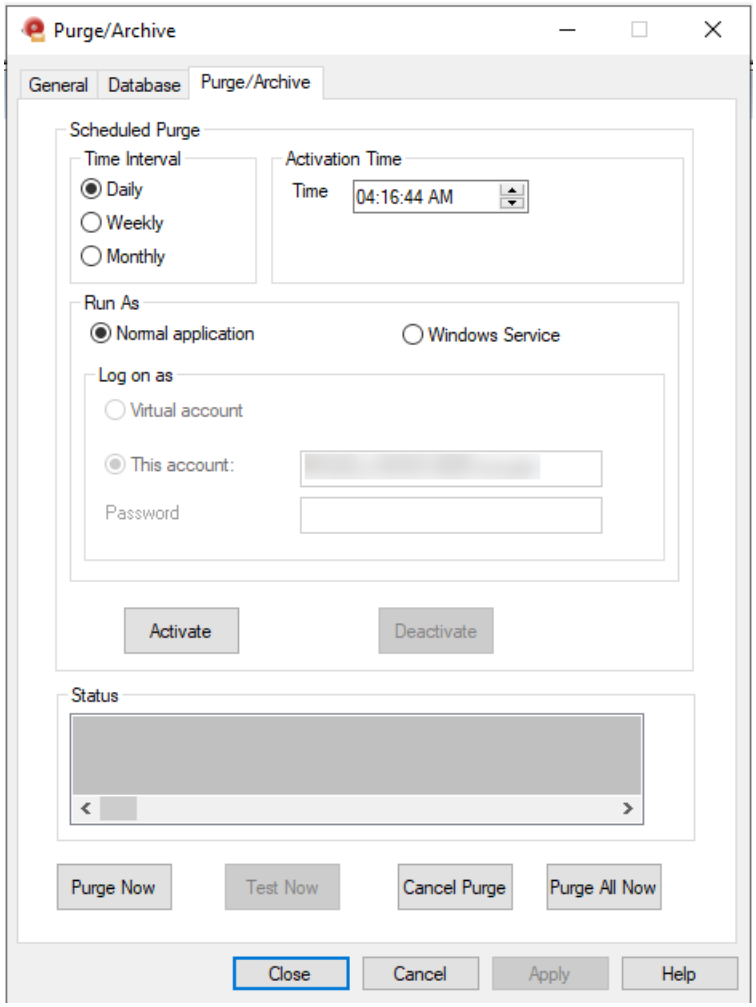
对于通过外部关键字约束条件链接到 AlarmMaster 等主表的表（如 ProviderSession、Query 及 Cause），则清除操作不会删除其中的项目。这些表中的相关记录会写入文件以保持数据的一致性，同时也会保留在数据库中。

警示：仅当 Alarm DB Logger 服务停止时，才可以手动清除所有记录（“立即全清”选项）。如果清除操作在 Alarm DB Logger 服务运行期间成功提交，则 Alarm DB Logger 服务会停止记录并开始缓存记录。

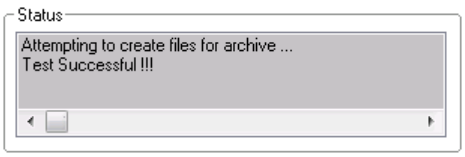
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

从报警数据库中手动清除与归档记录

1. 打开 **Alarm DB Purge-Archive** 实用程序。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Purge-Archive**。
2. 选择“清除/归档”选项卡。



3. 选择立即**测试**以执行清除测试，以便验证与数据库的连接及归档位置。
   清除测试会在指定的归档文件夹中创建一些空的归档文件。**状态**区域显示一条消息，指出测试成功。



   仅当已选择归档清除的记录后，立即**测试**按钮才可用。“归档”选项位于**常规**选项卡。
4. 清除数据库中的记录。执行以下任一操作：
   - 选择“立即清除”以清除所选的记录。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 选择“立即全清”以清除所有的记录。

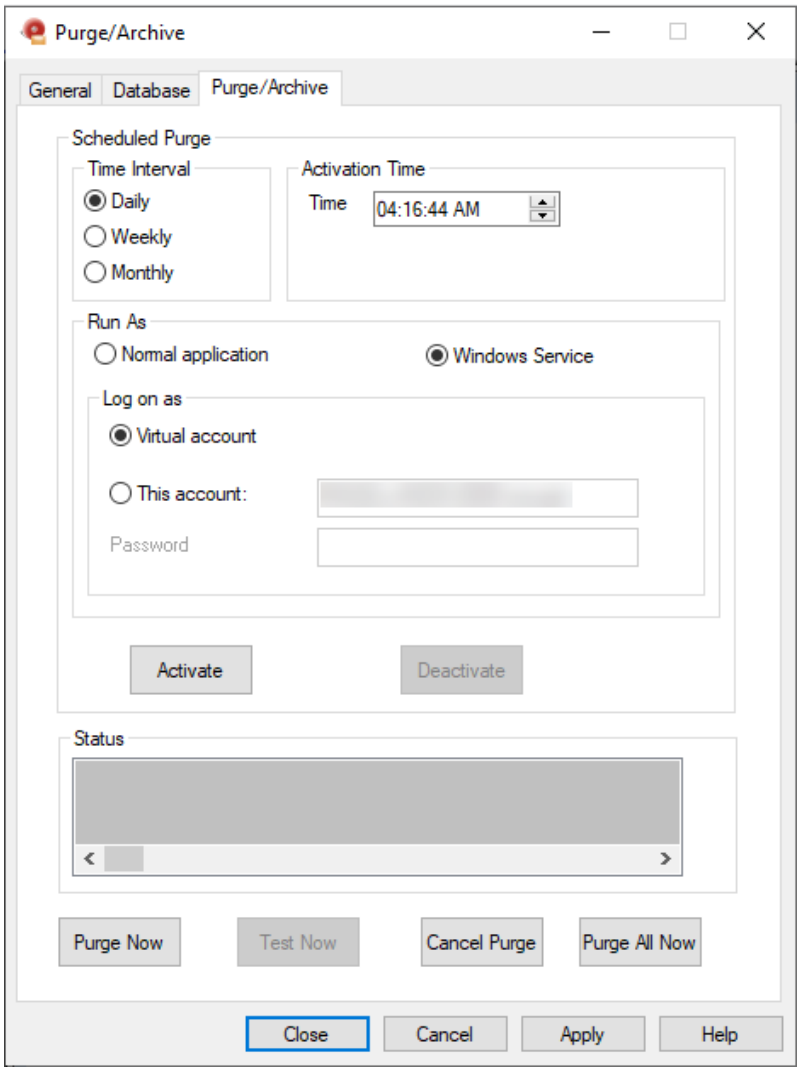
5. 要停止清除，选择“取消清除”。如果取消清除，则报警数据库会回滚到原始状态。

设置自动清除排程

Alarm DB Purge-Archive 实用程序可以按计划的间隔从报警数据库中自动清除或归档记录。您可以执行清除测试，以便验证同数据库及目标位置的连接，并可以启动与停止清除。

设置自动清除排程

1. 打开 **Alarm DB Purge-Archive** 实用程序。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Purge-Archive**。
2. 选择“清除/归档”选项卡。



3. 在“时间间隔”区域中，选择清除间隔（每天、每星期或每月）。

   如果选择每星期或每月，则在激活时间区域中出现日期框，供您指定星期几或月中的日期。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果选择每天，则在时间框中，可以配置希望开始清除/归档操作的时间。

4. 在运行方式区域中，选择正常应用程序可以将 Purge-Archive 实用程序作为应用程序运行，或者选择 Windows 服务可以将其作为服务运行。

   对于 Windows 服务，请选择虚拟帐户，或者在此帐户：区域下指定其它帐户的用户名和密码。

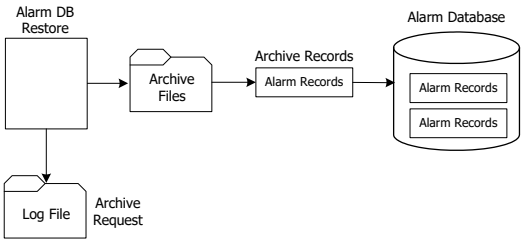
   注意：如需有关虚拟帐户的详细信息，请参阅使用虚拟帐户。

5. 选择"应用"以保存清除与归档设置。
6. 选择激活以实施 Alarm DB Purge-Archive 实用程序的自动清除计划。
7. 选择关闭。

   如果要停止自动清除计划并删除 Windows 服务，选择取消激活。短暂延迟之后，会将该服务从 Windows 服务中删除。

恢复报警数据库

Alarm DB Restore 实用程序可以将归档文件中归档的报警记录恢复到报警数据库中。下图概要介绍将报警记录恢复到数据库的步骤。



要恢复数据库，必须

- 配置与报警数据库的连接。
- 选择要恢复到报警数据库中的记录。
- 将归档的记录恢复到报警数据库。

最小化时，Alarm DB Restore 实用程序显示为系统工具栏中的一个图标。使用鼠标右键单击该图标时，会出现显示以下命令的菜单：

命令	描述
恢复	开始恢复过程。
取消恢复	取消恢复过程。
清除状态	清除状态窗口。
隐藏窗口	将 Alarm DB Restore 实用程序最小化为系统工具栏中的图标。
显示窗口	打开并最大化 Alarm DB Restore 实用程序。
退出	关闭 Alarm DB Restore 实用程序。

如果在 Alarm DB Restore 实用程序中单击鼠标右键，则会出现相同的菜单。

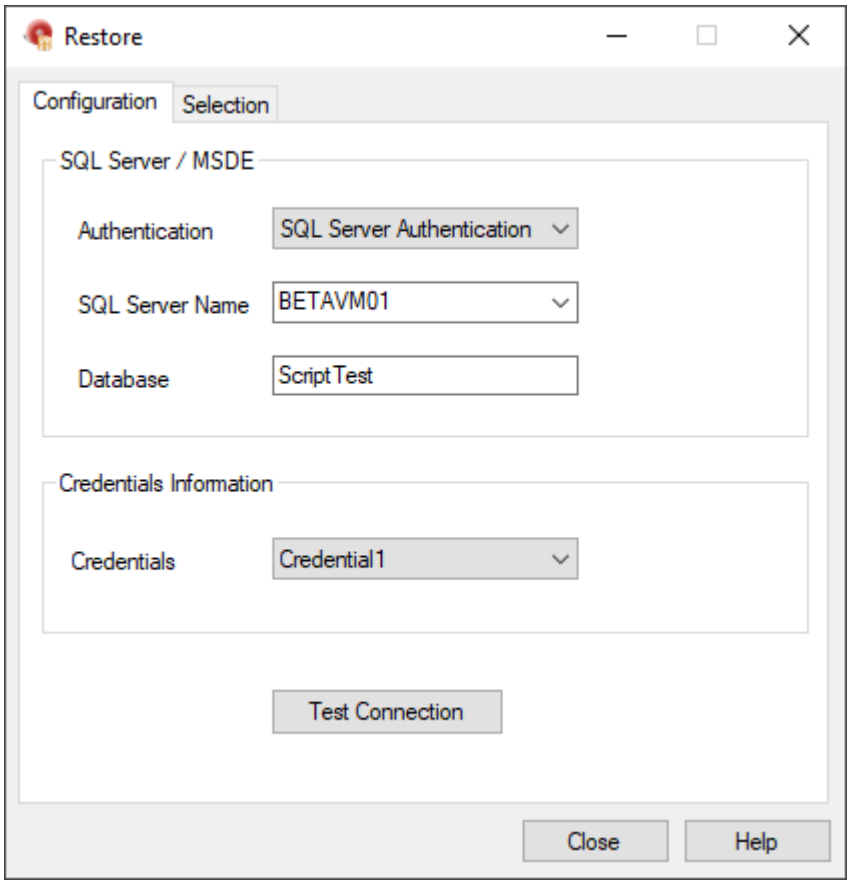
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置数据库连接

您必须选择要用于恢复归档数据的数据库。如果服务器上不存在指定的数据库，则程序会提示您使用缺省的服务器参数来创建新的数据库。

配置用于恢复归档数据的数据库

1. 打开 **Alarm DB Restore** 实用程序。执行以下操作：
   a. 在工具视图中，展开应用程序。
   b. 双击 **Alarm DB Restore**。
2. 选择配置选项卡。



3. 配置与报警数据库的连接。执行以下操作：
   a. 在身份验证列表中，选择身份验证方法：**SQL Server 身份验证**或 **Windows 身份验证**（缺省）。

   注意：与 SQL 身份验证相比，Windows 身份验证可提供更好的应用程序安全性。因此，如果从 Windows 身份验证切换到 SQL 身份验证，那么将显示一个弹出对话框，建议您使用 Windows 身份验证。如果您选择忽略此警告并继续使用 SQL 身份验证，请选择确定。在 Log Viewer 中将记录一条类似的消息。

   b. 在 **SQL Server 名**列表中，选择存放报警数据库的服务器的节点名。
   c. 在**数据库**框中，输入报警数据库的名称。
   d. 在**凭据信息**区域的**凭据**下拉列表中，选择用于身份验证的报警数据库凭据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

仅当选择 SQL Server 身份验证类型时，才会启用“凭据”字段。Windows 身份验证方法使用当前登录用户的凭据，并禁用“凭据”字段。

注意：对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发》中的使用凭据管理器。

a. 选择**测试连接**以测试与数据库的连接。此时出现一条消息，指出与报警数据库的连接是否成功。

b. 选择确定。

4. 选择**关闭**。

配置要恢复的文件

您可以选择要恢复的记录的时段，以及是否希望重建数据库表。

如果取消恢复，数据库会回滚到原始状态。

警示：如果尝试恢复数据库中已经存在的归档报警，则不会恢复这些归档的记录。这可以避免在数据库中出现重复的报警/事件项。与记录关联的“报警 GUID”或“事件 GUID”确定数据库中是否已经存在报警或事件。

选择要恢复的数据库记录

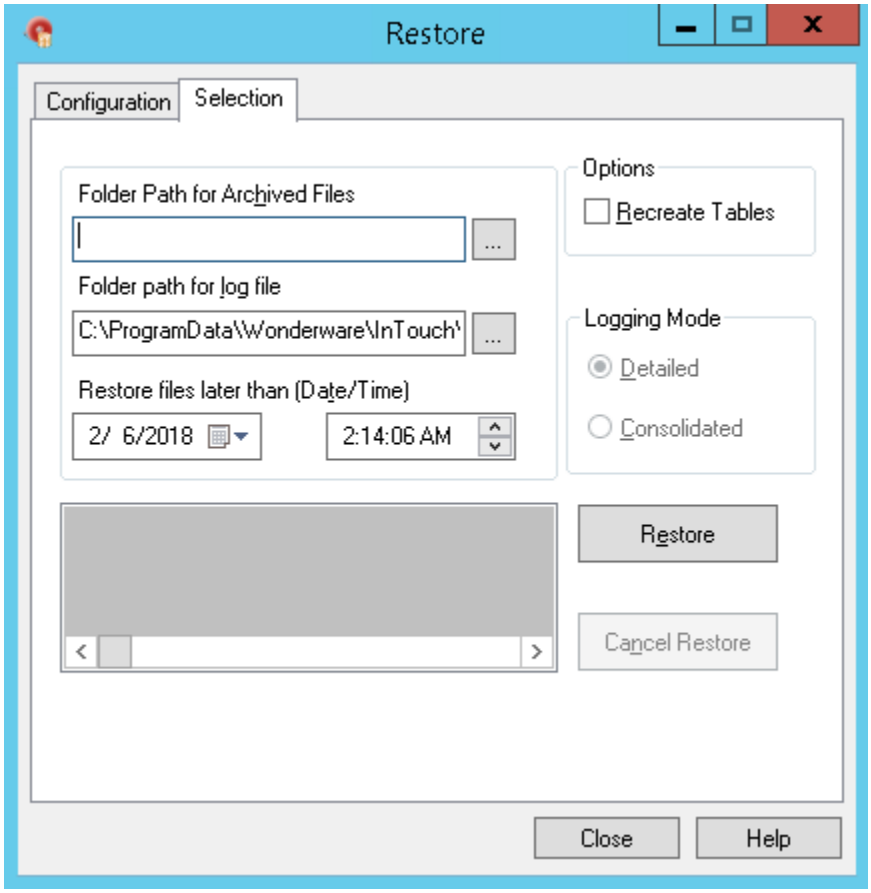
1. 打开 **Alarm DB Restore** 实用程序。执行以下操作：

   a. 在工具视图中，展开应用程序。

   b. 双击 **Alarm DB Restore**。

2. 选择**选择**选项卡。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**归档文件目录**框中，**输**入归档文件位置的完整路径（最多 255 个字母数字字符），或**选择**按**钮**以**查**找并**选择**存**储**归档文件的文件**夹**。
4. 在**恢复此后的文件(日期/时间)**区域中，**选择开**始将**记录**恢**复**到数据**库**的日期与**时间**。

   缺省情况下，**开**始日期与**时间设**置**为**当前日期与**时间**。
5. 在“日志文件目**录**”框中，**输**入用于**创**建并存**储**日志文件的位置的完整路径（最多 255 个字母数字字符），或点**击**按**钮**以**查**找并**选择**某个文件**夹**。
6. 如果**选**中**重建表复选**框，**则**会重建指定的**报**警数据**库**的表。
7. 根据**为**归档文件中包含的**报**警**记录选择**的**记录模式类**型，**选择**：
   - **详细** - 在**详细**模式下重建**报**警数据**库**表。
   - **合并** - 在合并模式下重建**报**警数据**库**表。

     重要事项：重建**这**些表**时**，会覆盖**报**警数据**库**中当前存**储**的所有**记录**。
8. **选择恢复**。

开始数据库恢复操作

在建立数据**库连**接并指定**归**档文件所在的文件**夹**及**时间过滤**器之后，便可以恢**复归**档的数据**库记录**。

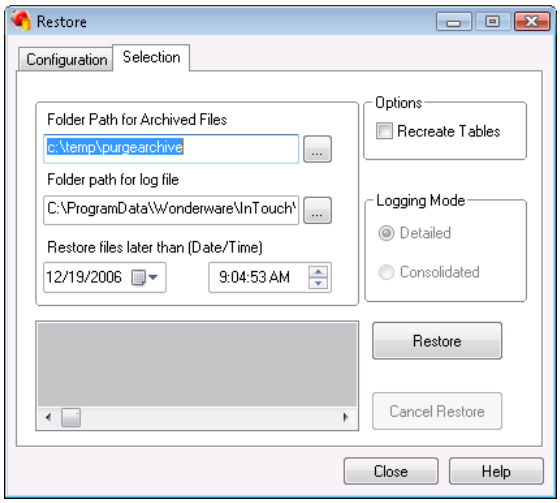
从归档文件中恢复数据库记录

1. 打**开 Alarm DB Restore** 实用程序。**执**行以下操作：

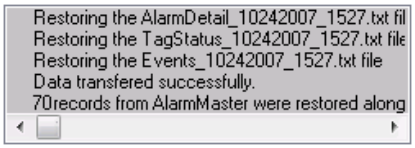
a. 在工具视图中，展开应用程序。

b. 双击 **Alarm DB Restore**。

2. 选择选择选项卡。



3. 选择恢复。此时出现一条消息，显示恢复是否成功，以及已恢复到数据库的记录数。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附录 AF 从 INTOUCH.ini 文件中自定义应用程序设置

第一次运行 InTouch 应用程序时，会在应用程序文件夹中创建 INTOUCH.ini 文件。创建 INTOUCH.ini 文件时，将值指定给确定单独 InTouch 应用程序运行特征的一组参数。

从 WindowMaker 或 WindowViewer 中继续配置应用程序时，会创建新的 INTOUCH.ini 参数或修改现有的参数。例如，从 WindowMaker 的历史记录属性对话框中配置记录功能时，记录参数会添加到 INTOUCH.ini 文件。

其它配置参数必须手工添加到 INTOUCH.ini 文件。

在自定义应用程序之后，可以将 INTOUCH.ini 文件复制到另一个应用程序的文件夹。这样，就可以在不重复所有自定义步骤的情况下为应用程序创建一致的运行特征。

自定义的 INTOUCH.ini 参数

下表列出了一组参数，您可以将这些参数手动输入到 INTOUCH.ini 文件中，为 InTouch 应用程序提供额外的自定义属性。

INTOUCH.ini 参数	用途
16PenTrendDrawMode	确定“16 笔趋势”按平均模式还是最小/最大模式显示数据值。
应用程序缩略图	设置应用程序略图文件的名称。
AllowPubAppEdit	设置应用程序标帜，以使其可以编辑发布的应用程序。如果该值为 1，则可以编辑发布的 InTouch 文件。
CommentRetentive 参数	确定是否保存运行时对“报警注释”域所作的更改。
CompileAllWindows	修改 CompileAllWindows = 1 并且启动 WindowViewer 时，所有窗口都将进行编译并创建 .wvw 文件。 当 CompileAllWindows = 0 并且启动 WindowViewer 时，仅编译主窗口并创建 .wvw 文件。以后每次打开窗口时，都会创建 .wvw 文件。
ForceLogCurrentValue	确定是否将所记录的标记的当前值按 ForceLogging 参数设置的间隔写入“历史日志”文件。
ForceLogging	设置将标记值定期写入“历史日志”文件（不论其当前值如何）的间隔长度。
LoopTimeOut	设置 InTouch 脚本中 FOR-NEXT 循环处理的超时时段。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

INTOUCH.ini 参数	用途
MarkAppReadOnlyNonRDS	在非 RDS 节点上，如果此参数设置为 1，它会将此节点视为只读节点，并将只读许可证用于 InTouchView 应用程序。
MouseMustBeOnObjectForOnKeyUp	确定是否必须将鼠标悬停在对象上才能触发“放开时”动作。缺省值为“1”。
NoKeyboardResize	确定是否按 WindowViewer 屏幕的分辨率来调整数字键盘的大小。
OldRightMouseBehavior	确定 WindowMaker 中是否启用鼠标右键。
PrintScreenWait	设置从 WindowViewer 中打印屏幕之前的等待时段。
PrintWindowWait	设置从 WindowViewer 中打印 InTouch 窗口之前的等待时段。
RemoteTagsLogEvents	确定 InTouch 应用程序是否记录远程引用的标记报警与事件。
RemoteTagsNoIOEvents	确定 InTouch 应用程序是否记录远程引用的标记报警。
ScaleForResolution	确定 InTouch 应用程序窗口在更改到具有不同屏幕分辨率的节点时是否自动调整大小。
ViewLicenseRetryCount	确定 WindowViewer 在启动期间以及没有可用的许可证时，在后台尝试获得许可证的次数。
WindowNameWithSpecialCharacters	如果将参数设置为 1，则可以使用窗口名中的特殊字符来创建新窗口。

设置自定义的记录属性

您可以在 INTOUCH.ini 文件中添加一组参数，用于指定标记值如何保存到 InTouch 历史日志文件。指定给这些参数的值确定记录频率以及是否记录远程引用的标记的值。

设置记录频率

InTouch HMI 基于以下两个条件将日志项写入历史日志文件：

- 只要标记值的变化超出记录死区值一个工程单位值，则 InTouch HMI 立即写入一个日志项。
- InTouch HMI 按固定间隔写入记录的所有标记的当前值。缺省的固定间隔为 60 分钟。

可以在 INTOUCH.ini 文件中添加两个参数来更改间隔。

- ForceLogging

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ForceLogging 以分钟为单位指定固定记录间隔的长度。ForceLogging 可以设置为 5 到 120 之间的值。缺省值是 ForceLogging=60。

- ForceLogCurrentValue

  ForceLogCurrentValue 强制 InTouch HMI 为记录的所有标记写入日志项，即使它们的当前值小于或等于记录死区范围时也是如此。缺省值是 ForceLogCurrentValue=0。

在下例中，当前标记值以 15 分钟为间隔或在标记值发生变化时写入“历史日志”文件中。

```
ForceLogging=15
ForceLogCurrentValue=1
```

记录远程引用的标记

缺省条件下，远程引用的标记不记录到事件日志文件。要记录远程引用的标记，必须启用事件记录功能，然后将 RemoteTagsLogEvents 参数添加到 INTOUCH.ini 文件。

```
RemoteTagsLogEvents=1
```

要将 I/O 标记排除在外不进行记录，请将 RemoteTagsNoIOEvents 参数添加到 INTOUCH.ini 文件。仅当 RemoteTagsLogEvents 参数设置为 1 时，才应用 RemoteTagsNoIOEvents 参数。

```
RemoteTagsNoIOEvents=1
```

禁用 WindowMaker 快捷菜单

缺省条件下，使用鼠标右键单击所选的对象时，WindowMaker 会显示一个快捷菜单。如果倾向于使用与较早版本的 HMI 相同的鼠标行为来开发应用程序，通过在 INTOUCH.ini 文件中将 oldrightmousebehavior 参数设置为 1，可以关闭 WindowMaker 的右击行为。

```
oldrightmousebehavior=1
```

设置自定义的 WindowViewer 属性

您可以添加一组 INTOUCH.ini 文件参数来设置 WindowViewer 的行为，以：

- 处理脚本循环。
- 针对不同屏幕分辨率来缩放 InTouch 窗口。
- 设置打印窗口或屏幕的等待时段。
- 将运行时更改记录到报警注释。
- 设置“16 笔趋势”的绘制模式。
- 调整数字小键盘的大小。
- 调整模拟与字符串用户输入链接的输入字段的大小。

添加脚本循环计时器

缺省条件下，InTouch 脚本中的 FOR-NEXT 循环必须在五秒钟内完成。如果 FOR-NEXT 循环处理在超时限制内没有完成，则 WindowViewer 自动停止脚本。这个超时限制可以防止由脚本错误而导致无限循环。

有时，可能需要编写 FOR-NEXT 循环代码处理超过五秒超时限制的脚本。通过将 LoopTimeout 参数添加到 INTOUCH.ini 文件，您可以更改超时限制的长度。

在本例中，循环处理持续的最长时间为 20 秒：

```
LoopTimeout=20
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

根据不同屏幕分辨率缩放 InTouch 窗口

将应用程序迁移到运行不同屏幕分辨率的其它节点时，通过在 INTOUCH.ini 文件中添加一个参数，可以保持 InTouch 窗口的当前分辨率。

ScaleForResolution 参数值确定运行 WindowViewer 的计算机上的分辨率发生更改之后，WindowMaker 是否自动缩放应用程序窗口。ScaleForResolution 参数不影响 WindowMaker 对话框的分辨率。
ScaleForResolution 参数设置为 1 时启用分辨率转换。

```
ScaleForResolution=1
```

设置打印等待时段的长度

选择要打印的窗口或屏幕时，WindowViewer 将所选窗口或屏幕加载到内存。WindowViewer 随后等待 10 秒，供窗口或屏幕中显示的所有 DDE 变量进行更新。在等待时段结束之后，WindowViewer 将窗口或屏幕发送到打印机。

通过将 PrintWindowWait 或 PrintScreenWait 参数添加到 INTOUCH.ini 文件，可以更改 WindowViewer 打印等待时段。任一参数的等待时段均以毫秒为单位来表示。

```
PrintWindowWait=15000
PrintScreenWait=20000
```

记录报警注释

操作员可以在确认报警时添加注释。要将运行时更改写入标记数据库中的报警注释域，请将下行代码添加到当前应用程序的 INTOUCH.ini 文件中。

```
CommentRetentive=1
```

设置“16 笔趋势”的绘制模式。

您可以根据 16PenTrendDrawMode 参数的值来选择“16 笔趋势”的线条绘制模式。

- 平均模式：`16PenTrendDrawMode=0`

  由于“16 笔趋势”的时间范围与缓冲区大小，趋势上的每个像素都可以代表几秒钟的数据。每个间隔都可能包含多个具有不同值的样本。结果是，趋势的数据点可能显示为间隔内观察到的最大值与最小值之间的一根垂直线。

  在绘制最小值到最大值的垂直线之后，趋势笔移到计算的间隔平均值。下一个间隔从平均值开始绘制线条，直到趋势上的下一个间隔结束。此时再次绘制最小值到最大值的垂直线，笔停在计算的间隔平均值上。此过程会为每个采样间隔重复。

  如果 INTOUCH.ini 文件中没有指定 16PenTrendDrawMode，则“平均”为“16 笔趋势”的缺省绘制模式。

- 最小/最大模式：`16PenTrendDrawMode=1`

  在“最小/最大”绘制模式中，趋势线通过直接连接每个数据采集间隔的端点来绘制。

调整数字小键盘的大小

您可以将一个参数添加到 INTOUCH.ini 文件，用于确定是否可以调整 InTouch 应用程序的数字小键盘的大小。在屏幕分辨率较高 (1280 x 1024) 的情况下放大数字小键盘，可以使得数字小键盘上出现的文本保持清晰。但是，应用程序可能对屏幕空间有所限制，因此对数字小键盘的大小也有实际的限制。

您可以将 NoKeyboardResize 参数添加到 INTOUCH.ini 文件中。缺省条件下不包含此参数。它的缺省值为：

```
NoKeyboardResize=0
```

缺省值允许数字小键盘根据屏幕分辨率调整大小。

可以指定给此参数的其它值有：

```
NoKeyboardResize=1
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在这种情况下，数字小键盘不根据屏幕分辨率调整大小，它的大小保持不变。

调整模拟与字符串用户输入链接的输入字段的大小

您可以将 ResizableInputLink 参数添加到 INTOUCH.ini 文件，以使用鼠标来调整“模拟”或“字符串”用户输入链接的输入框大小。ResizableInputLink 参数必须设置为非零值。

第一次调整“输入”字段的大小之后，WindowViewer 将 ResizableInputLinkWidth 与 ResizableInputLinkHeight 参数添加到 INTOUCH.ini 文件。这些参数以像素为单位指定输入框的宽度与高度。

示例：

```
Resizable InputLink = 1
Resizable InputLink Width=300
Resizable InputLink Height=50
```

同样，您也可以通过编辑 INTOUCH.ini 文件来手工修改指定给这些参数的值。

解决应用程序按钮卡住或显示值问题

可以向 InTouch.ini 文件添加一个参数，以解决 InTouch 应用程序按钮卡在按下位置或显示值不变的问题。按钮和值不响应重复的鼠标单击。

此问题的原因可能是由于某个带有 OnKeyDown 脚本的图形元素隐藏了窗口，OnKeyUp 脚本未运行。此外，当存在两个脚本（OnKeyDown 设置某位，OnKeyUp 清除该位）时，也可能导致按钮卡住问题。操作员单击按钮，但包含该按钮的窗口在释放鼠标前就已关闭。

通过以下操作解决这些问题

- 在 Intouch.ini 文件中插入 `UseLegacyOnKeyUp=1` 参数。
- 在 WindowViewer 属性对话框中选择使用内存窗口缓存复选框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Diagnose

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 33 QuickScript 疑难排解

通过使用 Log Viewer 来显示标记名的运行时值，可以排解 QuickScript 的问题。

将消息记录到 Log Viewer

使用 Log Viewer 可帮助调试 QuickScript。Operations Control Log Viewer 位于系统管理控制台中，随同 InTouch HMI 一起安装。

下面是调试 QuickScript 的一种方法：

1. 在 QuickScript 中设置检查点，以便将一些值记录到 Log Viewer。
2. 打开 Operations Control Log Viewer 来查看这些值。

另一种调试 QuickScript 的方法是创建一个将标记值记录到 Log Viewer 的“键脚本”。

在 QuickScript 中设置检查点

1. 打开怀疑可能导致错误的 QuickScript。
2. 找到希望设置检查点的行。
3. 将下面的代码段之一插入该行的后面：
   - LogMessage(messagetag);
     在此脚本中，messagetag 是要记录其值的消息标记名的名称。
   - LogMessage(StringFromIntg(inttag,10));
     在此脚本中，inttag 是要记录其值的整型标记名的名称。
   - LogMessage(Text(realtag,"#.#######"));
     在此脚本中，realtag 是要记录其值的实型标记名的名称。
   - LogMessage(DText(disctag,"TRUE","FALSE"));
     在此脚本中，disctag 是要记录其值的离散标记名的名称。
   - 在检查点将更多信息记录到 LogViewer，如标识符和/或标记名。例如，
     ```
     LogMessage("DEBUG tag:"+ind.name+" value:"+Text(ind,"#.####"));
     ```
     在此脚本中，`ind` 可以是间接模拟标记。

LogMessage() 函数

将用户定义的消息写入 Log Viewer。

类别

其它

语法

```
LogMessage("Message_Tag");
```

参数

Message_Tag
要记录到 Log Viewer 的字符串实际的字符串或消息标记名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

对于排解 InTouch 脚本的问题来说，这是一个功能非常强大的函数。通过策略性地将 LogMessage() 函数放入脚本，可以判断 QuickScript 的执行顺序、脚本的性能，以及确定在标记发生更改之前和受 QuickScript 影响之后的值。发送到 Log Viewer 的每条消息都会加上准确的日期和时间戳。

重要事项：百分号 (%) 字符会影响调试脚本时 Log Viewer 中显示的诊断消息。如果日志字符串或函数参数中出现 % 字符，则 WindowViewer 可能会停止响应。要消除 % 所导致的错误，请使用两个 %% 字符。

示例

```
LogMessage(“报告脚本正在运行”);
```

上面的语句会将以下内容打印到 Log Viewer：

94/01/14 15:21:14 WWSCRIPT 消息：报告脚本正在运行。

```
LogMessage（"MyTag 的值是 " + Text(MyTag, "#")）;
MyTag = MyTag + 10;
LogMessage（"MyTag 的值是 " + Text(MyTag, "#")）;
```

查看 Log Viewer 消息

Log Viewer 位于 Operations Control 管理控制台中，随同 InTouch HMI 一起安装。

查看 Log Viewer 中记录的值

1. 选择开始，指向程序，指向 **AVEVA**，然后选择 **System Platform 管理控制台**。此时出现系统管理控制台。
2. 在左侧窗格中，展开 **Log Viewer**，展开 **Default Group（缺省组）**，然后选择 **Local（本地）**。此时详细信息窗格中出现 Log Viewer 消息。
3. 找到 LogMessage() 函数所记录的值。

注意：如果在远程 InTouch HMI 节点上调试脚本，则必须将节点名添加到 Log Viewer 中的节点组，然后查看该节点的 Log Viewer 消息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 34 理解错误消息

当“标记查看器”无法执行请求时，它会显示错误消息。消息会显示在对话框中，其中包括错误描述。有关错误消息的更多详细信息，请参阅：

主窗口

添加标记引用

修改标记值

重命名观察窗口

主窗口

功能	描述	原因
快速搜索	未找到搜索的标记名/报警组。	系统无法找到匹配的标记名/报警组。
观察窗口	可以添加到观察窗口中的引用数上限为 2000 个。	您尝试向观察窗口中添加的项目超过 2000 个。
	观察窗口数的上限为 50 个。	您尝试添加的观察窗口超过 50 个。
	<File1> 不是有效的观察列表文件。	您在打开观察窗口时指定了无效的文件名。
	<File1> 已存在。是否要替换它？	您在保存观察窗口时指定了已存在的文件名。
	当前的 InTouch 应用程序中不存在以下引用，这些引用未添加到观察窗口中： <引用列表>	观察窗口中加载的某些引用无效，或不存在于当前的 InTouch 引用程序中。

添加标记引用

功能	描述	原因
添加标记引用	在 <abcd> 中无法找到 <f>。	您已在替换框中输入字符串 <f>，该字符串不是“标记引用”中字符串 <abcd> 的一部分。
	“<从>”值不得大于“<到>”值。	您在到框中输入的数字小于从框中的数字。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

功能	描述	原因
	请为“<从>”输入数字。 或者， 请为“<到>”输入数字。	您已在替换框中输入字符串，但“从”框或“到”框为空。
	当前的 InTouch 应用程序中不存在以下引用，这些引用未添加到观察窗口中： <引用列表>	您已输入一个无效引用，或者，您输入的多个引用中的部分引用无效。
	可以添加到观察窗口中的引用数上限为 2000 个。	您尝试向观察窗口中添加的项目超过 2000 个。

修改标记值

功能	描述	原因
修改整型值	<1a> 不是有效数值。	您已在值框中输入无效项目 <1a> 并选择应用。
	输入值超出整型数据类型的有效范围。	您输入的数字已超出允许的范围（-2147483648 到 2147483647）。
修改实型值	输入值超出实型数据类型的有效范围。	您输入的值已超出允许的范围（-3.402823466e38 到 3.402823466e38）。
标记引用对话框	该应用程序中不存在标记引用 <InvalidReference>。	您已在标记引用对话框中输入值 <InvalidReference> 并选择确定。
	待设置数据属于错误的数据类型。	引用是只读标记，但您尝试修改该标记的值。 或者， 您输入的值对于所选标记引用类型而言无效。例如，您在修改整型标记时输入了字符串类型的值。

重命名观察窗口

功能	描述	原因
观察窗口	观察窗口 <观察列表 1> 已存在。	您在重命名观察窗口时输入了已存在的名称。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

功能	描述	原因
	“ReadonlyReference”为只读, 无法修改。	您在观察窗口中添加了只读标记， 但单击该标记以修改其值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 35 管理 InTouch Web 客户端

本节包括：

InTouch Web 客户端错误处理

浏览器和移动支持

在 Web 浏览器中查看图形的疑难排解

无效证书错误

向 Chrome Web 浏览器添加扩展

InTouch Web 客户端错误处理

InTouch Web 客户端提供了统一的错误页面来显示预期错误。在以下情形将显示错误页面：

- **无应用程序** - Web 客户端无法检测到在 WindowViewer 上运行的上一个应用程序所需的支持文件。在 WindowMaker 中打开所需的应用程序。
- **权限遭拒绝** - 当前登录的用户没有足够的权限来查看网页。将用户添加到 aaInTouchUsers 或 aaInTouchRWUsers 用户组。
- **没有可用的许可证** - 没有为当前 Web 会话获得有效的许可证状态。确认有效的 Web 客户端许可证可用。
- **快速切换模式** - Web 客户端应用程序处于快速切换模式，但用户尝试使用 http://localhost/InTouch 以外的 URL 来访问 Web 客户端。

浏览器和移动支持

浏览器支持

注意：为让 InTouch Web 客户端正常工作，建议始终安装最新版本的浏览器或将浏览器升级到任何可用的最新版本。

InTouch Web 客户端已通过以下 Web 浏览器测试：

- Google Chrome
  - 适用于台式机的 Chrome V64.0.3282.186 或更高版本
  - 适用于 Android 手机和平板电脑的 Chrome V64.0.3282.137 或更高版本
- Microsoft Internet Explorer 11.0.9600.18861 或更高版本
- Microsoft Edge
  - 适用于 Windows 10 的 Microsoft Edge V41.16299.15.0 或更高版本
  - 适用于 Surface 的 Microsoft Edge V41.16299.248.0 或更高版本
- Apple Safari for iOS V11.2.6

如果 Web 浏览器不受支持，您将收到一条错误消息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

移动设备支持

InTouch Web 客户端支持以下移动设备及其缺省浏览器：

- Apple iPhone
- Apple iPad
- Android 手机和平板电脑
- Microsoft Surface

在 Web 浏览器中查看图形的疑难排解

如果 InTouch Web 客户端不显示，请验证以下设置：

- 确保 *InTouch Web 服务*正在节点上运行。
- 确保访问 InTouch Web 页面的用户帐户属于其中一个用户组。
- 如果同一台计算机上正在运行 IIS，可能与 InTouch Web 服务器冲突。
  - 如果 IIS 正在运行但未在使用中，请停止并禁用该服务。如果 IIS 必须运行，则配置为使用除 80 以外的端口，以避免冲突。

如果在尝试启动 InTouch Web 客户端时获得 *HTTP 错误 404.0 - 找不到*错误

- 确保所选节点上已启用 Web 客户端。联系管理员。

如果任何图形基元或动画不可见或未按预期工作

- 检查不受支持的功能的列表，以查看 Web 客户端是否尚不支持它。
- 通过 http://localhost/InTouch 在本地运行 Web 客户端，浏览到符号，然后查找兼容性图标以查看图形中不受支持项目的列表。
- 如果图形中使用了不受支持的基元或动画，则会删除该基元或动画，并在 Logger 中记录一条消息。记录的消息包含生成的 CSV 报告文件的路径。您可以查看记录器消息，并可以导航到报告查看更多信息。

如果所选图形已显示，但不显示任何数据或动画不更新

- 如果图形指向 InTouch 标记，确保 InTouch WindowViewer 正在运行。
- 如果图形指向应用程序对象引用，确保 Application Server 正在运行。
- 确保 InTouch iData 服务正在运行。

无效证书错误

错误：证书错误：您与此网站的连接不安全。

在访问 InTouch Web 客户端时，如果该客户端上没有可用的有效证书，浏览器即会显示此错误。

解决方案

1. 连接到服务器并启动 System Platform 配置器。
2. 从“系统管理服务器”页面中，选择**详细信息**。

   此时出现**证书**对话框。
3. 在**详细信息**选项卡上，选择**复制到文件**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 按照**证书导出向导**的说明进行操作并保存文件。
5. 将**证书**文件复制到客户机。
6. 在客户机中，启动 **Microsoft Management Console** (MMC)。
7. 导入**证书**。
8. 在导入**证书**之后，启动浏览器。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Manage

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 36 AVEVA Operations Control 连接体验

关于 Operations Control 连接体验

Operations Control 连接体验选项可通过 CONNECT 云功能在该节点的所有 Operations Control 产品中实现单一登录 (SSO) 体验。Operations Control 连接体验需要具有有效 Operations Control 订阅和用户管理的 CONNECT 帐户。

- 统一用户管理：在节点上启用 Operations Control 连接体验会将此节点上所有参与的 Operations Control 产品的行为都改为在启动节点上的第一个产品时需要使用 CONNECT 进行登录身份验证。节点上的产品在启动后将使用 CONNECT 进行身份验证，并且将为经过身份验证的用户启用单点登录 (SSO)。基于 CONNECT 身份验证的授权是 Operations Control 连接体验下唯一可用的安全模式。
- 跨节点的兼容性：必须在系统中的所有节点上都启用 Operations Control 连接体验。先前在未启用 Operations Control 连接体验的节点上构建的应用程序必须重新配置，才能在 Operations Control 连接体验环境中运行。

您可以随时禁用 Operations Control 连接体验，但必须在系统中的所有节点上都禁用 Operations Control 连接体验。在 Operations Control 连接体验下构建的应用程序必须重新配置（包括身份验证方法和产品许可），才能在非 Operations Control 连接体验环境下运行。

配置 Operations Control 连接体验

要在 AVEVA Operations Control 下操作 InTouch 和其它 System Platform 产品，无论是仅作为产品订阅，还是通过 Operations Control 连接体验，都必须配置已安装和许可的产品。仍然需要配置许可证服务器以使无头服务正常工作。

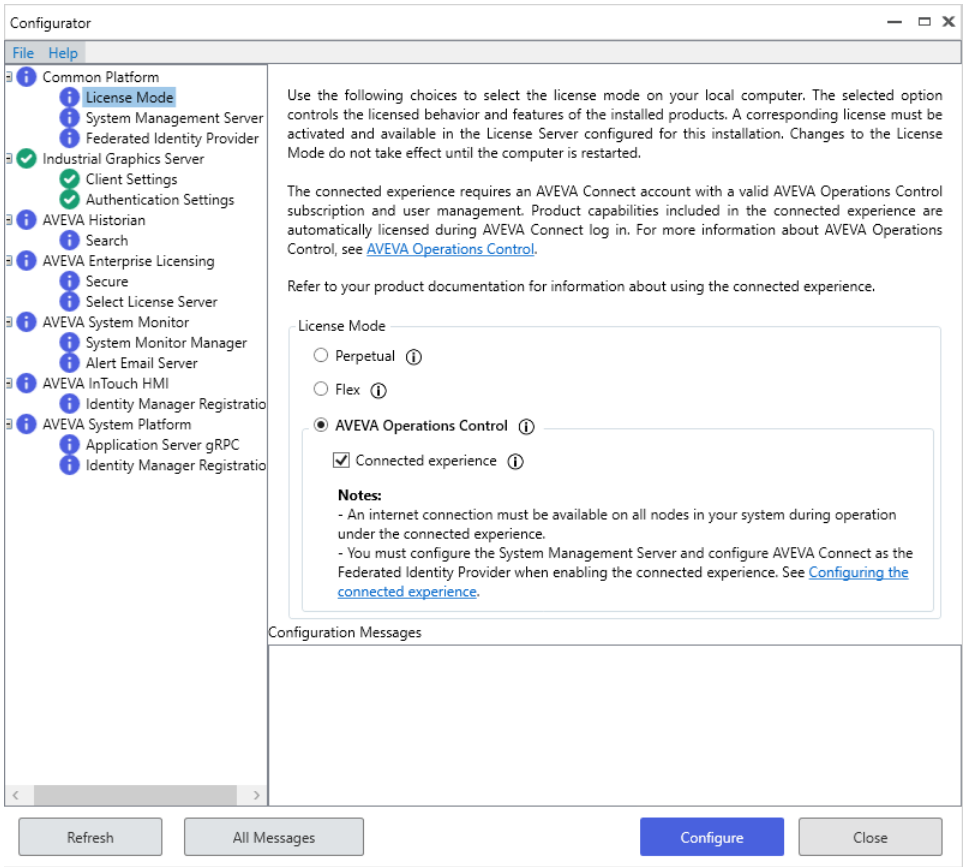
如需详细信息，请参阅：

- 配置器帮助
- AVEVA Identity Manager 帮助

要配置 Operations Control 连接体验：

1. 从**开始** > **AVEVA** > **配置器**打开配置器。
2. 在左侧窗格中**选择许可证模式**以配置许可证模式。
   a. **选择 AVEVA Operations Control 单选按钮**，以订阅两个 AVEVA Operations Control 软件包（Edge 和 Supervisory）中的至少一个，这包括您定义的用户组可无限使用该产品包中的所有产品。
   b. 选中**连接体验复选**框，以通过 CONNECT 云功能（默认在产品中已提供）在此节点上的所有 Operations Control 产品中实现单一登录 (SSO) 体验。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 要配置系统管理服务器，请在左侧窗格中的通用平台下选择系统管理服务器。

注意：如果系统提示您输入系统管理服务器的用户凭据，请使用以下格式输入用户名：DomainName\UserName。如果您具有域管理员权限但并不是本地计算机上的管理员，那么可能会显示输入用户凭据的提示。您必须是 Administrators 或 aaAdministrators 操作系统组的成员，才能配置系统管理服务器。

a. 要连接到现有的系统管理服务器，请选择连接到现有的系统管理服务器。从可用的系统管理服务器列表中，选择所需的系统管理服务器。请注意，该列表显示网络上已配置作为系统管理服务器运行的所有计算机。在大多数情况下，只有一台系统管理服务器。

b. 要将一台机器配置为冗余 SSO (RSSO) 服务器，请在连接到现有的系统管理服务器下选择将此机器配置为冗余 SSO 服务器。

或

c. 要将当前机器配置为系统管理服务器，请选择此机器为系统管理服务器选项。

d. 单击配置。

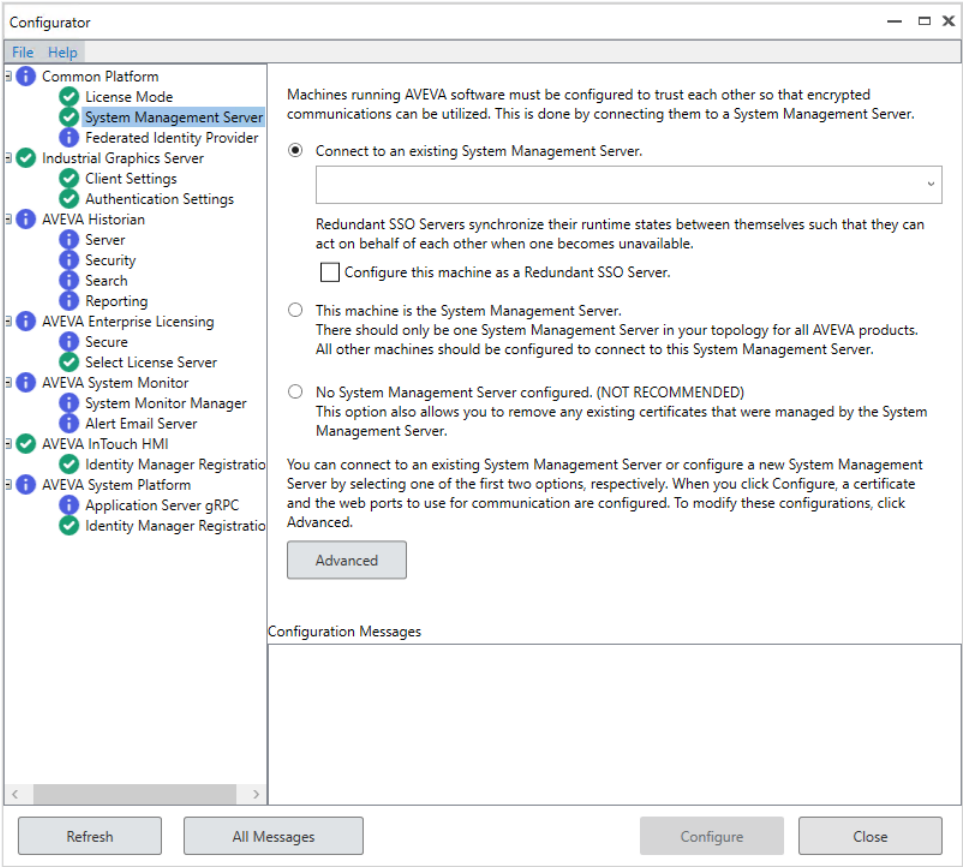
注意：

- Operations Control 连接体验不支持未配置任何系统管理服务器选项。您必须配置系统管理服务器，才能使用 Operations Control 连接体验。

- 只会将网络中的一台计算机标识并配置为系统管理服务器。然后，运行 AVEVA 产品的计算机可以连接到该单个系统管理服务器来建立信任并配置加密的通讯。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 配置冗余 SSO 不会配置单独的系统管理服务器。



4. 要配置联合标识供应器，请在左侧窗格中的通用平台下选择联合标识供应器。

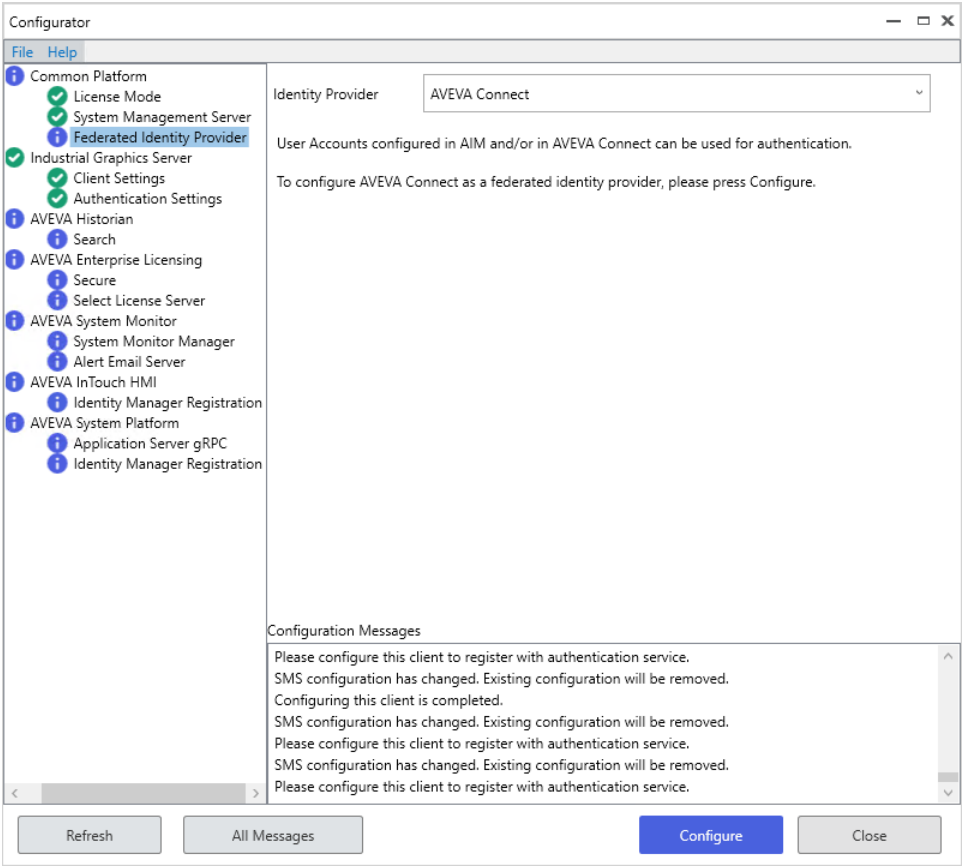
   您可以配置系统管理服务器 (SMS) 和冗余 SSO (RSSO) 计算机上可用的平台公共服务 (PCS) 框架中的 Identity Manager 组件来与 CONNECT 进行“联合”登录。这意味着用户可在身份标识管理器登录表中输入使用 CONNECT 帐户注册的电子邮件地址，随后他们将被重定向到 CONNECT，以便可以登录。目前，对于 Operations Control 连接体验，仅支持与 CONNECT 联合，但不支持与 Azure AD 联合。

注意：

- 对于联合标识连接，您需要一个有效的 CONNECT 帐户，并且您必须是该帐户的管理员。

- 此步骤仅适用于配置了“此机器为系统管理服务器”的节点。如果节点连接的是现有系统管理服务器，那么不需要配置联合标识供应器。

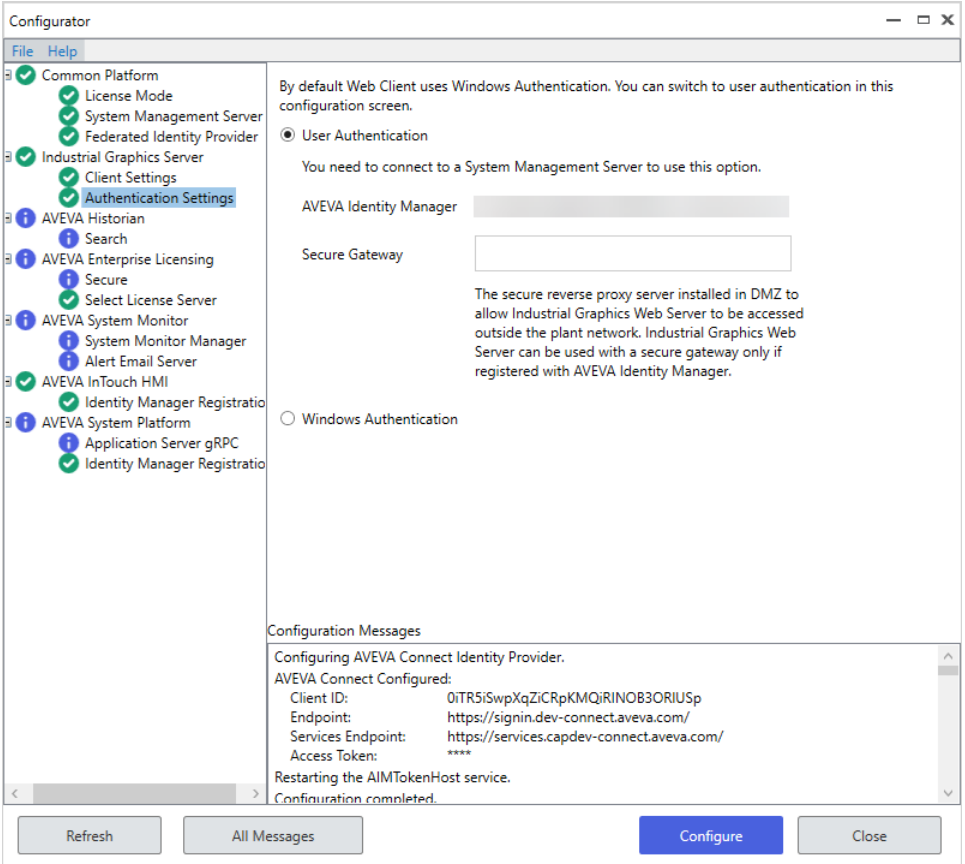
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



如果您有多个 CONNECT 帐户，那么在身份验证后，将显示一个帐户选择对话框，其中列出您所属的多个 CONNECT 帐户名称。选择您想要与之联合的帐户，从此以后，将使用所选帐户对该计算机和所有 Operations Control 产品进行身份验证或验证以订阅。如果您仅属于一个 CONNECT 帐户，则不会显示帐户选择对话框。

5. 要将 InTouch Web 客户端配置为使用用户身份验证，请在左侧窗格中的工业图形服务器 > 身份验证设置下选择用户身份验证，然后单击配置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

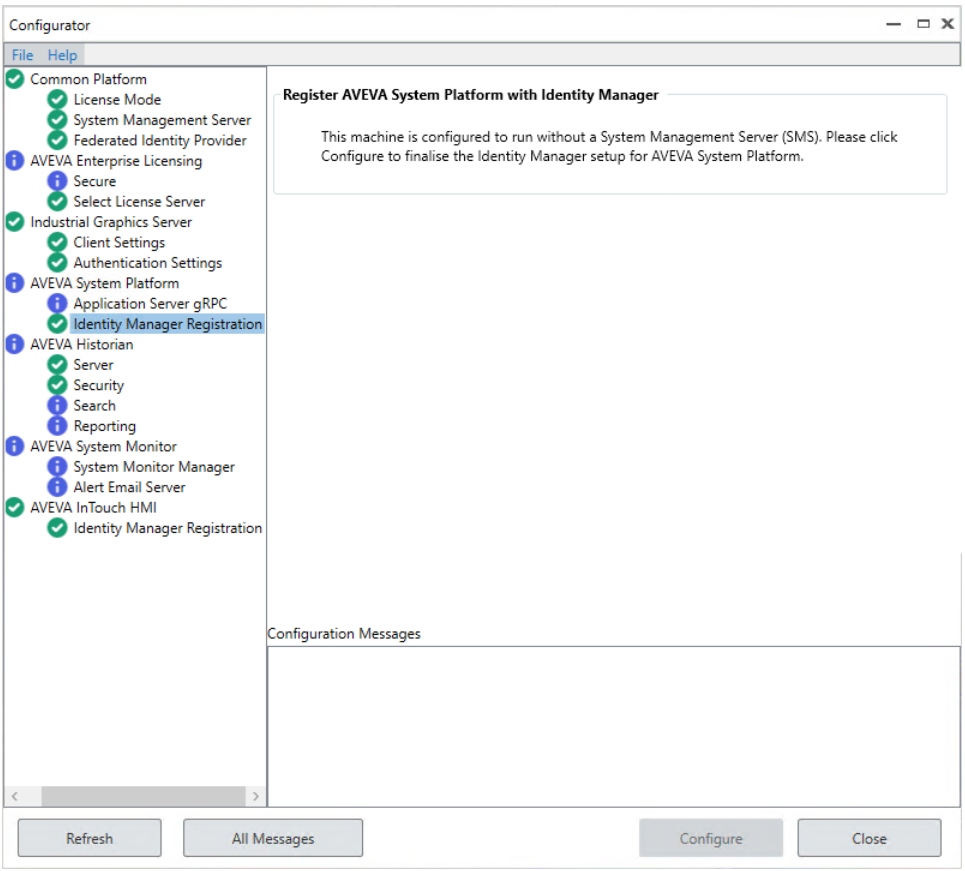


6. 要向 AVEVA Identity Manager 注册 System Platform，请在左侧窗格中的 **AVEVA System Platform** 下选择 **Identity Manager** 注册。

   只有在完成系统管理服务器配置后，才可以向 Identity Manager 注册 AVEVA System Platform。

   单击配置以向 Identity Manager 注册。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



7. 要向 AVEVA Identity Manager 注册 InTouch， **请**在左**侧**窗格中的 **AVEVA InTouch HMI** 下**选择 Identity Manager 注册**。

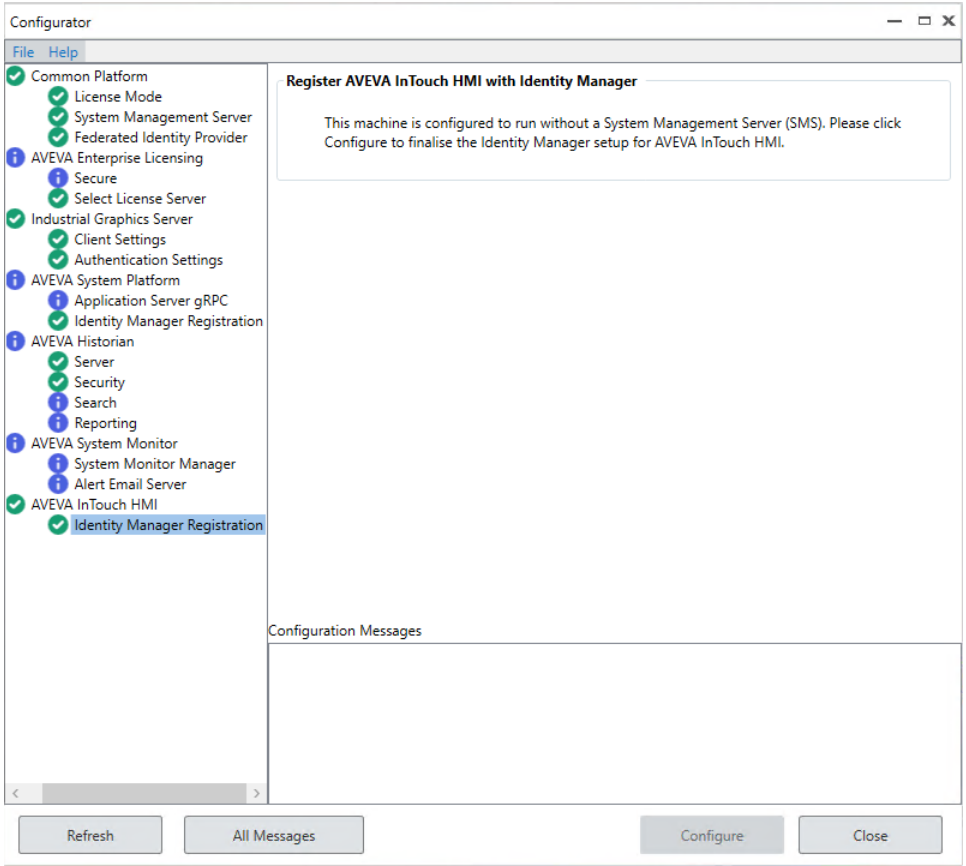
   只有在完成上述**步骤**后， 才可以向 Identity Manager 注册 InTouch HMI。

   - 您已启用 AVEVA Operations Control **连**接体**验**作**为**您的**许**可**证**模式。
   - 您已配置系**统**管理服**务**器。
   - 您已在**联**合**标识**供**应**器中**选择** CONNECT。

   单击配置以向 Identity Manager 注册。

   注意：如果您未配置系**统**管理服**务**器（不推荐）， 您仍然可以配置 InTouch， 但不能使用 Operations Control **连**接体**验**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



身份验证和权利

身份验证和权利以及授权是安全性元素，用于控制用户对您的机构使用的 AVEVA 产品的访问。

- *身份验证*是指通过使用一组唯一凭据（用户名和密码）验证用户的身份。
- *权利*是指您的机构订阅的产品以及向经过身份验证的用户授予访问权限的产品。如果用户没有有效的 Operations Control 订阅，即使他们经过身份验证，也可能无权使用特定产品。
- *授权*是指与经过身份验证的用户关联的权限级别。授权通常通过将用户分配至用户组或联合身份标识管理来完成。

  联合身份标识通过*联合身份标识提供者* (FIDP) 管理，并允许针对整个 AVEVA 产品组合的用户进行统一的用户管理和单点登录。这通过配置器启用，通常可在安装时完成，但随时可以重新配置。

您选择的许可证模式在系统平台环境中如何处理身份验证、权利以及授权方面至关重要。有关详细信息，请参阅许可证模式。

启用 Operations Control 连接体验后，用户每次登录系统平台组件（如 IDE、Historian、OMI ViewApp 等）时，该登录信息会作为事件另存为 CONNECT 审核日志中。

查看数据日志

通过审核日志，机构和 AVEVA 的授权用户可以通过 CONNECT 门户查看您的帐户中发生的事件日志或操作日志。这些日志显示您的机构帐户中在所选时间段内发生的所有操作。您必须是帐户管理员，或者必须具有报告查看者角色，才能查看审核日志。

注意：审核日志中的数据从您的帐户具有审核功能时起就可用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要查看审核日志，请转到 AVEVA Connect 网站。在站点导航菜单上，选择“审核”。此时会显示审核日志页面。

审核日志提供以下信息：

- 执行操作的时间戳（日期和时间）。
- 操作的详细信息，如用户名、计算机名称和应用程序名称（例如，AVEVA OMI 或系统平台 IDE）。记录的信息包括通过身份验证的用户访问的产品，以及他们的可用权利访问权限。
- 可以使用 CONNECT 中的**权利检查**函数过滤审核日志。此函数通过连接体验提供与产品使用情况相关的所有数据。

注意：在用户界面上显示数据时使用本地时间。但是，操作详细信息中显示 UTC 时间和数据导出时间。

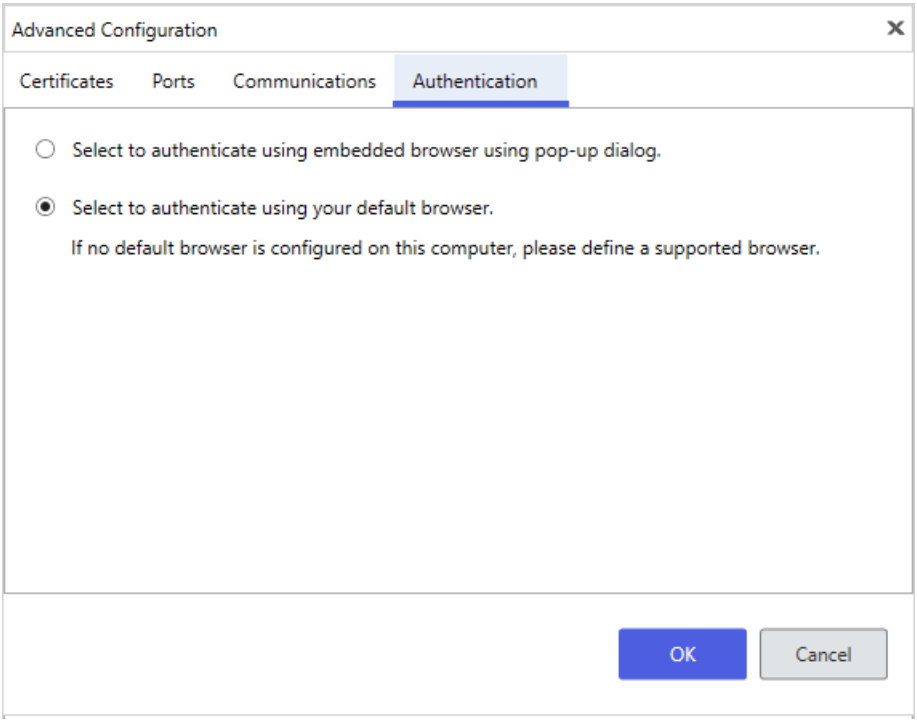
有关审核日志的更多信息，请参阅 CONNECT 帮助。

使用 AVEVA Identity Manager 进行身份验证

在配置器中启用 Operations Control 连接体验后，首次启动“应用程序管理器”、WindowMaker 或 WindowViewer 时，系统将提示您使用 AVEVA Identity Manager 进行身份验证。在后续的启动中，您将通过 SSO 进行身份验证。

在 Web 浏览器上使用 AVEVA Identity Manager 进行身份验证：

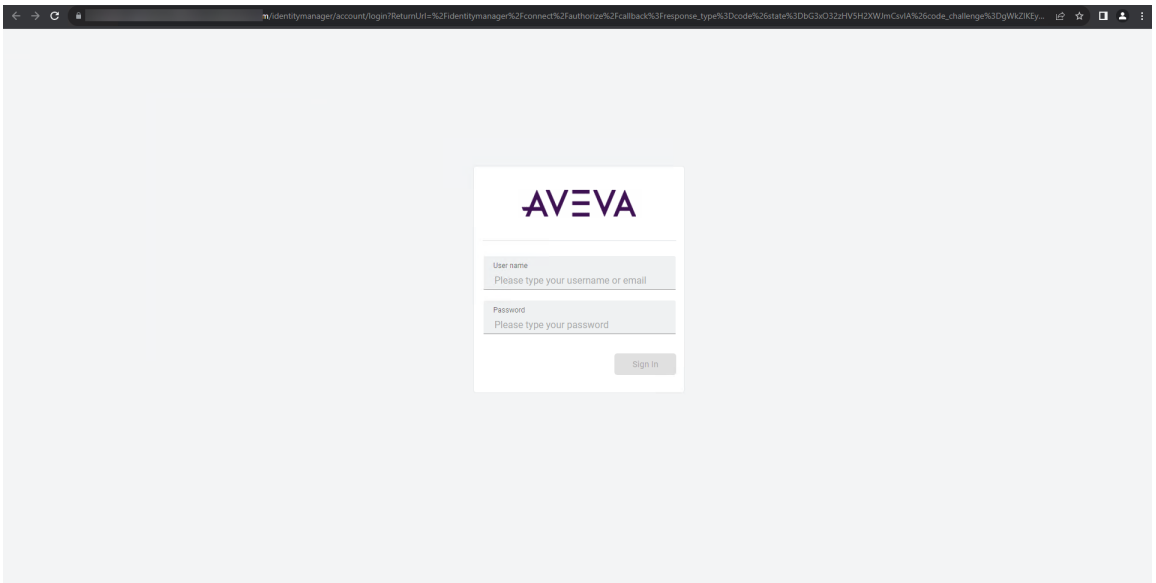
1. 要在 Web 浏览器上打开 AVEVA Identity Manager 登录屏幕，您必须在配置器中的**通用平台 > 系统管理服务器 > 高级 > 身份验证**下选择**选择使用缺省浏览器进行身份验证**选项。



2. 当您启动任何 AVEVA Operation Control 应用程序时，AVEVA Identity Manager 登录页面会在系统的缺省浏览器中打开。
3. 在 AVEVA Identity Manager 登录页面的**用户名**字段中，输入使用 CONNECT 帐户注册的电子邮件地址登录到 AVEVA Identity Manager。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

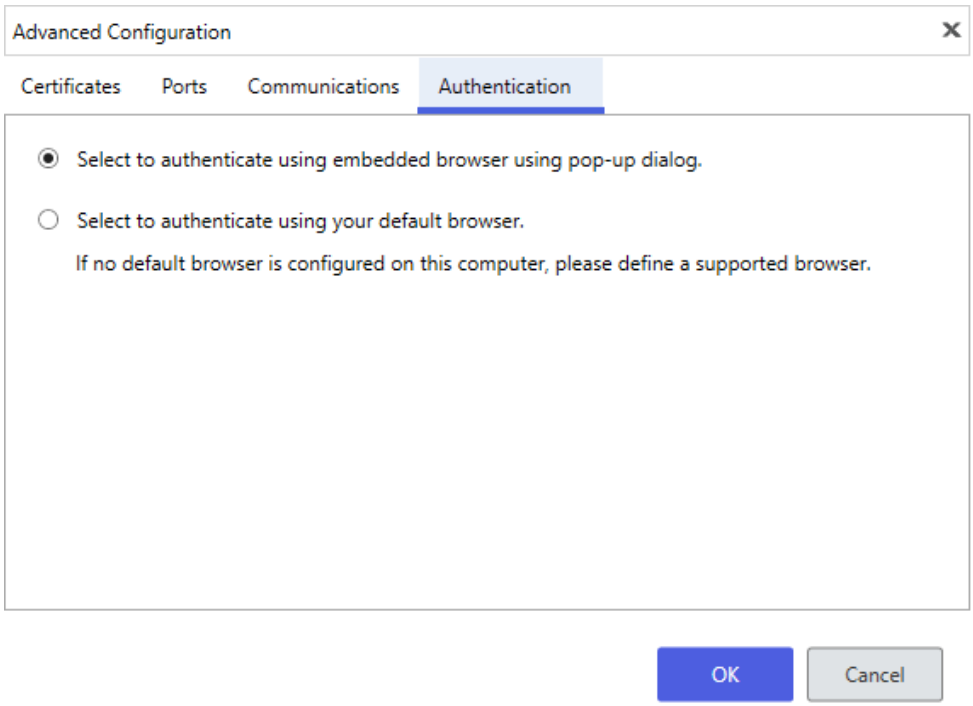
4. 在**密码**字段中，输入相应的**密码**。
5. 单击**登录**。



要在嵌入式浏览器上使用 AVEVA Identity Manager 进行身份验证：

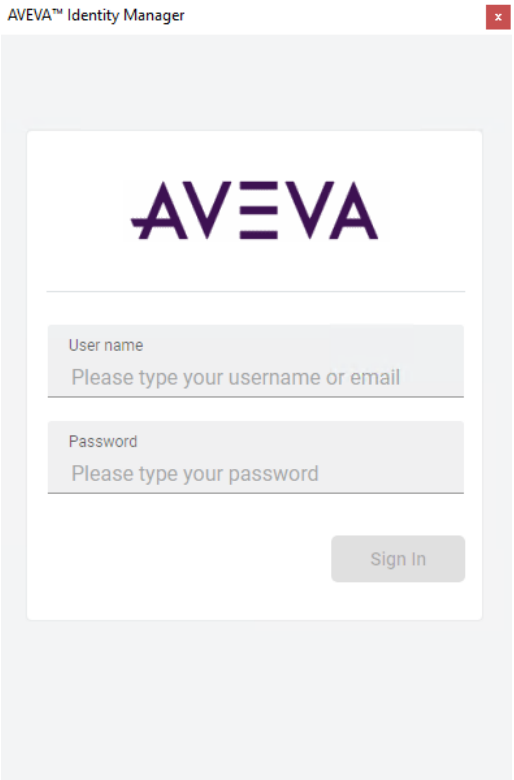
1. 要在嵌入式浏览器上打**开** AVEVA Identity Manager 登录屏幕，您必须在配置器中的**通用平台 > 系统管理服务器 > 高级 > 身份验证**下选择**选择在嵌入式浏览器中使用弹出式对话框进行身份验证**选项。默认条件下，该选项处于已**选择**状态。

   对于 InTouch Access Anywhere，出于安全原因，建议使用嵌入式浏览器选项。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 当您启动任何 AVEVA Operation Control 应用程序时， AVEVA Identity Manager 登录页面会在嵌入式浏览器中打开。
3. 在 AVEVA Identity Manager 登录页面的用户名字段中， 输入使用 CONNECT 帐户注册的电子邮件地址登录到 AVEVA Identity Manager。
4. 在密码字段中， 输入相应的密码。
5. 单击登录。



连接丢失期间的行为

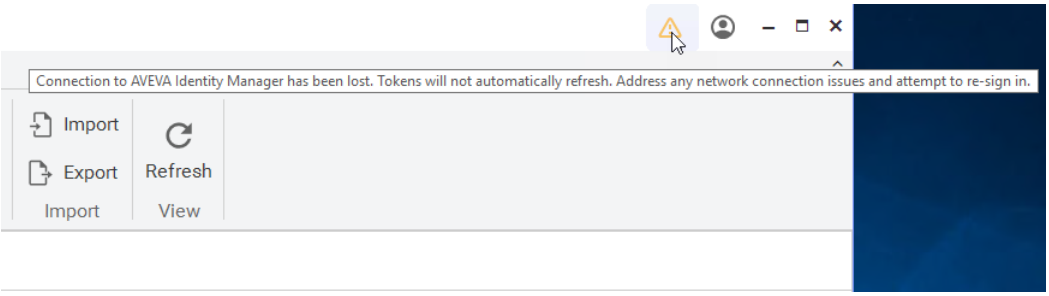
在连接丢失的情况下

- 身份验证成功且应用程序顺利运行后， 如果 AVEVA Identity Manager 与 CONNECT 之间的连接丢失， 内部部署功能将继续有效。
- 身份验证成功且应用程序顺利运行后， 如果 InTouch 应用程序与 AVEVA Identity Manager 之间的连接丢失， Operations Control 连接体验功能将失效。您必须使用登录选项重新登录。
- 在成功进行身份验证之后出现连接丢失的情况下， 如果您尝试在 WindowMaker 中打开应用程序， WindowMaker 将在演示模式下运行。
- 在成功进行身份验证之后出现连接丢失的情况下， 如果您尝试在 WindowViewer 中打开应用程序， WindowViewer 将在只读模式下运行。
- 如果您从未登录过 AVEVA Identity Manager，并且出现连接丢失的情况，当您尝试启动 WindowMaker 或 WindowViewer 时， WindowMaker 将不会运行， 而 WindowViewer 将在演示模式下运行。

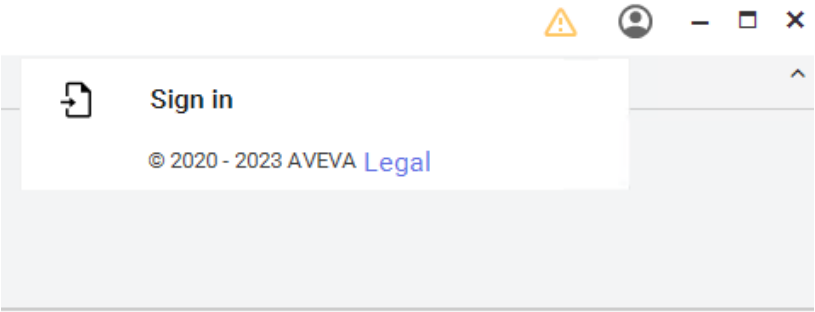
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

解决连接丢失问题

1. 如果与 CONNECT 或 AVEVA Identity Manager 的连接丢失，在“应用程序管理器”和 WindowMaker 中，将显示一个警告图标，并通知您连接状态。将光标悬停在警告图标上时，会出现一个含有连接丢失消息的工具提示。



2. 解决任何网络连接问题并单击配置文件图标，然后单击登录以再次登录。



恢复连接后，应用程序将重新启动，工具提示将不再可见。

许可和权利

InTouch 在 Operations Control 连接体验中的许可对于托管或独立应用程序而言是相同的。有两种可用的订阅：

AVEVA Operations Control Edge 订阅

当 InTouch 应用程序未配置为 ViewApp，并且许可证模式在配置器中设置为 Operations Control 连接体验时，需要此订阅才能启动 WindowViewer。

AVEVA Operations Control Supervisory 订阅

当 InTouch 应用程序配置为 ViewApp，并且许可证模式在配置器中设置为 Operations Control 连接体验时，需要此订阅才能启动 WindowViewer。

这两种订阅的会话数量不受限制，并且将启用 WindowMaker 的全部功能。

查看数据日志

审核日志使您能够查看帐户中的事件或操作日志。这些日志显示您的帐户中在所选时间段内发生的所有操作。您必须是帐户管理员，或者必须已被分配报告查看者角色，才能查看审核日志。

注意：审核日志中的数据从您的帐户具有审核功能时起就可用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

查看审核日志：

- 在站点导航菜单上，选择“审核”。此时会显示审核日志页面。

审核日志提供以下信息：

- 日期 – 执行操作的日期
- 消息 – 操作的详细信息，例如哪个用户执行了什么操作
- 功能 – 操作所影响的功能

注意：在用户界面上显示数据时使用浏览器的本地时间。但是，在将数据导出到 .csv 文件时使用 UTC 时区。

如需详细信息，请参阅 帮助。

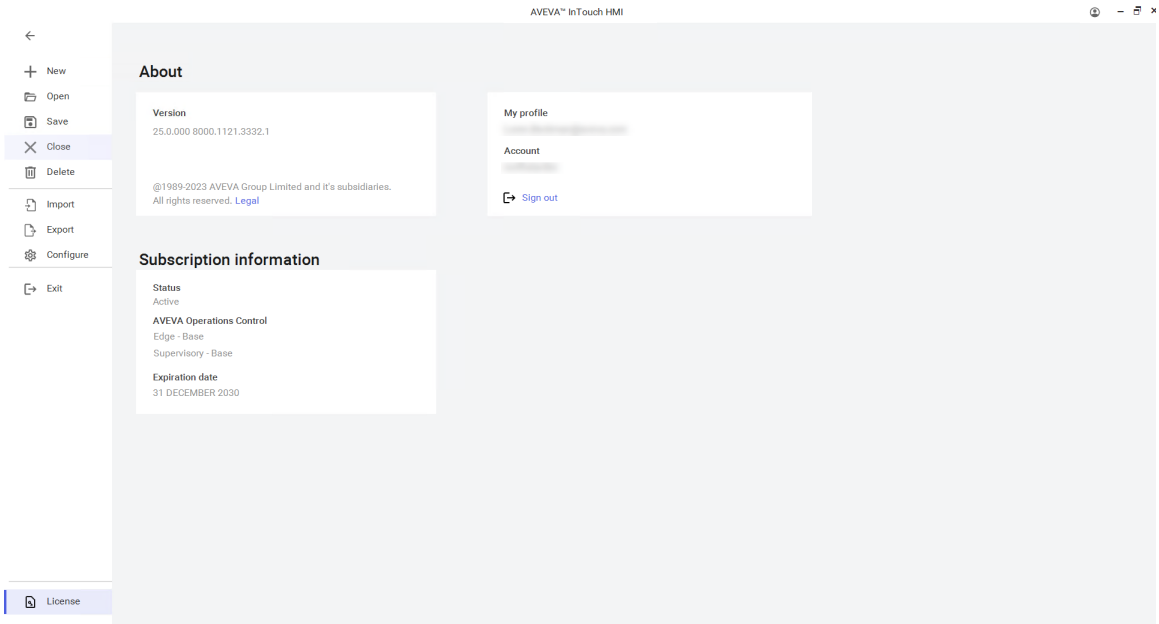
查看订阅信息

您可以查看 WindowMaker 或 WindowViewer 所使用的当前订阅的具体信息。

查看 WindowMaker 订阅信息

1. 打开 WindowMaker。
2. 在文件菜单上，选择屏幕左下角的许可证。

   此时出现关于屏幕。关于屏幕会显示 WindowMaker 版本和订阅信息。

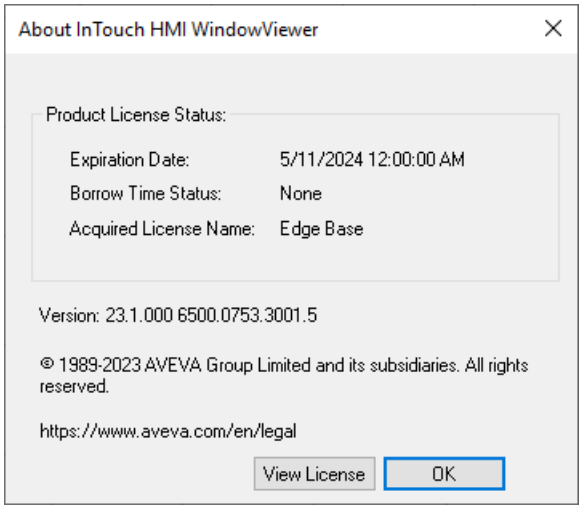


查看 WindowViewer 订阅信息：

1. 打开 WindowViewer。
2. 选择文件，然后单击关于 **WindowViewer**。

   此时出现关于 **WindowViewer** 对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 选择**查看许可证**以查看订阅的详细信息。

启动并运行 WindowMaker

本节介绍 WindowMaker 在 Operations Control 连接体验中的行为。

在 Operations Control 连接体验中使用 WindowMaker

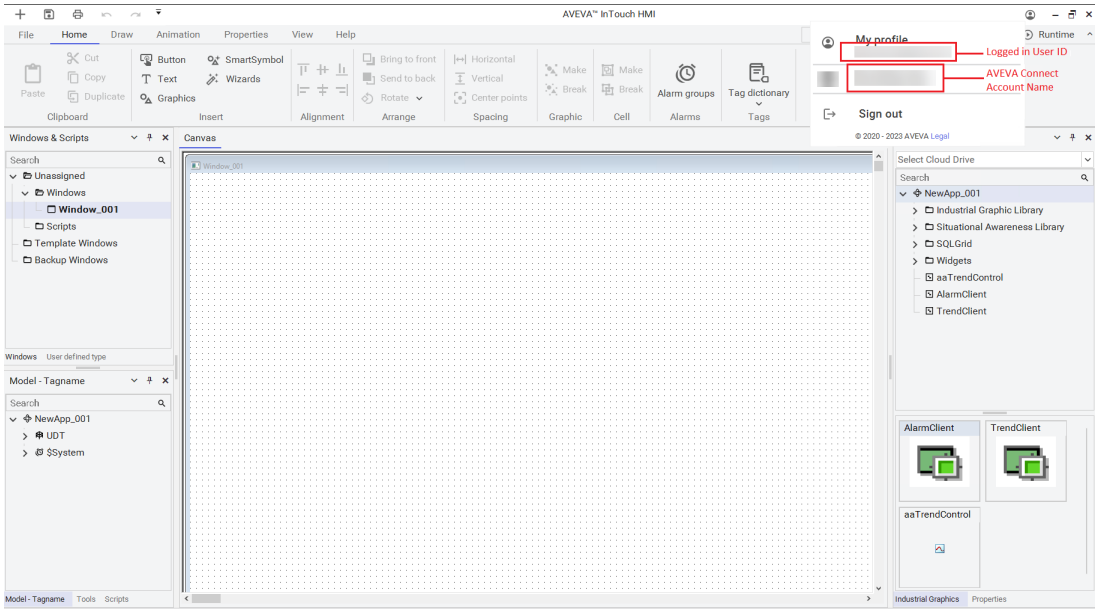
1. 启动 WindowMaker。

   如果您是首次在此节点上启动 WindowMaker，而且之前在同一节点上未启动过任何其它 Operations Control 产品并对其进行身份验证，那么系统会提示您在 Web 浏览器或嵌入式浏览器上（根据您在配置器的系统管理服务器中配置**身份验证选项**卡时做出的选择）使用 AVEVA Identity Manager 进行身份验证。如果您已经在此节点上启动 WindowMaker 并对其进行身份验证，或者之前在同一节点上启动了任何其它 Operations Control 产品并进行身份验证，那么将使用单一登录对您进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

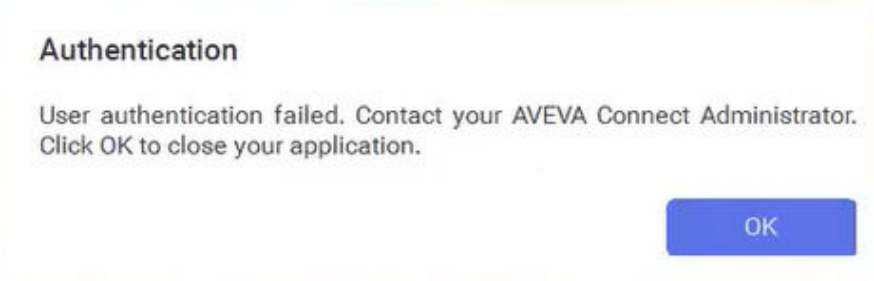
2. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。

   成功登录后，如果您有权使用 Edge 或 Supervisory 订阅，WindowMaker 会在标题栏上的用户配置文件中显示登录信息。“我的个人资料”将显示登录的用户 ID 和 CONNECT 帐户的名称。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 如果未提供有效凭据，或取消了身份验证，系统将以一条错误消息提示您。请与 CONNECT 管理员联系以获取凭据，然后尝试重新登录。单击确定后，WindowMaker 将关闭。



- 如果您关闭为使用 AVEVA Identity Manager 进行身份验证而启动的嵌入式浏览器登录页面，将立即显示身份验证失败对话框。
- 如果您不提供凭据或关闭为使用 AVEVA Identity Manager 进行身份验证而启动的 Web 浏览器，那么在 3 分钟后将显示身份验证失败对话框。

4. 当 WindowMaker 成功运行时，将检查是否获得有效的 AIM 和 CONNECT 刷新令牌。

   如果收到有效的 AIM 令牌，并且 AVEVA Identity Manager 与 CONNECT 断开连接，内部部署功能将起作用。您无法访问 CONNECT 工业图形驱动器。必须登录 AVEVA Identity Manager 才能访问 CONNECT 工业图形驱动器。

启动并运行 WindowViewer

本节介绍 WindowViewer 在 Operations Control 连接体验中的行为。

在 Operations Control 连接体验中使用 WindowViewer：

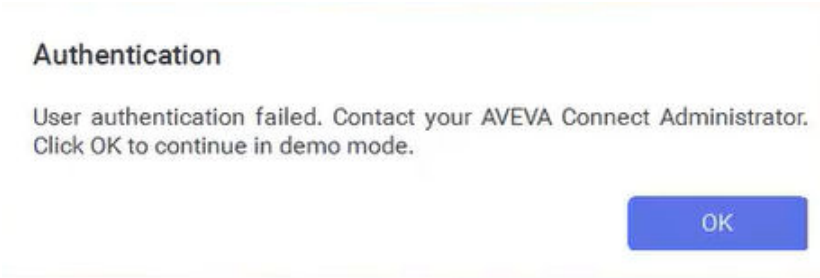
1. 启动 WindowViewer。

   如果您是首次在此节点上启动 WindowViewer，而且之前在同一节点上未启动过任何其它 Operations Control 产品并对其进行身份验证，那么系统会提示您在 Web 浏览器或嵌入式浏览器上（根据您在配置器的系统管理服务器中配置**身份验证选项**卡时做出的选择）使用 AVEVA Identity Manager 进行身份验

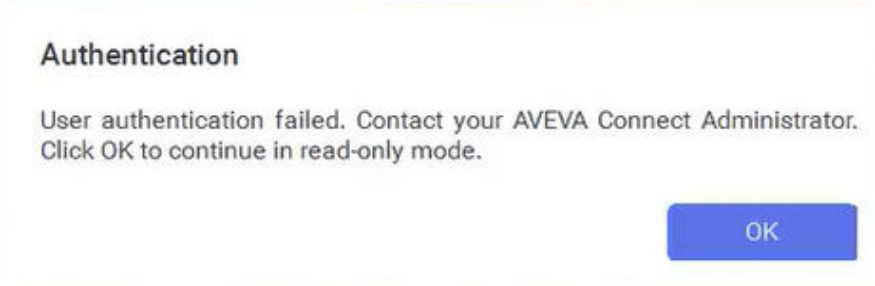
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

证。如果您已经在此节点上启动 WindowViewer 并对其进行身份验证，或者之前在同一节点上启动了任何其它 Operations Control 产品并进行身份验证，那么将使用单一登录对您进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

2. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。
3. 如果身份验证不成功，并且您之前从未登录，那么 WindowViewer 将在演示模式下启动。



4. 如果身份验证不成功或您已取消身份验证，但您之前在该节点上已成功进行过身份验证并登录至少一次，那么 WindowViewer 将在只读* 模式下启动。
   - 如果您关闭为使用 AVEVA Identity Manager 进行身份验证而启动的嵌入式浏览器登录页面，将立即显示身份验证失败对话框。
   - 如果您不提供凭据或关闭为使用 AVEVA Identity Manager 进行身份验证而启动的 Web 浏览器，那么在 3 分钟后将显示身份验证失败对话框。



5. 如果您与 AVEVA Identity Manager 断开连接，WindowViewer 将在只读* 模式下运行。
6. 成功登录后，如果您有 Edge 或 Supervisory 订阅，WindowViewer 会在标题栏上显示登录信息。
7. 如果您成功登录，但没有 Edge 或 Supervisory 订阅，那么：
   - 如果应用程序配置了 64 个或更少的标记，它将以演示模式打开。如果应用程序配置的标记数超过 64 个，它将不会以演示模式打开，并将退出 WindowViewer。
8. 当 WindowViewer 成功运行时，将检查是否获得有效的 AVEVA Identity Manager 和 CONNECT 刷新令牌。如果仅收到有效的 AVEVA Identity Manager 令牌，并且 AVEVA Identity Manager 与 CONNECT 断开连接：
   - WindowViewer 将在只读模式下运行。不会显示任何警告图标。您必须使用 GetTokenConnectionstatus() 函数脚本来了解连接的状态。您可以使用脚本编写方法来配置表示用户界面及其适当状态的图形。当出现连接丢失的情况时，配置的脚本将会运行，并且 WindowViewer 用户将看到与当前连接状态相关联的相应用户界面。如需有关 GetTokenConnectionstatus() 脚本的详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发指南》(ITBuild.pdf) 的 GetTokenConnectionStatus() function 一节。

注意：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 在 Operations Control 连接体验中，将会禁用 WindowMaker 中文件 > 配置 > **WindowViewer** > 窗口下的配置用户和更改密码选项。这些选项不适用于 Operations Control 连接体验，因此将不允许在 WindowMaker 中配置它们。缺省条件下，它们将处于取消选中和禁用状态，并且不能在 WindowViewer 中使用。

- WindowViewer 即服务不会使用来自 CONNECT 的订阅，而是使用传统的永久许可证。

***只读模式：**

只读表示无法写入 I/O 标记，但您仍然能够写入到内存标记。在只读模式下，您只能查看屏幕，无法进行任何更改。在此模式下，即使身份验证不成功，您也可以在 WindowViewer 中运行应用程序。这种只读模式是 WindowViewer 上的一种实现方式，不同于通常商业支持的只读模式。这种只读模式不适用于 InTouch 产品上的只读启用设置，并且与只读许可证无关。

如果您之前至少登录过一次且订阅尚未过期，并且有以下任一情况，WindowViewer 将在只读模式下运行：

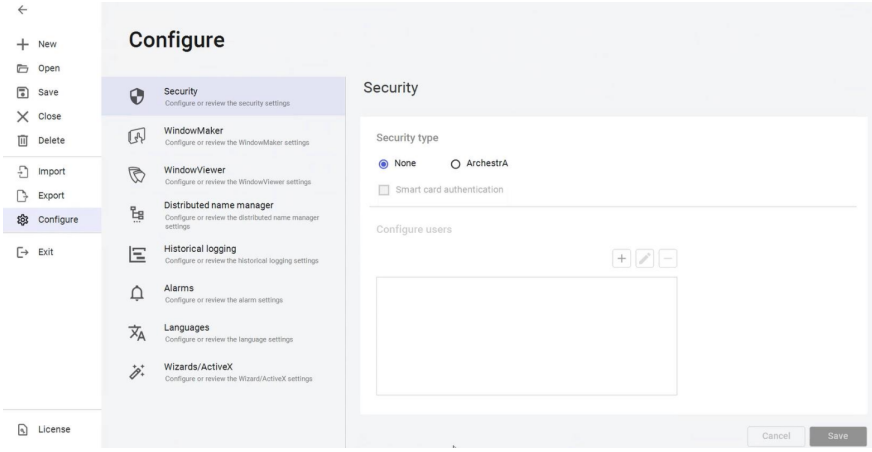
- 身份验证不成功
- 您不想进行身份验证
- 网络故障或超时

当订阅的合同日期结束时，WindowViewer 将在演示模式下运行。

在 Operations Control 连接体验中配置应用程序安全性

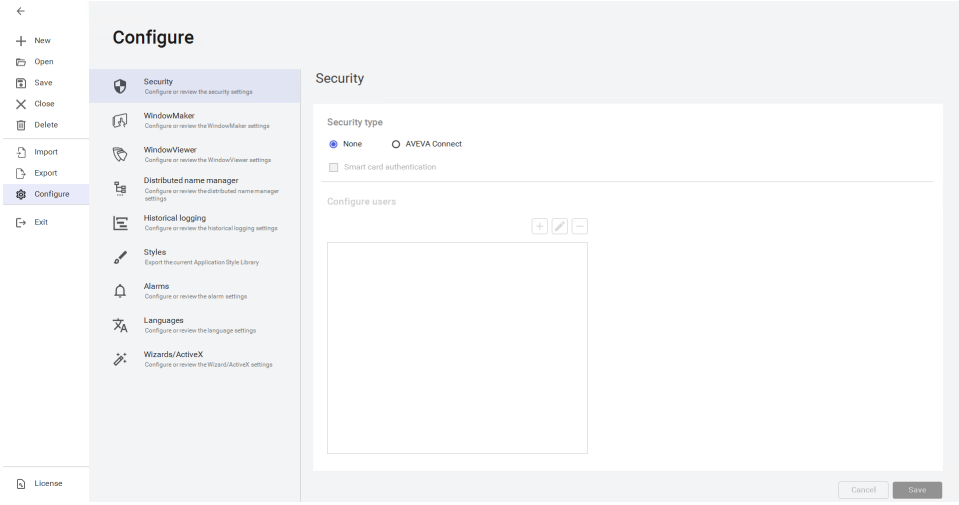
您必须在 WindowMaker 中配置以下任一安全性类型，才能在 Operations Control 连接体验模式下工作：

- 在托管应用程序中配置安全性
  - 无
  - ArchestrA（这需要在 System Platform IDE 上的身份验证模式选项卡下配置基于 CONNECT 的安全组配置。如需详细信息，请参阅在托管应用程序中配置安全性）



- 在独立应用程序中配置安全性
  - 无
  - CONNECT

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

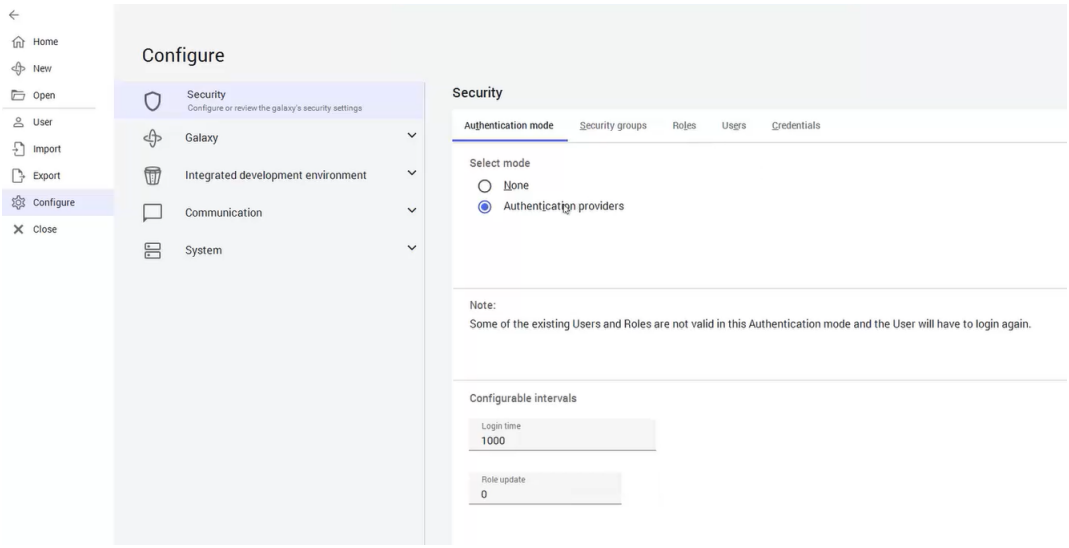


启用 Operations Control 连接体验模式并配置不兼容的安全性类型后，如果您打开 WindowViewer 或 WindowMaker，系统将显示一条提示消息，要求您使用上述任一支持的安全性类型重新配置 WindowMaker 中的安全性。WindowViewer 将在单击确定时关闭，并且在 WindowMaker 中可以选择编辑安全性类型。

在托管应用程序中配置安全性

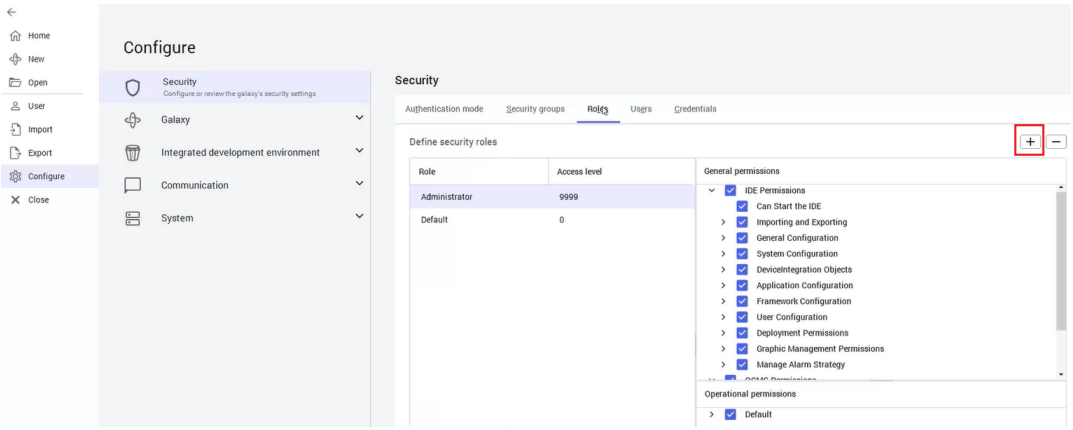
在托管应用程序中将安全性配置为 ArchestrA：

1. 启动 System Platform IDE。
2. 转到 **Galaxy** > 配置 > 安全性来打开安全性页面。在身份验证模式选项卡中，选择身份验证供应器选项。

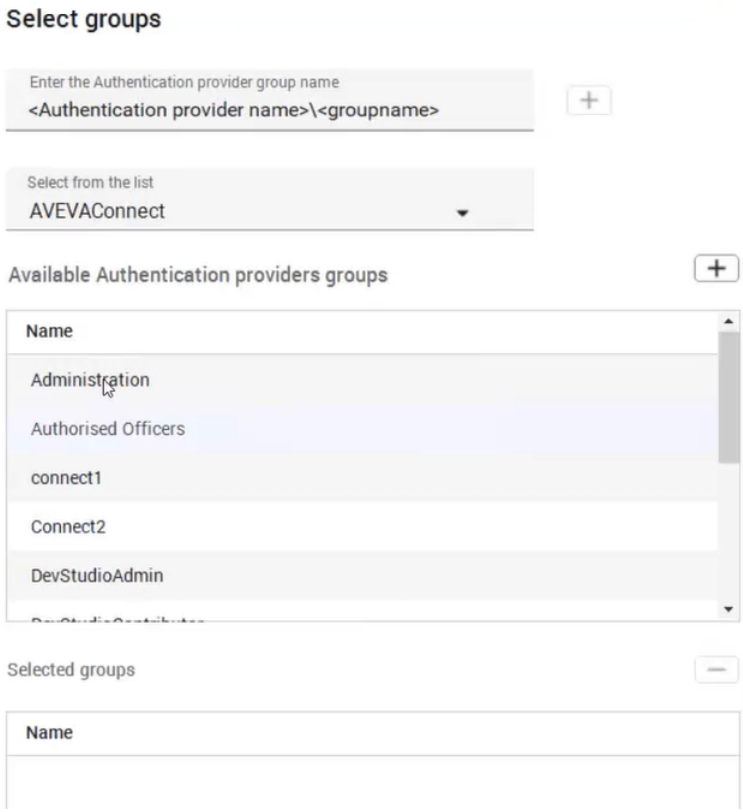


3. 要将访问级别映射到 CONNECT 组，请转到配置 > 安全性 > 角色，然后单击加号图标。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

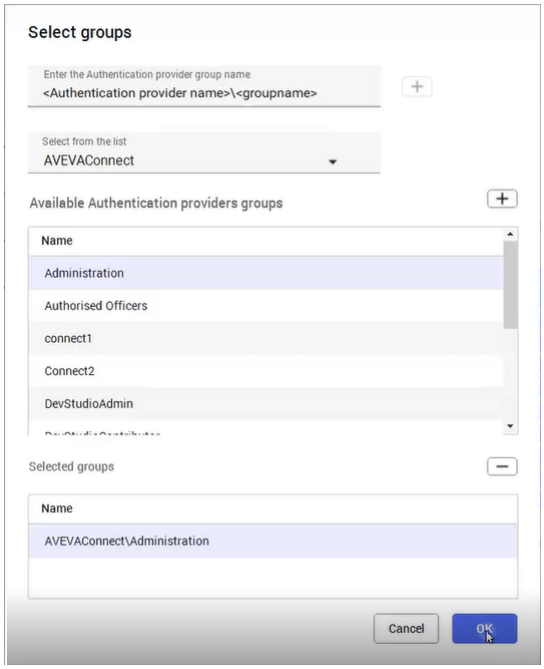


4. 验证在从列表中选择下拉列表中选择了 **CONNECT**。此时会显示您的 CONNECT 帐户中可用组的列表。



5. 从可用列表中添加所需的组，它们将显示在“所选组”列表中。在完成后单击确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



6. 根据适用情况配置组的**访问级别**、**一般权限**和**操作权限**，然后**单击保存**。
7. **选择** ViewApp 并启**动** WindowMaker。
8. 在 WindowMaker 中，在**文件** > **配置** > **安全性**下**选择 ArchestrA** 作为安全性**选项**，然后**单击保存**。

要在托管应用程序中将安全性配置为“无”：

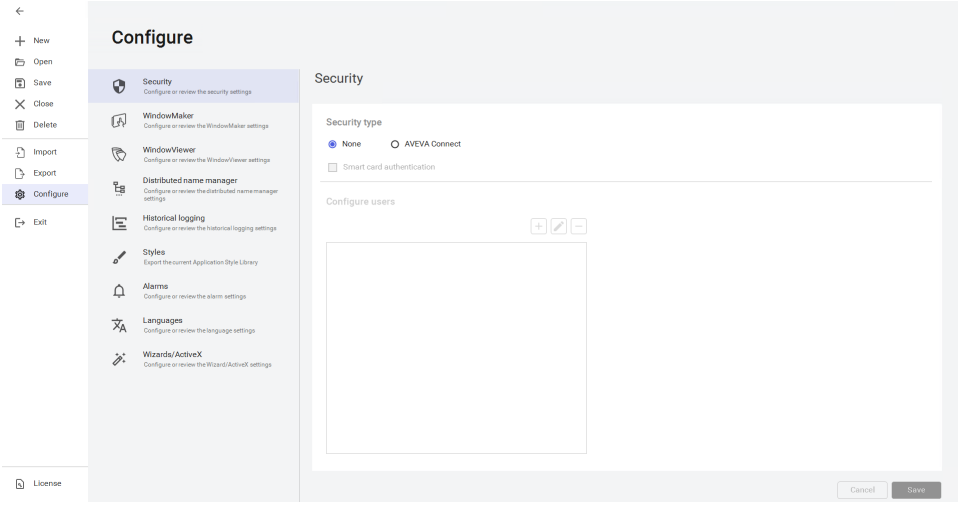
1. 启**动** System Platform IDE。
2. **选择** ViewApp 并启**动** WindowMaker。
3. 在**文件** > **配置** > **安全性**下**选择无**作为安全性**选项**，然后**单击保存**。

在独立应用程序中配置安全性

在独立应用程序中配置安全性：

1. 在 WindowMaker 中，在**文件** > **配置** > **安全性**下**选择无**或 CONNECT 作为安全性**选项**，然后**单击保存**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



启用 Operations Control 连接体验并配置不兼容的安全性类型后，如果您打开 WindowViewer 或 WindowMaker，系统将显示一条提示消息，要求您使用上述任一支持的安全性类型重新配置 WindowMaker 中的安全性。WindowViewer 将在单击确定时关闭，并且在 WindowMaker 中您可以选择编辑安全性类型。Operation Control 连接体验不支持 **InTouch** 和操作系统安全类型，而且您无法在非连接体验中使用 CONNECT 安全类型。

2. 要在独立 InTouch 中将访问级别映射到组，请使用脚本函数 **AddPermission()**。

   AddPermission() 方法在 Operations Control 连接体验中仅接受两个参数：

   - CONNECT 组
   - 访问级别

   Operations Control 连接体验中 AddPermission() 的脚本函数：

   ```
   DiscreteTag=AddPermission("", "CONNECT group", AccessLevel);
   ```

   如果运行时用户是 CONNECT 中多个组的成员，则访问级别将由具有最高访问级别的组确定。

在混合模式下操作

本节介绍了安全性类型在 Operations Control 连接体验中不兼容的情况下 WindowMaker 和 WindowViewer 的行为。

使用不兼容的安全性类型启动 WindowMaker：

1. 启动 WindowMaker。

   如果您是首次在此节点上启动 WindowMaker，而且之前在同一节点上未启动过任何其它 Operations Control 产品并对其进行身份验证，那么系统会提示您在 Web 浏览器或嵌入式浏览器上（根据您在配置器的系统管理服务器中配置身份验证选项卡时做出的选择）使用 AVEVA Identity Manager 进行身份验证。如果您已经在此节点上启动 WindowMaker 并对其进行身份验证，或者之前在同一节点上启动了任何其它 Operations Control 产品并进行身份验证，那么将使用单一登录对您进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

2. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。

3. 在成功登录并拥有有效权利后，如果安全性类型不兼容，系统将提示您配置安全性。

   注意：在 Operation Control 连接体验中不支持 **InTouch** 和 **OS** 安全性类型。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

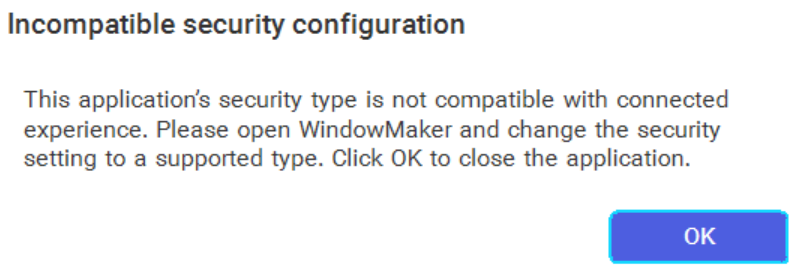
4. 单击是以继续并更改安全性类型，或单击否退出。

   如果您选择更改安全性类型，请在 WindowMaker 中配置安全性。如需有关如何配置安全性的信息，请参阅在 Operations Control 连接体验中配置应用程序安全性。

   注意：您必须检查您的脚本函数是否与 AVEVA Operations Control 连接体验兼容。如需详细信息，请参阅支持的 InTouch HMI 功能一节。

使用不兼容的安全性类型启动 WindowViewer：

1. 启动 WindowViewer。
2. 系统会提示您在 Web 浏览器中使用 AVEVA Identity Manager 进行身份验证。
3. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。
4. 在成功登录并拥有有效权利后，如果安全性类型不兼容，系统将提示您在 WindowMaker 中配置安全性。单击确定以退出 WindowViewer。



   注意：在 Operation Control 连接体验中不支持 **InTouch** 和 **OS** 安全性类型。

5. 在 WindowMaker 中配置安全性类型。如需有关如何配置安全性的信息，请参阅在 Operations Control 连接体验中配置应用程序安全性。
6. 然后，启动 WindowViewer。

启动并运行应用程序管理器

本节介绍“应用程序管理器”在 Operations Control 连接体验中的行为。

在 Operations Control 连接体验中使用“应用程序管理器”

1. 启动“应用程序管理器”。

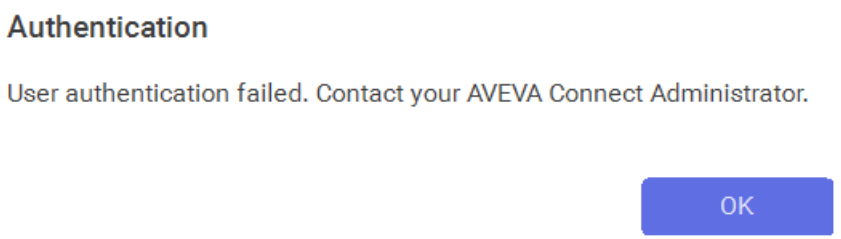
   如果您是首次在此节点上启动“应用程序管理器”，而且之前在同一节点上未启动过任何其它 Operations Control 产品并对其进行身份验证，那么系统会提示您在 Web 浏览器或嵌入式浏览器上（根据您在配置器的系统管理服务器中配置身份验证选项卡时做出的选择）使用 AVEVA Identity Manager 进行身份验证。如果您已经在此节点上启动“应用程序管理器”并对其进行身份验证，或者之前在同一节点上启动了任何其它 Operations Control 产品并进行身份验证，那么将使用单一登录对您进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

2. 输入注册用于登录到 AVEVA Identity Manager 的帐户的电子邮件地址和相应的密码。
   - 身份验证成功后：
     - 您的后续登录将通过单一登录进行身份验证。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- “应用程序管理器”功能区显示已登录用户的信息。
- 您将能够发布标记、访问 AVEVA 工业图形驱动器和凭证管理器、在只读或读/写模式下运行 View 应用程序，以及执行 DBLoad。

- 如果身份验证不成功或被取消：



  - “应用程序管理器”将在没有任何已登录用户的情况下保持运行。
  - 当您尝试启动 WindowMaker 或 WindowViewer 时，必须使用 AVEVA Identity Manager 进行身份验证。
  - 您只能在只读模式下运行 WindowViewer。在只读模式下，您只能查看应用程序，无法进行任何更改。WindowMaker 将退出。
- 如果您关闭为使用 AVEVA Identity Manager 进行身份验证而启动的嵌入式浏览器登录页面，将立即显示身份验证失败对话框。
- 如果您不提供凭据或关闭为使用 AVEVA Identity Manager 进行身份验证而启动的 Web 浏览器，那么在 3 分钟后将显示身份验证失败对话框。

从应用程序管理器中选择并运行应用程序

本节介绍在 Operations Control 连接体验中使用“应用程序管理器”打开应用程序的行为。

在 Operations Control 连接体验中使用“应用程序管理器”打开应用程序：

1. 启动“应用程序管理器”。要启动“应用程序管理器”，请单击开始，指向程序，指向 **AVEVA InTouch HMI**，然后单击 **InTouch HMI 应用程序管理器**。

   如果您是首次在此节点上启动“应用程序管理器”，而且之前在同一节点上未启动过任何其它 Operations Control 产品并对其进行身份验证，那么系统会提示您在 Web 浏览器或嵌入式浏览器上（根据您在配置器的系统管理服务器中配置**身份验证选项**卡时做出的选择）使用 AVEVA Identity Manager 进行身份验证。如果您已经在此节点上启动“应用程序管理器”并对其进行身份验证，或者之前在同一节点上启动了任何其它 Operations Control 产品并进行身份验证，那么将使用单一登录对您进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

   此时会打开 AVEVA 应用程序管理器。

2. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。一旦成功进行身份验证或身份验证不成功，“应用程序管理器”将会启动。

3. 选择所需的应用程序来运行。

   如果在上一步中身份验证不成功，系统将提示您使用 AVEVA Identity Manager 进行身份验证。

   身份验证成功后，如果应用程序的安全性类型与 Operations Control 连接体验不兼容，系统将显示一条消息，提示您安全性类型不兼容，是要更改安全性类型还是要退出。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4. 如果单击是，应用程序将在 WindowMaker 中打开。
5. 配置安全性类型。

如需有关安全性类型的详细信息，请参阅在 Operations Control 连接体验中配置应用程序安全性。

启动并运行 InTouch Access Anywhere

本节介绍 InTouch Access Anywhere 在 Operations Control 连接体验中的行为。

在 Operations Control 连接体验中使用 InTouch Access Anywhere

1. 使用 Web 浏览器启动 InTouch Access Anywhere，然后转到

   `http://ITAA_Server_Node_Name:8080/`

   或

   `http://ITAA_Server_IP_Address:8080/`

1. 在登录页面中，输入您的用户名、密码，然后从应用程序名称字段的下拉列表中选择要查看的 InTouch 应用程序。
2. 单击连接以启动 InTouch Access Anywhere 浏览器会话。

   如果您是首次在此节点上启动 InTouch Access Anywhere，而且之前在同一节点上未启动过任何其它 Operations Control 产品并对其进行身份验证，那么系统会提示您在 Web 浏览器或嵌入式浏览器上（根据您在配置器的系统管理服务器中配置身份验证选项卡时做出的选择）使用 AVEVA Identity Manager 进行身份验证。出于安全原因，建议使用嵌入式浏览器选项。如果您已经在此节点上启动 InTouch Access Anywhere 并对其进行身份验证，或者之前在同一节点上启动了任何其它 Operations Control 产品并进行身份验证，那么将使用单一登录对您进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

3. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。一旦成功进行身份验证，WindowViewer 将会启动。
4. 身份验证成功后：
   - 您的后续登录将使用单一登录进行身份验证。
   - WindowViewer 会验证是否有 Edge 权利访问权。
   - 如果已验证权利，WindowViewer 会在 CONNECT 网站的审核日志页面中记录 Edge 的使用情况。如需详细信息，请参阅 帮助。

启动网络应用程序开发 (NAD) 应用程序

网络应用程序开发 (NAD) 支持 Operations Control 连接体验。请注意，要使 NAD 在 Operations Control 连接体验中工作，您必须在系统的所有节点上启用 Operations Control 连接体验。

在 Operations Control 连接体验中使用 NAD 应用程序：

1. 启动 NAD 客户端。

   如果您是首次在此节点上启动 NAD，而且之前在同一节点上未启动过任何其它 Operations Control 产品并对其进行身份验证，那么系统会提示您在 Web 浏览器或嵌入式浏览器上（根据您在配置器的系统管理服务器中配置身份验证选项卡时做出的选择）使用 AVEVA Identity Manager 进行身份验证。如果您已经在此节点上启动 NAD 并对其进行身份验证，或者之前在同一节点上启动了任何其它 Operations

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Control 产品并进行身份验证，那么将使用单一登录对您进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

2. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。

   成功登录后，如果您有 Edge 或 Supervisory 订阅，那么将使用单一登录 (SSO) 进行登录，并且应用程序将运行。您无需在后续登录时重新输入凭据。

注意：

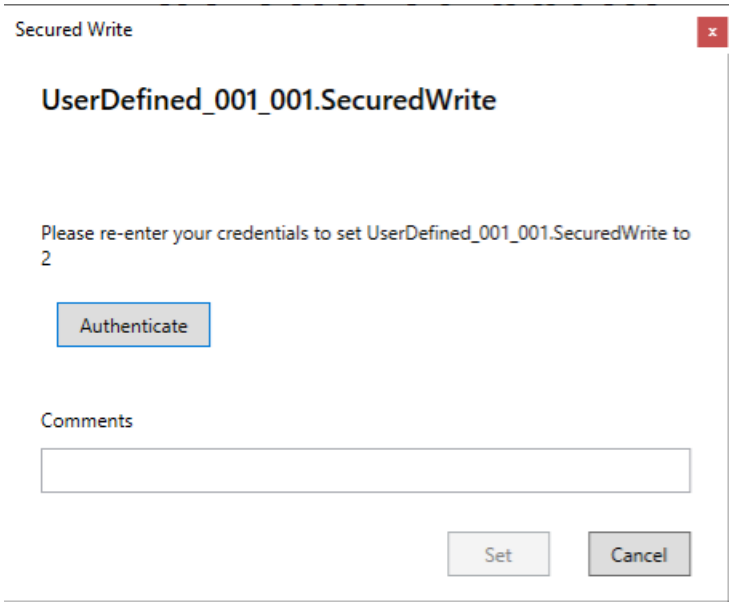
- 如果登录成功但没有有效权利，请与管理员联系以获取有效权利。如果登录不成功，系统将以一条错误消息提示您。请与管理员联系以获取有效的凭据。

- 如果计算机断开连接，您将无法再次进行身份验证，因此将无法运行应用程序。

执行安全和验证写入

执行安全写入：

1. 更改值。
2. 此时出现安全写入对话框。
3. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。

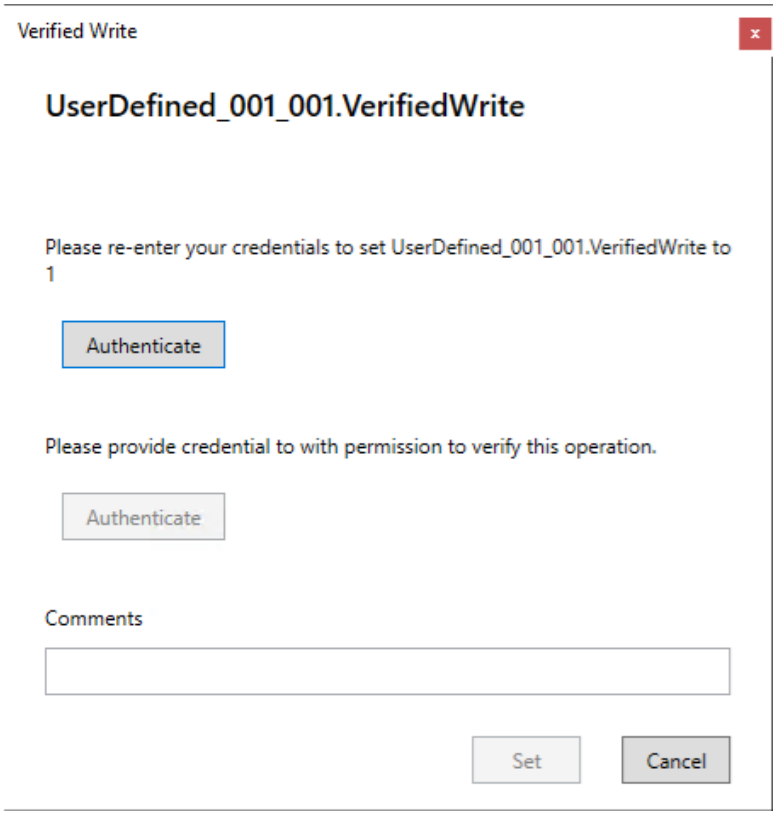


注意：安全写入操作不支持单一登录。也就是说，操作员必须明确地输入其凭据用于安全写入操作。

执行验证写入：

1. 更改一个值
2. 此时出现验证写入对话框。
3. 操作员单击身份验证，然后输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。
4. 验证者单击身份验证，然后输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



注意：验证写入操作不支持单一登录。也就是说，操作员和验证者都必须明确地输入其凭据用于验证写入操作。

启动并运行 Web 客户端（内部部署）

在配置器中启用 Operations Control 连接体验后，首次启动 Web 客户端时，系统将提示您使用 AVEVA Identity Manager 进行身份验证。您必须使用 AVEVA ID 和密码登录。在后续启动 Web 的过程中，您将通过单点登录 (SSO) 进行身份验证。如果您在启动其它 AVEVA 应用程序（例如“应用程序管理器”和 WindowMaker）时已经登录 AVEVA Identity Manager，并且您拥有有效的权利和权限，那么将使用 SSO 进行登录。请注意，仅当桌面应用程序使用系统浏览器进行身份验证时，桌面应用程序和 Web 客户端之间的 SSO 才会起作用。SSO 是特定于浏览器的。也就是说，如果您在一个浏览器（例如 Google Chrome）中输入了凭据，而在另一个浏览器（例如 Microsoft Edge）中打开 Web 客户端，或者以匿名模式打开 Web 客户端，必须再次输入凭据。

注意：如果您是从 System Platform 2023 升级到 System Platform 2023 R2，并且已经在 System Platform 2023 中配置了 AVEVA Identity Manager，那么在升级后，您必须通过切换到 Windows 身份验证，然后再切换到用户身份验证来重新配置用户身份验证，以使 Web 客户端能够在 Operations Control 连接体验中工作。也就是说，您必须先在**工业图形服务器 > 身份验证设置**下选择 **Windows 身份验证**，并单击**配置**。然后，再在**工业图形服务器 > 身份验证设置**下选择**用户身份验证**，并单击**配置**。

在 Operations Control 连接体验中使用 Web 客户端（内部部署）：

1. 要在 Operations Control 连接体验中启用 Web 客户端，请在配置器左侧窗格的**工业图形服务器 > 身份验证设置**下选择**用户身份验证**，然后单击**配置**。
2. 在浏览器中启动 Web 客户端。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果您是首次在此节点上启动 Web 客户端，而且之前在同一节点上未启动过任何其它 Operations Control 产品并对其进行身份验证，那么系统会提示您在 Web 浏览器或嵌入式浏览器上（根据您在配置器的系统管理服务器中配置“身份验证”选项卡时做出的选择）使用 AVEVA Identity Manager 进行身份验证。如果您已经在此节点上启动 Web 客户端并对其进行身份验证，或者之前在同一节点上启动了任何其它 Operations Control 产品并进行身份验证，那么将使用单一登录对您进行身份验证。如需详细信息，请参阅使用 AVEVA Identity Manager 进行身份验证一节。

3. 输入使用 CONNECT 帐户注册的用于登录到 AVEVA Identity Manager 的电子邮件地址和相应的密码。

   身份验证成功且具有有效权利后，Web 客户端将运行。您无需在后续登录时输入凭据，因为将使用单一登录 (SSO) 进行登录。

   注意：如果凭据不正确，系统将以一条错误消息提示您。请与管理员联系以获取凭据，然后重新输入正确的凭据。

无效权利的情形

本节介绍在权利无效时可能发生的不同情形。

- 如果您无权访问权利，WindowViewer 将在演示模式下启动。

  注意：如果应用程序配置了 64 个或更少的标记，那么将提供演示模式。如果应用程序配置了 64 个以上的标记，那么不会提供演示模式。

- 如果当前身份验证不成功，但您先前已在该节点上成功进行过身份验证并至少登录一次，则 WindowViewer 将在只读模式下启动。只读模式将继续运行，直到本地存储的过期日期过去。当过期日期过后，应用程序将在演示模式下运行。
- 如果您启动 WindowViewer 并取消身份验证，而且出现登录超时，那么 WindowViewer 将在演示模式或只读模式下启动。
- 如果您尝试进行身份验证但仅收到有效的 AVEVA Identity Manager 令牌，并且与 CONNECT 断开连接，那么内部部署功能将起作用，而云访问将失败。您需要登录才能重新获得云访问权限。
- 如果您尝试进行身份验证但未收到有效的 AVEVA Identity Manager 令牌，并且与 AVEVA Identity Manager 断开连接，那么 WindowViewer 将在演示模式和只读模式下启动。
- 当 WindowViewer 成功运行时，将检查是否获得有效的 AVEVA Identity Manager 和 CONNECT 刷新令牌。如果仅收到有效的 AVEVA Identity Manager 令牌，而 AVEVA Identity Manager 与 CONNECT 断开连接，那么内部部署功能将起作用，但云访问将失败。
- 如果与 AVEVA Identity Manager 断开连接，WindowViewer 将在演示模式或只读模式下启动。

不支持的功能

Web 客户端不支持以下功能：

- 在 Operations Control 连接体验中不支持 Windows 身份验证。
- 在 Operations Control 连接体验中不支持共享功能。
- Web 客户端在任何许可模式下均不支持冗余单点登录功能。

Operations Control 连接体验中 Web 客户端的许可和权利

如果您从其它许可模式切换到 Operations Control 连接体验，必须重新配置身份验证设置。

在 Operations Control 连接体验中，如果启用了 **InTouchView 应用程序复选**框，那么为启动 Web 客户端，您需要在帐户中拥有 Supervisory 订阅的权利。启动 Web 服务器后，如果修改 **InTouchView 应用程序复选**框，必须重新启动 CONNECT 工业图形 Web 服务器。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

启动并运行趋势控件

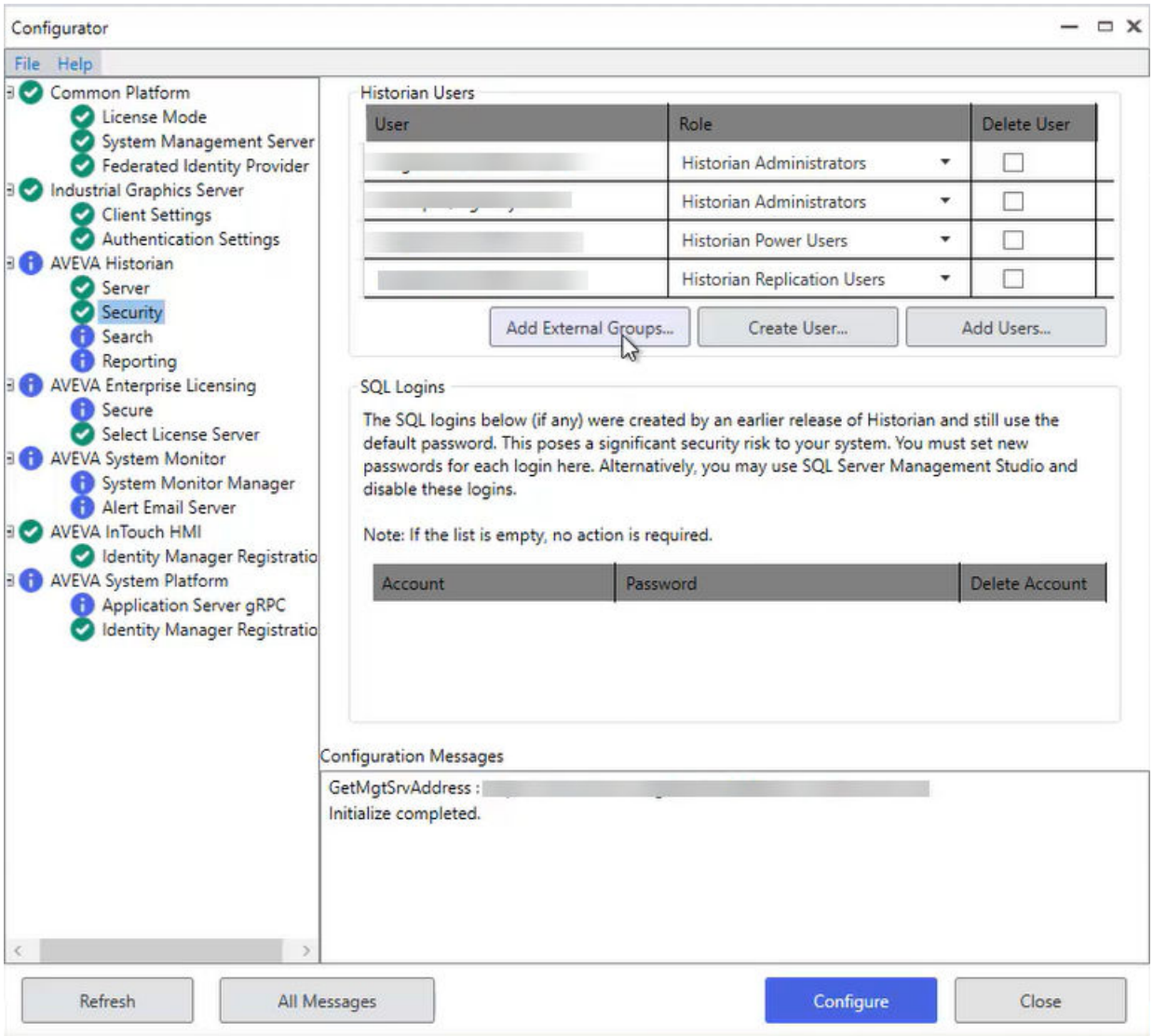
趋势控件利用单一登录功能，以便您可以从 WindowViewer 体验无缝单一登录。本节介绍趋势控件在 Operations Control 连接体验中的行为。

在 Operations Control 连接体验中使用趋势控件

配置器设置：

除了在配置 Operations Control 连接体验一节中提到的其它配置器设置之外，您还必须配置以下内容：

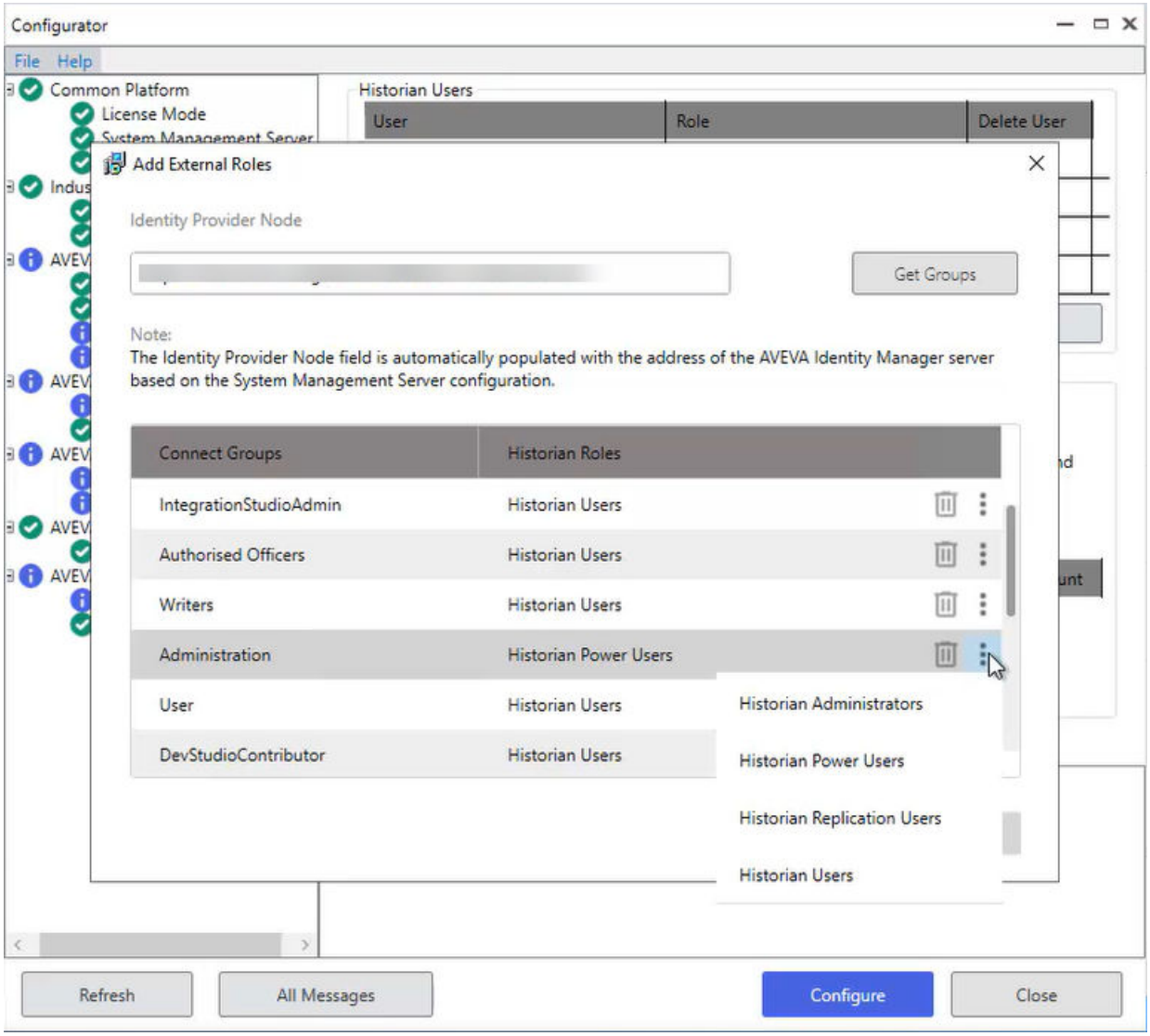
1. 配置 AVEVA Historian 服务器。如需详细信息，请参阅配置器帮助。
2. 在配置器中，转到 **AVEVA Historian** > 安全性。
3. 选择添加外部组。



4. 在添加外部角色窗口中，列出了 CONNECT 中的组。在 **Connect 组**列下，选择趋势控件用户所属的必需 CONNECT 组，然后单击垂直省略号（三个点）并选择 **Historian 管理员**或 **Historian 高级用户**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：为了使趋势控件能够在 Operations Control 连接体验中工作，趋势控件的用户必须属于 CONNECT 组，并且该组必须具有作为 **Historian 管理员**或 **Historian 高级用户** 的 **Historian 角色**。



5. 选择保存，然后选择配置。

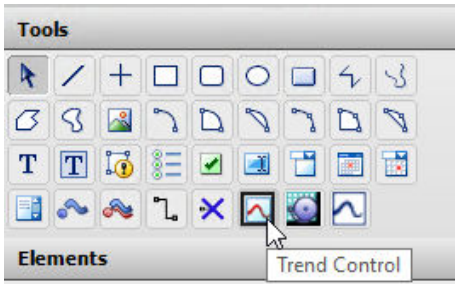
配置历史记录

配置配置器后，您必须在 WindowMaker 中配置**历史记录**。如需详细信息，请参阅《AVEVA™ InTouch HMI 应用程序开发指南》(ITBuild.pdf) 的配置常规记录属性存储到 Historian 一节。

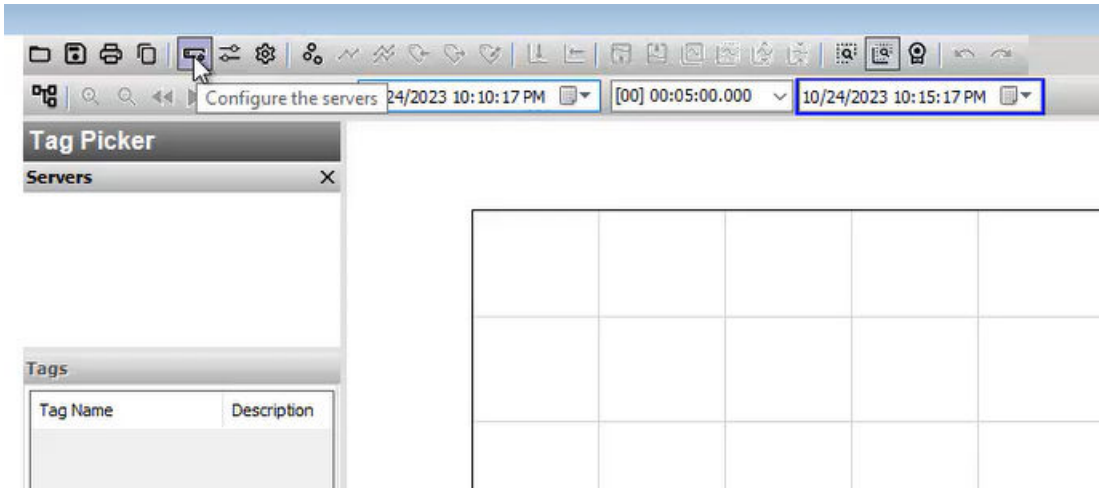
在服务器列表配置中将登录配置为单一登录：

1. 在 WindowMaker 的工业图形窗格下，使用鼠标右键单击并选择新建图形以创建图形。
2. 使用鼠标右键单击新创建的图形并选择打开，或双击该图形以在“工业图形编辑器”中将其打开。
3. 在工具窗格中，选择趋势控件并将其添加到画布中。保存并关闭图形。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

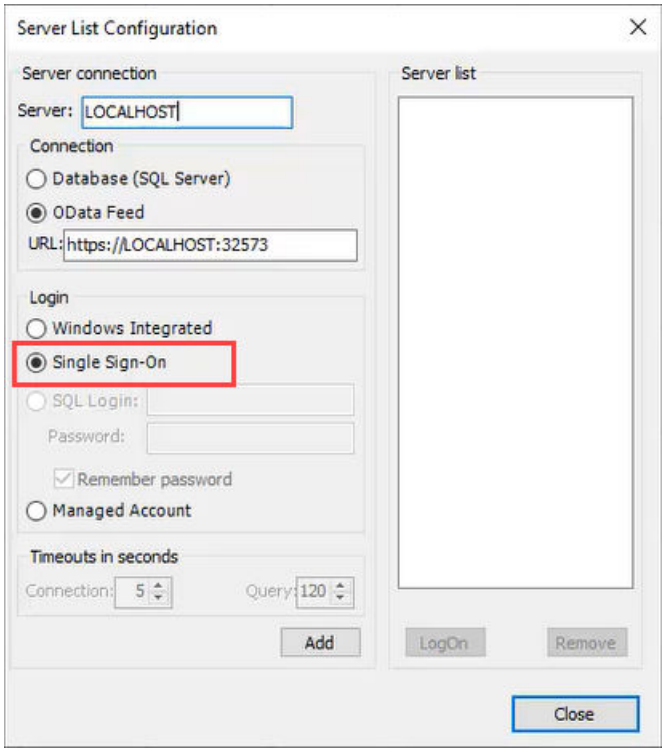


4. 在 WindowMaker 中，将新创建的图形添加到画布并保存。
5. 单击 WindowMaker 右上角的运行时，以在 WindowViewer 中打开该窗口。
6. 单击配置服务器。



7. 在服务器列表配置对话框中，如果输入服务器字段，**OData 源**下的 **URL** 字段会相应地进行更新。
8. 在登录部分下，选择单一登录，然后单击添加。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



此时会将该服务器添加到**服务器列表**中。

9. 选择登录。

从 AVEVA 应用程序注销

您可以通过单击配置文件图标下的**注销选项**，从 InTouch、ViewApp 或其它 Operations Control 连接体验应用程序中注销。从应用程序注销时：

- 注销是从 CONNECT 关闭会话。
- 当您注销桌面应用程序时，只会注销该特定的桌面应用程序。其它桌面应用程序将继续使用其当前登录的用户运行。
  - 如果您后续再启动桌面应用程序，系统将提示您重新进行身份验证。
- 使用嵌入式浏览器注销桌面应用程序时，Web 应用程序将继续运行。
- 关闭应用程序而不注销不会对会话执行任何操作。只会关闭该应用程序实例。会话仍位于 CONNECT 服务器上。当再次打开该应用程序时，您的会话可用，并且您将通过单一登录进行身份验证。
  - 如果会话已过期，那么在启动应用程序时，您将需要再次进行身份验证。
- 当您注销 WindowMaker 时，会话将关闭，并且 WindowMaker 也会退出。
- 当您注销 WindowViewer 时，WindowViewer 将在只读模式下运行。

支持的 InTouch HMI 功能

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

支持的客户端控件

在使用 AVEVA Identity Manager 身份验证的 Operations Control 连接体验模式中，支持以下 InTouch HMI 客户端控件。

- **报警客户端控件**：报警客户端控件利用 SSO 功能，以便您可以从历史块的视图应用程序体验无缝单一登录。通过身份验证并拥有权利的用户可以在 WindowViewer 中查看 Historian 报警。

  注意：为了使报警客户端控件能够在 Operations Control 连接体验中工作，您必须从 CONNECT 配置外部角色。

- **趋势控件**：趋势控件利用 SSO 功能，以便您可以从视图应用程序体验无缝单一登录。
- **趋势笔**：趋势笔利用 SSO 功能，以便您可以从视图应用程序体验无缝单一登录。通过身份验证并拥有权利的用户可以在 WindowViewer 中查看历史趋势。

支持的系统标记

在使用 AVEVA ID 的 Operations Control 连接体验中，支持以下 InTouch HMI 系统标记。

- **$AccessLevel**：$AccessLevel 系统标记已更新为与 AddPermission() 方法一起使用。如果运行时用户是 CONNECT 中多个组的成员，则访问级别将由具有最高访问级别的组确定。
- **$AccessTokenChanged**：$AccessTokenChanged 系统离散标记是一个只读标记。
- **$Operator**：$Operator 系统标记将显示从 CONNECT 登录的用户。
- **$OperatorName**：$OperatorName 系统标记将显示从 CONNECT 登录的用户。

脚本编写方法

InTouch HMI 还支持在 Operations Control 连接体验中使用脚本编写方法。已添加或修改以下脚本编写方法来支持 Operations Control 连接体验。

- GetAccessToken()
- GetAccessSecureToken()
- AddPermission()

  AddPermission() 方法在 Operations Control 连接体验中仅接受两个参数：

  - CONNECT 组
  - 访问级别

  Operations Control 连接体验中 AddPermission() 的脚本函数：

  ```
  DiscreteTag=AddPermission("", "CONNECT group", AccessLevel);
  ```

  如果运行时用户是 CONNECT 中多个组的成员，则访问级别将由具有最高访问级别的组确定。

- GetTokenConnectionstatus()
- PostLogonDialog() 函数
- SignedWrite()
- IsAssignedRole()
- Logoff()

注意：在 Operations Control 连接体验中不支持 ChangePassword() 脚本函数，将被禁用。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

CONNECT 工业图形驱动器

Operations Control 连接体验为 WindowMaker 中的 CONNECT 工业图形驱动器提供单一登录功能。如果您在连接体验下在 CONNECT 中进行身份验证，则可在启动 Window 时查看您的工业图形驱动器。这可让您在 WindowMaker 中开始使用 CONNECT 工业图形驱动器，而无需重新进行身份验证。因此，在 Operations Control 连接体验中，您将不会在 WindowMaker 的文件菜单下看到 **CONNECT** 选项卡。

- 在 Operations Control 连接体验中，当您启动 WindowMaker 时，如果您使用凭据或通过单一登录成功向 CONNECT 进行了身份验证，并拥有有效的订阅，那么 WindowMaker 将打开。
  - 您可以查看图形工具箱中提供的 CONNECT 工业图形驱动器。还可以使用下拉菜单导航到帐户中的其它租户。
  - 您可以根据自己在 CONNECT 工业图形驱动器上拥有的权限与工业图形云驱动器进行交互。

注意：您必须拥有“内容参与者”角色，才能访问 CONNECT 工业图形驱动器。此要求适用于 Operations Control 连接体验和非连接体验使用。

安全写入/验证写入

- WindowViewer 支持安全写入和验证写入，并通过 AVEVA ID 进行身份验证。
- 您必须通过 CONNECT 进行身份验证才能执行安全写入或验证写入动作。
- 安全写入或验证写入操作不支持 SSO。

网络应用程序开发 (NAD)

支持将 NAD 用于 Operations Control 连接体验和通过 AVEVA Identity Manager 进行身份验证。

InTouch Access Anywhere

支持将 InTouch Access Anywhere 用于 Operations Control 连接体验和通过 AVEVA Identity Manager 进行身份验证。

Web 客户端

- 在启动 Web 客户端 URL 时，系统将提示您使用 CONNECT 进行身份验证。
- 成功登录后，如果您具有有效的权利和权限，Web 客户端即会运行。

不支持的 InTouch HMI 功能

在此发行版中，Operations Control 连接体验中不支持以下 InTouch 产品和功能：

- Insight 发布者
- 小组件

在 Operations Control 连接体验中不支持以下 InTouch 产品和功能：

- Alarm DB Logger
- Alarm DB 清除/归档
- Alarm DB 恢复
- Alarm Hot Backup Manager
- 智能卡
- 应用程序管理器凭据管理器

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

章 37 在非 Operation Control 模式下使用 InTouch

本节介绍不受 AVEVA Operations Control 连接体验身份验证和权利控制的功能。

InTouch HMI 中的许可证

InTouch HMI 使用“AVEVA Enterprise 许可证服务器”使许可证可用于 InTouch。“AVEVA Enterprise 许可证管理器”管理一个或多个“许可证服务器”。

要使许可证可用于 InTouch HMI，请完成以下步骤：

1. 导入购买该许可证时收到的授权 XML 文件。
2. 使用**许可证管理器**界面，选择要在许可证服务器上激活授权上的哪些许可证。
3. 许可证激活之后，就会在 WindowMaker 或 WindowViewer 启动时变为可用。

激活的许可证会出现在“许可证管理器”的许可证网格下。

在以下情况下，InTouch 会释放消耗的许可证并将其返回到许可证服务器：

- 运行 InTouch 的计算机关闭或
- InTouch 应用程序关闭

注意：如果 InTouch HMI 异常关闭，将不会返回许可证。必须重新启动 InTouch HMI，然后手动关闭才能释放许可证。

安装 InTouch HMI 的同时会安装“许可证管理器”和“许可证服务器”。缺省情况下，InTouch HMI 将指向您的本地“许可证服务器”。您可以在安装后配置程序中更改此配置。如需有关详细程序，请参阅《*AVEVA Enterprise 许可证指南*》。

InTouch 许可证基于应用程序运行时可以使用的可变标记数。您需要理解 InTouch 许可证方案如何计算标记。您可以使用一组函数来计算应用程序中预期的远程引用标记数。

重要事项：许可证数量可随时更改。

理解许可证标记计数

在 InTouch 应用程序运行时，标记句柄存储在内存数据库中。每个标记都必须有一个指定的句柄。标记句柄由 WindowViewer 初始化与使用，但在应用程序停止之后，标记句柄不会永久保存到磁盘。

InTouch HMI 现在支持无限制的标记数量。这意味着运行时数据库理论上可以存储无限制标记数量许可证，包括本地标记与引用远程标记源的标记。但是，我们仅测试了最多 300,000 个标记句柄 (300K)。

InTouch 应用程序使用的同时活动标记数永远不会超过这个运行时数据库中的可用句柄数。InTouch 许可证确定可以给多少个本地与远程标记指定运行时数据库中的句柄。

此外，应用程序中的最大活动标记数受限于运行时数据库的功能限制。实际最大标记计数小于理论最大标记句柄。最大潜在标记计数中需要减去一组常数。

- 保留无效标记句柄位，以指出 WindowViewer 运行时数据库中是否出现无效的句柄值。
- InTouch 10 包含 34 个系统标记，这些标记无法由用户自定义的标记来替换。如果将 7.11 或更早版本的应用程序移植到当前版本的 InTouch，则系统标记计数是 37。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 在配置时，会保留 4096 个数据库句柄以存储占位符标记。在配置期间导入窗口、脚本或符号时，占位符标记会指定给这个内存段。在运行时，所有这 4096 个占位符句柄都可以指定给远程引用标记。

InTouch 许可证选项基于应用程序中可以使用的最大本地标记和远程引用标记数。

如果 InTouch 标记许可证允许的标记数少于 60K，则实施粘滞标记许可方案。粘滞标记是一种远程标记引用，在 WindowViewer 接收到远程引用的数据改变通知时，它才进行绑定。WindowViewer 在运行时更新远程标记引用，最多可以达到 InTouch 许可证的最大限制。WindowViewer 不会更新超出许可证限制的任何额外远程标记引用。关闭窗口时，WindowViewer 不会减少远程引用标记计数。在应用程序运行时，每个远程引用标记计数都保持其粘滞性。

超出 InTouch 许可证的最大远程引用标记计数时，会出现一条消息。达到许可证最大值之后，与无效的远程引用标记关联的值不会在应用程序中进行更新。您必须停止后再重新启动应用程序，然后才可以打开包含一个或多个远程标记引用的其它窗口，这些引用尚未同根据许可证限制而统计的那些标记关联起来。

允许 60K 或更多标记的许可证意味着不实施粘滞标记，并且不限制可在 InTouch 应用程序内使用的标记数量或本地和远程标记组合的数量。

例如，使用 60K 许可证时，实现的本地与远程引用标记限制是：

- 可能的本地标记总数

  61404 = 65535 – (4096 + 37 + 1)

- 可能的远程引用标记总数

  最大值 = 65535 –（37 + 1 + 本地标记数）

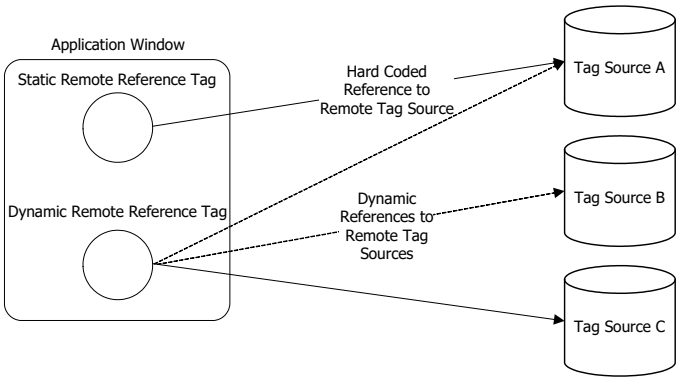
因此，使用 60K 许可证运行应用程序时，可以有效地使用每个潜在的远程引用来交换本地标记数据库中的每个用户自定义标记。在运行时配置中，可能的远程引用标记永远不会少于 4096 个。对于允许 100000 个标记的许可证，最大值 = 100000 – (37 + 1 + 本地标记数)

理解 InTouch 远程引用限制

InTouch 远程引用标记分为两种类型。*静态*远程引用在您从“标记名字典”中定义标记时明确指定到固定的远程地址。应用程序开始运行时，静态远程引用指定给标记数据库中的标记句柄。在应用程序运行时，静态远程引用标记计数保持其粘滞性。

*动态*远程引用在应用程序运行时解析目标地址。如果动态远程引用标记指定了数据库句柄，则通过在脚本中使用 .Reference 点域或 IOSetRemoteReference() 函数，可以在运行时更改目标地址。

下图显示 300K 许可证下运行的一个 InTouch 应用程序示例，它没有粘滞的远程引用标记。在应用程序运行时，静态远程引用标记计数会保持粘滞性。但是，动态远程引用标记计数仅用于活动标记源。以前与远程标记源的连接不粘滞，也不计入远程引用计数或总计标记计数。



© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

InTouch 60K 许可证不使用在动态远程标记引用数上施加限制的粘滞标记计数。这使得应用程序可以在运行期间动态访问 60K 以上的标记。动态远程引用的标记使用计数随着包含远程引用的窗口的打开与关闭而上下波动。但是，应用程序同时使用的活动标记数不能比运行时标记数据库所实施的限制更多。

动态引用标记计数仅在 WindowViewer 使用磁盘储存空间来保存运行中应用程序的内容时才上下波动。如果 WindowViewer 配置为缓存 InTouch 窗口或工业图形，则远程引用标记数可能不会减少，除非从缓存中删除窗口。

窗口不可见并不意味着远程标记引用尚未与其来源绑定。但是，如果完全禁用窗口缓存，并且未指定任何高优先级窗口，则 WindowViewer 的运行方式将非常类似于旧有的“始终从磁盘加载”情形。在这种情况下，所有窗口在关闭后都将从内存中删除，并且动态远程标记引用在 60K 许可证环境下回收重用。

从 I/O 标记的远程引用不包含在 InTouch 许可证的粘滞远程引用计数中。I/O 标记的远程引用可以更改无限次，而不会根据粘滞远程引用限制进行统计。

在发生故障转移而切换到辅助标记源时，应用程序可以从辅助标记源访问相同的项目，而不增加许可的标记计数。在故障转移之后，从辅助标记源访问新的项目会增加标记计数。进行故障返回切换到主标记源之后，这些项目仍可以访问。

在标记计数达到许可的最大数目时，无论是从主标记源还是从辅助标记源访问，都不能激活更多的项目。

可用于 InTouch HMI 的许可证

InTouch HMI 提供不同类型的许可证来管理各种情形。许可证根据各种参数来确定，例如：

1. **控制台类型**：指定控制台类型；远程桌面服务/终端服务或非 RDS 节点。RDS/TSE 是运行在配置了终端服务器的计算机上的控制台，而非 RDS 是运行在未配置终端服务器的计算机上的控制台。如需有关详细信息，请参阅部署和使用终端服务与远程桌面服务。
2. **访问类型**：指定将节点配置成的访问类型 - 只读或读写。如需有关详细信息，请参阅配置远程会话中运行的应用程序的用户访问权限。
3. **数据源**：指定应用程序将使用的数据源 - Galaxy 或 InTouch 标记服务器。如需详细信息，请参阅 InTouchView 应用程序。

InTouch HMI 无限制 RDS 许可证

InTouch HMI 无限制 RDS 许可证由 WindowViewer 用于无限制客户端会话，仅用于启用了 RDS 的计算机。当 WindowViewer 获取许可证时，应用程序即被授权。读取/写入访问权限取决于远程访问配置。同一无限制许可证将同时提供只读和读写访问权限。如果未获取许可证，则授权将取决于现有 RDS 处理和远程访问配置。

RDS 和非 RDS 环境中的 InTouch 许可证

如果 InTouch 应用程序在启用了远程桌面服务 (RDS) 的服务器节点上运行，那么该控制台的行为方式将与 RDS 客户端会话的行为方式相同。每个会话都将使用一个许可证。每个会话也将使用一个单独的 InTouch 开发许可证。

在此情况中，InTouch 应用程序的读写功能由其远程访问配置定义，并由所使用的许可证确认。例如，在 RDS 客户端会话中，如果在 WindowViewer 中启动具有只读远程访问配置的应用程序，它将查找只读 InTouch 许可证。如果许可证服务器中未提供 RDS 只读许可证，则启动许可证验证将失败。

在未启用 RDS 的节点上，您还可以通过操作系统允许的 RDS 客户端会话登录。如果 InTouch 应用程序在此非 RDS 环境中运行，那么该客户端会话的行为方式将与控制台的行为方式相同。在这种情况下，应用程序的远程访问配置不确定读写访问权限。只有在非 RDS 环境下，读写访问权限才由许可证确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在非 RDS 环境中，当应用程序在自由模式下运行时，它将始终在读/写模式下运行。

关于 InTouchView 应用程序许可证

InTouchView 应用程序显示专为在“应用程序服务器”环境中使用而设计的可视化界面。如需有关此应用程序类型的详细信息，请参阅 InTouchView 应用程序。

InTouchView 应用程序所使用的许可证类型取决于其是否在 RDS 环境中运行。

如果 InTouchView 应用程序在 RDS 客户端会话中运行，根据该应用程序的远程访问类型配置，它将寻找只读或读写客户端连线许可证。

每个 RDS 会话将仅使用一个连线会话。

注意：如果配置了 Galaxy 数据源，InTouchView 应用程序与 OMI ViewApp 应用程序使用相同的许可证。

InTouchView 应用程序许可证

您可以将 InTouch HMI 应用程序配置为 InTouchView 应用程序，这样它可用作 InTouch 标记服务器或 Galaxy 的客户端。

如果将 InTouchView 应用程序配置为连接到 InTouch 标记服务器中的数据，则可用的许可证为：

	InTouch HMI 客户端读写许可证	**InTouch HMI 客户端只读许可证**	**InTouch HMI 无限制客户端许可证***
远程访问	读写	只读	读写和 只读
RDS/TSE	是	是	是
非 RDS	是	是	否
支持 MarkAppReadOnlyNonRDS	是	是	不适用

* 此许可证支持无限制数量的 RDS 客户端。

如果将 InTouchView 应用程序配置为连接到 Galaxy 中的数据，则可用的许可证为：

	监督客户端读写许可证	**监督客户端只读许可证**	**监督客户端服务器许可证**
远程访问	读写	只读	读写和 只读
RDS/TSE	是	是	是
非 RDS	是	是	否
与 OMI 共享	是	是	是

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

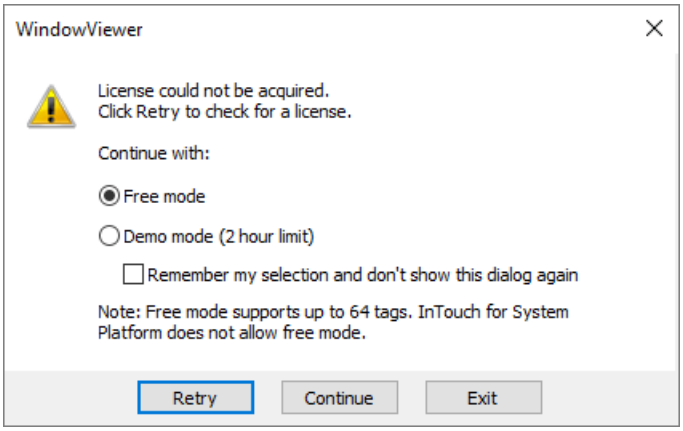
支持 MarkAppReadOnlyNonRDS	是	是	不适用

WindowMaker 许可

WindowMaker 不需要任何许可证。WindowMaker 没有许可证并且无限制地运行。

WindowViewer 许可

打开 WindowViewer 时，将进行许可证检查。如果获取合适的许可证，WindowViewer 将正常打开。如果未获得许可证，则会显示以下对话框。



您可以执行以下操作之一：

- 获取有效许可证并选择重试。
- 选择自由模式，然后选择继续以自由模式继续。有关更多信息，请参阅以自由模式和演示模式运行。

  注意：如果 InTouch 应用程序被配置为 InTouchView 应用程序或标记计数大于 64，则将禁用可用模式选项。

- 选择演示模式，然后选择继续，以在接下来的 120 分钟内继续演示模式。
- 选择退出，以关闭 WindowViewer。

如果选中记住我的选择并不再显示此对话框复选框，后续启动启动 WindowViewer 时将自动选择此模式，并且不会再次显示此对话框。如果您要重置此复选框选项，可以针对以下项编辑应用程序文件夹中的 INTOUCH.INI 文件，并将其设置为 0。

```
LicenseRememberOption=0
```

在服务器上管理 InTouch 许可证的最佳做法

管理 InTouch 许可证时有几种最佳做法可供遵循，这些做法将确保必定使用 InTouch 许可证。必定使用许可证允许您针对特定系统，按需使用适当的许可证。这种使用许可证的方式让使用基于服务器的 AVEVA Enterprise 许可证系统管理 InTouch 许可证变得更容易。

必定使用许可证的两种最佳方式是许可证保留和浮动许可证。如需有关详细信息，请参阅下面几个部分。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

保留许可证

您可以在“许可证管理器”中将许可证保留给特定设备。将许可证保留给特定设备可确保该许可证不会被其它 InTouch 应用程序获得，也不会打断或阻止您的应用程序运行。

基于用户的许可证保留

在 AVEVA Enterprise 的“许可证管理器”上的许可证保留页面中，可以标记将某个许可证保留给特定的用户。虽然保留页面允许此特殊配置，但您一定要知道，InTouch OMI ViewApp 和 InTouch HMI ViewApp 都不支持基于用户的许可证保留。最终将导致软件无法获得保留的许可证。因此，只能使用基于设备的监督客户端许可证保留。

基于设备的许可证保留

为特定设备保留监督客户端许可证时，“设备名称”必须是运行 InTouch HMI/OMI ViewApp 的计算机名称。如果 ViewApp 在 RDS 或终端服务器内运行，则“设备名称”需要遵循以下命名模式：

<RDS 主机名>-<RDP 客户端名>-<索引>

其中“RDS 主机名”是 RDS 或终端服务器的名称，“RDP 客户端名”是运行 RDP 客户端软件的 PC 机名称，而“索引”是 1；但如果单一客户端机器中有多个 RDP 会话，那么为该特定 RDP 客户端保留的每个许可证的索引应从 1 开始递增，最大为该特定 RDP 客户端中全部 RDP 会话的总数。

示例 1： 主机名为 "ControlRoomA” 的计算机运行 InTouch OMI

设备名称： "ControlRoomA"

示例 2： 主机名为 "ControlRoomB” 的计算机运行单一远程桌面客户端 (RDP)，并连接到主机名为 "PrimaryRDS” 的远程桌面服务器（即：终端服务器）

设备名称： "PrimaryRDS-ControlRoomB-1"

示例 3：主机名分别为 "SupervisorPC1” 和 "LineMgrA” 的两台计算机，各自运行一个远程桌面客户端 (RDP)，并连接到主机名为 "PrimaryRDS” 的远程桌面服务器（即：终端服务器）

设备名称：

许可证保留 1：“PrimaryRDS-SupervisorPC1-1”

许可证保留 2：“PrimaryRDS-LineMgrA-1”

示例 4： 主机名为 "ExecutiveDesktop” 的计算机运行四 (4) 个远程桌面客户端 (RDP)，并连接到主机名为 "PrimaryRDS” 的远程桌面服务器（即：终端服务器）

设备名称：

许可证保留 1："PrimaryRDS-ExecutiveDesktop-1"

许可证保留 2："PrimaryRDS-ExecutiveDesktop-2"

许可证保留 3："PrimaryRDS-ExecutiveDesktop-3"

许可证保留 4："PrimaryRDS-ExecutiveDesktop-4"

如需 RDS 负载平衡支持，可以在多个 RDS 客户端会话可指向的单一许可证服务器上激活所有 RDS 许可证。服务器上的许可证必须具有相同的功能，以便各个许可证可以在每个 RDS 客户端会话之间共享。内部参数具有相同值的许可证可视为具有相同功能。这种情况无需保留许可证。如果需要不同 RDS 客户端会话的不同许可证类型，那么必须在每个 RDS 服务器上安装许可证服务器。

有关详细的许可证保留程序，请参阅 *《AVEVA Enterprise Licensing 指南》*。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

关于浮动许可证

浮动许可证不保留给任何机器。建议将相同产品名称和功能的浮动许可证放在一台许可证服务器上。例如，您所拥有的许可证服务器可以具有多个功能相同且已激活的 InTouch 2017 Runtime 60K 标记许可证。同样，您所拥有的许可证服务器也可以具有多个功能相同且已激活的 InTouch 2023 Runtime Unlimited 标记许可证。这是确保必定使用许可证的推荐做法。

不过，不建议在同一许可证服务器上激活具有相同产品名称但不同功能的许可证。例如，您可以在同一许可证服务器上混合使用 InTouch 2023 Runtime Unlimited 标记激活许可证和 InTouch 2017 Runtime 500 标记激活许可证。在此情形中，无法确保哪个 WindowViewer 实例将使用标记计数更高的许可证。

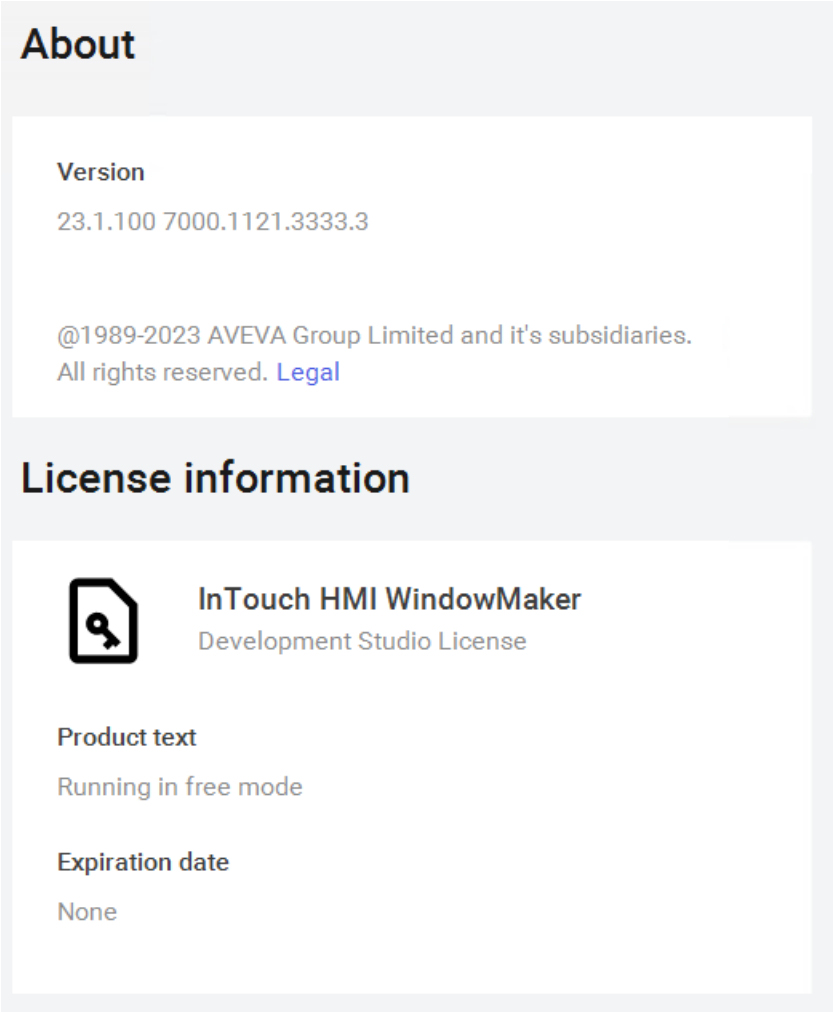
查看许可证信息

您可以查看 WindowMaker 或 WindowViewer 当前所使用的特定信息。

查看 WindowMaker 许可证信息

1. 打开 WindowMaker。
2. 在文件菜单上，选择屏幕左下角的**许可证**。

   此时出现关于屏幕。关于屏幕会显示 WindowMaker 版本和许可证信息。



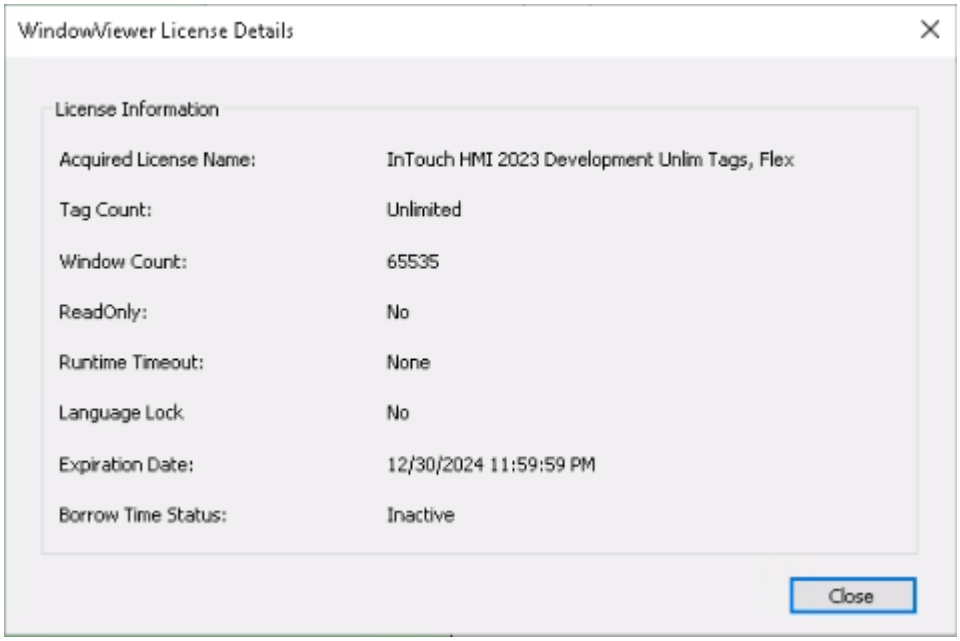
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此对话框中不会显示该公司名称和许可证序列号。此信息会出现在“AVEVA Enterprise 许可证管理器”界面。

查看 WindowViewer 许可证信息

1. 打开 WindowViewer。
2. 选择文件，然后单击关于 **WindowViewer**。

   此时出现关于 **WindowViewer** 对话框。
3. 选择查看许可协议，可查看“最终用户许可协议”。
4. 选择查看许可证，可查看该许可证的详细信息。此时出现“许可证信息”对话框。



许可证信息中所显示的参数如下所示：

- **获得的许可证名称**：来自许可证服务器的产品所使用许可证的全名。
- **标记计数**：所使用许可证允许的标记数。
- **窗口计数**：所使用许可证允许的窗口数。
- **只读**：显示许可证允许的 I/O 读/写权限。“否”表示应用程序可以写入 I/O 标记。
- **运行时超时**：许可证分配的应用程序运行时。超时期限过后，InTouch 会话将结束。
- **语言锁**：仅适用于中文操作系统上的 InTouch 的许可证。要在中文操作系统上运行中文版 InTouch，必须使用中文许可证或英语许可证。

  语言锁限制不会请求连接许可证。
- **过期日期**：所使用许可证的过期日期。
- **犹豫时间状态**：用于通知用户许可证的借用期已过 50%（不管分配的借用时间是什么时候）。当达到 50% 时，状态将变为活动。如果该许可证的借用期完全过期后没有续订，那么 InTouch 将变为无许可证状态。

注意：“InTouch 应用程序管理器”的查看许可证选项处于禁用状态。因为“应用程序管理器”不使用许可证，所以您只能查看 EULA。关于应用程序管理器对话框不会显示任何许可证相关的信息。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

管理启动后不同许可证的使用

作为标准升级和维护活动的一部分，可以在“许可证管理器”上停止或激活不同的许可证。如果 WindowMaker 或 WindowViewer 在启动时间后使用有效的许可证且无法对该许可证进行续订，它可以使用与最初指定或使用的许可证不同的其它许可证。如果发生这种情况，那么 InTouch 必须验证新许可证的功能是否合适。上一合法有效的许可证与当前使用的许可证之间的功能比较将确定如果不进入宽限期，WindowMaker 或 WindowViewer 是否可以继续运行。如需有关如何退出宽限期的详细信息，请参阅利用宽限期。

启动后所使用的另外一种许可证可能会出现的两种情形如下所述。

情形 1：启动后，所使用的许可证的功能减少

在此情形中，启动后所使用的许可证与上一合法有效的许可证相比，功能变少。这被视为许可证降级，将导致 WindowMaker 或 WindowViewer 进入宽限期。

如果新许可证的参数落入以下检查范围，那么会将该许可证视为降级：

将会触发宽限期的参数变化如下所述：

- 标记计数：如果标记计数减少，则触发宽限期。
- 窗口计数：如果窗口计数减少，则触发宽限期。
- 运行时超时：如果“超时”值从无变为其它任何值，则触发宽限期。
- 语言锁：如果“语言锁”值从否变为是（或者相反），则触发宽限期。
- 只读：如果 ReadOnly 参数从否变为是（或者相反），则触发宽限期。

如果发生许可证降级的情况，则降级或“发生冲突”的参数会显示在查看许可证对话框中。例如，如果 WindowMaker 最初使用六万个标记计数的许可证，降级为三万个标记计数的许可证，那么标记计数参数将显示如下：

重要事项：该应用程序将继续使用上一合法有效的许可证的功能运行。在此例中，进入宽限期后，仍将保持最初的六万个标记计数。

情形 2：启动后，所使用的许可证的功能增加

在此情形中，启动后所使用的许可证与原始许可证相比，功能相同或增加。这被视为许可证降级，将不会导致 WindowMaker 或 WindowViewer 进入宽限期。

如果发生许可证升级，那么查看许可证对话框中的所有参数都将更新以显示更高的功能值。

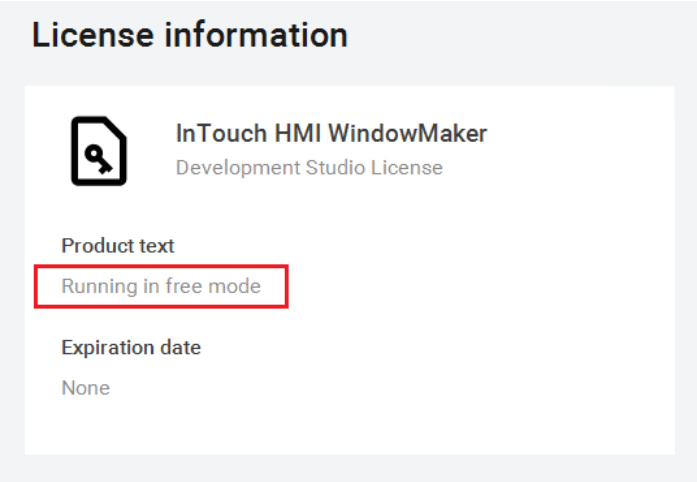
以自由模式和演示模式运行

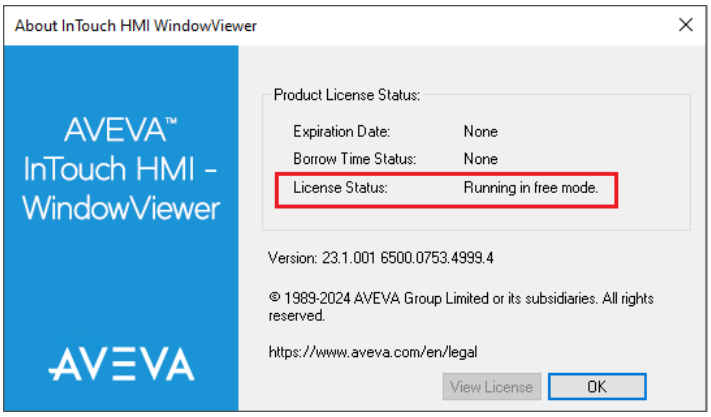
虽然 WindowViewer 正在自由模式下运行，但会从运行时中删除 Galaxy 访问名称。在以下情况下，无法使用自由模式：

- 标记数超过 64 个或
- 应用程序被配置为 InTouchView 应用程序

注意：在 Operations Control 连接体验中不支持自由模式。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。





在演示模式下运行时，没有标记计数或窗口计数限制。不过，WindowViewer 在演示模式下只能运行两个小时；超过两小时后，将会超时。演示模式超时后，将会提示退出。

注意：在演示模式下，即使您激活了有效许可证，也需要先退出 WindowViewer，然后重新启动 WindowViewer，才能使用该有效许可证。

在 InTouch.ini 文件中配置 ViewLicenseRetryCount 关键码，将会指示 WindowViewer 继续在后台按照参数中指定的次数尝试获得许可证。如果已获得许可证，对话框将会关闭，并且启动 WindowViewer。

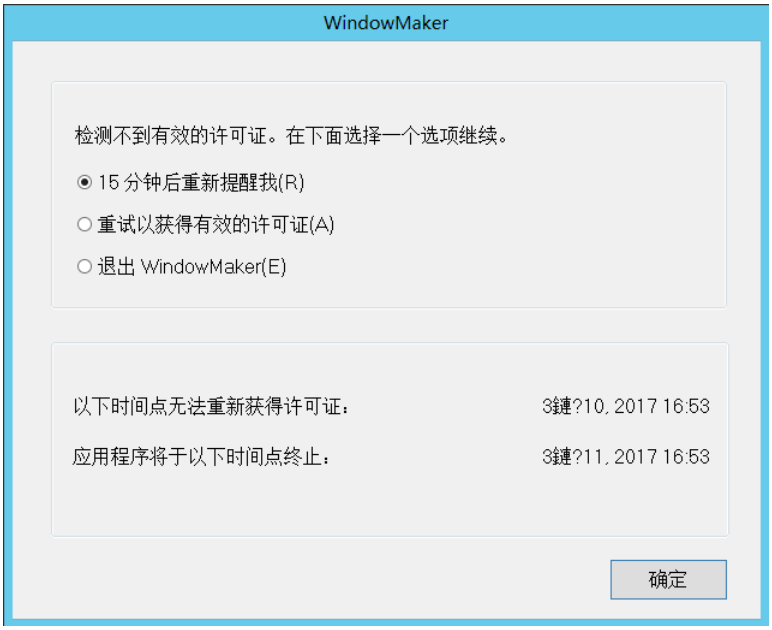
利用宽限期

宽限期为 24 个小时，在此期间，虽然某些状况已经发生，InTouch 仍然可以继续使用上一合法有效许可证的功能运行。WindowMaker 和 WindowViewer 都可以进入宽限期。当宽限期结束时，如果 InTouch 在期限内未重新获得有效许可证，将会终止。

以下情境将触发宽限期。如果在二十四小时内未重新获得有效许可证，那么 WindowMaker 或 WindowViewer 将终止。

当 WindowMaker 或 WindowViewer 进入宽限期时，系统将显示以下对话框提示您：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



您可以选择再次提醒，重试获得许可证或退出应用程序。

情形 1：所使用的许可证丢失

如果 WindowMaker 或 WindowViewer 使用来自许可证服务器的有效许可证，并且于产品仍在运行时停用该许可证，那么 WindowMaker 或 WindowViewer 将进入宽限期。

要退出宽限期并恢复正常运行，请在许可证服务器上激活有效许可证。使用该许可证后，WindowMaker 或 WindowViewer 将退出宽限期，恢复正常运行。

情形 2：许可证过期

如果 WindowMaker 或 WindowViewer 使用来自许可证服务器的有效许可证，但该许可证过期，那么 WindowMaker 或 WindowViewer 将进入宽限期。要退出宽限期并恢复正常运行，请在许可证服务器上激活有效许可证。

如果 WindowMaker 或 WindowViewer 未能获得许可证，则会再次显示第一个对话框。

情形 3：许可证降级

“AVEVA Enterprise 许可证管理器”是基于服务器的许可证系统，这意味着需要定期续订许可证。如果 WindowMaker 或 WindowViewer 许可证在此续订期间降级而导致功能减少，那么 WindowMaker 或 WindowViewer 将进入宽限期。如需降级情形的详细说明，请参阅管理启动后不同许可证的使用。

要退出宽限期并恢复正常运行，请重新尝试获得上一个合法有效的许可证或者获得更加适用的许可证。

重要事项：在宽限期模式下，上一个合法有效的许可证启用的功能仍可使用。

在上述所有情形中，如果选择重新尝试获得许可证并且成功获得合适的许可证，您将看到一条消息，指示已成功获得许可证。

如果 InTouch 无法获得许可证，则会再次显示第一个对话框。

注意：在 InTouch.ini 文件中配置 ViewLicenseRetryCount 关键码，将会指示 WindowViewer 继续在后台按照参数中指定的次数尝试获得许可证。如果已获得许可证，对话框将会关闭，并且启动 WindowViewer。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

远程标记计数函数

InTouch HMI 包含一组函数，可以验证应用程序是否符合许可证的远程标记要求。您可以编写一些包含这些函数的临时脚本，在部署到生产环境之前测试与确定应用程序是否有任何潜在的许可证问题。验证应用程序不存在任何许可证问题之后，请删除这些脚本。

IORRGetSystemInfo() 函数

IORRGetSystemInfo() 函数返回正在运行的 InTouch 应用程序的标记计数。IORRGetSystemInfo() 函数根据参数值返回一个数值，该数值可以是：

- InTouch 许可证指定的最大远程标记地址数。
- 在 InTouch 应用程序运行期间根据许可证统计的远程标记地址数。
- InTouch 应用程序中当前激活的远程标记数。
- 正在运行的 InTouch 应用程序中可用的远程标记数。
- 当前禁用的远程引用标记数。
- 正在运行的 InTouch 应用程序中的本地标记数。
- 如果函数调用期间发生错误，或 *Option* 参数指定为无效的值，则为 -1。

类别

其它

语法

```
IORRGetSystemInfo(Option);
```

参数

选项

整型标记或整型常数，指定要返回的远程引用标记计数的类型。可能的值有：

1	根据 InTouch 许可证返回允许的最大远程标记地址数。本地 I/O 标记不记入远程标记计数。 在 InTouch 应用程序运行时，此数字为常数。
2	返回根据许可证限制统计的独特远程标记地址数，这些地址是在 InTouch 应用程序运行时激活的。本地 I/O 标记不记入远程标记计数。 如果许可证允许使用 60K 以上的远程引用标记，则不论激活了多少远程标记地址，此数字都可能是 0。在无限制许可证下运行时，WindowViewer 不统计激活的远程标记地址。 在具有远程标记限制的许可证下运行应用程序时，此计数会递增，直至达到远程标记许可证计数的限制。达到远程标记限制之后，便无法激活更多的远程标记地址。此时仅能重新激活当前已激活的地址。通过使用将 *Option* 参数设置为 3 的 IORRWriteState，可以获取根据许可证限制统计的远程引用标记的列表。
3	返回 InTouch 应用程序中当前已激活的远程引用标记数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4	返回可以在 InTouch 应用程序中激活且不会导致缺少远程标记句柄的远程引用标记数。应用程序在运行时，此计数通常会发生变化。停止与启动脚本、打开与关闭包含远程引用标记的窗口时，都会影响远程引用标记计数。 此数字可以少于许可证上剩余的数字，尤其是在许可证没有限制的情况下。发生这种情况是因为存在一个内部限制来确定同时可以有多少个活动的远程引用标记。
5	返回 InTouch 应用程序中当前处于禁用状态的远程标记数。
6	返回 InTouch 应用程序中当前的本地标记数。

示例

以下示例返回 InTouch 应用程序正在运行时根据许可证限制统计的远程引用标记数。返回的远程引用标记计数被指定 RRTagCount 整型标记的值。

```
RRTagCount = IORRGetSystemInfo(2);
```

IORRWriteState() 函数

IORRWriteState() 函数将应用程序远程标记的当前状态信息保存到文本文件。如果不存在该文件，此函数会创建它。每次运行包含此函数的脚本时，都会将新的信息附加到该文件中。

您可以指定将哪些远程标记信息保存到文件。另外，此函数的返回值会指出信息是否已成功添加到文件中。

类别

其它

语法

```
IORRWriteState(FilePath, Option, " ");
```

参数

FilePath
文本文件的完整文件夹路径，该文件包含有关应用程序远程标记的信息。*FilePath* 参数可以是字符串常量或消息标记。

选项
整型标记或整型常数，指定写入文件的远程标记计数信息。可能的值有：

1	当前远程标记地址的列表。此信息还包括每个远程标记的状态、激活时间及停用时间。
2	当前所有活动地址的列表，包括激活时间。
3	已激活并根据许可证进行统计的所有远程引用标记地址的列表。 激活新的远程标记地址时，新的项目会添加到列表中。达到许可证限制时，不再向列表中添加更多的项目。 不过，如果远程标记许可证限制是无限的，则不向此列表中添加任何地址。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

4	由于远程引用标记计数超出许可证限制，或由于达到了内部标记句柄限制而未激活的当前地址的列表。 如果远程标记许可证限制是无限的，则不会返回与许可证相关的地址。 如果许可证是无限的，则列表包含由于实施限制而当前未处于活动状态的所有项目。如果此列表中的任何项目变为活动状态，则它会从列表中删除。某个项目由于许可证限制而停用时，它会从列表中删除。此列表在 InTouch 应用程序运行时进行更新。

"" （空字符串）
此参数保留下来供将来使用，但在脚本的 IORRWriteState() 函数中必须包含它。

结果

下面是一些示例，用来解释由 IORRWriteState() 函数调用保存到文件的信息。

当前地址列表

输出文件中的下面这行代码显示完全激活且可以更新的远程引用标记的示例。这行代码显示地址 65535 指定给了 TestProt:di000 远程引用标记：

```
65535 <TestProt:di000> (RAA) {C:5/23/2007 9:58:35 AM} {A:5/23/2007 9:58:35 AM}
```

远程引用标记名后面是圆括号括起来的三个标帜：

- 第一个标帜 R 表示标记的远程引用已经成功解析。如果标记的远程引用仍然等待处理，则指定给第一个标帜的值是 X。
- 第二个标帜表示远程标记当前处于活动状态 A，还是处于不活动状态 D。
- 第三个标帜表示由于许可证限制该地址是允许的 A，还是不允许的 D。

C 后面的日期表示创建远程标记的时间。A 后面的日期显示最近激活标记的时间。如果远程引用标记已在 InTouch 应用程序运行时停用，这行代码会在最后包含一个停用时间。

输出文件中的下面这行代码显示超出 InTouch 许可证标记计数限制的活动远程引用标记的示例。标记的远程引用已成功解析，且标记当前处于活动状态：

```
65414 <TestProt:di121> (RAD) {C:5/23/2007 9:58:35 AM} {A:5/23/2007 9:58:35 AM}
```

但 InTouch 应用程序中不更新标记值，这是因为指定给远程引用标记的地址超出了 InTouch 许可证的标记计数限制。

活动地址列表

输出文件中的下面这行代码显示完全激活的远程标记的示例，指定给该标记的值会更新 InTouch 应用程序：

```
65429 <TestProt:di106> (A) {C:5/23/2007 9:58:35 AM} {A:5/23/2007 9:58:35 AM}
```

第一个数字是远程标记句柄，其后是地址，然后是表示允许的标帜 A，或表示不允许的标帜 D。输出行中标帜的后面是创建时间、最近激活时间以及最近停用时间。如果该远程标记在应用程序运行期间尚未停用过，则停用时间不会出现。

输出文件中的下面这行代码显示超出许可证限制的活动远程标记的示例：

```
65342 <TestProt:di193> (D) {C:5/23/2007 9:58:35 AM} {A:5/23/2007 9:58:35 AM}
```

许可的地址

列表中的下面这行代码显示指定给远程引用标记的地址以及它添加到列表的时间。

```
<testprot:di000> {C:5/23/2007 9:58:36 AM}
```

拒绝的地址

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

由于实施限制或由于标记计数超出许可证最大值，被拒绝的地址出现在列表中。

本例显示超出许可证限制的远程标记地址：

```
testprot:di125 [1] (L) {F:5/23/2007 9:58:39 AM} {R:5/23/2007 9:58:39 AM}
```

地址与计数一起列出，指出尝试了多少次去引用该项目。标帜指出该地址是由于超出许可证限制 (L)，还是由于内部实施限制 (I) 而出现在列表中。两个时间分别代表它第一次添加到列表的时间与最近访问的时间。

示例

本例将当前激活的远程标记地址写入到 c:\intouch\data 文件夹下的文件中。ReturnValue 标记会被指定给一个整数，指出函数调用是否成功地将远程标记信息写入文件中。

```
ReturnValue = IORRWriteState("c:\intouch\data", 2, "");
```

IORRGetItemActiveState() 函数

IORRGetItemActiveState() 函数返回指定的远程标记地址的状态。

类别

其它

语法

```
IORRGetItemActiveState(ItemPath, Option);
```

参数

ItemPath
ItemPath 是代表所关注的地址的字符串。*ItemPath* 可以是字符串常量或消息标记。

选项
整型标记或整型常数，指定要返回的远程引用标记计数的类型。可能的值有：

1	确定当前远程标记地址目前是否处于活动状态。 如果是当前的活动地址，则返回值为 1。如果不是当前地址，则返回值为 -1。 如果是当前地址但并非处于活动状态，则返回值为 0。
2	确定当前远程标记地址在应用程序运行期间是否激活过。 如果是当前地址，并且至少激活过一次，则返回值为 1。如果不是当前地址，则返回值为 -1。如果是当前地址但从未激活过，则返回值为 0。
3	确定当前远程标记地址是否停用过。 如果不是当前地址，则返回值为 -1。 如果是当前地址但从未停用过，则返回值为 0。
4	确定当前远程标记地址是否已禁用。 如果是当前地址，且至少停用过一次，则返回值为 1。如果不是当前地址，则返回值为 -1。如果地址未禁用，则返回值为 0。如果是当前地址，并且已禁用，则返回值为 1。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

5	确定地址是否在允许的列表中。 如果地址不在列表中，则返回值为 0。 如果地址在列表中，则返回值为 1。
6	确定地址是否在拒绝的列表中。 如果地址不在列表中，则返回值为 0。 如果地址在列表中，则返回值为 1。

示例

本例确定 TestProt:di000 远程标记地址当前是否处于活动状态：

```
ReturnValue = IORRGetItemActiveState("TestProt:di000", 1);
```

本例确定 TestProt:di121 远程标记地址当前是否已禁用：

```
ReturnValue = IORRGetItemActiveState("TestProt:di121", 4);
```

本例确定当前是否根据许可证限制来统计 TestProt:di001 远程标记地址。

```
ReturnValue = IORRGetItemActiveState("TestProt:di001", 5);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

授权 InTouch Web 客户端

您需要有效的 Web 服务器许可证才能从 Web 浏览器登录和查看应用程序图形。许可证还扩展至将应用程序图形托管在外部网站上。如需有关许可的详细信息，请参阅 *AVEVA Enterprise Licensing 帮助*。Web 服务器提供不同类型的许可证，允许您连接到无限个会话以在 Web 浏览器中查看应用程序图形并与之交互，包括：

- InTouch Web 服务器*弹性许可证*
- InTouch Web 服务器*连接读写许可证*
- InTouch Web 服务器*无限制读写许可证*
- InTouch Web 服务器*无限制只读许可证*

许可证状态更改

如果 Web 服务器已获得有效许可证，但后来未能续订许可证，Web 客户端将处于宽限期模式。通知页面中将显示一条通知消息。每次 Web 客户端尝试续订许可证并失败时，都会记录该消息。

对于 InTouch HMI 应用程序，WindowViewer 必须使用读/写许可证运行，以便 Web 客户端连接到 WindowViewer 并接收 InTouch 标记数据。

获得 InTouch 应用程序图形的许可证

获取许可证是一个两阶段过程。Web 服务器启动后，将先对用户进行身份验证，然后确定用户的授权。如果已启用匿名访问，则会绕过身份验证步骤，并且使用“Guest”用户在“只读”模式下启动 InTouch Web 客户端。

身份验证

1. InTouch Web 客户端支持 Windows 身份验证。Web 服务器将验证用户是否属于“aaInTouchUsers”或“aaInTouchRWUsers”才能通过身份验证来使用 Web 客户端。安装过程中将创建这两个用户组。对于远程身份验证服务器，必须在服务器上创建域用户组。
2. 安装 Web 客户端时的登录用户将自动添加到这两个用户组。其他用户可以稍后添加。将新用户添加到用户组后，新用户必须注销再重新登录，以使更改生效。

收购

启动时，在用户经过身份验证后，Web 服务器按照以下工作流程获得许可证。

1. Web 服务器将检查配置器中是否启用了“弹性”选项。

   如果未启用“弹性”选项，Web 服务器将尝试获得无限制读写许可证。（转到第 4 步）
2. 如果在配置器中启用了“弹性”和 Enterprise 选项，Web 服务器将尝试获得 Enterprise 许可证。

   如果在配置器中启用了“弹性”选项但未启用 Enterprise 选项，Web 服务器将尝试获得弹性许可证。（转到第 3 步）
3. 未获得许可的 Web 服务器即会尝试获得弹性许可证。

   如果弹性许可证可用，Web 服务器将获得弹性许可证。

   如果弹性许可证不可用，那么 Web 服务器将被认为是处于未获得许可的状态。如果 Web 客户端在 Web 服务器未获得许可时尝试连接，将显示许可证错误页面。
4. 未获得许可的 Web 服务器将尝试获得无限制读写许可证。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

如果无限制读写许可证可用，Web 服务器将获得无限制读写许可证。

如果无限制读写许可证不可用，Web 服务器将尝试获得连接读写许可证。

5. 如果连接读写许可证可用，Web 服务器将获得连接读写许可证。

   如果连接读写许可证不可用，Web 服务器将尝试获得无限制只读许可证。

6. 如果无限制只读许可证可用，Web 服务器将获得无限制只读许可证。

   如果无限制只读许可证不可用，Web 服务器将进入只读单会话模式。

许可证功能

InTouch Web 客户端支持以下许可证功能：

- 弹性许可证
- 读写许可证
- 无限制只读许可证

每种模式的功能如下所述：

已获得许可证	支持的功能	条件
弹性许可证	• 读写许可证允许无限个连接。 • 支持 InTouch 和 AppServer 域名空间的个人工作区	请参阅下面的备注。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

读写许可证	• 写入外部引用，如 AppServer 属性或 InTouch 标记 • 确认报警及其用户凭证将被记录为确认报警的操作员。 • 连接读写许可证允许与 Web 服务器建立多个 InTouch Web 客户端连接（例如，一组 5 个、10 个、15 个等）。	如果满足以下条件，则可以访问“写入”和“报警确认”功能： • Web 服务器已获得弹性许可证、Brand Neutral R/W 许可证、无限制读写许可证或连接读写许可证 • InTouch Web 客户端会话在允许的读/写连接内 • 登录到 InTouch Web 客户端会话的用户属于 Web 客户端读写组。
只读许可证	• 客户端会话不能写入外部引用或确认报警。	只有在以下情况下，用户才能访问只读模式： • Web 服务器已获得弹性许可证、Brand Neutral R/W 许可证、无限制读写许可证或连接读写许可证，但登录的用户不属于 InTouch Web 客户端 aaInTouchRWUsers 用户组，或者 • Web 服务器已获得无限制只读许可证，或者 • Web 服务器未获得任何许可证，并且此 Web 客户端会话是第一个客户端会话。

注意：弹性许可证不允许回退。如果 Web 服务器无法获得弹性许可证，将会取消该 Web 服务器的许可。在 Web 服务器未获得许可的状态下，如果 InTouch Web 客户端尝试连接，将显示许可证错误页面。

单一会话模式

如果 Web 服务器在启动时未获得任何许可证，它将在单一 Web 会话模式下运行，这根据先到先得的原则进行分配。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 如果有效的许可证不可用，但已获得第一个会话许可证，那么 InTouch Web 客户端将通知用户没有可用的许可证。
- 如果 Web 服务器已获得无限制许可证，但随后失去该许可证，则 Web 服务器允许会话存在一个宽限期。

订阅 Application Server 数据需要监管许可证。如果您没有监管许可证，将在 2 个小时后终止订阅 Application Server 数据。

宽限期

Web 服务器将在以下情况下进入宽限期：

- 如果获得了有效许可证，但许可证后来到期。
- 如果获得了有效许可证，但后来无法获得。

在宽限期中，对于已连接会话和新会话，所有 InTouch Web 客户端功能都可使用。Web 客户端页面将显示一个可视通知，提示 Web 服务器处于宽限期模式。宽限期为 14 天。宽限期结束后，Web 服务器将继续工作，并立即切换到先到先得的单会话许可证。如果获得同一许可证，则 Web 服务器将退出宽限期模式。

定期续订

获得有效许可证后，将定期续订许可证。定期续订许可证操作将只检查并确保可以续订同一许可证。如果无法续订同一许可证，则 Web 服务器将进入宽限期模式。

续订频率取决于如下所列出的许可证类型：

许可证类型	续订频率
弹性许可证 无限制读写许可证 无限制只读许可证	每 7 分钟
连接读写许可证	每 24 小时

续订期间许可证状态不会升级或降级。要获得不同的许可证，请重新启动 Web 服务器，然后启动许可证获取的启动序列。

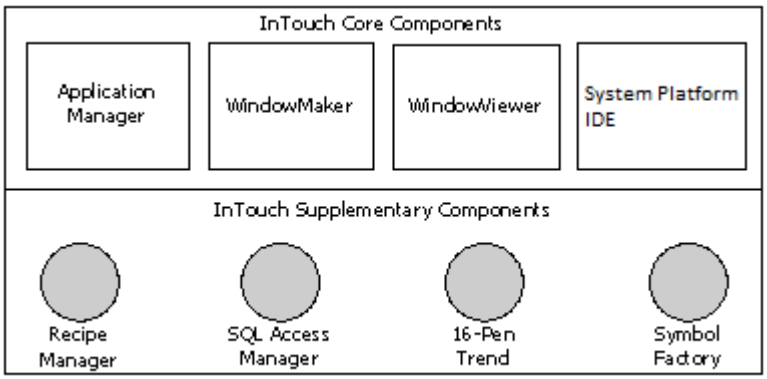
许可证释放

如果 Web 服务器关闭，Web 服务器将释放其获得的所有许可证，无论客户端会话数和 Web 服务器许可证的当前状态如何。Web 服务器还将立即终止所有 Web 客户端会话。

关于辅助组件

您可以选择随 InTouch HMI 核心组件一起安装四个辅助组件。这些辅助组件为 InTouch HMI 应用程序提供附加功能。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

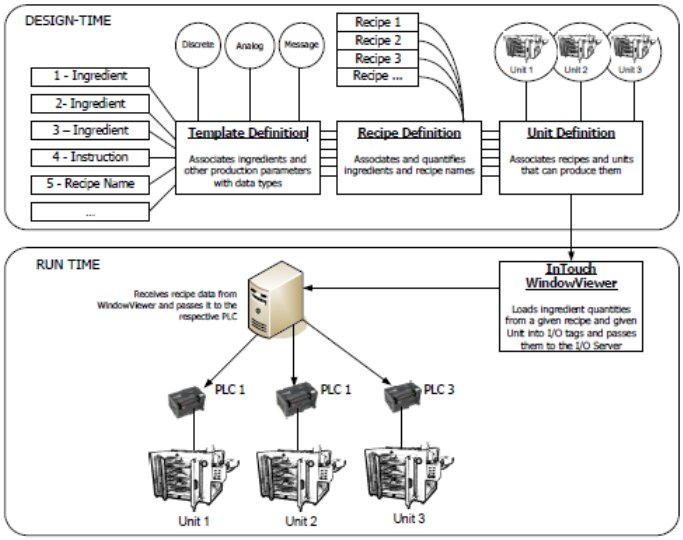


- Recipe Manager 包含一套电子表格与脚本函数，可用于创建生产配方。
- “SQL 访问管理器”由一个程序和一套 SQL 函数组成，可用于将 InTouch 数据存储到数据库。
- “16 笔趋势”包含一个趋势向导与许多脚本函数，可用于创建实时与历史趋势。
- Symbol Factory 提供一套工业符号，可放入 InTouch 应用程序中代表过程组件。

使用 Recipe Manager

制造业根据使用标准数量原材料的可重复过程来生产产品。从本质上说，产品是根据配方进行生产的。配方描述原材料、原材料数量以及它们如何组合起来，从而生产出最终的产品。最直观的情况是，面包房根据列出所有成分与程序化步骤的基本配方来制作饼干。

Recipe Manager 是 InTouch HMI 的辅助组件，可用于简化创建生产配方的过程。下图概要介绍 Recipe Manager 如何从配方模板中获取信息，以便管理产品生产的过程。



Recipe Manager 概述

Recipe Manager 可以作为可选组件随 InTouch 一起安装。Recipe Manager 由 Recipe Manager 实用程序和一套 InTouch 配方脚本函数组成。

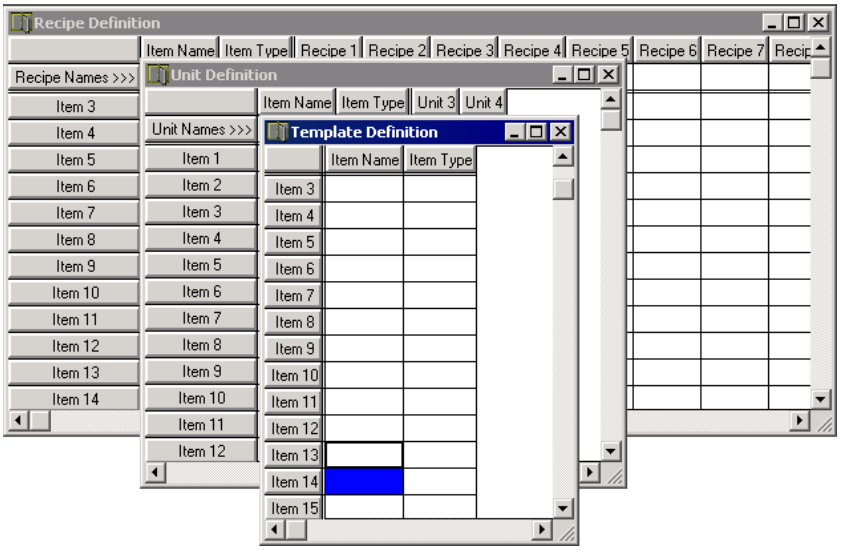
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您可以从 WindowMaker 中访问 Recipe Manager 实用程序，也可以从 C:\Program Files (x86)\Wonderware\InTouch\ 中启动 Recipe.exe 文件来访问该实用程序。Recipe Manager 实用程序包含一个可用于创建与编辑配方模板的界面。Recipe Manager 将模板保存在配方文件中。

通常，与生产过程关联的标记使用 QuickScript 来访问配方模板文件中的数据。Recipe Manager 包含一套 QuickScript 函数，可用于选择、加载、修改、创建以及删除模板文件中包含的生产配方。

Recipe Manager 实用程序

Recipe Manager 实用程序提供一个类似于电子表格的用户界面，可用于创建与维护配方模板文件。一个文件由三个模板组成。通过在每个模板的电子表格的单元格中添加或修改数据，可以创建与编辑这些模板。



这些模板保存在逗号分隔值（CSV）文件中。使用任何支持 .csv 文件格式的程序<1>（</1>如<2>“</2>记事本<3>”</3>或 Excel<4>），</4>都可以创建与编辑配方模板定义<5>。</5> 不过，Recipe Manager 提供了一些预先设置好格式的电子表格和一套编辑工具，可以轻松、可靠地创建与维护这些模板。

配方模板文件

Recipe Manager 模板文件包含以下信息：

- 配方中使用的成分的名称及其数据类型。
- 将 InTouch 标记与配方成分值关联起来的单元名。
- 配方名，它包含配方实例中使用的每种成分的数量或值。

模板定义

<1>“</1>模板定义<2>”</2>模板定义所有的配方成分<3>。</3> 每个配方成分都有数据类型与之关联。成分数据类型可以是模拟、离散或消息。成分名不要求是 InTouch 标记。

单元定义

“单元定义”模板将 InTouch 标记与配方成分关联起来。您可以创建许多种不同的加载定义。这些定义称为单元。通过使用 RecipeLoad() 函数，可以将配方的特定实例加载到关联的 InTouch 标记。“单元定义”可以由文件中定义的所有成分，或者仅是其中的一部分成分组成。

备注：单元标记可以是能够在 InTouch 窗口中进行查看与编辑的内存型，或是能够直接加载到 PLC 的 I/O 标记。

配方定义

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

“配方定义”模板指定每个配方的名称及其使用的各种成分的数量。配方实例可以在运行时通过配方函数进行修改、创建或删除。

在 Recipe Manager 中编辑配方数据

通过完成一系列的任务，可以创建生产配方。下面的列表给出创建配方需要完成的 Recipe Manager 任务以及应该遵循的顺序：

- 配置 Recipe Manager 编辑网格。
- 编辑模板中的数据。
- 给<1>“</1>模板定义<2>”</2>模板指定成分名与单元类型<3>。</3>
- 在“单元定义”模板中，将 InTouch 标记映射到成分。
- 在<1>“</1>配方定义<2>”</2>模板中<3>，</3>将值指定给配方成分<4>。</4>

配置 Recipe Manager 编辑网格

在创建生产配方之前，应该配置 Recipe Manager。配置 Recipe Manager 编辑功能有两项任务：

- 设置模板项目数的最大限制。
- 设置 ENTER 键滚动功能。

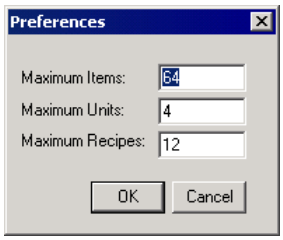
在创建配方之前，需要配置在配方模板中可以输入的最大项数。您必须为项目、单元以及配方名指定一组最大限制。

模板最多可以包含 9999 个项目、单元以及配方名。不过，很大的最大限制可能会影响系统性能。此外，如果设置的最大限制所需的内存超出计算机的可用内存，则可能会看到错误消息。

配置配方模板的最大限制

1. 通过以下方法之一启动 Recipe Manager：
   - 启动 WindowMaker。在工具视图中，展开应用程序，然后选择 **Recipe Manager**。

   此时出现“Recipe Manager” 对话框。
2. 在选项菜单上，选择首选项。此时出现“首选项”对话框。



3. 在最大项目框中，输入“模板定义”模板中允许的最大项目名数量。
4. 在最大单元框中，输入“单元定义”模板中允许的最大单元数量。
5. 在最大配方框中，输入“配方定义”模板中允许的最大配方名数量。

警示：在首选项对话框中设置的值应用于创建的所有配方模板文件。修改这些值时，也会修改现有的所有配方模板文件。

1. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Recipe Manager 包含一个可以简化在配方模板中输入数据的选项。选择按回车键自动向下选项时，按 Enter 键可以将光标向下移动到列中的下一个单元格。

设置 ENTER 键模板滚动功能

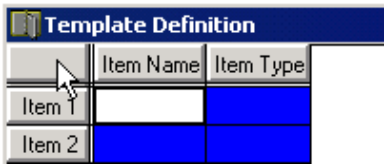
1. 打开 Recipe Manager。

   缺省条件下，Recipe Manager 不自动滚动到模板电子表格中的下一个单元格。

2. 在选项菜单上，选择按回车键自动向下以设置单元格滚动功能。
3. 如果希望关闭单元格滚动功能，请选择按回车键自动向下。

使用编辑网格

Recipe Manager 包含一组编辑命令，可用于在模板中添加、修改或删除数据。通常，您在模板中选择要编辑的数据，然后执行编辑操作。下表介绍配方模板用于输入与选择模板数据的常用功能。

功能	描述
输入框	文本输入框用于为所选的模板单元格输入数据。选择模板单元格时，其内容显示在 Recipe Manager 对话框顶部附近的文本输入框中。
选择所有单元格	选择模板左上角的单元格可以选择所有单元格。 
选择整行	选择模板的行名可以选择该行中的所有单元格。
选择整列	选择模板的列标题可以选择该列中的所有单元格。
自动调整所有列的大小	双击模板可以按各列中最长输入项的宽度来自动调整模板中各个列的大小。
自动调整一列的大小	双击标题可以按照列中最长输入项的宽度来自动调整该列的大小。 注意：“模板定义”模板中的“项目类型”列的大小无法自动调整。

编辑模板时，可以执行以下操作：

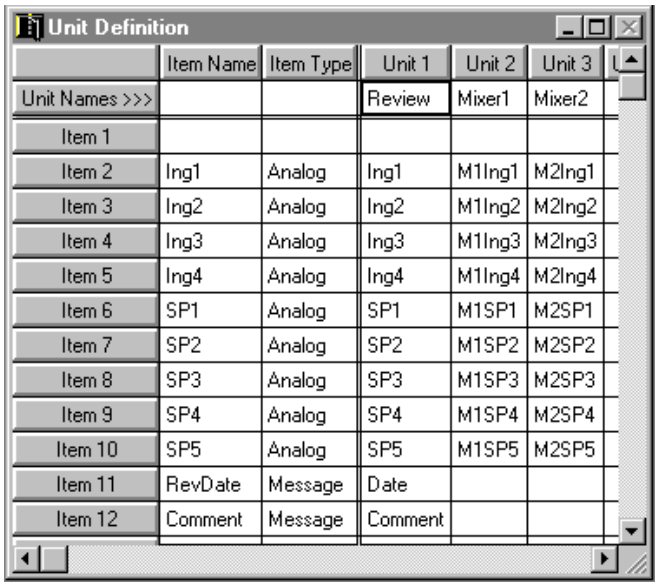
- 清除一定范围单元格中的数据。
- 将一定范围单元格中的数据复制到邻近所选范围的单元格。
- 在模板定义模板中插入一行。
- 在模板中插入一列。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 从**模板定义**模板中**删**除一行。
- 从模板中**删**除一列。

清除一定范围单元格中的数据

1. 从模板中**选择**一定范围的**单**元格。

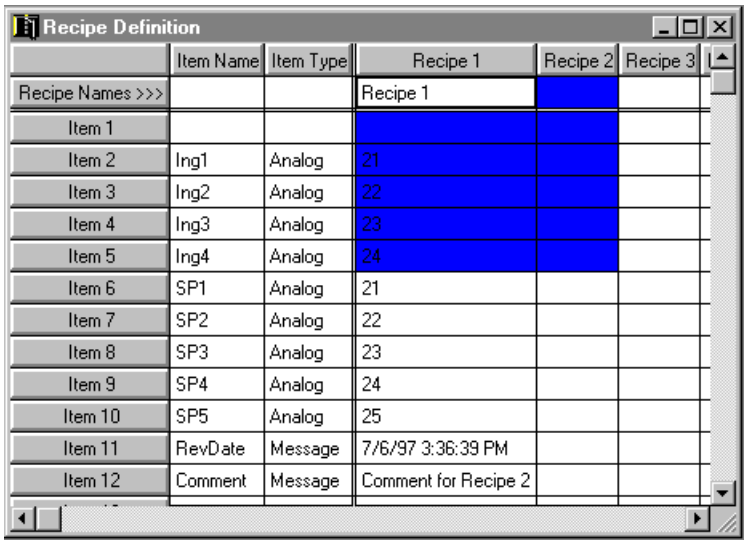


	Item Name	Item Type	Unit 1	Unit 2	Unit 3
Unit Names >>>			Review	Mixer1	Mixer2
Item 1					
Item 2	Ing1	Analog	Ing1	M1Ing1	M2Ing1
Item 3	Ing2	Analog	Ing2	M1Ing2	M2Ing2
Item 4	Ing3	Analog	Ing3	M1Ing3	M2Ing3
Item 5	Ing4	Analog	Ing4	M1Ing4	M2Ing4
Item 6	SP1	Analog	SP1	M1SP1	M2SP1
Item 7	SP2	Analog	SP2	M1SP2	M2SP2
Item 8	SP3	Analog	SP3	M1SP3	M2SP3
Item 9	SP4	Analog	SP4	M1SP4	M2SP4
Item 10	SP5	Analog	SP5	M1SP5	M2SP5
Item 11	RevDate	Message	Date		
Item 12	Comment	Message	Comment		

2. 在**编辑**菜单上，**选择清除**。此时出现一条消息，要求确认是否应清除所选范围的单元格。
3. **选择是**。此时模板清除所选范围单元格的数据。

将一定范围的单元格复制到邻近的所选范围

1. **选择**要复制的单元格或一定范围的单元格。
2. **选择**要将数据复制到的邻近单元格范围。



	Item Name	Item Type	Recipe 1	Recipe 2	Recipe 3
Recipe Names >>>			Recipe 1		
Item 1					
Item 2	Ing1	Analog	21		
Item 3	Ing2	Analog	22		
Item 4	Ing3	Analog	23		
Item 5	Ing4	Analog	24		
Item 6	SP1	Analog	21		
Item 7	SP2	Analog	22		
Item 8	SP3	Analog	23		
Item 9	SP4	Analog	24		
Item 10	SP5	Analog	25		
Item 11	RevDate	Message	7/6/97 3:36:39 PM		
Item 12	Comment	Message	Comment for Recipe 2		

所选的范围必须大小相同，可以位于原始所选单元格范围的上方、下方、左侧或右侧。

3. 在**编辑**菜单上，**选择**适当的填充命令。此时数据复制到所选范围的单元格。

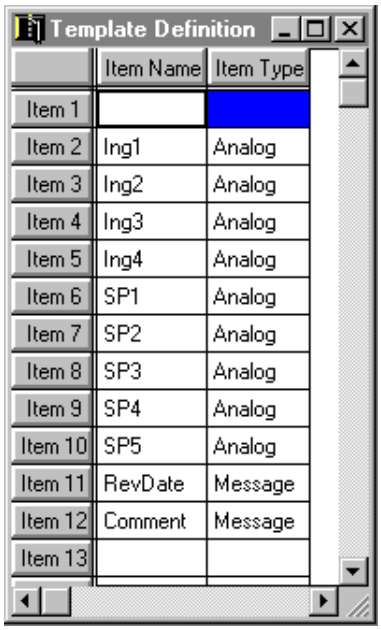
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：如果数据复制到其中的新列不足以容纳最长的输入项，请双击列标题以根据最长的输入项来调整其宽度。

在“模板定义”模板中插入一行

1. 选择模板定义模板。
2. 在“模板定义”模板中，通过单击项目 # 来选择一行，以便在其上方插入新的行。

   您无法直接在配方定义或单元定义模板中插入一些行。相反，配方定义与单元定义模板会自动继承对模板定义所作的全部修改。
3. 在编辑菜单上，选择插入。此时新行正好插入所选行的上方，同时所有后续的行自动重新编号。



注意：如果已经达到给 Recipe Manager 首选项配置的最大值，则插入命令处于非活动状态。您必须增加指定的数值<1>，</1>才能给配方模板添加<2>“</2>项目/单元/配方<3>”。</3>
修改首选项时，所作的更改应用于现有的全部配方模板文件。

插入列

1. 选择单元 # 或配方 # 以选择某个列，此列将位于所插入列的右侧。

   您无法在“配方定义”或“单元定义”模板中插入列。
2. 在编辑菜单上，选择插入。此时新列插入到所选列的左侧。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Unit Definition

	Item Name	Item Type	Unit 1	Unit 2	Unit 3
Unit Names >>>			Review		Mixer2
Item 1					
Item 2	Ing1	Analog	Ing1		M2Ing1
Item 3	Ing2	Analog	Ing2		M2Ing2
Item 4	Ing3	Analog	Ing3		M2Ing3
Item 5	Ing4	Analog	Ing4		M2Ing4
Item 6	SP1	Analog	SP1		M2SP1
Item 7	SP2	Analog	SP2		M2SP2
Item 8	SP3	Analog	SP3		M2SP3
Item 9	SP4	Analog	SP4		M2SP4
Item 10	SP5	Analog	SP5		M2SP5
Item 11	RevDate	Message	Date		
Item 12	Comment	Message	Comment		

在本例中，**Mixer2** 数据移到**单元 3** 列中，新的列作为**单元 2** 插入。

删除列

1. 选择**单元 #** 或**配方 #** 列标题，以选择要删除的列。

   您无法从“配方定义”或“单元定义”模板中删除列。

2. 在**编辑**菜单上，选择**删除**。此时出现一个确认消息对话框，要求确认是否删除。
3. 选择**是**。此时该列从模板中删除，剩余的列重新编号。

删除行

1. 选择**模板定义**模板。

   您可以从模板定义模板中删除一些行，但无法从**配方定义**或**单元定义**模板中进行删除。

2. 选择**项目 #** 行标题，以便选择要删除的行。
3. 在**编辑**菜单上，选择**删除**。此时出现一个确认消息对话框，要求确认是否删除。
4. 选择**是**。此时该行从模板中删除。

定义成分名与数据类型

“模板定义”模板列出配方的各种成分以及与每种成分关联的项目类型。您必须先完成“模板定义”模板，然后才能将数据添加到其它配方模板。

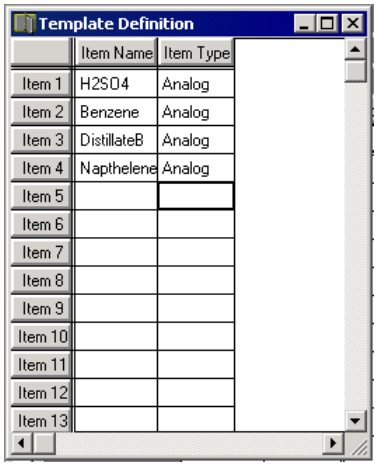
要定义“模板定义”模板

1. 启动 Recipe Manager。
2. 在“文件”菜单上，选择**新建**。此时出现三个 Recipe Manager 模板。
3. 选择**模板定义**标题栏以选择模板窗口。
4. 在**项目名**列单元格中，输入为配方成分选择的名称。

   每个单元格只能输入一种成分。

5. 在**项目类型**列单元格中，为相应的配方成分输入有效的项目类型。

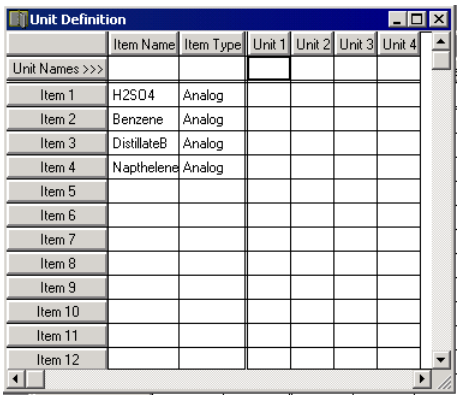
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



有效的项目类型是：模拟、离散或消息。输入 A 代表 analog（模拟），D 代表 discrete（离散），M 代表 message（消息）。按 **Enter** 键时，Recipe Manager 自动完成项目类型的剩余字符。

将 InTouch 标记映射到成分

单元定义模板将 InTouch 标记与给定单元的配方成分关联起来。如下图所示，**单元定义**模板的头两列会列出来自**模板定义**模板的“项目名”与“项目类型”。



给单元定义的标记可以是内存标记，或是从“I/O 服务器”获取 PLC 数据的远程标记。

在 InTouch QuickScript 中使用 RecipeLoad() 函数时，必须指定“单元名”。随后在运行 QuickScript 时，该“配方名”定义中包含的值会加载到在“单元名”中指定的标记。

定义“单元定义”模板

1. 选择**单元定义**模板的标题栏以选择模板窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Unit Definition

	Item Name	Item Type	Unit 1	Unit 2	Unit 3
Unit Names >>>			Review	Mixer 1	Mixer 2
Item 1	Ing1	Analog	Ing1	M1Ing1	M2Ing1
Item 2	Ing2	Analog	Ing2	M1Ing2	M2Ing2
Item 3	Ing3	Analog	Ing3	M1Ing3	M2Ing3
Item 4	Ing4	Analog	Ing4	M1Ing4	M2Ing4
Item 5	SP1	Analog	SP1	M1SP1	M2SP1
Item 6	SP2	Analog	SP2	M1SP2	M2SP2
Item 7	SP3	Analog	SP3	M1SP3	M2SP3
Item 8	SP4	Analog	SP4	M1SP4	M2SP4
Item 9	SP5	Analog	SP5	M1SP5	M2SP5
Item 10	RevDate	Message	Date		
Item 11	Comment	Message	Comment		

2. 在**单元名**行中，**输**入要定义的**每**个**单**元的名称。
3. 在**单元 #** 列**单**元格中，使用以下方法之一，**为每**个相**应**的配方成分**输**入 InTouch **标记**的名称：
   - **输**入**标记**名。
   - 如果 WindowMaker 正在运行，**请**双**击单**元格以**显**示**选择标记**对**话**框。然后，双**击**所需的**标记**将它插入到**单**元格中，或是**选择**它并**选择确定**。
4. 对于**每**个“**单**元/配方”**组**合，重**复**此操作程序。

为不同配方中的成分定义值

<1>“</1>配方定义<2>”</2>模板指定**每**个配方的名称及其使用的各**种**成分的数量<3>。</3> “配方定义”模板**显**示先前定义的“模板定义”模板中的“**项**目名”与“**项**目**类**型”信息。

在 InTouch QuickScript 中**执**行 **RecipeLoad()** 函数**时**，成分**值**加**载**到 InTouch **标记**中。

定义“配方定义”模板

1. **选择配方定义**模板的**标题栏**以**选择**模板窗口。
2. 在“配方名”行中，**输**入要定义的**每**个配方的名称。

Recipe Definition

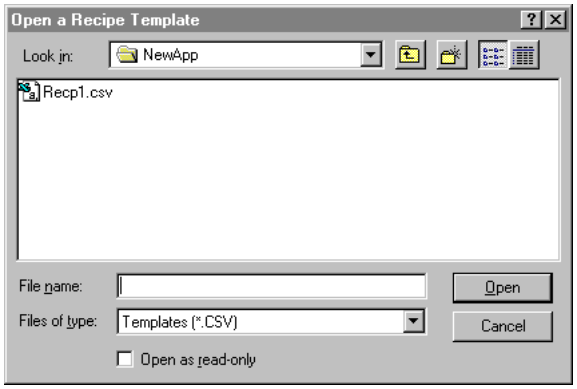
	Item Name	Item Type	Recipe 1	Recipe 2
Recipe Names >>>			Recipe 1	Recipe 2
Item 1	Ing1	Analog	11	21
Item 2	Ing2	Analog	12	22
Item 3	Ing3	Analog	13	23
Item 4	Ing4	Analog	14	24
Item 5	SP1	Analog	11	21
Item 6	SP2	Analog	12	22
Item 7	SP3	Analog	13	23
Item 8	SP4	Analog	14	24
Item 9	SP5	Analog	15	25
Item 10	RevDate	Message	7/15/97 3:19:56 PM	7/6/97 3:36:39 PM
Item 11	Comment	Message	A Comment	Comment for Recipe 2

3. 在**配方 #** 列**单**元格的“**项**目名”列中，分**别为每**个配方成分**输**入**值**。
4. 在**文件**菜**单**上，**选择保存**以保存配方模板文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

打开现有的配方模板文件

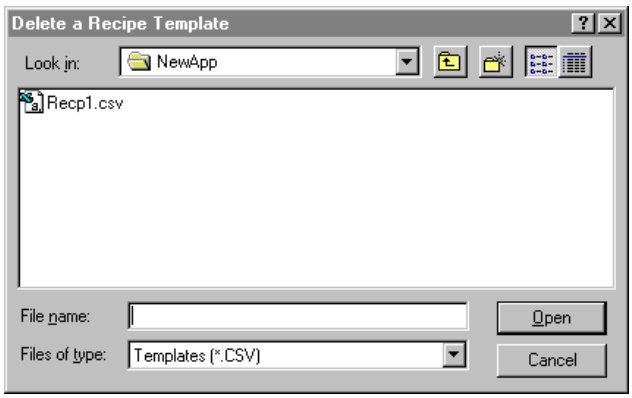
1. 打开 **Recipe Manager**。
2. 在文件菜单上，选择打开。此时出现打开配方模板对话框。



3. 找到并选择“配方”文件，然后选择打开。文件中的三个配方模板出现在 Recipe Manager 中，此时可以进行编辑。

删除配方模板文件

1. 在文件菜单上，选择删除。此时出现删除配方模板对话框。



2. 找到并选择配方文件，然后选择打开，或双击文件名。此时出现一个消息框，要求确认删除。

注意：打开的配方模板文件无法删除。

3. 选择是以删除该文件。

在其它应用程序中编辑配方数据

使用任何支持逗号分隔数据的程序，都可以创建与编辑配方模板定义。您可以使用 Microsoft Excel 或“记事本”来创建与编辑 Recipe Manager 模板文件。

使用 Excel 处理配方模板文件

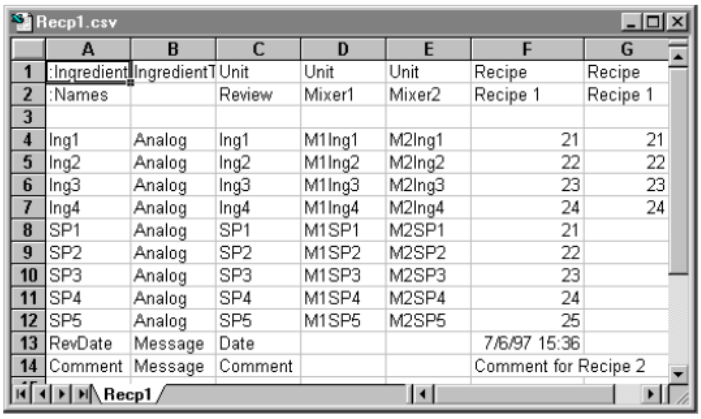
如果不希望使用 Recipe Manager 实用程序，则可以使用 Excel 来创建或编辑配方模板。使用 Excel 创建或编辑的 Recipe Manager 模板必须保存到文件扩展名为 .csv 的文件。

在 Microsoft Excel 中打开现有的配方模板文件

1. 启动 Excel。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

2. 在文件菜单上，选择打开。此时出现打开对话框。
3. 找到并选择 .csv 文件，然后选择打开，或双击文件名。此时 Excel 显示该文件的内容。

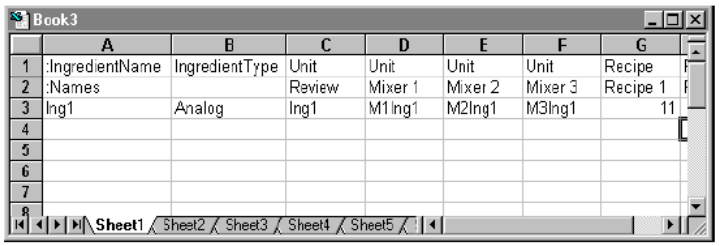


	A	B	C	D	E	F	G
1	:Ingredient	IngredientT	Unit	Unit	Unit	Recipe	Recipe
2	:Names		Review	Mixer1	Mixer2	Recipe 1	Recipe 1
3							
4	Ing1	Analog	Ing1	M1Ing1	M2Ing1	21	21
5	Ing2	Analog	Ing2	M1Ing2	M2Ing2	22	22
6	Ing3	Analog	Ing3	M1Ing3	M2Ing3	23	23
7	Ing4	Analog	Ing4	M1Ing4	M2Ing4	24	24
8	SP1	Analog	SP1	M1SP1	M2SP1	21	
9	SP2	Analog	SP2	M1SP2	M2SP2	22	
10	SP3	Analog	SP3	M1SP3	M2SP3	23	
11	SP4	Analog	SP4	M1SP4	M2SP4	24	
12	SP5	Analog	SP5	M1SP5	M2SP5	25	
13	RevDate	Message	Date			7/6/97 15:36	
14	Comment	Message	Comment			Comment for Recipe 2	

4. 编辑配方文件的内容，然后保存更改。

在 Excel 中创建新的配方模板文件

1. 启动 Excel。
2. 创建新的工作簿。
3. 在电子表格中输入配方数据，如下图所示。



	A	B	C	D	E	F	G
1	:IngredientName	IngredientType	Unit	Unit	Unit	Unit	Recipe
2	:Names		Review	Mixer 1	Mixer 2	Mixer 3	Recipe 1
3	Ing1	Analog	Ing1	M1Ing1	M2Ing1	M3Ing1	11
4							
5							
6							
7							
8							

各个项目必须按照图中所示的顺序进行输入。“单元名”必须在包含“配方名”的列左侧的列中定义。

4. 使用 .csv 文件扩展名保存电子表格。

使用记事本处理配方模板文件

如果不希望使用 Recipe Manager 实用程序，则可以使用“记事本”来创建或编辑配方模板。使用“记事本”创建或编辑的 Recipe Manager 模板必须保存到文件扩展名为 .csv 的文件。

在“记事本”中打开现有的配方模板文件

1. 启动“记事本”。
2. 在文件菜单上，选择打开。此时出现“打开”对话框。
3. 找到并选择配方文件，然后选择打开，或双击文件名。
4. 编辑配方文件的内容，然后保存更改。

在“记事本”中创建新的配方模板文件

1. 启动“记事本”。
2. 在文件菜单上，选择新建。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

3. 按下面这种格式输入以下数据：

```
:IngredientName,IngredientType[,Unit]...[,Recipe]...
:Names,,[,UnitName]...[,RecipeName]...
IngredientName,{Analog,Discrete,Message},[,tag]...[,value]
```

注意：所有的“单元名”必须先于“配方名”在文件中进行定义。

4. 使用 .csv 文件扩展名保存文件。

通过嵌套配方来创建复杂的结构

使用 InTouch QuickScript 可以将多个配方模板文件互相链接起来，从而创建复杂的应用。通过定义一个与“单元名”模板中的消息标记关联的成分名，便可以链接配方模板文件。然后，您使用配方的名称来加载该消息标记。

通过链接配方模板文件，可以创建主配方模板文件，这些文件定义不同配方文件中的各种配方所使用的机器配置参数。将这类信息保存在一个中心文件中，可以大幅减少数据发生更改时进行维护与更新所需的时间。

在下图中，“项目名”Setup 标记定义为消息型，并且单元中包含此项目的 Setup 消息标记。每个配方都包含在另一个配方文件（选择配方时，此文件加载到 Setup 标记）中定义的第二个配方名。

RECFILEA.CSV

	1	2	3	4	5	6	7	8	9
1	:Item Name	Item Type	Unit	Unit	Unit	Unit	Recipe	Recipe	Recipe
2	:Names		**Review**	Mixer 1	Mixer 2	Mixer 3	Recipe 1	**Recipe 2**	Recipe 3
3	Ing1	Analog	Ing1	M1Ing1	M2Ing1	M3Ing1	11	21	31
4	Ing2	Analog	Ing2	M1Ing2	M2Ing2	M3Ing2	12	22	99
5	Ing3	Analog	Ing3	M1Ing3	M2Ing3	M3Ing3	13	23	66
6	Ing4	Analog	Ing4	M1Ing4	M2Ing4	M3Ing4	14	24	34
7	SP1	Analog	SP1	M1SP1	M2SP1	M3SP1	11	21	31
8	SP2	Analog	SP2	M1SP2	M2SP2	M3SP2	12	22	32
9	SP3	Analog	SP3	M1SP3	M2SP3	M3SP3	13	23	33
10	SP4	Analog	SP4	M1SP4	M2SP4	M3SP4	14	24	34
11	SP5	Analog	SP5	M1SP5	M2SP5	M3SP5	15	25	35
12	**Setup**	Message	**Setup**	LinkFile	LinkFile	LinkFile	Setup2A	**Setup3A**	Setup1A
13									

为此，将输入以下脚本：

```
RecipeName="Recipe2";
RecipeLoad("c:\recipe\recfilea.csv", "Review", RecipeName);
```

此脚本运行时，**Setup** 标记的值变为 Setup3A，并加载到 Review 单元。随后，通过运行以下脚本来执行下一次配方加载操作，以便将机器设置参数加载到为 PLC1 单元定义的标记，并将 Setup 标记的值用作“配方名”。

```
RecipeLoad("c:\recipe\machine.csv", "PLC1", Setup);
```

MACHINE.CSV

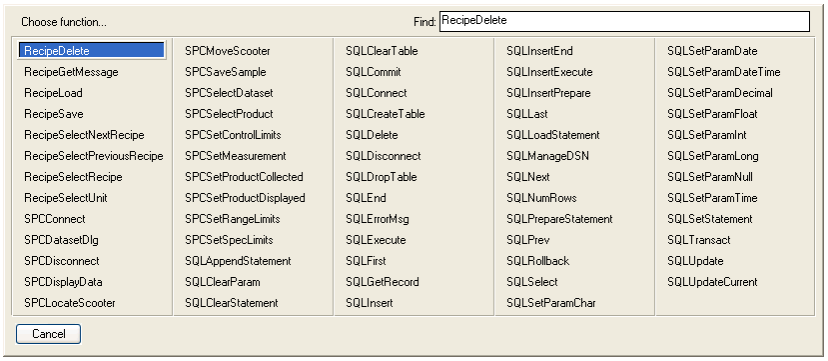
	1	2	3	4	5	6	7	8
1	:Item Name	Item Type	Unit	Recipe	Recipe	Recipe		
2	:Names		**PLC1**	Setup1A	Setup2A	**Setup3A**		
3	PARM1	Analog	PARM1	11	21	31		
4	PARM2	Analog	PARM2	12	22	99		
5	PARM3	Analog	PARM3	13	23	66		
6	PARM4	Analog	PARM4	14	24	34		
7	PARM5	Analog	PARM5	11	21	31		
8	PARM6	Analog	PARM6	12	22	32		
9	PARM7	Analog	PARM7	13	23	33		
10	PARM8	Analog	PARM8	14	24	34		
11	PARM9	Analog	PARM9	15	25	35		
12								

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在 InTouch 中使用配方

您可以使用 InTouch QuickScript 与配方模板文件进行交互。Recipe Manager 包含一套可插入 QuickScript 中的脚本函数。通过使用包含这些函数的脚本，可以在配方模板文件中选择、修改、插入或删除记录。

使用 InTouch 脚本编辑器可以将配方函数添加到任何类型的脚本。下图显示列出了配方函数的 InTouch 脚本编辑器窗口。所有的配方函数都通过将前缀 Recipe 用作函数名的一部分来进行标识。



InTouch 配方函数直接读取和写入配方模板文件。因此，Recipe Manager 程序不需要处在运行状态，配方函数便可以在 InTouch QuickScript 中正常运行。

如果配方模板文件正由 InTouch HMI 使用，则创建的任何新配方，或是对现有配方所作的任何更改，都无法写入配方模板文件。Recipe Manager 仅用于创建配方模板文件。创建配方模板文件之后，关闭 Recipe Manager。

将配方函数自动插入到脚本中

1. 打开 InTouch 脚本编辑器。
2. 将光标放到脚本中要插入配方函数的位置。
3. 在函数下，选择附件。 此时出现“选择函数”对话框。
4. 选择要插入到 QuickScript 中的配方函数。 此时对话框关闭，该函数插入到光标位置。

在配方文件中加载与保存配方数据

Recipe Manager 提供单独的 InTouch QuickScript 函数在配方文件中加载与保存配方数据。

RecipeLoad() 函数

RecipeLoad() 函数将配方数据加载到 InTouch 应用程序中特定标记的单元。

类别

配方

语法

```
RecipeLoad("Filename","UnitName","RecipeName");
```

参数

FileName
配方模板文件的名称。与 FileName 关联的值可以是字符串常数，也可以是包含配方模板文件名的消息标记。

UnitName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

指定的配方模板文件中特定单元的名称。RecipeSelectUnit() 函数将值返回给此参数。与 UnitName 关联的值可以是字符串常数，也可以是包含该单元名的消息标记。

RecipeName
指定的配方模板文件中特定配方的名称。与 *RecipeName* 关联的值可以是字符串常数，也可以是包含配方名的消息标记。

示例

下面的语句将 Recipe1 配方的值加载到为 Unit1 定义的标记中：

```
RecipeLoad("c:\recipe\recfile.csv", "Unit1", "Recipe1");
```

RecipeSave() 函数

RecipeSave() 函数将新配方或是对现有配方所作的更改保存到指定的配方模板文件。

类别

配方

语法

```
RecipeSave("Filename","UnitName","RecipeName");
```

参数

FileName
配方模板文件的名称。与 FileName 关联的值可以是字符串常数，也可以是包含配方模板文件名的消息标记。

UnitName
指定的配方模板文件中此函数将要使用的特定单元的名称。RecipeSelectUnit() 函数将值返回给此参数。与 UnitName 关联的值可以是字符串常数，也可以是包含该单元名的消息标记。

RecipeName
指定的配方模板文件中特定配方的名称。与 *RecipeName* 关联的值可以是字符串常数，也可以是包含配方名的消息标记。

示例

下例将对 Recipe3 配方所作的更改保存到 recfile.csv 文件。如果 recfile.csv 文件中当前不存在 Recipe3，则会创建它。这些值用于设置为 Unit2 定义的标记的值：

```
RecipeSave("c:\recipe\recfile.csv", "Unit2", "Recipe3");
```

从配方文件中删除配方

使用 **RecipeDelete** 函数，可以从指定的配方模板文件中删除配方。

RecipeDelete() 函数

RecipeDelete 函数从指定的配方模板文件中删除配方。

类别

配方

语法

```
RecipeDelete("Filename","RecipeName");
```

参数

FileName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配方模板文件的名称。与 FileName 关联的值可以是字符串常数，也可以是包含配方模板文件名的消息标记。

RecipeName
指定的配方模板文件中特定配方的名称。与 *RecipeName* 关联的值可以是字符串常数，也可以是包含配方名的消息标记。

示例

下面的语句从 recfile.csv 文件中删除 Distlt1 配方：

```
RecipeDelete("c:\recipe\recfile.csv", "Distlt1");
```

选择单元（标记成分映射）

使用 RecipeSelectUnit() 函数，可以选择要加载当前配方值的标记所对应的单元。

RecipeSelectUnit() 函数

RecipeSelectUnit() 函数打开选择单元对话框，供运行时用户选择单元。所选的单元名返回到消息标记。RecipeSelectRecipe() 与 RecipeSelectUnit() 函数都与 RecipeLoad() 函数配合使用。

类别

配方

语法

```
RecipeSelectUnit("Filename","UnitName",Number);
```

参数

FileName
配方模板文件的名称。FileName 参数可以是字符串常量，也可以是包含配方模板文件名的消息标记。

UnitName
写入所选单元名称的消息标记。不带英文引号的实际消息标记，或是字符串。

编号
返回给参数的最大字符串长度。在 InTouch 中，字符串（消息）标记的最大长度是 131 个字符。除非已缩短 InTouch 标记的最大字符串长度，否则此参数使用 131。数字或整型标记。

示例

下面的语句打开选择单元对话框：

```
RecipeSelectUnit("c:\recipe\recfile.csv", UnitName, 131);
```

选择<1>“</1>单元<2>”</2>之后<3>，</3>其名称返回给 UnitName 标记<4>。</4>

从配方文件中选择单独的配方

Recipe Manager 包括一套可以从配方文件中选择单独配方的函数。在脚本中使用这些函数时，可以根据名称从文件中选择配方，也可以根据文件中的顺序选择上一个或下一个配方。

RecipeSelectRecipe() 函数

RecipeSelectRecipe() 函数打开选择配方对话框，供运行时用户选择配方。所选的配方名返回到消息标记。

类别

配方

语法

```
RecipeSelectRecipe("Filename","RecipeName", Number);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

FileName
配方模板文件的名称。FileName 参数可以是字符串常量，也可以是包含配方模板文件名的消息标记。

RecipeName
写入所选配方名称的消息标记。不带英文引号的实际消息标记，或是字符串。

编号
返回给参数的最大字符串长度。InTouch 消息标记的最大长度是 131 个字符。除非已缩短 InTouch 标记的最大字符串长度，否则此参数使用 131。数字或整型标记。

示例

下面的语句会打开“选择配方”对话框：

```
RecipeSelectRecipe("c:\recipe\recfile.csv", RecipeName, 131);
```

从对话框中选择配方之后，其名称返回给 **RecipeName** 标记。

RecipeSelectNextRecipe() 函数

RecipeSelectNextRecipe() 函数选择配方模板文件中的下一个配方。

类别

配方

语法

```
RecipeSelectNextRecipe("Filename", "RecipeName", Number);
```

参数

FileName
配方模板文件的名称。FileName 参数可以是字符串常量，也可以是包含配方模板文件名的消息标记。

RecipeName
消息标记，包含用作起点的配方名（执行函数之前）与所选的配方名（执行函数之后）。不带英文引号的实际消息标记，或是字符串。

编号
返回给参数的最大字符串长度。在 InTouch 中，字符串（消息）标记的最大长度是 131 个字符。除非已缩短 InTouch 标记的最大字符串长度，否则此参数使用 131。数字或整型标记。

示例

下面的语句读取 RecipeName 标记的当前值并返回文件中的下一个配方。如果 RecipeName 的值为空或是无法找到，则返回文件中的第一个配方。如果 RecipeName 当前包含文件中的最后一个“配方名”，则原封不动地返回该配方名。配方按照它们的创建顺序保存在配方模板文件中。

```
RecipeSelectNextRecipe("c:\recipe\recfile.csv",
RecipeName, 131);
```

RecipeSelectPreviousRecipe() 函数

RecipeSelectPreviousRecipe() 函数选择配方模板文件中定义的上一个配方。

类别

配方

语法

```
RecipeSelectPreviousRecipe("Filename","RecipeName",Number);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

FileName
配方模板文件的名称。FileName 参数可以是字符串常量，也可以是包含配方模板文件名的消息标记。

RecipeName
消息标记，包含用作起点的配方名（执行函数之前）与所选的配方名（执行函数之后）。不带英文引号的实际消息标记，或是字符串。

编号
返回给此参数的最大字符串长度。在 InTouch 中，消息标记的最大长度是 131 个字符。除非已缩短 InTouch 标记的最大字符串长度，否则此参数使用 131。数字或整型标记。

示例

下面的语句会导致系统读取 RecipeName 标记的当前值，并返回文件中的上一个“配方名”。返回的这个字符串将存储在 RecipeName 中，并且将覆盖掉当前值。如果 RecipeName 的值为空或是无法找到，则返回文件中的最后一个配方。如果 RecipeName 当前包含文件中的第一个“配方名”，则原封不动地返回该配方名。配方按照它们的创建顺序进行保存。

```
RecipeSelectPreviousRecipe("c:\recipe\recfile.csv", RecipeName, 131);
```

理解配方脚本函数返回的错误消息

通过使用配方函数返回的诊断错误码，可以排解配方应用程序的错误。本节包含配方函数错误码列表，并介绍如何使用 RecipeGetMessage() 函数来显示与错误码关联的消息。

如果 RecipeLoad() 函数成功执行，它会将 ErrorCode 模拟标记的值设置为 0。如果 RecipeLoad() 失败，它会将 ErrorCode 标记设置为特定错误条件的编号。

要检索配方函数的错误码，必须将它赋给某个 InTouch 模拟标记。下例显示一个可以返回配方函数错误码的脚本语句：

```
ErrorCode = RecipeLoad(FileName, UnitName, RecipeName);
```

显示错误码消息

每个配方函数都会返回一个代表该函数的错误条件的编号。通过在 InTouch 的“数据改变”脚本中使用 RecipeGetMessage() 函数，可以将相应的错误码写入一个模拟标记，并将关联的错误码消息写入一个消息标记。

下面的代码示例显示一个“数据改变”脚本。

```
RecipeGetMessage(ErrorCode, ErrorMessage, 131);
```

只要 ErrorCode 标记的值发生改变，此脚本便会自动运行。此脚本运行时，RecipeGetMessage() 函数读取 ErrorCode 标记的当前数值，并将与该值关联的消息返回给 ErrorMessage 标记。

下表列出可能的错误码及其对应的错误消息与描述：

值	错误消息	描述
0	成功	调用的配方函数成功执行。
-1	无此配方模板	指定的配方模板文件不存在。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

值	错误消息	描述
-2	View 没有激活	由于 WindowViewer 不在运行，另一程序所调用的配方函数无法执行。
-3	内存不足	内存不足，无法完成当前的活动。
-4	在配方模板文件中的行太长	配方模板文件中的某一行超出最大允许长度。
-5	配方模板文件中的行被截断	配方模板文件中的某一行被截断。
-6	不是一个有效的配方模板文件	指定的文件不是一个有效的配方模板文件。
-7	期望“单元”或“配方”	配方模板文件中遗失单元名或配方名。
-8	配方模板文件中未定义单元	配方文件的“单元定义”模板中尚未定义任何单元。
-9	配方模板文件中未找到配方名	指定的配方在配方模板文件中未定义。
-10	配方模板文件中未找到单元名	指定的单元名在单元定义模板文件中未定义。
-12	期望是“模拟”、“离散”或“消息”	给配方模板文件中的某个项目输入的类型不正确。有效的类型是 Analog（模拟）、Discrete（离散）或 Message（消息）。
-13	标记类型与“模拟”、“离散”或“消息”不匹配	指定的标记与项目类型不匹配。例如，将某个配方项目定义为模拟类型，同时又已将某个消息标记定义为它的单元。
-14	无效的离散值，期望是“0”或“1”	给配方模板文件中的某个离散标记输入了不正确的值。离散标记的有效值仅限 0 或 1。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

值	错误消息	描述
-15	无法打开临时文件	临时文件无法打开，可能是由于空闲磁盘空间不足。
-16	保存配方模板文件时发生写入错误	保存配方模板文件期间发生错误。
-17	没有选定用户	用户在“选择配方”对话框中没有选择配方名，而是选择了“取消”。
-19	另一应用程序正在使用配方模板，或者当前运行查看操作的用户没有足够的安全权限来打开该文件	指定的配方模板文件已经打开，或者用户没有足够的权限，因此无法通过 WindowViewer 访问。

RecipeGetMessage() 函数

RecipeGetMessage() 函数接受错误号（由其它一些配方函数返回），并返回该错误号的纯文本错误消息。

类别

配方

语法

```
RecipeGetMessage(Analog_Tag,Message_Tag,Number);
```

参数

Analog_Tag
要获取其错误消息的错误号。

Message_Tag
不带英文引号的实际消息标记，或是字符串。

编号
Number 参数设置由 *Message_Tag* 参数返回的字符串的最大长度。缺省条件下，InTouch 消息标记设置为最大长度，即 131 个字符。除非已经缩短“InTouch 标记名字典”中 *Message_Tag* 标记的最大字符串长度，否则此参数使用 131。*Number* 参数可以是常数，也可以是包含数字的整型标记。

示例

通过在 InTouch 的“数据改变”QuickScript 中使用 RecipeGetMessage() 函数，可以将错误码写入 ErrorCode 标记，将关联的错误码消息写入 ErrorMessage 标记：

```
Data Change Script Tagname[.field]:ErrorCode
Script body:RecipeGetMessage(ErrorCode, ErrorMessage,131);
```

只要 ErrorCode 标记的值发生改变，此 QuickScript 便自动运行。此 QuickScript 运行时，RecipeGetMessage() 函数读取 **ErrorCode** 标记的当前数值，并将与该值关联的消息返回给 **ErrorMessageTag**。

```
ErrorCode = RecipeLoad ("c:\App\recipe.csv","Unit1","cookies");
RecipeGetMessage(ErrorCode, ErrorMessageTag, 131);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

向配方应用安全机制

通过在配方模板文件中定义一个“项目名”，由它设置加载、保存或删除配方时所需的最低安全访问级别，可以控制对配方的访问权限。

在下面的文件示例中，SecurityLevel 项目名定义为模拟标记。Review 单元包含此项目的 SecurityLevel 模拟标记。每个配方定义一个值，在该配方加载到 Review 单元时，这个值加载到 SecurityLevel 标记中。

MACHINE.CSV

	1	2	3	4	5	6	7
1	:Item Name	Item Type	Unit	Unit	Recipe	Recipe	Recipe
2	:Names		**Review**	PLC1	Setup1A	Setup2A	**Setup3A**
3	PARM1	Analog		PARM1	11	21	31
4	PARM2	Analog		PARM2	12	22	99
5	PARM3	Analog		PARM3	13	23	66
6	PARM4	Analog		PARM4	14	24	34
7	PARM5	Analog		PARM5	11	21	31
8	PARM6	Analog		PARM6	12	22	32
9	PARM7	Analog		PARM7	13	23	33
10	PARM8	Analog		PARM8	14	24	34
11	PARM9	Analog		PARM9	15	25	35
12	**SecurityLevel**	Analog	**SecurityLevel**		2000	5000	**7000**

您可以创建一个包含<1>“</1>访问被拒绝<2>”</2>消息的窗口<3>；</3>只要用户的安全访问级别对所选的配方而言无效<4>，</4>便显示该消息<5>。</5> 为此，所选的配方必须加载到仅包含一个模拟标记的单元，而所选配方的安全级别值将加载到此模拟标记中进行验证。

例如：

```
RecipeSelectRecipe("c:\recipe\machine.csv",MyRecipe, "131");
```

此时出现“选择配方”对话框。选择<1>“</1>配方名<2>”</2>之后<3>，</3>它返回给 RecipeName 标记<4>，</4>同时脚本继续运行<5>。</5>

```
RecipeLoad( "c:\Recipe\Machine.csv", "Review", MyRecipe );
IF SecurityLevel <= $AccessLevel THEN
   Status =RecipeLoad( "c:\Recipe\Machine.csv", "PLC1", MyRecipe );
   ELSE
   Show "Access Denied";
ENDIF;
```

此脚本运行时，如果访问级别大于或等于 7000，则所选配方的值加载到 PLC1 单元的标记中。否则出现“访问被拒绝”窗口，并且配方不会加载到 PLC1 中。

从 InTouch 中使用 SQL 数据库

数据库在共享共同的属性或字段的表中存储信息。“结构化查询语言”（Structured Query Language，简称 SQL）是用来以查询的形式访问该类信息的语言。“SQL 访问管理器”让您可以使用查询来访问、修改、创建以及删除数据库表。

“SQL 访问管理器”是可以随 InTouch 一起安装的可选程序。通过“SQL 访问管理器”可以：

- 创建与运行复杂查询。这些查询可以动态构建，也可以保存在外部文件中。此外，这些查询可以包含要在运行时传递给查询的参数。
- 运行数据库支持的 SQL 语句并从查询中检索结果。您还可以给“SQL 访问管理器”使用存储过程，尽管并非完全支持所有的存储过程。

通过在 QuickScript 编辑器对话框中选择“附件”按钮，可以自动将 SQL 函数插入 InTouch QuickScript。SQL 函数自动插入到脚本中的当前光标位置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您可以使用“SQL 访问管理器”将数据（如批次配方）从 SQL 数据库传输到 InTouch 应用程序。“SQL 访问管理器”还可以用来将运行时数据、报警状态或历史数据从 InTouch 传输到数据库。例如，在完成一个机器周期之后，公司希望保存几组数据，每组针对一个不同的应用程序。SQL 数据库提供了在一个或多个第三方应用程序之间轻松传输信息的功能。“SQL 访问管理器”使这些数据可以在任何 InTouch 应用程序中进行访问与显示。

“SQL 访问管理器”由一个程序和一套 SQL 函数组成。“SQL 访问管理器”程序创建数据库列，并将它们与 InTouch 标记关联起来。将数据库列与 InTouch 数据库标记关联起来的过程称为绑定。通过将 InTouch 数据库标记绑定到数据库列，“SQL 访问管理器”可以直接操纵数据库中存储的 InTouch 数据。

“SQL 访问管理器”在逗号分隔变量文件 SQL.DEF 中保存数据库字段名及其关联的对象。此文件位于 InTouch 应用程序文件夹，可以使用“SQL 访问管理器”或任何文本编辑器（如“记事本”）进行查看或修改。“SQL 访问管理器”还可以创建“表模板”来定义 InTouch 使用的数据库的结构与格式。

SQL 函数可在脚本中使用，根据操作员的输入或标记值的变化，或存在一组特定的条件时自动运行。您可以使用这些函数在选择访问的表中选择、修改、插入或删除记录。例如，如果存在某个报警条件，则应用程序可以运行包含 SQLInsert() 或 SQLUpdate() 函数的脚本，以便保存所有适当的数据点以及报警的状态。

设置数据源

“SQL 访问管理器”是一个与 ODBC 兼容的应用程序，可以同支持可用 ODBC 驱动程序或 OLE DB 供应器的任何数据库进行通讯。

配置数据库连接字符串有多种方法：

- 使用“Microsoft ODBC 管理器”程序来配置要给“SQL 访问管理器”使用的 ODBC 驱动程序。
- 运行 SQLConnect() 函数并使用参数值来指定 OLE DB 供应器。如需详细信息，请参阅 SQL Server 数据库应用程序。
- 使用外部 UDL 文件来设置数据库连接字符串。

配置 ODBC 驱动程序

1. 运行“Microsoft ODBC 管理器”程序。
2. 选择驱动程序或数据源，然后选择添加新名称、设置缺省值或配置。此时出现 **ODBC 驱动程序设置**对话框。

选项	描述
数据源名	用于确定数据源的用户自定义名称。
描述	数据源的用户自定义描述。
数据库目录	确定包含数据库文件的文件夹。如果未指定，则使用当前工作文件夹。

3. 选择确定。

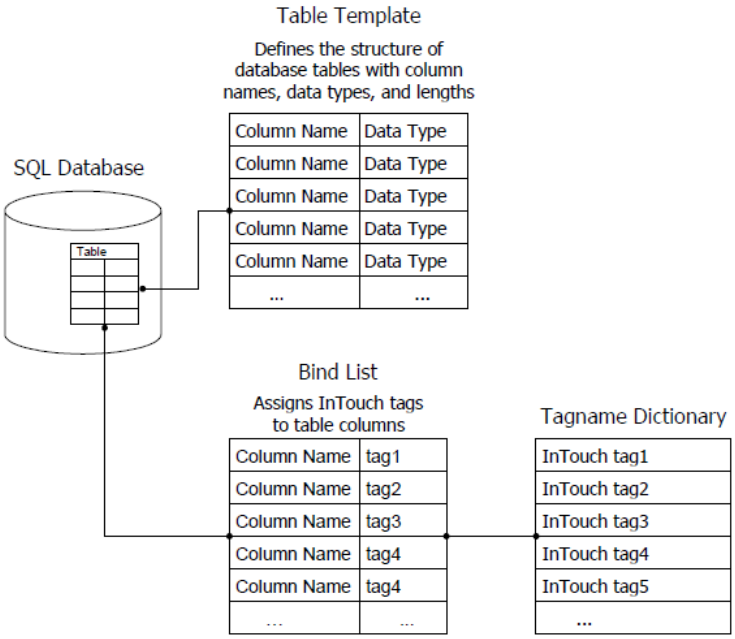
注意：创建“ODBC 数据源”时，会在 Windows 目录中创建一个 ODBC.INI 文件。您可以手工编辑 ODBC.INI 文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将 InTouch 标记映射到数据库列

您可以将 InTouch 标记映射到数据库列。这是通过“绑定列表”来完成的。大多数 SQL Access 函数都通过“绑定列表”来使得 InTouch 标记可以访问 SQL 数据库表中的数据。

“绑定列表”将数据库表列与 InTouch 的“标记名字典”中的标记关联起来。“绑定列表”还包含一个描述数据库表格式的“表模板”。



运行包含 SQLInsert()、SQLSelect() 或 SQLUpdate() 函数的脚本时，“绑定列表”更新为指定使用了哪些 InTouch 标记，以及同这些标记关联的是哪些表格列。

创建新的“绑定列表”

1. 在工具窗格中，展开 **SQL 访问管理器**，然后选择**绑定列表**。

   此时出现一条确认 SQL.DEF 文件创建的消息。

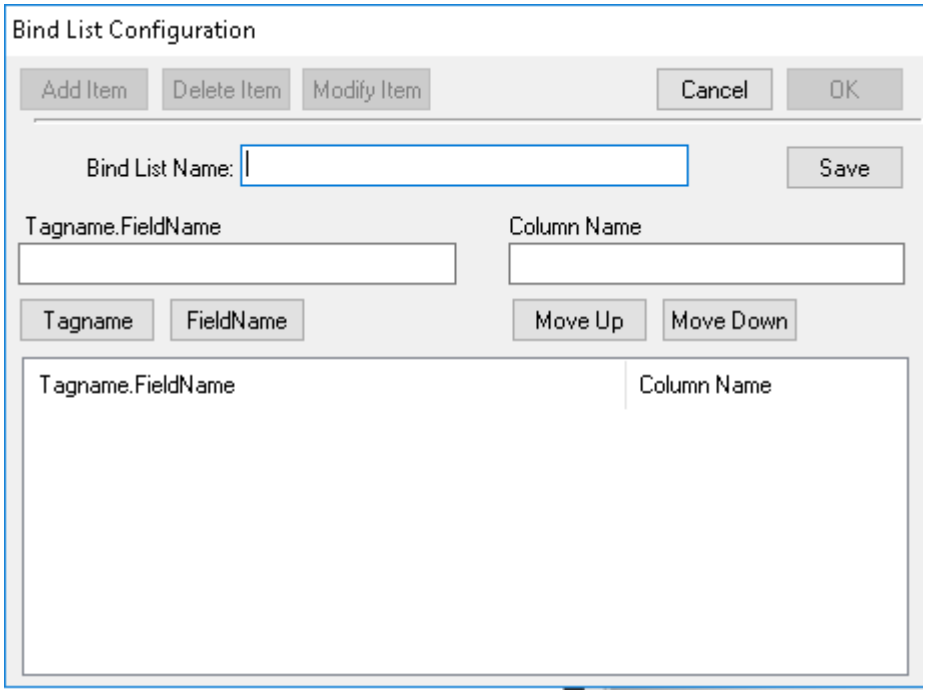
2. 单击是以创建 SQL.DEF 文件。

   此时出现**选择绑定列表对话**框。

3. 选择新建。

   此时出现**绑定列表配置对话**框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 在**绑定列表名**框中，输入“绑定列表名”。

   “绑定列表名”的最大长度是 32 个字符。

5. 要为该“绑定列表”定义标记，执行以下操作之一：
   - 在**标记名.域名**框中，输入 InTouch 标记名。您也可以按 *tag_name.dotfield_name* 的形式添加可选的标记点域。
   - 选择**标记名**以选择一个现有的标记。此时出现**选择标记**对话框。从列表中选择一个标记。

   注意：WindowViewer 一启动，应用程序中未使用但在“SQL 访问绑定列表”中指定的 I/O 型标记便会立即激活（提示 DAServer）。

6. 通过执行以下操作之一选择要附加到标记的点域：
   - 在**标记名.域名**框中，在标记名后面输入一个英文句点和点域名。
   - 选择 **FieldName**。此时出现“选择域名”对话框。选择要附加到标记的点域。

7. 在**列名**框中，输入列的名称。

   列名的最大长度是 30 个字符。如果列名包含空格，则在“绑定列表”与脚本中使用时，请使用方括号将列名括起来。例如：

```
WHERE EXPR= "[Valve ID] = " + text (tagname,"#");
```

8. 通过执行以下操作之一调整标记在“绑定列表”中的位置：
   - 选择**向上移动**将所选的标记在列表中上移一级。
   - 选择**向下移动**将所选的标记在列表中下移一级。

9. 选择**添加项**将新的 Tagname.FieldName 与“列名”添加到“绑定列表”。

10. 选择**确定**以保存新的“绑定列表”配置并关闭对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置绑定列表中的 SQL Server 字符串分隔符

SQLInsert() 与 SQLUpdate() 函数使用缺省格式, 即使用英文单引号将消息字符串括起来。有些 SQL 数据库期望收到由其它类型的分隔符括起来的消息字符串。例如， Oracle 8.0 期望收到由方括号括起来的日期字符串。发生这种情况时，必须使用 Delim() 函数。

在“绑定列表配置”对话框的“列名”字段中，在列名后面使用 delim() 函数。输入关键字 "delim" 时，后面必须跟着：

- 英文左圆括号
- 左分隔符
- 系统区域设置中定义的列表分隔符
- 右分隔符
- 英文右圆括号

英文系统的示例：`datestring delim (',')`

德文系统的示例：`datestring delim (';')`

要使用相同的左分隔符与右分隔符，请如下例所示用圆括号指定分隔符（不含列表分隔符）：

```
datestring delim (' ')
```

修改“绑定列表”

1. 在工具窗格中，展开 SQL 访问管理器，然后选择绑定列表。

   此时出现一条确认 SQL.DEF 文件创建的消息。

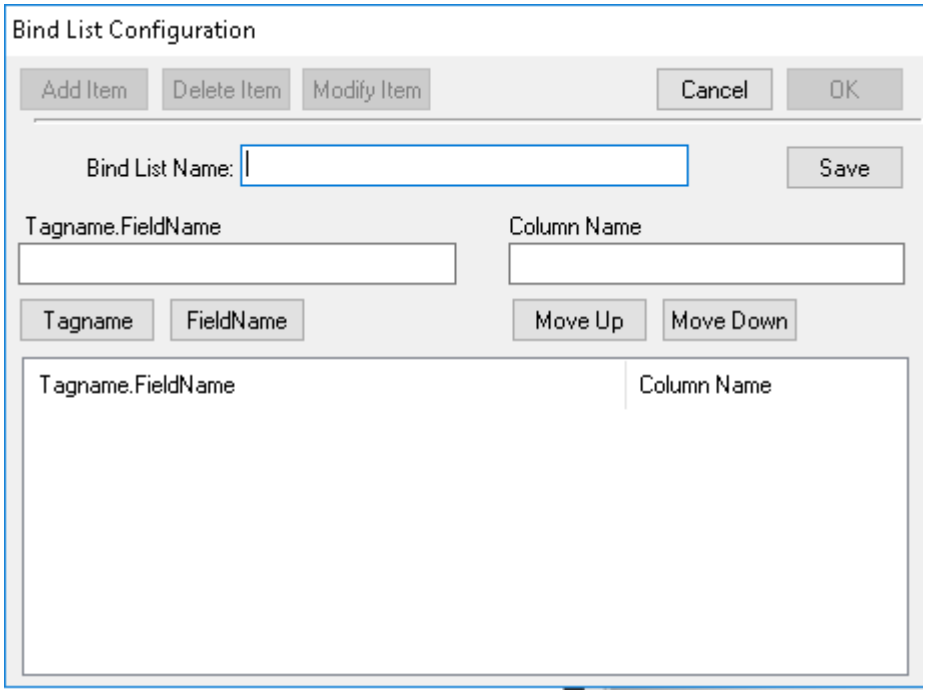
2. 单击是以创建 SQL.DEF 文件。

   此时出现选择绑定列表对话框。

3. 选择要更改的“绑定列表”名称，然后选择修改。

   此时出现绑定列表配置对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



4. 修改所需的项目。
5. 选择确定以保存更改并关闭对话框。

使用 Excel 修改“绑定列表”

“SQL 访问管理器”将“绑定列表”与“表模板”的配置信息保存到 SQL.DEF 文件。此文件采用逗号分隔值（CSV）格式。

SQL.DEF 文件可以使用任何支持“逗号分隔值”文件的产品（如 Excel）来修改。

数据出现在该文件中的形式如下：

```
:BindListName, BindListName
Tagname1.FieldName,ColumnName1
Tagname2.FieldName,ColumnName2
Tagname3.FieldName,ColumnName3
:TableTemplateName, TableTemplateName
ColumnName1,ColumnType,[ColumnLength],Null,Index
ColumnName2,ColumnType,[ColumnLength],Null,Index
ColumnName3,ColumnType,[ColumnLength],Null,Index
```

删除“绑定列表”

1. 在工具窗格中，展开 **SQL 访问管理器**，然后选择**绑定列表**。

   此时出现一条确认 SQL.DEF 文件创建的消息。

2. 单击是以创建 SQL.DEF 文件。

   此时出现**选择绑定列表对话**框。

3. 选择要删除的“绑定列表”名称。
4. 选择**删除**。此时出现一条消息，要求确认是否删除“绑定列表”。
5. 选择是以删除所选“绑定列表”。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

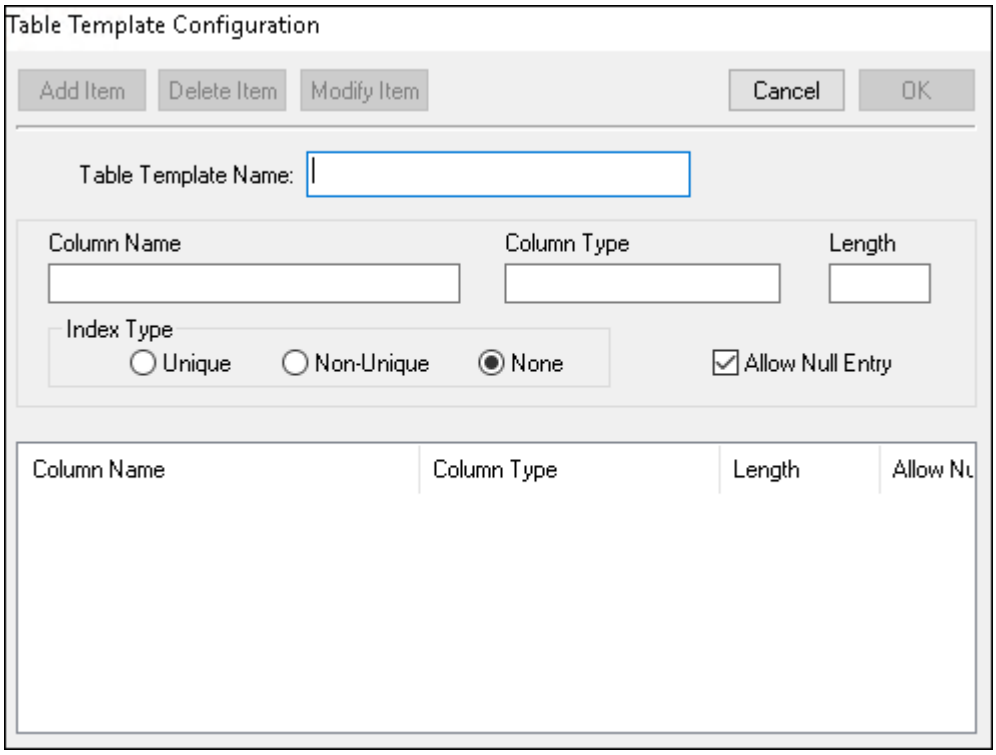
定义新表的结构

“表模板”定义数据库中创建的新表的结构与格式。“表模板”存储在 SQL.DEF 文件中。

创建新的“表模板”

1. 在工具窗格中，展开 SQL 访问管理器，然后选择表模板。
2. 选择新建。

   此时出现表模板配置对话框。



3. 在表模板名框中，输入“表模板”的名称。

   没有索引时，“表模板名”的最大长度是 32 个字符。如果要创建索引（不论是否唯一），则“表模板名”不得超过 24 个字符。
4. 在列名框中，输入“表模板”的列名。

   列名的最大长度是 30 个字符。
5. 在列类型框中，输入列的数据类型。根据使用的数据库的不同，数据类型选项会有所不同。
6. 在索引类型区域中，选择以下选项之一：
   - 唯一：每个列值必须是唯一的。
   - 不唯一：每个列值不要求是唯一的。
   - 无：无索引。

   注意：运行包含 SQLCreateTable() 函数的脚本时，自动创建一个索引文件。
7. 选择允许项目为空以允许在此列中输入空数据。

   注意：InTouch 不支持空数据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

插入数据时，如果尚未给标记输入值，则根据标记类型指定空值。

数据类型	值
离散	0
整型	0
消息	没有字符的字符串

8. 选择添加项，以将新的列名、列类型、长度以及索引类型添加到“表模板”。
9. 选择确定，以保存新的“表模板”配置并关闭对话框。

修改“表模板”

1. 在工具窗格中，展开 SQL 访问管理器，然后选择表模板。

   此时出现选择表模板对话框。
2. 选择要修改的“表模板”名，然后选择修改。

   此时出现表模板配置对话框。
3. 修改所需的项目。
4. 选择确定以保存更改并关闭对话框。

删除“表模板”

1. 在工具窗格中，展开 SQL 访问管理器，然后选择表模板。

   此时出现选择表模板对话框。
2. 选择要删除的“表模板”名称。
3. 选择删除。此时出现一条消息，要求确认是否删除“表模板”。
4. 选择是。此时再次出现“表模板配置”对话框。
5. 选择“确定”以关闭对话框。

使用数据库应用程序

“SQL 访问管理器”支持 Oracle、Microsoft SQL Server 以及 Microsoft Access 数据库。每种数据库都有一些独特的要求。本节包含几个单独的小节，分别介绍如何配置每种数据库与“SQL 访问管理器”之间的连接。

SQL Server 数据库应用程序

您可以在 InTouch QuickScript 中使用 SQLConnect() 函数来连接 Microsoft SQL Server 数据库。SQLConnect() 函数将用户登录到 SQL Server 数据库并打开一个连接。SQLConnect() 函数使用的连接字符串的格式如下：

(SQLConnect(*ConnectionId*,"<*attribute*>=<*value*>;
<*attribute*>=<*value*>;...");

ConnectionID 参数是包含会话编号的整型标记。几乎每个其它 SQL Access 函数都要使用这个会话编号来引用与 SQL Server 数据库的连接。每调用一次 SQLConnect() 函数，会话编号递增 1。

下表介绍 Microsoft SQL Server 使用的 **SQLConnect()** 函数属性：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

属性	值
供应器	SQLOLEDB
数据源	安装数据库的服务器名
初始目录	数据库名
用户 ID	登录 ID，区分大小写
密码	口令，区分大小写

```
"Provider=SQLOLEDB.1;User ID=UserIDStr; Password=PasswordStr;Initial
Catalog=DatabaseName;Data Source=ServerName;"
```

“SQL 访问管理器”将四种类型的 InTouch 标记（离散、整型、实型以及消息）同其它 SQL Server 数据库数据类型关联起来。

数据类型	长度	范围	标记类型
char	8,000 个字符	1 到 131	消息
Int		-2,147,483,648 到 2,147,483,647	整型
浮点	15 位数	-1.79E+308 到 1.79E+308	实型

char 数据类型包含固定长度的字符串。InTouch 消息标记要求使用 char 数据类型。字段长度必须指定。Microsoft SQL Server 数据库支持最大长度为 8000 个字符的 char 字段。不过，InTouch 消息标记限制为最多 131 个字符。如果消息标记值包含的字符超过给数据库字段指定的长度，则在将该字符串插入数据库时会截断它。

int 数据类型代表 InTouch 整型标记。如果未指定字段长度，则长度设置为数据库的缺省值。如果指定了长度<1>，</1>则它的形式是<2>“</2>宽度<3>”。</3> <1>“</1>宽度<2>”</2>确定列的最大位数<3>。</3>

float 数据类型代表 InTouch 实型标记。字段长度设置是由数据库固定下来的。此数据类型不要求指定字段长度。

Microsoft Access 数据库应用程序

要与 Microsoft Access 通讯，必须通过在 InTouch QuickScript 中执行 SQLConnect() 函数来连接它。

SQLConnect() 函数用于连接到 Microsoft Access 数据库。通过运行包含 SQLConnect() 函数的脚本，可以登录到数据库服务器，并打开一个连接，以便运行其它 SQL 函数。SQLConnect() 使用的连接字符串的格式如下：

SQLConnect(*ConnectionId*,"<*attribute*>=<*value*>;
<*attribute*>=<*value*>;...");

DSN 是 Microsoft Access 使用的一个独特属性<1>，</1>用于确定在<2>“</2>Microsoft ODBC 管理器<3>”</3>中配置的数据源的名称<4>。</4>

```
SQLConnect(ConnectionId,"DSN=MSACC");
```

“SQL 访问管理器”支持的有效数据类型取决于所使用的 ODBC 驱动程序的版本。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型	长度	缺省值	范围	标记类型
文本	255 个字符	--	--	消息
数字	--	--	--	整型
数字	--	--	--	实型

文本数据类型包含固定长度的字符串，并且用于 InTouch 消息标记。长度必须指定。Microsoft Access 数据库支持最大长度为 255 个字符的文本字段。InTouch 消息标记限制为 131 个字符。如果消息变量包含的字符超过给数据库字段指定的长度，则在将该字符串插入数据库时会截断它。Microsoft Access ODBC 驱动程序支持每个列名最多 17 个字符。使用 SQLSetStatement(Select Col1, Col2, ...) 时，支持的最大列数为 40。

Oracle 数据库应用程序

要在 SQL Access 与 Oracle 数据库之间进行通讯，必须通过运行包含 SQLConnect() 函数的脚本来连接它。

与 Oracle 8.0 数据库进行通讯

1. 确认运行 InTouch 的计算机上是否安装了 Oracle OLEDB Provider (MSDAORA.DLL) 文件。此文件是由随 InTouch 一起安装的 MDAC 来安装的。
2. 通过在 InTouch 动作脚本中执行 **SQLConnect()** 函数连接到 Oracle。

SQLConnect() 函数使用的连接字符串的格式如下：

SQLConnect(*ConnectionId*,"<*attribute*>=<*value*>;
<*attribute*>=<*value*>;...");

下表介绍 Oracle 使用的函数属性：

属性	值
供应器	MSDAORA
用户 ID	用户名
密码	密码
数据源	Oracle Server 机器名

```
SQLConnect(ConnectionId, "Provider=MSDAORA; Data Source=OracleServer; User ID=SCOTT;
Password=TIGER;");
```

下表列出“SQL 访问管理器”针对 Oracle 数据库所支持的有效数据类型。

数据类型	长度	缺省值	范围	标记类型
char	2000 个字符	1 个字符		消息
数字	38 位	38 位		整型

要将日期与时间记录到 Oracle 8.0 日期字段中，必须使用 delim() 函数配置绑定列表。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将日期与时间记录到 Oracle 日期字段

1. 在“应用程序浏览器”下的“SQL 访问管理器”中，选择“绑定列表”。此时出现“绑定列表配置”对话框。
2. 在“标记名.域名”框中，输入要使用的标记。例如，DATE_TIME_TAG。
3. 在“列名”框中，输入 Oracle 日期字段的名称。如果使用 Oracle 8.0，请使用 delim() 函数指定任何分隔符。如果使用 Oracle 9.2 或更高版本，则不需要 delim() 函数。
4. 在 InTouch 应用程序中，创建一个 QuickScript，以准备根据当前日期与时间得到的输入数据。例如：
   ```
   DATE_TIME_TAG = "TO_DATE('" + $DateString + " " + StringMid($TimeString,1,8) +
   "','mm/dd/yy hh24:mi:ss')";
   ```
   此 QuickScript 在 WindowViewer 中运行之后，日期按以下格式出现：
   ```
   TO_DATE('08/22/06 23:32:18' ,'mm/dd/yy hh24:mi:ss')
   ```

在 InTouch 中执行常见的 SQL 操作

InTouch 使用 SQL Access 函数与数据库中存储的信息进行交互。通过使用这些 SQL Access 函数，可以编写脚本来选择、修改、插入或删除数据库记录。

SQL 动作是同步型的。从 InTouch 应用程序中运行数据库 QuickScript 时，只有在函数请求的数据库操作完成之后，控制权才返回给 InTouch。

SQL Access 函数遵循一些标点符号标准，用以描述与函数关联的参数的类型。使用英文双引号括起的脚本字符串来输入参数 ("Arg1") 时，会完全按照原样使用该字符串。如果未使用英文双引号，则将参数值视位标记名，同时将标记的当前值与该参数关联起来。

大多数 SQL 函数都会返回结果码。如果结果码不为零，则函数失败，此时应该采取其它措施。结果码可以由 SQLErrorMsg() 函数使用。

通过使用 InTouch QuickScript 编辑器，可以将 SQL 函数插入到 QuickScript 中。将 SQL 函数插入到脚本中的一般操作程序包括以下步骤：

将 SQL 函数添加到脚本中

1. 启动 InTouch WindowMaker。
2. 使用 QuickScript 编辑器打开 QuickScript。
3. 将光标放到脚本中要插入 SQL 函数的位置。
4. 在函数区域中，单击附件以显示选择函数对话框。
5. 选择要插入到 QuickScript 中的 SQL 函数。此时该脚本更新并显示所插入的 SQL 函数。

与 SQL Access 函数关联的参数包括：

- *BindList*

  对应于 SQL.DEF 文件中定义的“绑定列表”名。

- *ConnectionID*

  ConnectionID 参数引用某个内存整型标记的名称，该标记存放由 **SQLConnect()** 函数指定给每个数据库连接的编号 (ID)。

- *ConnectString*

  ConnectString 确定数据库系统以及任何附加的登录信息。它按照以下格式输入：
  ```
  "DSN=data source name[;attribute=value
  [;attribute=value]...]"
  ```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

Microsoft SQL Server 连接字符串

- Microsoft OLE DB Provider for SQL Server（建议使用）。
```
"Provider=SQLOLEDB.1;User ID=sa; Password=;Initial Catalog=MyDB;Data
Source=MyServer;"
```
  OLE DB Provider for SQL Server 是 sqloledb。
- Microsoft OLE DB Provider for SQL Server（建议使用）
```
"Provider=SQLOLEDB.1;uid=sa;pwd=;Database=MyDB"
```
- Microsoft OLE DB Provider for ODBC（使用缺省供应器 MSDASQL for SQL Server）：
```
"DSN=Pubs;UID=sa;PWD=;"
```
- Microsoft OLE DB Provider for ODBC（使用缺省供应器 MSDASQL for SQL Server）：
```
"Data Source=Pubs;User ID=sa;Password=;"
```

Oracle 连接字符串

- Microsoft OLE DB Provider for Oracle（建议使用）
```
"Provider=MSDAORA;Data Source=ServerName;User ID=UserIDStr;
Password=PasswordStr;"
```

Microsoft Access 连接字符串

Microsoft OLE DB Provider for Microsoft Jet（建议使用）。Microsoft.Jet.OLEDB.4.0 是本机 OLE DB Provider for Microsoft Jet（Microsoft Access Database 引擎）。
```
"Provider=Microsoft.Jet.OLEDB.4.0;Data Source=d:\DBName.mdb;User
ID=UserIDStr;Password=PasswordStr;"
```

Microsoft OLE DB Provider for ODBC（使用缺省供应器 MSDASQL for MS Access）：
```
"Provider=MSDASQL;DSN=DSNStr;UID=UserName; PWD=PasswordStr;"
```

- *ErrorMsg*

  包含对错误的文本描述的消息变量。

- *FileName*

  包含信息的文件的名称。

- *MaxLen*

  与参数关联的列的最大长度。此参数确定数据是可变字符型还是长可变字符型。如果 MaxLen 小于或等于数据库允许的最大字符串长度，则数据是可变字符型。如果大于该数值，则数据是长可变字符型。

- *OrderByExpression*

  定义各个列以及是按升序还是降序进行排列。只有列名可用于排序。表达式必须采用以下格式：

  ColumnName [ASC|DESC]

  要按列名的升序方式给所选的表排序：
```
"manager ASC"
```
  要按多个列进行排序，则表达式的格式为：

  ColumnName [ASC|DESC],

  ColumnName [ASC|DESC]

  要按某个列名（如 temperature）的升序、另一个列名（如 time）的降序给所选的表排序：
```
"temperature ASC, time DESC"
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- *ParameterNumber*

  语句中的实际参数编号。

- *ParameterType*

  指定的参数的数据类型。有效值是：

类型	描述
Char	用空格填充的固定长度字符串
Var Char	可变长度字符串
十进制	BCD 数
整型	4 字节带符号整数
小整数	2 字节带符号整数
浮点	4 字节的浮点数
双精度浮点数	8 字节的浮点数
日期时间	8 字节日期时间值
日期	4 字节日期时间值
时间	4 字节日期时间值
无类型	无数据类型

- *ParameterValue*

  要设置的实际值。

- *精度*

  是十进制值的精度、字符的最大个数，或日期时间值的字节长度。

- *RecordNumber*

  要检索的实际记录号。

- *ResultCode*

  大多数 SQL 函数都会返回的整型变量。如果函数执行成功，则返回的 ResultCode 是零 (0)；如果失败，则返回负整数。

- *刻度*

  是十进制值的小数位数。仅当适用于要设置为空的参数时，才需要使用此值。

- *StatementID*

  使用高级功能语句时，SQL 返回供内部使用的 StatementID。

- *SQLStatement*

  实际的语句，例如：

```
ResultCode=SQLSetStatement(ConnectionID, "Select LotNo, LotName from LotInfo");
```

- *TableName*

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

TableName 参数包含要在数据库中访问或创建的表的名称。

- *TemplateName*

  TemplateName 参数是 SQL.DEF 文件中定义表的模板的名称。

- *WhereExpr*

  为表中的任一行定义的可以为真或为假的条件。函数仅提取表中条件为真的那些行。表达式必须采用以下格式：

  ColumnName comparison_operator expression

注意：如果列的数据类型是字符，则表达式必须使用英文单引号括起来。

下例选择 Name 列包含 EmployeeID 值的所有行：

```
Name='EmployeeID'
```

下例选择包含部件号 100 到 199 的所有行：

```
partno>=100 and partno<200
```

下例选择 temperature 列包含的值大于 350 的所有记录：

```
temperature>350
```

连接与断开数据库

通过在脚本中使用 SQLConnect() 与 SQLDisconnect() 函数，可以连接到 SQL 数据库，以及断开与 SQL 数据库的连接。

SQLConnect() 函数

您可以在 InTouch QuickScript 中使用 **SQLConnect()** 函数来连接到 ConnectString 参数指定的数据库。

SQLConnect() 返回一个值给 ConnectionID 参数，后续的所有 SQL 函数都将它用作参数。在脚本中使用 **SQLConnect** 函数之前，必须先在应用程序文件夹中定义“绑定列表”。

类别

SQL

语法

```
[ResultCode=]SQLConnect(ConnectionID, "ConnectString");
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

ConnectString
确定 SQLConnect() 函数中使用的数据库以及任何附加登录信息的字符串。

附注

应用程序文件夹中必须存在“绑定列表”（SQL.DEF 文件）。否则此函数无法正常工作。

如果在 win.ini 文件的 [InTouch] 部分定义了 SQLTrace=1，则每次成功执行 **SQLConnect** 时，都会将 ADO、供应器以及数据库系统的版本信息记录到 Log Viewer。

示例

下面的语句连接到 IBM OS/2 Database Manager 与名称为 SAMPLE 的数据库：

```
[ResultCode=]SQLConnect(ConnectionID,"DSN=OS2DM;
DB=SAMPLE");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

此函数返回一个值给 ConnectionID 变量，所有后续的“SQL 函数”都要将此变量用作参数。

```
"DSN=data source name[;attribute=value
[;attribute=value]...]"
```

SQLConnectEx() 函数

在使用类型为“用户名和密码”的凭据进行 SQL 身份验证时，可以使用 InTouch QuickScript 中的 **SQLConnectEx()** 函数连接到由 ConnectString 参数指定的数据库。

SQLConnectEx() 返回一个值给 ConnectionID 参数，后续的所有 SQL 函数都将它用作参数。在脚本中使用 **SQLConnectEx** 函数之前，必须先在应用程序文件夹中定义“绑定列表”。

类别

SQL

语法

```
[ResultCode=]SQLConnectEx(ConnectionID, "ConnectString", "Credential Name");
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnectEx() 函数指定给每个数据库连接的编号 (ID)。

ConnectString
确定 SQLConnectEx() 函数中使用的数据库以及任何附加登录信息的字符串。

凭证名称
凭据管理器中存储的凭据的名称。对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。

附注

应用程序文件夹中必须存在“绑定列表”（SQL.DEF 文件）。否则此函数无法正常工作。

如果在 win.ini 文件的 [InTouch] 部分定义了 SQLTrace=1，则每次成功执行 **SQLConnectEx** 时，都会将 ADO、供应器以及数据库系统的版本信息记录到 Log Viewer。

示例

下面的语句连接到 IBM OS/2 Database Manager 与名称为 SAMPLE 的数据库：

```
[ResultCode=]SQLConnectEx(ConnectionID,"DSN=OS2DM;
DB=SAMPLE");
```

此函数返回一个值给 ConnectionID 变量，所有后续的“SQL 函数”都要将此变量用作参数。

```
"DSN=data source name[;attribute=value
[;attribute=value]...]"
```

SQLDisconnect() 函数

SQLDisconnect() 函数断开与数据库的连接，并清理为 **SQLPrepareStatement()** 与 **SQLInsertPrepare()** 函数获取且尚未释放的所有资源。

类别

SQL

语法

```
[ResultCode=]SQLDisconnect(ConnectionID);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ConnectionId
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

另请参阅

SQLConnect()

新建表

通过在 InTouch QuickScript 中使用 SQLCreateTable() 函数，可以使用指定的“表模板”中的参数在数据库中创建表。

SQLCreateTable() 函数

通过在 InTouch QuickScript 中使用 SQLCreateTable() 函数，可以使用指定的“表模板”中的参数在数据库中创建表。“表模板”在 SQL.DEF 文件中定义，该文件包含数据库表的结构。

类别

SQL

语法

```
[ResultCode=]SQLCreateTable(ConnectionID, TableName,TemplateName);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

TableName
要创建的数据库表的名称。

TemplateName
要使用的模板定义的名称。

示例

下面的 **SQLCreateTable()** 函数示例使用 OutputVal 模板中定义的列名和数据类型来创建一个名为 BATCH1 的表：

```
ResultCode=SQLCreateTable(ConnectionID,"BATCH1",
"OutputVal");
```

另请参阅

SQLConnect()

删除表

您可以在 InTouch QuickScript 中使用 **SQLDropTable()** 函数从数据库中删除某个表。

SQLDropTable() 函数

您可以在 InTouch QuickScript 中使用 **SQLDropTable()** 函数从数据库中删除某个表。包含 **SQLDropTable()** 函数的 QuickScript 执行完毕之后，将不再能够识别该表，它也不会对任何 SQL 语句作出响应。

类别

SQL

语法

```
[ResultCode=]SQLDropTable(ConnectionID, TableName);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

TableName
要从数据库中删除的表的名称。

示例

下面的 **SQLDropTable()** 函数示例从数据库中删除 BATCH1 表：

```
ResultCode=SQLDropTable(ConnectionID,"BATCH1");
```

另请参阅

SQLConnect()

从表中检索数据

您可以在脚本中使用一组 SQL 函数从数据库中检索数据，并将值写入 InTouch 标记。

- **SQLSelect()** 函数从表中检索信息，并将这些信息以记录的形式放入内存中创建的临时“结果表”。
- **SQLGetRecord()** 函数从当前的选择缓冲区中检索 RecordNumber 所指定的记录。
- **SQLNumRows()** 函数返回表中满足前面的 **SQLSelect()** 函数所指定的准则的行数。
- SQLFirst() 函数检索由上一个 **SQLSelect()** 函数创建的“结果表”的第一条记录。
- SQLNext() 函数检索由上一个 **SQLSelect()** 函数创建的“结果表”的下一条记录。
- SQLPrev() 函数检索逻辑表中上一行的数据，获取该行的值并将其放入 InTouch 标记。
- SQLLast() 函数检索逻辑表的最后一行，获取该行的值并将其放入 InTouch 标记。
- SQLEnd() 函数将用于存储同 ConnectionId 关联的“结果表”内容的内存释放出来。

SQLFirst()、**SQLPrev()**、**SQLNext()**、**SQLLast()** 以及 **SQLGetRecord()** 函数从逻辑表中指定的行检索数据，并将它保存为 InTouch 标记值。如果某个字段为 NULL，则根据标记是模拟型还是消息型，将关联的 InTouch 标记的值设置为零或是零长字符串。

如果数据库中字符串的长度超过 131 个字符，则只有头 131 个字符从数据库中复制到关联的 InTouch 消息标记。

SQLSelect() 函数

SQLSelect() 函数从表中检索记录。包含 SQLSelect() 函数的脚本处理完毕之后<1>, </1>检索到的记录放入内存中的临时<2>“</2>结果表<3>”。</3> 这些记录可以使用 SQLFirst()、SQLLast()、SQLNext() 以及 SQLPrev() 函数进行浏览。

重要事项：在包含 SQLSelect() 函数的脚本结束之后，务必调用 SQLEnd() 函数以释放结果表占用的内存。

类别

SQL

语法

```
[ResultCode=]SQLSelect(ConnectionID,TableName, BindList,WhereExpr,OrderByExpression);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

TableName
要访问的数据库表的名称。

BindList
定义所使用的 InTouch 标记以及这些标记关联的数据库列。

WhereExpr
为表中的任一行定义的可以为真或为假的条件。SQLSelect() 函数仅从 *WhereExpr* 条件为真的那些行中提取数据。表达式必须采用以下格式：

ColumnName comparison_operator expression。

注意：如果使用字符表达式进行比较，则表达式必须使用英文单引号括起来。

下例选择 name 列包含 EmployeeID 值的所有行：

name='EmployeeID'
下例选择包含部件号 100 到 199 的所有行：

```
partno>=100 and partno<200
```

下例选择 temperature 列包含的值大于 350 的所有行：

```
temperature>350
```

WhereExpr - 内存消息标记

```
OrderByExpr - 内存消息标记
Speed_Input - 内存实型 - 用户输入模拟
Serial_Input - 内存消息 - 用户输入字符串
```

模拟示例

```
WhereExpr = "Speed = " + text
(Speed_Input,"#.##");
```

Speed_Input 是数字，因此必须转换为文本才能与 WhereExpr 字符串串联。
字符串示例

```
WhereExpr = "Ser_No = ‘" +
Serial_input + "’";
```

Serial_Input 是字符串，因此必须用英文单引号括住其值，例如：WhereExpr = "Ser_No='125gh'" ；
使用 like 语句的字符串示例

```
WhereExpr = "Ser_No like ‘-’"
```

使用 Like 比较运算符时，可以将 % 字符用作通配符。
使用布尔 AND 运算符的字符串与模拟示例

```
WhereExpr = "Ser_No = ‘" + Serial_input + "’" + " and " + "Speed = " +
text(Speed_Input,"#.##");OrderByExpr = "";
```

如果顺序无关紧要，请使用如上所示的空字符串。
使用 WhereExpr 标记的 SQLSelect

```
ResultCode = SQLSelect(Connect_Id,TableName,
BindList,
WhereExpr,OrderByExpr);
Error_msg = SQLErrorMsg( ResultCode );
```

内置 WhereExpr 的 SQLSelect 函数

```
ResultCode = SQLSelect(Connect_Id,TableName,
BindList,
"Ser_No = ‘" + Serial_input + "’", OrderByExpr);
Error_msg = SQLErrorMsg( ResultCode );
```

OrderByExpr
定义表列中数据的排序方向。只有列名可用于排序，并且表达式必须采用以下格式：
ColumnName [ASC|DESC]

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

下例按照 manager 列中数据的升序对表进行排序：

```
"manager ASC"
```

您也可以按多个列进行排序，此时表达式采用以下格式：
ColumnName [ASC|DESC],
ColumnName [ASC|DESC]
下例按 temperature 列的升序与 time 列的降序给所选的表排序：

```
"temperature ASC,time DESC"
```

示例

下面的语句使用绑定列表 List1 从 BATCH 表中选择列名类型包含 cookie 这个值的记录。它将按 amount 列的升序与 sugar 列的降序来显示信息：

```
ResultCode=SQLSelect(ConnectionID,"BATCH", "List1","type='cookie'","amount ASC,sugar
DESC");
```

下面的语句未指定 WhereExpr 与 OrderByExpr 的值，因此会选择数据库中的所有数据：

```
ResultCode=SQLSelect(ConnectionID,"BATCH", "List1", "","");
```

另请参阅

SQLFirst()、SQLConnect()、SQLLast()、SQLNext()、SQLPrev()、SQLEnd()

SQLGetRecord() 函数

SQLGetRecord() 函数从当前的选择缓冲区中检索 RecordNumber 参数所指定的记录。

类别

SQL

语法

```
[ResultCode=]SQLGetRecord(ConnectionID, RecordNumber);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

RecordNumber
要检索的实际记录号。

示例

```
ResultCode=SQLGetRecord(ConnectionID,3);
```

另请参阅

SQLConnect()

SQLNumRows() 函数

SQLNumRows() 指出有多少行满足在上一个 SQLSelect() 函数中指定的准则。例如，如果使用 WhereExpr 参数来选择所有列名为 AGE（其中 AGE 等于 45）的行，则返回的行数可能是 40 或 4000。这可以确定接下来要处理的函数。

类别

SQL

语法

```
SQLNumRows(ConnectionID);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

示例

下面的语句将选择的行数返回给 NumRows 整型标记：

```
NumRows=SQLNumRows（ConnectionID）;
```

另请参阅

SQLConnect()

SQLFirst() 函数

SQLFirst() 函数选择上一个 SQLSelect() 函数创建的“结果表”中的第一条记录。

类别

SQL

语法

```
[ResultCode=]SQLFirst(ConnectionID);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

另请参阅

SQLConnect()、SQLSelect()

SQLNext() 函数

SQLNext() 函数选择上一个 SQLSelect() 函数创建的<1>“</1>结果表<2>”</2>序列中的下一条记录<3>。</3>在脚本中运行 SQLNext() 函数之前，必须先处理 SQLSelect() 函数。

类别

SQL

语法

```
[ResultCode=]SQLNext(ConnectionID);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

示例

```
ResultCode=SQLNext(ConnectionID);
```

另请参阅

SQLConnect()、SQLSelect()

SQLPrev() 函数

SQLPrev() 函数选择上一个 SQLSelect() 函数创建的“结果表”中的上一条记录。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

SQL

语法

```
[ResultCode=]SQLPrev(ConnectionID);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

附注

使用此命令之前，必须先处理 SQLSelect() 函数。

示例

```
ResultCode=SQLPrev(ConnectionID);
```

另请参阅

SQLConnect()、SQLSelect()

SQLLast() 函数

SQLLast() 函数选择上一个 SQLSelect() 函数创建的“结果表”中的最后一条记录。

类别

SQL

语法

```
[ResultCode=]SQLLast(ConnectionID);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

示例

```
ResultCode=SQLNext(ConnectionID);
```

另请参阅

SQLConnect()、SQLSelect()

SQLEnd() 函数

SQLEnd() 函数在 SQLSelect() 函数之后运行，以释放用于存储“结果表”内容的内存。

类别

SQL

语法

```
[ResultCode=]SQLEnd(ConnectionID);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

SQLConnect()、SQLSelect()

向表中写入新记录

通过使用 **SQLInsert()** 函数，可以将新记录插入到数据库中。**SQLInsert()** 函数使用 InTouch 标记的当前值向表中插入一条记录。**SQLInsert()** 函数通过一个步骤来完成预备、插入和结束语句的操作。

如果与 InTouch 消息标记关联的字符串长度超过为表中相应得文本字段定义的大小，则按照为该字段定义的大小使用消息标记中的字符数。

注意：InTouch 标记不得为 NULL。如果“绑定列表”包含相应的字段，则无法使用这些函数更新数据库中的 NULL 值，或是将 NULL 值插入数据库。通过在不包含某个字段的 INSERT 语句（此语句应该已经定义为允许使用 NULL 值）上使用 **SQLExecute**，可以将 NULL 值插入该字段。

SQL Access 提供三个其它函数，分别用于预备、插入以及记录插入后的清理。同时使用这些函数时，可以编写包含单个预备和结束语句的脚本，然后根据需要添加任意数量的记录插入语句。如果使用单独的函数而不是 **SQLInsert()** 函数来插入数据，则可以减少计算机上的资源占用量。

SQLInsert() 函数

SQLInsert() 函使使用所提供的<1>“</1>绑定列表<2>”</2>中标记的值<3>，</3>向引用的表中插入一条新记录<4>。</4> BindList 参数定义要使用哪些 InTouch 标记，以及同这些标记关联的是哪些数据库列。

使用 SQLInsert() 函数可以完成预备、插入和结束语句的操作。

类别

SQL

语法

```
[ResultCode=]SQLInsert(ConnectionID, TableName, BindList);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

TableName
要访问的数据库表的名称。

BindList
定义要使用哪些 InTouch 标记，以及同这些标记关联的是哪些数据库列。

示例

下面的语句使用 List1 中指定的标记值向 ORG 表中插入一条新记录：

```
ResultCode=SQLInsert(ConnectionID,"ORG","List1");
```

SQLInsertPrepare() 函数

SQLInsertPrepare() 函数在每次运行时创建并预备一条 Insert 语句。此时不处理该 Insert 语句。StatementID 参数是一个整型标记，在处理该语句之后，它将包含一个值。

类别

SQL

语法

```
[ResultCode=]SQLInsertPrepare (ConnectionID, TableName, BindList, StatementID);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

TableName
要访问的数据库表的名称。

BindList
定义要使用哪些 InTouch 标记，以及同这些标记关联的是哪些数据库列。

StatementID
使用 SQLPrepareStatement() 函数时 SQL 返回的整数值。

另请参阅

SQLConnect()、SQLPrepareStatement()

SQLInsertExecute() 函数

SQLInsertExecute() 函数运行先前预备好的 Insert 语句，此语句由 SQLInsertPrepare() 函数指定。

SQLInsertExecute() 函数使用 InTouch 标记的当前值向上一个 SQLInsertPrepare() 函数所确定的表中插入一行。如果 BindList 参数包含 MS SQL Server 表的 Identity 关键字段，则必须在运行 SQLInsertExecute() 之前设置 IDENTITY_INSERT 选项。

StatementID 参数包含一个整数值，该值是在运行脚本中的上一个 SQLInsertPrepare() 函数时由 SQL 返回的值。

类别

SQL

语法

```
[ResultCode=]SQLInsertExecute(ConnectionID, BindList, StatementID);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

BindList
定义要使用哪些 InTouch 标记，以及同这些标记关联的是哪些数据库列。

StatementID
使用 SQLPrepareStatement() 函数时 SQL 返回的整数值。

另请参阅

SQLConnect()、SQLPrepareStatement()

SQLInsertEnd() 函数

SQLInsertEnd 函数清理与 **SQLInsertPrepare** 所创建的 StatementID 函数关联的资源。

类别

SQL

语法

```
[ResultCode=]SQLInsertEnd(ConnectionID, StatementID);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

StatementID
使用 SQLPrepareStatement() 函数时 SQL 返回的整数值。

示例

下例显示了在脚本中应该如何指定多个插入函数。

```
ResultCode = SQLSetStatement(ConnectionId, "SET IDENTITY_INSERT Products ON");
ResultCode = SQLExecute(ConnectionId, "", 0);
ResultCode = SQLInsertPrepare(ConnectionId, TableName, Bindlist, StatementID);
ResultCode = SQLInsertExecute(ConnectionId, Bindlist, StatementID);
ResultCode = SQLInsertEnd(ConnectionId, StatementID);
```

另请参阅

SQLConnect()、SQLPrepareStatement()

更新表中现有的记录

SQL Access 提供两个函数来使用 InTouch 标记的值更新表记录：

- SQLUpdate()
- SQLUpdateCurrent()

SQLUpdate() 函数

SQLUpdate() 函数使用 InTouch 标记的当前值来更新表中与 WhereExpr 参数设置的条件相匹配的所有行。

类别

SQL

语法

```
[ResultCode=]SQLUpdate(ConnectionID, TableName, BindList, WhereExpr);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

TableName
要访问的数据库表的名称。

BindList
定义要使用哪些 InTouch 标记，以及同这些标记关联的是哪些数据库列。

WhereExpr
为表中的任一行定义的可以为真或为假的条件。此函数仅更新表中条件为真的那些行。表达式必须采用以下格式：
ColumnName comparison_operator expression。

注意：如果列的数据类型是字符，则表达式必须用英文单引号括起来。

示例

下例选择 name 列包含 EmployeeID 值的所有行：

```
name='EmployeeID'
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

下例选择包含部件号 100 到 199 的所有行：

```
partno>=100 and partno<200
```

下例选择 temperature 列包含的值大于 350 的所有行：

```
temperature>350
```

下面的语句将 BATCH 表中批号为 65 的所有记录更新为 BindList "List1" 中所指定的标记的当前值：

```
ResultCode=SQLUpdate(ConnectionID,"BATCH", "List1","lotno=65");
```

注意：务必确保所有记录都是唯一的。如果表中存在完全相同的记录，则所有相似的记录都会更新。

另请参阅

SQLConnect()

SQLUpdateCurrent() 函数

SQLUpdateCurrent() 函数使用映射到表字段的 InTouch 标记来更新逻辑表中的当前行，映射由 SQLSelect() 或 SQLExecute() 函数语句中指定的“绑定列表”来完成。如果有其它行与当前行完全相同，则所有这些行都会更新。

一次可以更新最多 54 条完全相同的记录。如果有太多完全相同的行要在 **SQL Access** 中更新，SQLUpdateCurrent() 函数会返回错误。错误消息类似于："Microsoft Cursor Engine: Key column information is insufficient or incorrect. Too many rows were affected by update"。

要避免此错误，请在表中创建一个唯一的键字段，使得每行都是唯一的。强烈建议让 SQL Access 使用的所有的表都有唯一的键。对于没有键的表，建议将类型为 AutoNumber (Access) 的字段，或是将用作行标识 (SQL Server) 的整型字段用作主键，以便 SQLUpdateCurrent() 函数一次只更新一行。这个主键字段不必包含在<1>“</1>绑定列表<2>”</2>中<3>。</3>

类别

SQL

语法

```
[ResultCode=]SQLUpdateCurrent(ConnectionID);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

示例

```
ResultCode=SQLUpdateCurrent(ConnectionID);
```

另请参阅

SQLConnect()

从表中删除记录

您可以使用两个 SQL 函数从数据库表中删除记录。

SQL Access 提供两个函数来删除表记录：

- SQLClearTable() 从表中删除记录
- SQLDelete() 从表中删除与指定的条件匹配的记录。

SQLClearTable() 函数

SQLClearTable() 函数从表中删除所有的记录。它不从数据库中删除表。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

SQL

语法

```
[ResultCode=]SQLClearTable(ConnectionID, "TableName");
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

TableName
要清除其所有记录的表的名称。

示例

在下例中 SQLClearTable() 函数从 BATCH1 表中清除所有的记录。

```
ResultCode=SQLClearTable(ConnectionID,"BATCH1");
```

另请参阅

SQLConnect()、SQLClearStatement()

SQLDelete() 函数

SQLDelete() 函数从表中删除与 *WhereExpr* 参数指定的条件匹配的所有记录。

类别

SQL

语法

```
[ResultCode=]SQLDelete(ConnectionID, TableName, WhereExpr);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

TableName
要从中清除记录的表的名称。这些记录满足 *WhereExpr* 参数指定的条件。

WhereExpr
为表中的任一行定义的可以为真或为假的条件。**SQLDelete()** 函数仅会删除 *WhereExpr* 条件为 True 的记录行。表达式必须采用以下格式：
ColumnName comparison_operator 表达式

注意：SQLDelete() 函数不得包含空的 *WhereExpr* 参数。

示例

下面的语句删除 BATCH1 表中批号等于 65 的所有记录：

```
ResultCode=SQLDelete(ConnectionID,"BATCH1", "lotno=65");
```

注意：如果列的数据类型为字符，则必须使用英文单引号将表达式括起来，如 "MachineID='AG_LX7_2'"。

另请参阅

SQLConnect()

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

执行参数化语句

使用 SQLSetStatement() 与 SQLAppendStatement() 函数可以建立动态查询。SQLSetStatement() 函数开始一个新的 SQL 语句。它可以是任何有效的 SQL 语句，包括存储过程的名称。SQLAppendStatement() 函数使用字符串的内容来追加 SQL 语句。

SQLSetStatement() 函数

SQLSetStatement() 函数在已建立的连接 ConnectionID 上，使用 SQLStatement 的内容创建一个 SQL 语句缓冲区。每个 ConnectionID 可以有一个 SQL 语句缓冲区。错误在函数的返回值中返回。

类别

SQL

语法

```
[ResultCode=]SQLSetStatement(ConnectionID, SQLStatement);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

SQLStatement
实际的 SQL 语句，请参阅以下示例。

示例

```
ResultCode=SQLSetStatement(ConnectionID,"Select LotNo, LotName from LotInfo");
```

在下例中，StatementID 设置为零，因此在运行语句之前，语句不必调用 SQLPrepare(Connect_Id, StatementID)。由于 SQLPrepare 没有创建 StatementID 来正确结束这个 select 语句，因此请使用 SQLEnd() 函数而不是 SQLClearStatement() 函数。

```
SQLSetStatement( Connect_Id, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLExecute(Connect_Id,0);
```

在下例中，StatementID 由 SQLPrepareStatement() 函数创建，并用在 SQLExecute() 函数中。要结束这个 SELECT 语句，请使用 SQLClearStatement() 函数释放资源与句柄。

```
SQLSetStatement( Connect_Id, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLPrepareStatement(Connect_Id,StatementID);
SQLExecute(Connect_Id,StatementID);
SQLSetStatement( Connect_Id, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLPrepareStatement(Connect_Id,StatementID);
SQLExecute(Connect_Id,StatementID);
```

另请参阅

SQLConnect()

SQLAppendStatement() 函数

SQLAppendStatement() 函数使用字符串的内容来追加 SQL 语句。返回值指出在函数调用过程中是否发生了错误。

InTouch 标记可以支持的最大字符串长度是 131 个字符。通常使用 SQLAppendStatement() 函数将附加的字符串串联到语句。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

SQL

语法

```
[ResultCode=]SQLAppendStatement(ConnectionID, "SQLStatement");
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

SQLStatement
要追加的实际语句。

示例

```
ResultCode=SQLAppendStatement(ConnectionID, "where tablename.columnname=TR-773-01");
```

另请参阅

SQLConnect()、SQLClearStatement()

创建语句或从文件中加载现有的语句

您可以使用其它第三方数据库工具来创建查询，然后使用 SQL Access 运行该查询。首先，您必须从第三方数据库工具创建的 .SQL 查询文件中加载 SQL 语句。

```
ResultCode = SQLLoadStatement (ConnectionID, "c:\myappdir\lotquery.sql");
```

您使用 SQLoadStatement() 函数加载 SQL 查询。现在便可以运行该语句了。

SQLLoadStatement() 函数

SQLLoadStatement() 函数从文件中读取 SQL 语句。

每个文件只能包含一条语句。不过，如果尚未调用 SQLPrepareStatement() 或 SQLExecute() 函数，则可以使用 SQLAppendStatement() 函数给语句追加内容。

类别

SQL

语法

```
[ResultCode=]SQLLoadStatement(ConnectionID, FileName);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

FileName
包含 SQL 语句的文件的名称。

附注

加载语句并获取语句句柄之后，使用 QLPrepareStatement() 函数准备执行语句。

示例

SQL.txt 文件包含以下 SQL 语句：

```
Select ColumnName from TableName where ColumnName>100;
```

SQLLoadStatement() 函数从文件中加载该语句。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
ResultCode=SQLLoadStatement(ConnectionID,
"C:\SQL.txt")
```

另请参阅

SQLConnect()、SQLAppendStatement()、SQLExecute()、SQLPrepareStatement

预备语句

使用以下函数，您可以创建任何需要的参数化语句，然后逐个动态填充参数。例如，您可以将常规语句保存到文件、使用 SQLLoadStatement() 函数加载它、使用 QLPrepareStatement() 函数准备它以获取语句标识，然后使用以下函数填充语句中的各个参数：

- **SQLPrepareStatement()**
- **SQLSetParamChar()**
- **SQLSetParamDate()**
- **SQLSetParamDateTime()**
- **SQLSetParamDecimal()**
- **SQLSetParamFloat()**
- **SQLSetParamInt()**
- **SQLSetParamLong()**
- **SQLSetParamNull()**
- **SQLSetParamTime()**
- **SQLClearParam()**
- **SQLClearStatement()**

要在 SQL 语句上执行参数替换，请在 SQL 语句中要指定后续参数的位置放置一个 "?" 号。此时会预备该语句，在语句中设置参数，然后再运行该语句。

SQLPrepareStatement() 函数

SQLPrepareStatement() 函数预备要运行的 SQL 语句。它不运行该语句，而只是激活该语句以便设置参数值。

类别

SQL

语法

```
SQLPrepareStatement(ConnectionId, StatementID)
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

附注

预备缺省语句，并返回 StatementID（1、2、3 等）。对于需要使用 SQLSetParam{Type} 函数来设置参数的语句，这个预备操作非常有用。

设置语句参数

“SQL 访问管理器”提供一组函数来修改给 SQL 语句中包含的某个参数指定的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

SQLSetParamChar() 函数

在脚本中，SQLSetParmChar() 函数可以用于将指定参数的值设置为指定的字符串。此函数可以在执行之前调用多次，导致参数值设置为发送的所有值的串联结果。长度 0（零）会被忽略。

类别

SQL

语法

```
SQLSetParamChar(StatementID, ParameterNumber, ParameterValue, MaxLength);
```

参数

StatementID
使用 SQLPrepareStatement() 函数时 SQL 返回的整数值。

ParameterNumber
语句中的参数编号。

ParameterValue
要设置为参数值的值。

最大长度
与此参数关联的列的最大宽度。此设置确定参数是可变字符类型还是长可变字符类型。如果 *MaxLength* 小于或等于数据库所允许的最大字符串长度，则此参数是可变字符类型。如果大于，则是长可变字符类型。

另请参阅

SQLPrepareStatement()

SQLSetParamDate() 函数

SQLSetParamDate() 函数将某个参数的值设置为指定的日期。

类别

SQL

语法

```
SQLSetParamDate(StatementID, ParameterNumber, "Value");
```

参数

StatementID
确定查询中的 SQL 语句的整数值。

ParameterNumber
整数值，用于确定 *StatementID* 参数所确定的 SQL 语句中的参数。

值
指定给参数的日期（用英文双引号括起来的文字形式），或是值为日期的标记的名称。指定给日期的时间是 0:00:00。

示例

本例将第三个语句的第二个参数设置为与 NewDate 标记关联的日期。

```
SQLSetParamDate(3, 2, NewDate);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

SQLPrepareStatement()

SQLSetParamDateTime() 函数

SQLSetParamDateTime() 函数将参数的值设置为指定的日期与时间。

类别

SQL

语法

```
SQLSetParamDateTime(StatementID, ParameterNumber, Value, Precision);
```

参数

StatementID
确定查询中的 SQL 语句的整数值。

ParameterNumber
整数值，用于确定 *StatementID* 参数所确定的 SQL 语句中的参数。

值
指定给 ParameterNumber 参数所确定的参数的日期与时间。

精度
整数，用于指定日期时间值的字符数，此日期时间值指定为参数的值。

另请参阅

SQLPrepareStatement()

SQLSetParamDecimal() 函数

SQLSetParamDecimal() 函数将参数的值设置为十进制数。

类别

SQL

语法

```
SQLSetParamDecimal(StatementID, ParameterNumber, Value, Precision, Scale);
```

参数

StatementID
确定查询中的 SQL 语句的整数值。

ParameterNumber
整数值，用于确定 *StatementID* 参数所确定的 SQL 语句中的参数。

值
Value 可以是代表十进制数 (123.456) 的字符串或 InTouch 消息标记，或是 InTouch 内存实型标记。
建议使用消息标记而不要使用实型标记，以确保参数的精度。不过，如果 Value 必须是浮点数（例如，从 DAServer 收到的实数值），则函数仍然起作用。但是由于浮点表示法的限制，可能无法保证很高的精度。

精度
指定数字中总计位数的整数。

刻度
指定小数点后面的位数的整数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

本例将第三条 SQL 语句的第二个参数设置为 123.456。精度是六位，范围到小数点右侧三位。

```
SQLSetParamFloat(3, 2, 123.456, 6, 3);
```

另请参阅

SQLPrepareStatement()

SQLSetParamFloat() 函数

SQLSetParamFloat() 函数将参数的值设置为 64 位有符号的浮点值。

类别

SQL

语法

```
SQLSetParamFloat(StatementID, ParameterNumber, Value);
```

参数

StatementID
确定查询中的 SQL 语句的整数值。

ParameterNumber
整数值，用于确定 *StatementID* 参数所确定的 SQL 语句中的参数。

值
64 位带符号浮点数，要指定为所指定的参数的值。

示例

本例将第三条 SQL 语句的第二个参数设置为 -5。

```
SQLSetParamFloat(3, 2, -5);
```

另请参阅

SQLPrepareStatement()

SQLSetParamInt() 函数

SQLSetParamInt() 函数将参数的值设置为 16 位有符号的整数。

类别

SQL

语法

```
SQLSetParamInt(StatementID, ParameterNumber, Value);
```

参数

StatementID
确定查询中的 SQL 语句的整数值。

ParameterNumber
整数值，用于确定 *StatementID* 参数所确定的 SQL 语句中的参数。

值
16 位带符号整数，要指定为所指定的参数的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

本例将第三条 SQL 语句的第二个参数设置为 -5。

```
SQLSetParamInt(3, 2, -5);
```

另请参阅

SQLPrepareStatement()

SQLSetParamLong() 函数

SQLSetParamLong() 函数将参数的值设置为 32 位有符号的模拟数字。

类别

SQL

语法

```
SQLSetParamLong(StatementID, ParameterNumber, Value);
```

参数

StatementID
确定查询中的 SQL 语句的整数值。

ParameterNumber
整数值，用于确定 *StatementID* 参数所确定的 SQL 语句中的参数。

值
32 位带符号模拟数字，要指定为所指定的参数的值。

示例

本例将第一条语句的第三个参数设置为 2.1e9。

```
SQLSetParamLong(1, 3, 2.1e9);
```

另请参阅

SQLPrepareStatement()

SQLSetParamNull() 函数

SQLSetParamNull() 函数将 SQL 语句中的指定参数值设置为 NULL。

类别

SQL

语法

```
SQLSetParamNull(StatementID, ParameterNumber, ParameterType, Precision, Scale)
```

参数

StatementID
确定查询中的 SQL 语句的整数值。

ParameterNumber
整数值，用于确定 *StatementID* 参数所确定的 SQL 语句中的参数。

ParameterType
整数值，用于指定与 *ParameterNumber* 参数所指定的参数关联的数据的类型。*ParameterType* 参数可以赋以下这些值：
0：字符串

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1：日期/时间
2：整数
3：浮点数
4：十进制数

精度
与参数的数据类型关联的数据的精度。

刻度
十进制值的小数位数。仅当适用于要设置为空的参数时，才需要使用此值。

附注

与 NULL 值的比较由 SQL Server 中的 ANSI_NULLS 选项控制。在 SQL Server 7.0 中，此选项在创建对象（而不是执行查询）时解析。在 SQL Server 7.0 中创建存储过程时，此选项缺省条件下为 ON，因此 "WHERE MyField = NULL" 之类的子句总是返回 NULL (FALSE)，并且使用此子句的 SELECT 语句不会返回任何行。

要让 = 或 <> 比较操作返回 TRUE 或 FALSE，在创建存储过程时，必须将该选项设置为 OFF。如果 ANSI_NULLS 没有设置为 OFF，则 SQLSetParamNull() 无法按预期执行。在这种情况下，与 NULL 值的比较操作应该使用以下语法："WHERE MyField IS NULL" 或 "WHERE MyField IS NOT NULL"。

示例

此事务集返回 Products 表中 ProductName 不为 NULL 的所有行。

```
SET ANSI_NULLS OFF
GO
CREATE PROCEDURE sp_TestNotNull @ProductParam varchar(255)
AS SELECT * FROM Products WHERE ProductName <> @ProductParam
GO
SET ANSI_NULLS ON
GO
```

InTouch 可以运行以下 SQL Access 脚本。

```
ResultCode = SQLSetStatement(ConnectionId, "sp_TestNotNull");
ResultCode = SQLPrepareStatement(ConnectionId, StatementID);
ResultCode = SQLSetParamNull(StatementID, 1, 0, 0, 0);
ResultCode = SQLExecute(ConnectionId, BindList, StatementID);
ResultCode = SQLFirst(ConnectionId);
ResultCode = SQLClearStatement(ConnectionId, StatementID);
```

另请参阅

SQLPrepareStatement()

SQLSetParamTime() 函数

SQLSetParamTime() 函数将指定的时间参数值设置为指定的字符串。

类别

SQL

语法

```
SQLSetParamTime(StatementID, ParameterNumber, Value)
```

参数

StatementID
确定查询中的 SQL 语句的整数值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ParameterNumber
StatementID 参数所确定的 SQL 语句中的实际参数编号。

值
要设置的实际值。将 ParameterNumber 参数所指定的参数设置为时间值。运行该函数的计算机的当前日期同指定的时间包含在一起。

示例

本例将第四条 SQL 语句的第二个参数设置为 10:00 AM。

```
ResultCode=SQLSetParamTime(1, 3, "10:00:00 AM");
```

另请参阅

SQLPrepareStatement()

清除语句参数

SQLClearParam() 函数清除所指定的参数的值。

SQLClearParam() 函数

SQLClearParam() 函数清除所指定的参数的值。在调用 **SQLExecute()** 函数以运行查询之前，必须再次调用 **SQLSetParam()***xxx***()** 函数中的一个以重新加载参数。

类别

SQL

语法

```
[ResultCode=]SQLClearParam(StatementID,ParameterNumber);
```

参数

StatementID
SQLPrepareStatement() 函数运行时返回的整数值。

ParameterNumber
ParameterNumber 参数确定 SQL 语句中要修改的实际参数。将与 *StatementID* 关联的 ParameterNumber 的值设置为零或零长字符串，具体取决于该参数是数值还是字符串。

另请参阅

SQLPrepareStatement()、SQLExecute()

执行语句

在 InTouch 脚本中，可以使用 SQLExecute() 函数在运行时运行 SQL 查询。

SQLExecute() 函数

SQLExecute 函数在脚本中运行 SQL 查询。如果语句包含 SELECT，则 BindList 参数指定用于绑定数据库列与 InTouch 标记的“绑定列表”的名称。如果“绑定列表”为 NULL，则不进行任何关联标记的操作。

类别

SQL

语法

```
SQLExecute(ConnectionID,BindList,StatementID);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

BindList
BindList 参数可以是零长字符串。如果 *StatementID* 与返回行的查询关联，则使用 **SQLExecute()** 的结果更新逻辑表。如果指定实型“绑定列表”，则结果与 BindList 参数关联。预先知道 StatementID 不与返回行的查询关联时，零长“绑定列表”非常有用。

StatementID
使用 SQLPrepareStatement() 函数时 SQL 返回的整数值。

附注

错误在函数的返回值中返回。如果语句已经预备好，则应该传递 prepare 语句返回的语句句柄。如果语句尚未预备好，则该语句的句柄应该是零。

注意：对于尚未预备好的语句，SQLExecute() 函数只能调用一次。如果语句已经预备好，则可以调用它多次。

缺省语句与连接 ID 关联。它可以是文本型 SQL 语句（SELECT、INSERT、DELETE 或 UPDATE）、查询（带或不带参数）的名称（对于 MS Access），或存储过程（带或不带参数）的名称（对于 MS SQL Server）。

缺省语句由 **SQLLoadStatement()**、**SQLSetStatement()** 以及 **SQLAppendStatement()** 函数修改。只要指定 StatementID = 0，**SQLExecute()** 便使用缺省语句。

示例

本例从 lotquery.sql 文件中加载 SQL 语句，并将 SELECT 语句的结果放入“绑定列表”指定的 InTouch 标记。

```
ResultCode = SQLLoadStatement (ConnectionID, "c:\myappdir\lotquery.sql");
ResultCode = SQLExecute(ConnectionID, "BindList", 0);
ResultCode = SQLNext (ConnectionID);
```

对于复杂查询以及超过 131 个字符的字符串表达式, 必须使用这个 **SQLSetStatement()** 函数。字符串表达式超过 131 个字符时，请使用 **SQLAppend()** 函数。

```
SQLSetStatement(ConnectionID, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLExecute(ConnectionID, "BindList", 0);
```

在上例中，StatementID 参数设置为零；因此在执行语句之前不必调用 SQLPrepareStatement(Connection_Id, StatementID)。

由于 SQLPrepare 语句没有创建 StatementID 来正确结束此 SELECT 语句，因此请使用 SQLEnd() 函数而不是 **SQLClearStatement()** 函数。

```
SQLSetStatement(Connection_Id, "Select Speed, Ser_No from tablename where Ser_No =’" +
Serial_input + "’");
SQLPrepareStatement(Connection_Id, StatementID);
SQLExecute(Connection_Id, StatementID);
```

在上例中，StatementID 由 SQLPrepareStatement 函数调用创建，并由 SQLExecute 函数使用。要结束此 SELECT 语句，请在脚本中使用 SQLClearStatement() 函数调用，以释放资源与 StatementID。

SQLExecute() 函数支持某些存储过程。例如，假设在数据库服务器 "LotInfoProc" 上创建了一个存储过程，它包含以下 select 语句："Select LotNo, LotName from LotInfo"。

您编写 InTouch QuickScript 根据正在使用的数据库类型运行该存储过程。下例显示了运行 SQL Server 数据库的存储过程的脚本语句。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
ResultCode = SQLSetStatement (ConnectionID,"LotInfoProc");
ResultCode = SQLExecute(ConnectionID, "BindList", 0);
ResultCode = SQLNext (ConnectionID);
{Get results of Select}
```

下例显示了运行 Oracle 数据库的存储过程的脚本语句。

```
ResultCode = SQLSetStatement (ConnectionID, "{CALL LotInfoProc}");
ResultCode = SQLExecute(ConnectionID, "BindList", 0);
ResultCode = SQLNext (ConnectionID);
{Get results of Select}
```

另请参阅

SQLConnect()、SQLPrepareStatement()

释放占用的资源

SQLClearStatement 函数释放与 *StatementID* 所指定的语句关联的数据库资源。

SQLClearStatement() 函数

SQLClearStatement() 函数释放与 *StatementID* 参数指定的语句关联的数据库资源。

类别

SQL

语法

```
[ResultCode=]SQLClearStatement(ConnectionID, StatementID);
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

StatementID
使用 **SQLPrepareStatement()** 函数时 SQL 返回的整数值。

另请参阅

SQLConnect()、SQLPrepareStatement()

使用事务集

SQL Access 包含一系列可从数据库中更改、插入、更新或删除记录的事务函数。通常，这些事务在脚本中按事务集的形式分组。事务集同时提交。

SQLTransact() 函数

SQLTransact() 函数定义一组称为事务集的 SQL 语句的开始。事务集的处理方式与单个事务相同。
SQLTransact() 函数运行之后，在 **SQLCommit()** 函数成功运行之前，所有的后续操作都不会提交给数据库。

类别

SQL

语法

```
[ResultCode=]SQLTransact(ConnectionID)
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

示例

此示例事务集包含三个 insert 语句。

```
ResultCode = SQLTransact(ConnectionID);
ResultCode = SQLInsertPrepare(ConnectionID, TableName, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertEnd(ConnectionID, StatementID);
ResultCode = SQLCommit(ConnectionID);
```

另请参阅

SQLCommit()、SQLRollback()

SQLCommit() 函数

SQLCommit() 函数定义事务集的结束。**SQLTransact()** 函数运行之后，在 **SQLCommit()** 函数成功运行之前，事务集中所有的后续 SQL 语句都不会提交给数据库。

注意：编写包含 SQLCommit() 函数的 QuickScript 时，请务必小心。处理时间随着事务集中 SQL 语句数量的增加而增加。

类别

SQL

语法

```
[ResultCode=]SQLCommit(ConnectionID)
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

示例

此示例脚本包含执行三次数据库插入操作的事务集。

```
ResultCode = SQLTransact(ConnectionID);
ResultCode = SQLInsertPrepare(ConnectionID, TableName, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertEnd(ConnectionID, StatementID);
ResultCode = SQLCommit(ConnectionID);
```

另请参阅

SQLRollback()、SQLTransact()、SQLCommit()

SQLRollback() 函数

SQLRollback() 函数反转或回滚最新的事务集。事务集是在 **SQLTransact()** 与 **SQLCommit()** 函数之间发出的一组命令。

事务集的处理方式与单个事务相同。SQLTransact() 函数运行之后，所有的后续操作都不会提交给数据库。SQLCommit() 函数运行之后，才发生对数据库的查询更改。如果在 SQLCommit() 函数之前运行 SQLRollback() 函数，则它可以回滚事务集。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

SQL

语法

```
[ResultCode=]SQLRollback(ConnectionID)
```

参数

ConnectionID
内存整型标记的名称，用于存放 SQLConnect() 函数指定给每个数据库连接的编号 (ID)。

示例

本例回滚脚本中 **SQLTransact** 函数之前的数据库值。

```
ResultCode =SQLTransact(ConnectionID);
ResultCode = SQLInsertPrepare(ConnectionID, TableName, BindList, StatementID);
ResultCode = SQLInsertExecute(ConnectionID, BindList, StatementID);
ResultCode = SQLInsertEnd(ConnectionID, StatementID);
ResultCode =SQLRollback(ConnectionID);
```

另请参阅

SQLCommit()、SQLTransact()

在运行时打开 ODBC 管理器对话框

使用 **SQLManageDSN()** 函数在 InTouch 应用程序运行期间运行“Microsoft ODBC 管理器”。

SQLManageDSN() 函数

SQLManageDSN 函数在 InTouch 应用程序运行期间运行“Microsoft ODBC 管理器”设置程序。在脚本中，**SQLManageDSN()** 可用于添加、删除及修改 SQL Server 或 Access 数据库现有的数据源名。

类别

SQL

语法

```
SQLManageDSN(ConnectionId)
```

参数

ConnectionId
ConnectionId 不使用。保留它是为了与旧版 SQL Access 保持后向兼容性。因此可以给函数传递任何数字。在运行此函数以打开“ODBC 数据源管理器”之前，不需要建立数据库连接。

示例

```
SQLManageDSN(0);
```

理解 SQL 错误消息

本节介绍如何排解使用 SQL Access 函数的 InTouch 应用程序的疑难问题。第一部分介绍 SQLErrorMsg() 函数；并且包含一个表格，列出 SQL 结果码及对应的错误消息。第二部分包含一些列出具体数据库错误消息的表格。

SQLErrorMsg() 函数

所有的 SQL 函数都会返回一个可用于排解疑难的结果码。SQLErrorMsg() 函数返回与结果码关联的错误消息，并将它指定为 InTouch 消息标记的值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

SQL

语法

```
ErrorMsg = SQLErrorMsg(ResultCode);
```

参数

ResultCode
上一个 SQL 函数返回的整数值。SQLErrorMsg() 函数将 InTouch 消息标记的值设置为同结果码关联的消息。如需有关同结果码关联的错误消息的详细信息，请参阅理解 SQL 错误消息。

附注

对于此处未记录的结果码，请参阅数据库的文档。此外也可以到 Log Viewer 中浏览任何额外的错误消息。

win.ini 文件的 [InTouch] 部分中定义的 SQLTrace=1 标帜对于调试 SQL Access 脚本非常有用。

示例

本例将同“SQL 访问管理器”结果码关联的错误消息指定给 ErrorMsg 消息标记。

```
ErrorMsg = SQLErrorMsg(ResultCode)
```

另请参阅

SQLConnect()
SQL 访问管理器结果码与消息

下表列出一些常见的 SQL Access 结果码及相应的错误消息和描述：

ResultCode	错误消息	描述
0	没有发生错误	SQL 函数成功运行，没有发生错误。
-1	<来自数据库供应器的消息>	来自供应商数据库的特定错误消息。
-2	无法执行空语句	SQLExecute(ConnectionId, BindList, 0) 运行之前未调用含非空语句的 SQLSetStatement 或 SQLLoadStatement。
-4	返回值为空	从数据库中读取的整数或实数值为空。这仅仅是发送到 Log Viewer 的警告消息。
-5	已无更多的行可供获取	已经到达表中的最后一条记录。
-7	参数标识无效	使用有效的参数标识调用 SQLSetParamChar()、SQLSetParamDate()、SQLSetParamDateTime()、SQLSetParamDecimal()、SQLSetParamFloat()、SQLSetParamInt()、SQLSetParamLong()、SQLSetParamNull() 或 SQLSetParamTime() 函数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ResultCode	错误消息	描述
-8	参数列表无效	无效参数列表的示例：1, 2, 3, 5（缺 4）。
-9	NULL 参数类型无效	使用无效 *Type* 参数值调用了 SQLSetParamNull 函数。
-10	不允许更改参数的数据类型	使用不同类型的现有参数调用 SQLSetParamChar()、SQLSetParamDate()、SQLSetParamDateTime()、SQLSetParamDecimal()、SQLSetParamFloat()、SQLSetParamInt()、SQLSetParamLong()、SQLSetParamNull() 或 SQLSetParamTime() 函数。
-11	语句成功执行后不允许再添加参数。	在语句成功运行之后，针对新参数标识调用 SQLSetParamChar()、SQLSetParamDate()、SQLSetParamDateTime()、SQLSetParamDecimal()、SQLSetParamFloat()、SQLSetParamInt()、SQLSetParamLong()、SQLSetParamNull() 或 SQLSetParamTime() 函数。
-12	日期时间格式无效	遇到无效的日期时间格式，例如在执行 SQLSetParamTime()、SQLInsertExecute() 或 SQLUpdateCurrent() 时。
-13	小数格式无效	遇到无效的小数格式， 例如在执行 SQLSetParamDecimal()、SQLInsertExecute() 或 SQLUpdateCurrent() 时。
-14	货币格式无效	遇到无效的货币格式，例如在执行 SQLInsertExecute() 或 SQLUpdateCurrent() 时。
-15	此操作的语句类型无效	使用 SQLPrepareStatement() 创建的语句标识调用 SQLInsertEnd， 或是使用 SQLInsertPrepare() 创建的语句标识调用 SQLClearStatement()。
-1001	内存不足	没有足够的内存来运行此函数。
-1002	无效连接	传递给 *ConnectionId* 参数的值无效。
-1003	未找到绑定列表	指定的<1>“</1>绑定列表<2>”</2>名不存在<3>。</3>
-1004	未找到模板	指定的<1>“</1>表模板<2>”</2>名不存在<3>。</3>

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ResultCode	错误消息	描述
-1005	内部错误	发生内部错误。致电<1>“</1>技术支持<2>”。</2>
-1006	字符串为空	警告 - 从数据库中读取的字符串为空。这仅仅是发送到 Log Viewer 的警告消息。
-1007	字符串被截断	警告 - 从数据库中读取的字符串超过 131 个字符，在选择时被截断。此警告发送到 Log Viewer。
-1008	无位置子句	Delete 语句中没有 Where 子句。
-1009	连接失败	如需有关数据库连接失败的详细信息，请查看 Log Viewer。
-1010	在连接字符串的 DB= 部分上指定的数据库不存在	指定的数据库不存在。
-1011	没有选定行	在没有先运行 SQLSelect() 或 SQLExecute() 函数的情况下， 调用了 SQLNumRows()、SQLFirst()、SQLNext()、SQLLast() 或 SQLPrev() 函数。
-1013	无法找到要加载的文件	使用找不到的文件名调用了 SQLLoadStatement() 函数。

来自供应商数据库的错误消息返回的 ResultCode 是 -1。SQL Access 函数的 ResultCode 总是 -1，但消息则原封不动地从数据库供应器中复制而来。

对于使用 Oracle 数据库时发生的错误消息，请参阅 Oracle Server 文档以了解特定的错误消息与解决方案。

下表列出使用 Microsoft SQL Server 或 Access 数据库时可能发生的常见错误消息。

错误消息	解决方案
一次不能有多条语句处于活动状态	在调用 SQLSelect() 函数之后，试图运行 SQL 命令。运行 SQLEnd() 以释放 SQLSelect() 占用的系统资源，或是给第二条语句使用单独的 *ConnectionId*。
没有足够的内存来处理命令	尝试重新启动客户端工作站。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

错误消息	解决方案
对象名 [tablename] 无效	正在使用的数据库中没有该表名。尝试使用 DB=数据库名。

查看 Microsoft SQL Server 文档以了解特定的错误消息与解决方案。

保留字列表

本节列出一些关键字，这些关键字不得用于 SQL Access 的“绑定列表”、“表模板”以及 ODBC 接口。

如果将某个保留关键字用作“绑定列表”或“表模板”中的“列名”，则 Log Viewer 会记录一条错误消息。错误的类型取决于所用 ODBC 驱动程序以及该关键字出现的位置。例如，最常见的错误之一是在“绑定列表”或“表模板”中将 DATE 与 TIME 用作“列名”。为避免此错误，请使用略有不同的名称，例如“aDATE”与“aTIME”。

保留关键字定义 InTouch SQL Access 所用的“结构化查询语言”（Structured Query Language，简称 SQL）。正在使用的特定 ODBC 驱动程序也会识别出这些关键字。如果无法正确解释 SQL 命令，则“SQL 访问管理器”会生成一条错误消息，这可以在 Log Viewer 中查看到。

下面按字母顺序的列表显示 SQL Access 与 ODBC 的保留关键字：

ABSOLUTE	ADA	添加
所有	ALLOCATE	ALTER
与	ANY	ARE
AS	ASC	ASSERTION
AT	AUTHORIZATION	AVG
BEGIN	BETWEEN	BIT
BIT_LENGTH	BY	CASCADE
CASCADED	CASE	CAST
CATALOG	CHAR	CHAR_LENGTH
CHARACTER	CHARACTER_LENGTH	CHECK
CLOSE COALESCE	COBOL	COLLATE
COLLATION	COLUMN	COMMIT
CONNECT	CONNECTION	CONSTRAINT
CONSTRAINTS	CONTINUE	CONVERT
CORRESPONDING	COUNT	CREATE
CURRENT	CURRENT_DATE	CURRENT_TIME
CURRENT_TIMESTAMP	CURSOR	DATE
DAY	DEALLOCATE	DEC
DECIMAL	DECLARE	DEFERRABLE

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

DEFERRED	删除	DESC
DESCRIBE	DESCRIPTOR	DIAGNOSTICS
DICTIONARY	DISCONNECT	DISPLACEMENT
DISTINCT	DOMAIN	DOUBLE
DROP	ELSE	END
ESCAPE	EXCEPT	EXCEPTION
EXEC	EXECUTE	EXISTS
EXTERNAL	EXTRACT	FALSE
FETCH	FIRST	FLOAT
FOR FOREIGN	FORTRAN	FOUND
FROM FULL	GET	GLOBAL
GO	GOTO	GRANT
GROUP	HAVING	HOUR
IDENTITY	IGNORE	IMMEDIATE
IN	INCLUDE	INDEX
INDICATOR	INITIALLY	INNER
INPUT	INSENSITIVE	插入
INTEGER	INTERSECT	INTERVALL
INTO	IS	ISOLATION
JOIN	KEY	LANGUAGE
LAST	LEFT	LEVEL
LIKE	LOCAL	LOWER
MATCH	MAX	MIN
MINUTE	MODULE	MONTH
MUMPS	NAMES	NATIONAL
NCHAR	NEXT	NONE
非	NULL	NULLIF
NUMERIC	OCTET_LENGTH	OF
关闭	打开	ONLY
OPEN	OPRN	OPTION
或	ORDER	OUTER

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

输出	OVERLAPS	PARTIAL
PASCAL	PLI	POSITION
PRECISION	PREPARE	PRESERVE
PRIMARY	PRIOR	PRIVILEGES
PROCEDURE	PUBLIC	RESTRICT
REVOKE	RIGHT	ROLLBACK
ROWS	SCHEMA	SCROLL
SECOND	SECTION	选择
SEQUENCE	设置	SIZE
SMALLINT	SOME	SQL
SQLCA	SQLCODE	SQLERROR
SQLSTATE	SQLWARNING	SUBSTRING
SUM	系统	TABLE
TEMPORARY	THEN	TIME
TIMESTAMP	TIMEZONE_HOUR	TIMEZONE_MINU
TO	TRANSACTION	TRANSLATE
TRANSLATION	TRUE	UNION
UNIQUE	UNKNOWN	UPDATE
UPPER	USAGE	USING
值	VALUES	VARCHAR
VARING	VIEW	WHEN
WHENEVER	WHERE	WITH
WORK	YEAR	

使用 16 笔趋势向导

您可以使用 InTouch 向导来创建实时与历史趋势，这些趋势最多可显示 16 个标记的数据。16 笔趋势是可以在 InTouch 安装期间安装的辅助组件。

16 笔趋势向导的配置同其它 InTouch 图表向导的非常类似。16 笔趋势向导允许配置以下趋势属性：

- 指定给每个趋势笔的标记
- 趋势线的宽度与颜色
- 历史趋势的起始和结束日期与时间
- 实时趋势的更新速率与时间跨度

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 指定给趋势标记的最小与最大工程单位
- 主、副趋势时间刻度
- 主、副趋势值刻度

创建 16 笔趋势

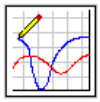
通过从 WindowMaker 中选择 16 笔趋势向导，可以创建趋势。

创建“16 笔”实时或历史趋势

1. 从 WindowMaker 中打开一个窗口以放置“16 笔趋势”。
2. 在**绘图**菜单上的**插入组**中，**选择向导**。

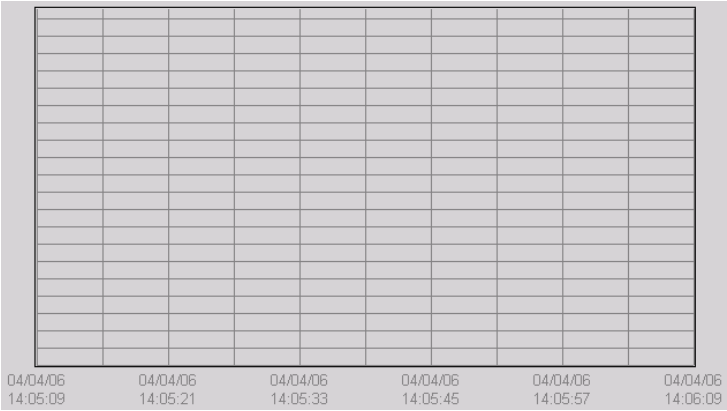
   此时出现**向导选择**对话框。
3. 从向导列表中选择**趋势**。

   此时**向导选择**对话框的右侧面板显示一组趋势向导图标。
4. 选择 **16 笔趋势**向导，然后选择**确定**。

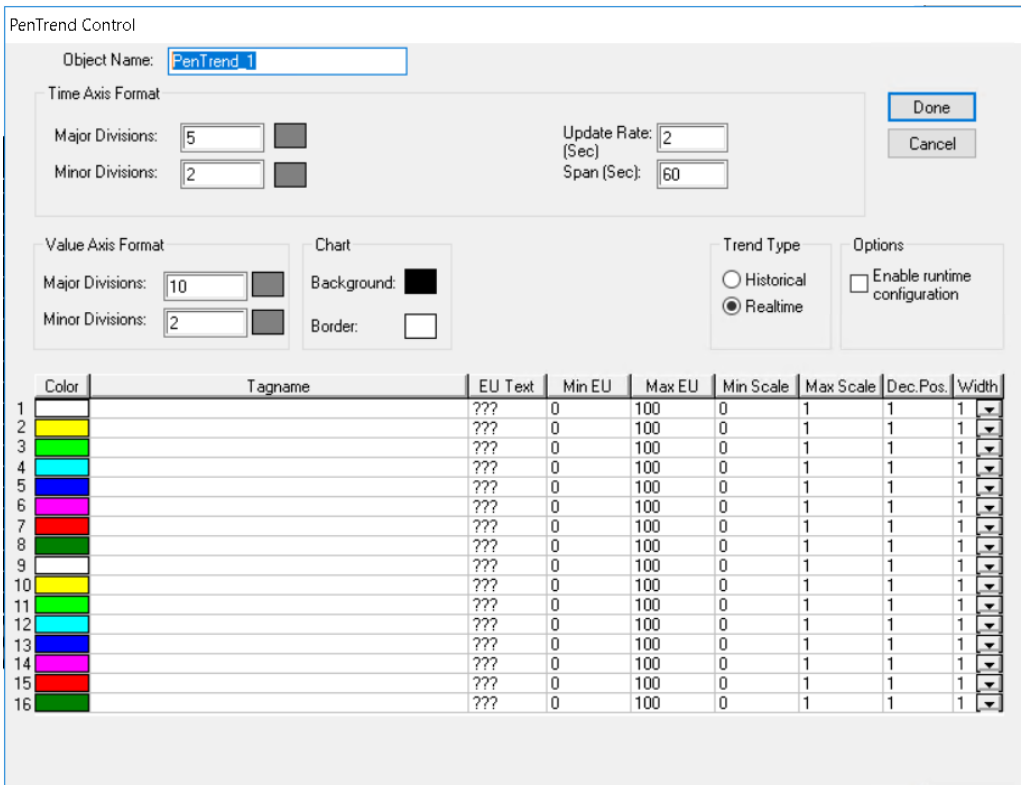
   此时**向导选择**对话框关闭，您的窗口再次出现。
5. 在窗口中单击以放置“16 笔趋势”。

   向导在窗口中放置一个“16 笔趋势”模板。

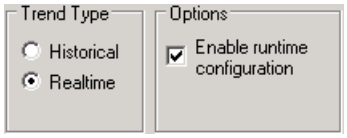
   

6. 双击该“16 笔趋势”模板，以打开 **PenTrend** 控件对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



7. 在**趋势类**型区域中，将**历史**或**实时选**作要创建的**趋势**的类型。



PenTrend 控件对话框会根据所选的趋势类型自动显示适当的时间与更新选项。

8. 在**选项**区域中，选择或清除**允许运行时配置选项**。

选择此选项时，操作员可以在“16 笔趋势”运行期间修改它的某些属性。

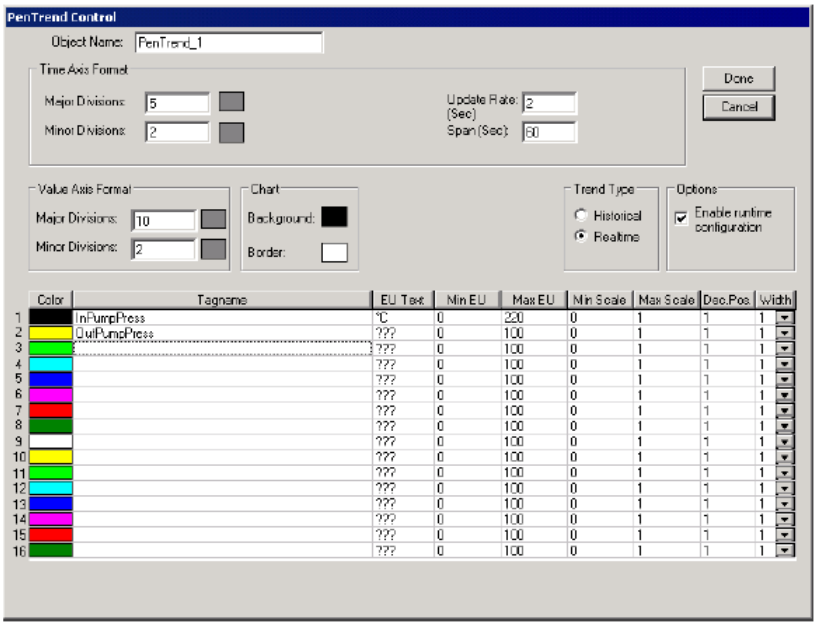
配置要在趋势图上显示的标记

您可以使用 16 笔趋势向导将标记指定给趋势笔。16 笔趋势向导包括一些列，可以指定要在趋势上显示的标记属性。这些列使用指定给标记的缺省属性值（来自“标记名字典”）。通过在配置趋势时指定其它值，可以覆盖指定的这些标记值。

配置“16 笔趋势”标记

1. 打开包含“16 笔趋势”模板的窗口。
2. 双击“16 笔趋势”。此时出现 **PenTrend** 控件对话框，底部有一块网格区域，可用于指定与趋势笔关联的标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 在**对象名**框中，给“16 笔**趋势**”指定一个名称。

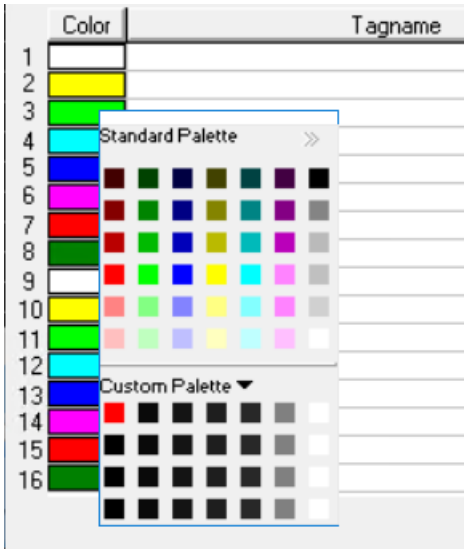
   缺省名称是 PenTrend_1，**每创**建一个新的**趋势**，名称中的数字便**递**增 1。

4. 在**标记名**框中，**输**入要与网格左**侧**列出的笔号**关联**的**标记**名。

   双**击标记名**下的**单**元格内部**时**，**显**示**选择标记**对话框。通过从**选择标记**对话框中**选择标记**，可以将**标记**指定**给**笔。

   注意：通**过选择**包含**标记**名的“**标记**名”框，然后按**键盘**上的空格**键**，可以**删**除**标记**。

5. 在**颜色**列中，**单击每**个**颜**色框以打**开调**色板。**给**笔**选择**一**种颜**色。



6. 在**工程单位文本**列中，**输**入要在运行**时**最初使用的文本，以用作**每**支笔的笔**轴**的**标题**文本。

   此文本是笔**设置为**活**动时**的**轴**文本。**工程单位文本**列的初始指定**为**“**标记**名字典”中**标记**的工程**单**位。您可以覆盖“16 笔**趋势**”的**这**些缺省工程**单**位。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

7. 在最小工程单位列中，输入指定给笔的最小工程单位值。

   最小工程单位列最初显示标记的最小工程单位值（来自“标记名字典”）。您可以指定另一个仅应用于“16 笔趋势”的最小工程单位值。

8. 在最大工程单位列中，输入指定给笔的最大工程单位。

   最大工程单位列最初显示标记的最大工程单位值（来自“标记名字典”）。您可以指定另一个仅应用于“16 笔趋势”的最大工程单位值。

注意：“最小/最大工程单位”对于显示历史趋势数据非常重要。历史趋势显示的工程单位刻度范围是 0-100%。

---

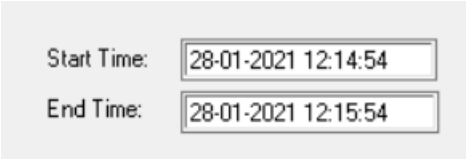
1. 在最小刻度值列中，输入要在运行时最初使用的百分比，以用于计算相应工程单位刻度的最小笔轴网格。
2. 在最大刻度值列中，输入要在运行时最初使用的百分比，以用于计算相应工程单位刻度的最大笔轴网格。
3. 在小数位数列中，输入在标记笔轴网格时最初要在运行时使用的小数位数。
4. 在宽度列中，选择笔画宽度（以像素单位）以用于绘制趋势上显示的数据值。
5. 继续执行下一个操作程序，以更新“16 笔趋势”的趋势时间与更新速率。

配置趋势的时间跨度与更新速率

根据正在创建实时还是历史“16 笔趋势”，**PenTrend** 控件对话框显示不同的选项。对于历史趋势，可以设置时间跨度；对于实时趋势，可以设置更新速率。

配置 16 笔历史趋势的时间跨度

1. 双击窗口中的“16 笔”历史趋势。此时出现 **PenTrend** 控件对话框，其中包含用于设置趋势起始和结束日期与时间的选项。



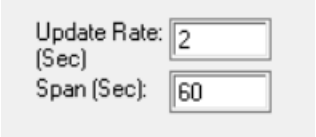
2. 设置历史趋势的起始和结束日期与时间。

   起始和结束日期与时间都采用以下格式：

   DD-MM-YYYY HH:MM:SS AM/PM

配置“16 笔”实时趋势的更新速率

1. 双击窗口中的“16 笔”历史趋势对象。此时出现 **PenTrend** 控件对话框，其中包含用于设置实时趋势更新速率和跨度的选项。



2. 在更新速率框中，输入历史趋势每次刷新间隔的秒数。
3. 在跨度框中，输入趋势中显示的实时间隔的秒数。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

配置趋势的显示选项

通过使用“16 笔趋势”向导，可以配置趋势的外观效果。

配置“16 笔趋势”的显示选项

1. 在 WindowMaker 中双击 **16 笔趋势**。

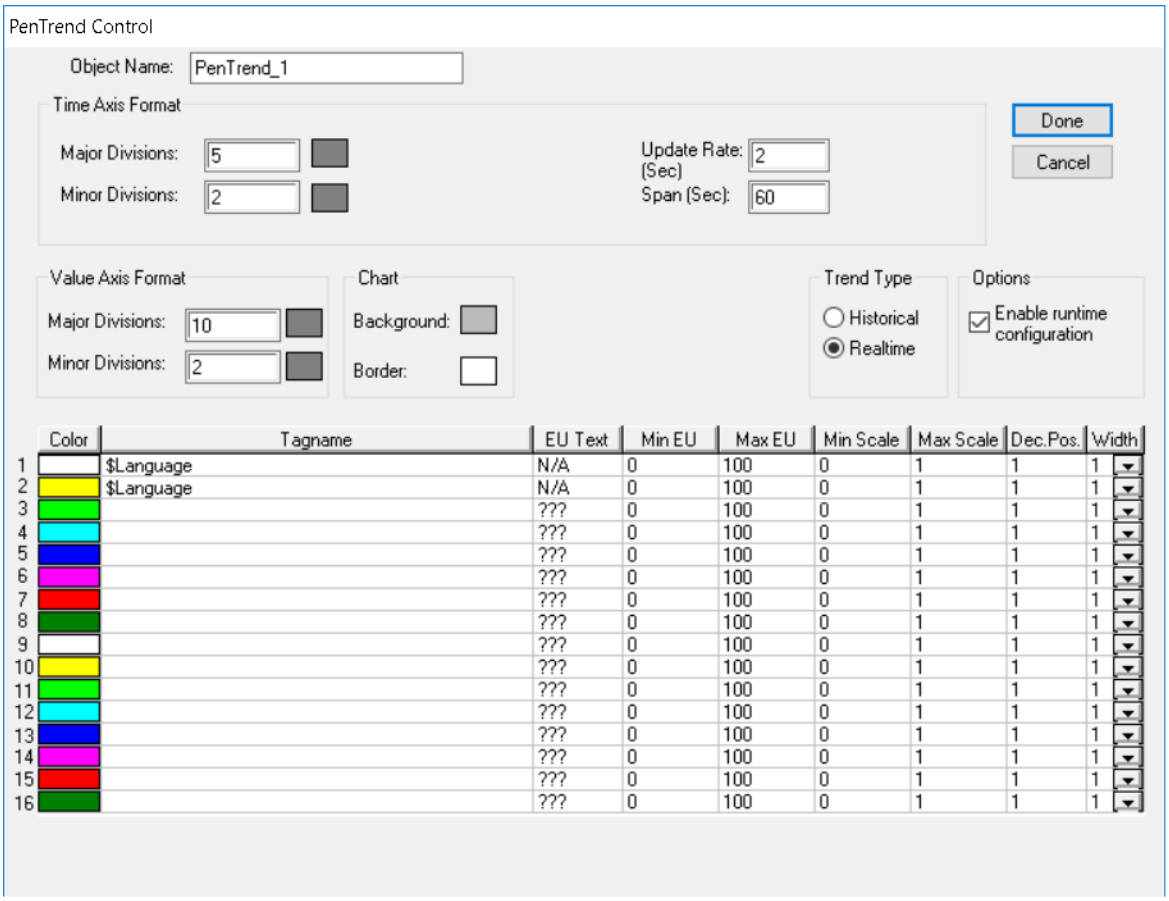
   此时出现 **PenTrend 控件对话**框。
2. 在**时间轴格式**区域的**主刻度**中，输入主**时间**刻度数。此**选项设置趋势**横坐**标轴**的主**时间**刻度数。
3. 如果要为主**时间**刻度**线**指定另一**种颜**色，**请选择主刻度**右**侧**的**颜**色框，以打**开调**色板并**选择**一**种颜**色。否则会跳过此**步骤**并接受**给**主**时间**刻度**线**指定的缺省**颜**色。
4. 在**副刻度**框中，输入**趋势**横坐**标轴**上**显**示的副**时间**刻度数。
5. **给**副**时间**刻度**线选择**一**种颜**色。
6. 在**值轴格式**区域中的**主刻度**中，输入**趋势**的主**时间**刻度数。此**选项设**置**纵**数**值轴**上**显**示的主刻度数。
7. **设**置主**值**刻度的**颜**色。
8. 在**副刻度**框中，输入**趋势纵**坐**标轴**上**显**示的副数**值**刻度数。
9. **设**置副数**值**刻度的**颜**色。
10. 在**图表**区域中，**选择趋势**的背景与**边**框**颜**色。
11. **选择完成**将对配置所作的更改保存到“16 笔**趋势**”中。

在运行时更改趋势的配置

如果**选择**了 **PenTrend 控件对话**框中的**启用运行时配置选项**，则操作**员**可以在**应**用程序运行期**间**更改“16 笔**趋势**”的一些特征。

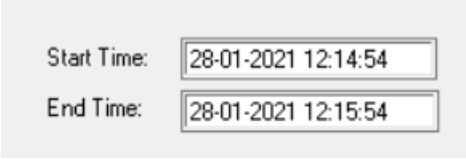
在运行**时对**“16 笔**趋势**”所作的更改不是永久性的。如果操作**员关闭** WindowViewer，然后再次启**动应**用程序窗口，则“16 笔**趋势**”将保留原来在 WindowMaker 中定**义**的配置。下**图显**示**选择**正在运行的“16 笔**趋势**”时出**现**的 **PenTrend 控件对话**框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



在运行时可以更改以下内容：

- 给趋势笔指定的标记或表达式
- 趋势标记或表达式的特征
- 趋势的类型（“历史”或“实时”）
- 历史趋势的日期与时间范围



- 实时趋势的更新速率与频率。

单击完成之后，趋势会在当前 WindowViewer 应用程序会话期间保留所作的配置更改。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

使用脚本控制 16 笔趋势向导

您可以在 QuickScript 中使用一组函数在运行时控制“16 笔趋势”对象。例如，可以将笔连接到图表上、将新的事件添加到图表、删除或重新绘制网格，以及删除或重新绘制指示器。

ptGetTrendType() 函数

ptGetTrendType() 函数可以在脚本中用于返回一个值，此值指出“16 笔趋势”的当前模式是显示历史还是实时数据。

类别

笔趋势

语法

```
ptGetTrendType(TrendName);
```

参数

TrendName
趋势的名称。*TrendName* 必须是字符串常量或消息标记。

返回值

返回趋势类型：

0 = 历史趋势

1 = 实时非滚动趋势

2 = 实时趋势

示例

下例返回一个值，指出 PumpPress 趋势是显示历史还是实时数据。

```
ptGetTrendType("PumpPress");
```

ptLoadTrendCfg() 函数

在脚本中，ptLoadTrendCfg() 函数可用于从文件中加载趋势配置值。

类别

笔趋势

语法

```
ptGetTrendCfg(TrendName,FileName);
```

参数

TrendName
趋势对象的名称。指定给 *TrendName* 的值必须是字符串常量或消息标记。

FileName
配置文件的名称。配置文件的文件夹路径必须包含在 *FileName* 参数中。

示例

使用 C:\TrendCfg.txt 文件中的值来配置 TankFarm 趋势。

```
ptLoadTrendCfg("TankFarm","C:\TrendCfg.txt");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

ptPanCurrentPen() 函数

在脚本中，ptPanCurrentPen() 函数可用于沿纵数值轴向上或向下滚动“16 笔趋势”的笔。垂直滚动由主趋势单位与副趋势单位数（作为参数值指定）确定。

类别

笔趋势

语法

```
ptPanCurrentPen(TrendName,MajorUnits, MinorUnits);
```

参数

TrendName
趋势对象的名称。*TrendName* 必须是字符串常量或消息标记。

MajorUnits
系数，确定按主刻度线定义的单位滚动的次数。负数表示沿纵轴向下滚动。

MinorUnits
系数，确定按副刻度线定义的单位额外滚动的次数。负数表示沿纵轴向下滚动。

示例

本例将笔向上滚动 1 个主刻度线。

```
ptPanCurrentPen("TrendName", 1, 0);
```

本例将笔向上滚动 0.5 个副趋势刻度。

```
ptPanCurrentPen("TrendName", 0, 0.5);
```

本例将笔向下滚动 2 个主刻度线加上 0.5 个副刻度线。

```
ptPanCurrentPen("TrendName", -2, -0.5);
```

本例向上滚动 1 个主刻度线并向下滚动 2 个副刻度线。

```
ptPanCurrentPen("TrendName", 1, -2);
```

ptPanTime() 函数

在脚本中， ptPanTime() 函数可用于根据指定的主或副趋势单位数，沿时间横轴向左或向右滚动“16 笔趋势”的笔。

类别

笔趋势

语法

```
ptPanTime(TrendName, MajorUnits, MinorUnits);
```

参数

TrendName
趋势对象的名称。*TrendName* 必须是字符串常量或消息标记。

MajorUnits
系数，确定按横向的主刻度线滚动的次数。负数表示将趋势向左平移。

MinorUnits
系数，确定按副刻度线定义的单位额外滚动的次数。负数表示将趋势向左平移。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附注

开发期间在 **PenTrend** 控件中指定的“主刻度”与“副刻度”的设置是计算滚动量的基础。对于时间跨度为 120 秒、主刻度值为 10、副刻度值为 2 的趋势，产生的趋势是每 12 秒有一条主刻度线、每 6 秒有一条副刻度线。ptPanTime("TrendName",1,0.5) 函数将时间轴滚动 1*12 + 0.5*6 = 15 秒。

示例

本例将笔沿趋势的横坐标轴向右滚动 1 个主刻度。

```
ptPanTime("TrendName", 1, 0);
```

本例将笔沿趋势的横坐标轴向右滚动 0.5 个副刻度。

```
ptPanTime("TrendName", 0, 0.5);
```

本例将笔沿趋势的横坐标轴向左滚动 2.5 个主刻度。

```
ptPanTime("TrendName", -2, -0.5);
```

本例将笔沿趋势的横坐标轴向右滚动 1 个主刻度并向左滚动 2 个副刻度。

```
ptPanTime("TrendName", 1, -2);
```

ptPauseTrend() 函数

在脚本中，ptPauseTrend() 函数可用于使“16 笔趋势”暂时停止更新图形。趋势保持停止状态，直到使用 0 值再次调用 ptPauseTrend。

类别

笔趋势

语法

```
ptPauseTrend(TrendName, Value);
```

参数

TrendName
趋势对象的名称。*TrendName* 必须是字符串常量或消息标记。

值
值为 1 时暂停更新趋势。值为 0 时恢复更新趋势。

示例

本例在 *Value* 参数为 1 期间，暂停对“16 笔趋势”的任何进一步更新。

```
ptPauseTrend ("TrendName",1);
```

ptSaveTrendCfg() 函数

在脚本中，ptSaveTrendCfg() 函数可用于将趋势的当前配置值保存到文件中。

类别

笔趋势

语法

```
ptSaveTrendCfg(TrendName,FileName);
```

参数

TrendName
趋势对象的名称。必须是字符串常量或消息标记。

FileName

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要用于保存趋势配置值的文件的名称。配置文件的文件夹路径可以通过 FileName 参数进行指定。

示例

ptSaveTrendCfg() 函数将 PumpTrend“16 笔趋势”的值保存到 C:\Config.txt 文件中。

```
ptSaveTrendCfg ("PumpTrend", "C:\Config.txt")
```

ptSetCurrentPen() 函数

在脚本中，ptSetCurrentPen() 函数可用于通过指定的编号来选择笔以控制笔轴。

类别

笔趋势

语法

```
ptSetCurrentPen(TrendName, PenNum);
```

参数

TrendName
趋势的名称。必须是字符串常量或消息标记。

PenNum
要指定为当前趋势笔的笔的编号 (1-16)。

示例

ptSetCurrentPen() 函数将 2 号笔指定为 PumpPress 趋势当前的笔。

```
ptSetCurrentPen("PumpPress",2);
```

ptSetPen() 函数

在脚本中，ptSetPen() 函数可用于将标记指定给趋势笔。

类别

笔趋势

语法

```
ptSetPen(TrendName,PenNum,TagName);
```

参数

TrendName
趋势对象的名称。必须是字符串常量或消息标记。

PenNum
要指定为当前趋势笔的笔号。

TagName
要指定给趋势笔的标记的名称。

示例

ptSetPen() 函数将 **PumpInP** 标记指定给 PumpPress 趋势的 2 号笔。

```
ptSetPen ("PumpPress",2,"PumpInP");
```

ptSetPenEx() 函数

在脚本中，ptSetPenEx() 可用于将标记指定给特定的趋势笔，并覆盖在“标记名字典”中给标记指定的配置值。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

笔趋势

语法

```
ptSetPenEx(TrendName, PenNum, TagName, minEu, maxEU, minPercent, maxPercent, Decimal, EU);
```

参数

TrendName
趋势对象的名称。必须是字符串常量或消息标记。

PenNum
要指定为当前趋势笔的笔号。

TagName
要指定给趋势笔的标记的名称。

minEU
指定的标记的最小工程单位值。

maxEU
指定的标记的最大工程单位值。

minPercent
在运行时最初用于计算相应工程单位刻度的最小笔轴网格的百分比。

maxPercent
在运行时最初用于计算相应工程单位刻度的最大笔轴网格的百分比。

十进制
趋势中标记值的小数精度。

EU
标记工程单位的标签。

示例

ptSetPenEx() 函数将 **PumpInP** 标记指定给 PumpPress 趋势的 2 号笔。标记工程单位范围设置为 0 到 1500，其单位是 PSI。网格的百分比范围是 0 到 1，小数精度设置为 2。

```
ptSetPenEx ("PumpPress", 2, "PumpInP", 0, 1500, 0, 1, 2, "PSI");
```

ptSetTimeAxis() 函数

在脚本中，ptSetTimeAxis() 函数可用于设置趋势的起始和结束日期与时间。

类别

笔趋势

语法

```
ptSetTimeAxis(TrendName,StartDateTime,EndDateTime);
```

参数

TrendName
趋势对象的名称。必须是字符串常量或消息标记。

StartDateTime
趋势起始的日期与时间。起始日期和时间的格式为：dd/mm/yyyy hh:mm:ss AM/PM。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

EndDateTime
趋势结束的日期与时间。起始日期和时间的格式为：dd/mm/yyyy hh:mm:ss AM/PM。

示例

ptSetTimeAxis() 函数将趋势的起始和结束日期与时间设置为从 2007 年 5 月 22 日 8:30 开始，时间跨度是 25 小时。

```
ptSetTimeAxis ("PumpPress", "05/22/2007 08:30:00 AM", "05/23/2007 09:30:00 AM");
```

ptSetTimeAxisToCurrent() 函数

在脚本中，ptSetTimeAxisToCurrent() 可用于计算图表当前的时间跨度以及图表的结束时间。

类别

笔趋势

语法

```
ptSetTimeAxisToCurrent(TrendName);
```

参数

TrendName
趋势对象的名称。*TrendName* 必须是字符串常量或消息标记。

示例

ptSetTimeAxisToCurrent() 函数将 PumpPress 趋势的结束日期与时间设置为当前日期与时间。

```
ptSetTimeAxisToCurrent("PumpPress");
```

ptSetTrend() 函数

在脚本中，ptSetTrend() 函数可用于暂停或重新开始更新“16 笔趋势”。

类别

笔趋势

语法

```
ptSetTrend(TrendName, EnableUpdates);
```

参数

TrendName
趋势对象的名称。必须是字符串常量或消息标记。

EnableUpdates
值为 1 时开始更新趋势。值为 0 时停止更新趋势。

示例

ptSetTrend() 函数更新 PumpPress 趋势。

```
ptSetTrend("PumpPress",1);
```

ptSetTrendType() 函数

在脚本中，ptSetTrendType() 函数可用于指定趋势是显示历史还是实时数据。

类别

笔趋势

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

语法

```
ptSetTrendType(TrendName, TrendType);
```

参数

TrendName
趋势对象的名称。必须是字符串常量或消息标记。

TrendType
值为 1 时指定历史趋势。值为 2 时指定实时趋势。

示例

ptSetTrendtype() 函数指定 PumpPress 趋势显示实时数据。

```
ptSetTrendType("PumpPress",2);
```

ptZoomCurrentPen() 函数

在脚本中，ptZoomCurrentPen() 函数可用于更改趋势 Y 轴上显示值的范围。趋势纵数值轴的范围可以按指定的缩放比率增加或减少。

类别

笔趋势

语法

```
ptZoomCurrentPen(TrendName,ZoomFactor);
```

参数

TrendName
趋势对象的名称。必须是字符串常量或消息标记。

ZoomFactor
指定大于 1.0 的数字时，可以增加趋势值的范围，具体是将当前的范围限制乘以缩放因子。指定小于 1.0 的缩放因子时，可以减少趋势纵坐标轴上显示值的范围。

附注

缩放比率应用于当前笔 Y 轴范围的现有跨度。例如，如果趋势 Y 轴范围开始时为 –50 到 50，则按 2.0 的比率缩放之后，新的范围是 –100 到 100。如果再次按 2.0 的比率进行缩放，则新的范围是 –200 到 200。缩放比率应用于当前生效的范围，而不是原始的 Y 轴范围。

在运行时，每个趋势笔的缩放比率会持久保持。使用 **ptSetCurrentPen()** 函数从一支笔切换到另一支时，Y 轴值范围会反映所选笔的当前刻度范围。

示例

ptZoomCurrentPen 函数将 "PumpPress" 趋势中当前标记的 Y 轴范围加倍。

```
ptZoomCurrentPen("PumpPress",2);
```

ptZoomTime() 函数

在脚本中，ptZoomTime() 函数可用于更改趋势横轴上显示的时间跨度。

类别

笔趋势

语法

```
ptZoomTime(TrendName,Zoom);
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

参数

TrendName
趋势对象的名称。必须是字符串常量或消息标记。

Zoom
指定大于 1.0 的数字时，可以增加趋势横轴上显示的时间段。指定小于 1.0 的数字时，可以减少横轴上显示的时间段。

示例

ptZoomTime() 函数将按 17% 增加趋势横轴上显示的时间段。

```
ptZoomTime("PenTrend_1",1.17);
```

ptZoomTime() 函数按 50% 减少趋势横轴上显示的时间段。例如，如果原始的时间范围设置为 1 小时，则 **ptZoomTime()** 函数将趋势的时间段减少到 30 分钟。

```
ptZoomTime("PenTrend_1", 0.5);
```

Symbol Factory

Symbol Factory 汇集了 4000 多个工业符号，这些符号可用作 InTouch 应用程序窗口中的可视化元素。Symbol Factory 是可以在 InTouch HMI 安装期间安装的辅助组件。

注意：通过使用“工业图形编辑器”，可以为与 Application Server 交互的 InTouch 应用程序创建可视化的元素。您还可以使用“图形编辑器”为应用程序创建独立于 InTouch HMI 的可视化智能元素。如需有关“工业图形编辑器”的详细信息，请参阅 Application Server 文档。

AVEVA 对于此产品的任何部分都不提供任何形式的担保。您可以将问题报告给“全球技术支持”。我们强烈建议，在安装或使用任何新的实用程序或应用程序之前，务必备份应用程序与数据。

符号类型

Symbol Factory 包含四种类型的向导：

- 图片向导
- 位图向导
- 纹理向导
- InTouch 对象

图片向导

Symbol Factory 图片向导是设备或流程图的矢量图像。创建应用程序时，可以通过执行以下操作来修改图片向导图像：

- 给图像指定动画
- 水平或垂直翻转图像
- 更改图像的水平与垂直透视图
- 绕轴旋转图像
- 更改图像的填充颜色与图案
- 更改图像线条的尺寸、样式及颜色

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

位图向导

位图向导是位图图像，如窗口图标或文本块。创建应用程序时，可以通过执行以下操作来修改图片向导图像：

- 给位图指定动画
- 水平或垂直翻转图像
- 更改位图的水平与垂直长度
- 放置边框或是在位图边框周围放置阴影
- 以 90 度为增量绕轴旋转位图图像
- 定义透明色
- 使用其它颜色替换位图中的最多三种颜色

纹理向导

纹理向导类似于位图向导，只是它的大小可以调整，以便形成连续图案。纹理向导通常用于创建窗口的背景，或是用作图形对象的填充。您可以从 Symbol Factory 的“纹理”类别中选择纹理向导。

InTouch 对象

InTouch 对象是指“按照原样”存储在 Symbol Factory 中的 InTouch 单元或向导。

将 InTouch 对象从 Symbol Factory 中粘贴到 WindowMaker 之后，就不能再在 Symbol Factory 中对其进行编辑。

选择 WindowMaker 中的对象时，如果该对象是单元，则出现替换标记名对话框；如果该对象是单独的图形对象，则出现动画链接选择对话框。

使用 Symbol Factory

使用 Symbol Factory 向导与使用其它向导非常类似。选择向导，将它放置到窗口中，然后配置它。

快速入门

如果熟悉 Symbol Factory，请重温这些提示以快速完成入门：

- 要配置向导的选项，请在窗口中双击向导，然后选择 **Reichard Software 出品的 Symbol Factory** 对话框中的**选项**。
- 要将向导复制到另一个类别，请将它的略图拖到**类别**窗口中，并放到另一个类别上。要进行移动，请按住 SHIFT 键。
- 要编辑向导的描述，请使用鼠标右键单击向导略图。要编辑类别的描述，请使用鼠标右键单击类别。
- 要删除向导，请使用鼠标右键单击向导略图，然后选择删除符号。
- 通过配置 Symbol Factory，一组开发人员可以通过网络使用与充实相同的向导库。
- 特定类别中的所有向导都存储在使用 .cat 扩展名的文件中。Symbol Factory 类别文件通常位于 c:\program files\wonderware\intouch\symfac 文件夹中。您可以将该文件复制到另一台计算机的 c:\program files\wonderware\intouch\symfac 文件夹中。

在窗口中放置 Symbol Factory 向导

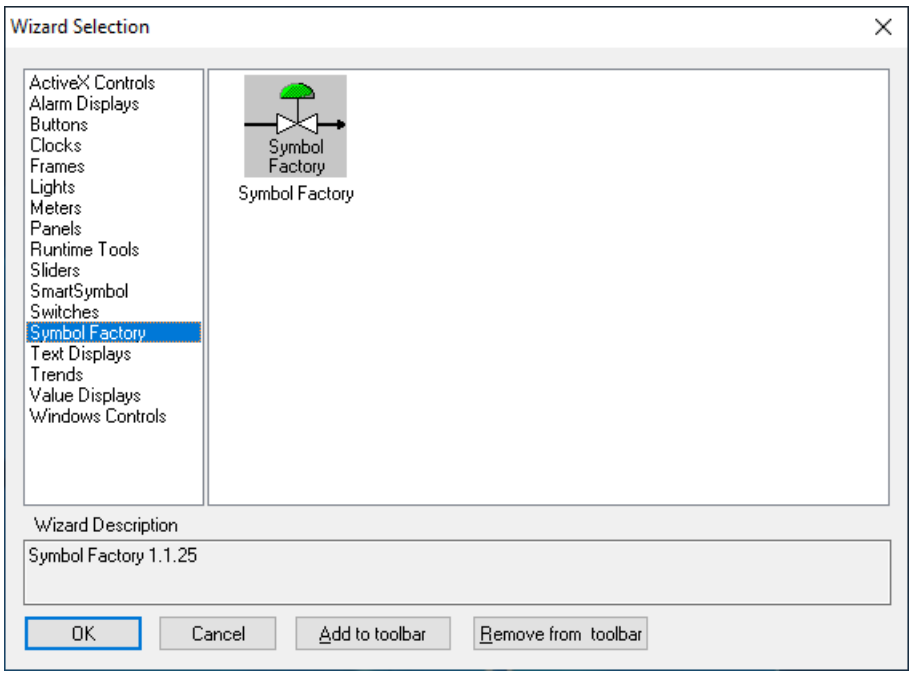
在窗口中放置 Symbol Factory 向导与放置其它向导类似。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

将 Symbol Factory 向导放入窗口

1. 在 WindowMaker 中打开一个应用程序。
2. 在**绘图**菜单上的**插入组**中，**选择向导**。

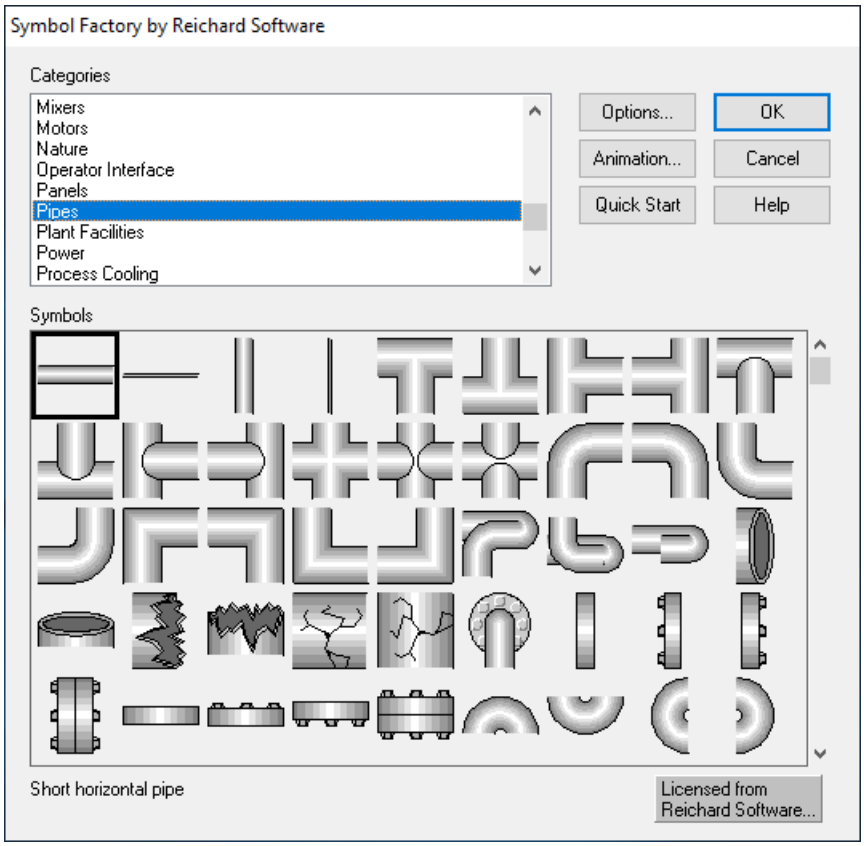
   此时出现向导**选择**对话框。



3. 在左侧窗格显示的向导列表中，**选择 Symbol Factory**。
4. **选择显**示区域中的 Symbol Factory 向导，然后**选择确定**。
5. 在窗口中**单击**以放置向导。

   此时出现 **Reichard Software** 出品的 **Symbol Factory** 对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



6. 在**类别**列表中，选择一个类别。此时符号窗口显示所选类别中的各个向导。
7. 选择要放置的向导，然后选择确定。

配置符号选项

向导的选项随向导的不同而有所不同。由于更改颜色时，每个像素都必须经过扫描并且有可能发生更改，因此将影响位图与纹理的绘制速度。

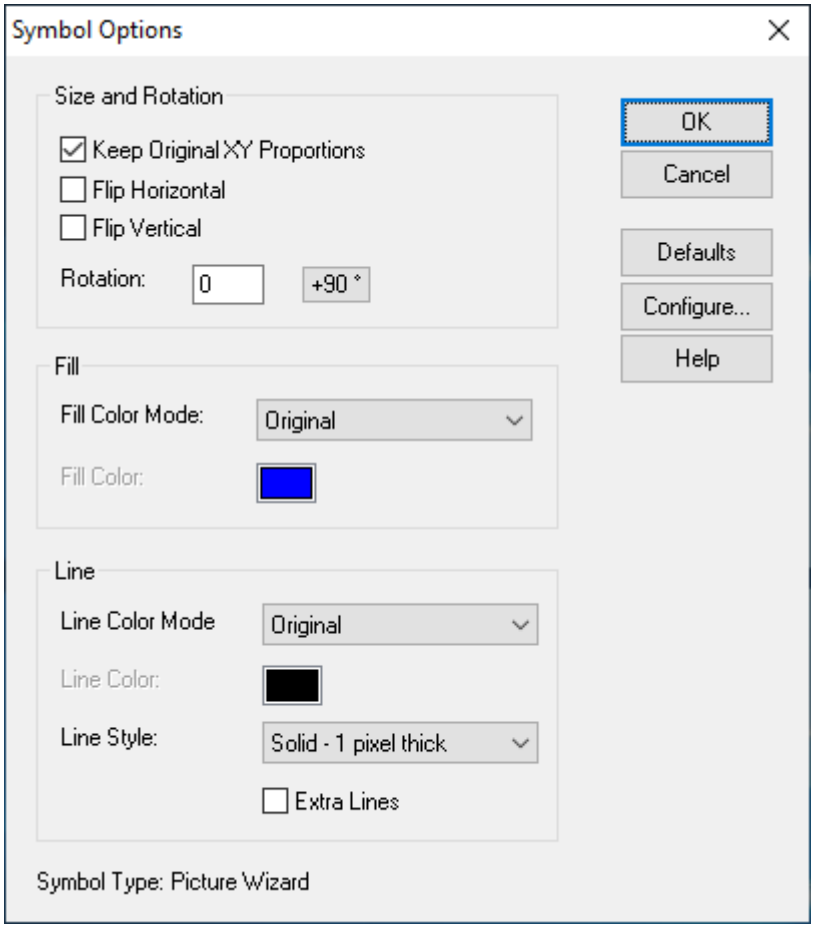
配置向导选项

1. 打开包含 Symbol Factory 向导的窗口。
2. 双击向导。

   此时出现 **Reichard Software** 出品的 **Symbol Factory** 对话框。
3. 保持选中该向导，然后选择**选项**。

   此时出现符号**选项**对话框。符号**选项**对话框上显示的图像属性根据选择要编辑的向导类型而有所不同。下例显示了选择图片向导符号时提供的选项。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



提示：如果选择了配置 Symbol Factory 对话框中的启用鼠标右键的替代按钮选项，则会显示“编辑符号”按钮，单击它可以配置所选的向导。

4. 在**大小与旋转**区域中，执行以下任一操作：
   - 选中**保持原始的 XY 比例复选**框，以保持向导的原始纵横比。
   - 选中**水平翻转复选**框，以水平翻转向导。
   - 选中**垂直翻转复选**框，以垂直翻转向导。
   - 在**旋转**框中，输入要旋转向导的度数。图片向导可以旋转到任何角度。位图与纹理向导仅能以 90 度为增量（0、90、180 或 270）进行旋转。选择按钮可以自动将旋转角度增加 90 度。
5. 如果正在配置图片向导，在**线条**与**填充**区域中，执行以下任一操作：
   - 在**填充颜色模式**列表中，选择一种填充类型。单击**填充颜色**框以打开调色板。
   - 在**线条颜色模式**列表中，选择一种线条颜色。单击**线条颜色**框以访问调色板。
   - 在**线条样式**列表中，选择一种线条样式。
   - 选中**额外线条复选**框，以在向导中的渐变边框处添加线条。
6. 如果正在配置位图或纹理向导，在**效果**与**更改颜色**区域中，执行以下任一操作：
   - 选中**包含边框复选**框，以在向导周围创建黑色边框。
   - 选中**包含阴影复选**框，以在向导后面创建暗灰色的阴影。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 选择每个颜色框，以便打开调色板来更改向导中的各种颜色。

7. 选择确定。

为向导设置动画效果

您可以给任何 Symbol Factory 向导设置动画效果。通过 Symbol Factory，可以访问最常用的动画链接。如果需要其它类型的动画链接，则必须分解向导，然后使用标准的 InTouch 动画链接给它设置动画效果。

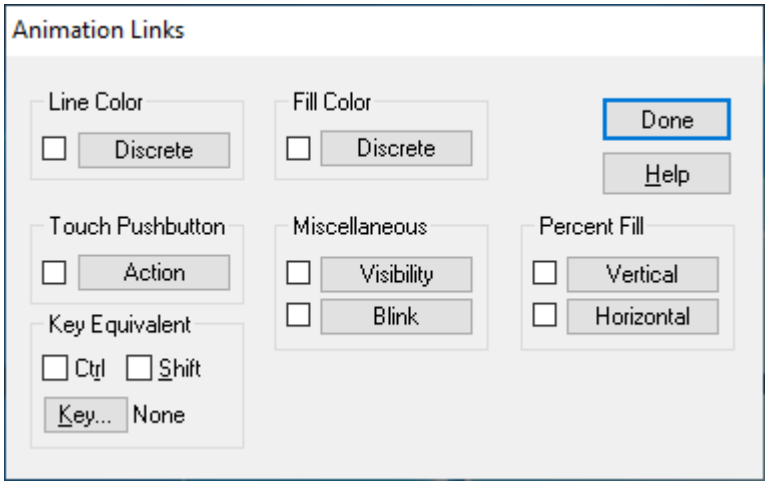
为向导设置动画效果

1. 在 Symbol Factory 中选择一个向导；如果已将向导粘贴到窗口中，则双击该向导。

   此时出现 **Reichard Software 出品的 Symbol Factory** 对话框。

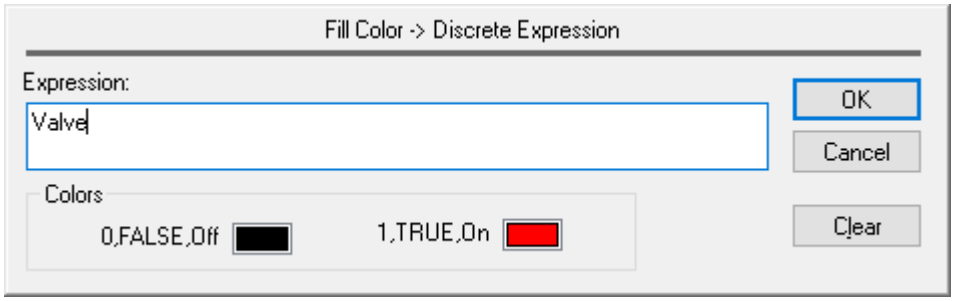
2. 选择动画。

   此时出现动画链接对话框。



3. 选择要应用于所选向导的每种类型的动画链接所对应的按钮。

   此时出现表达式对话框。



4. 在表达式窗口中，输入表达式。

   在窗口中双击以打开选择标记对话框。如果使用现有的标记，则可以双击表达式中的标记打开标记名字典，以查看该标记的定义。

   如果使用未定义的标记，则关闭表达式对话框时，程序会提示您定义它。

5. 配置动画链接类型的详细信息。

6. 选择确定。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

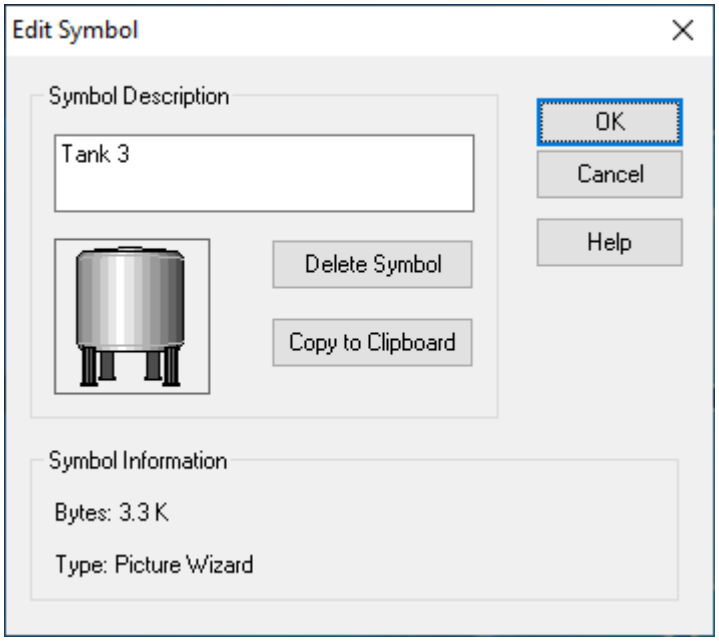
编辑符号

您可以更改向导工具提示中的描述，删除向导，或是将向导复制到 Windows 剪贴板。

编辑向导

1. 在 **Reichard Software** 出品的 **Symbol Factory** 对话框中，选择包含所需向导的类别。
2. 使用鼠标右键单击向导。

   此时出现编辑符号对话框。



3. 编辑向导。执行以下任意操作：
   - 在符号描述框中，输入工具提示文本。描述的最大长度是 80 个字符。
   - 选择删除符号以删除向导。
   - 选择复制到剪贴板以将向导复制到 Windows 剪贴板。如果该向导是图片向导，则将作为 Windows 元文件复制。如果该向导是位图向导或纹理向导，则将作为 Windows 位图复制。
4. 选择确定。

分解向导以便编辑

您可以分解 Symbol Factory 向导以便单独编辑各项。不过在分解向导之后，向导属性将会丢失。如果不小心分解了向导，可以使用“撤消”工具来重新组装。

分解向导

1. 打开 WindowMaker。
2. 在动画菜单上的单元组中，选择分解。

在网络上共享一个类别的符号

通过配置 Symbol Factory，可以允许多个开发人员通过网络使用与充实向导的类别文件。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

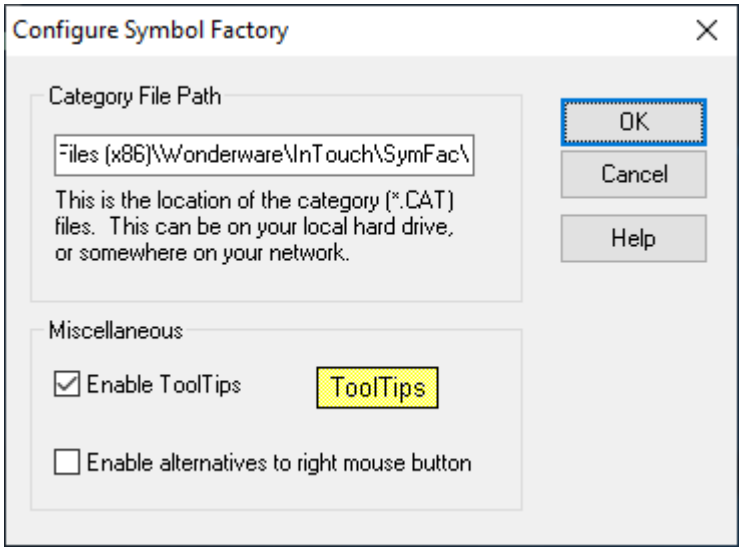
将类别文件移到网络文件夹中

1. 在 Symbol Factory 对话框中，选择**选项**。

   此时出现符号**选项**对话框。

2. 选择配置。

   此时出现配置 **Symbol Factory** 对话框。



3. 在**类别文件路径**框中，输入要用于保存类别文件的网络文件夹的完整路径。
4. 选择确定。

使类别只读

将向导文件存储到网络文件夹上时，可能会希望使类别成为只读的，以防止其他用户移动或重命名向导。

使类别只读

- 在 Windows 的<1>“</1>资源管理器<2>”</2>中<3>，</3>将文件设置为只读<4>。</4>

查看类别属性

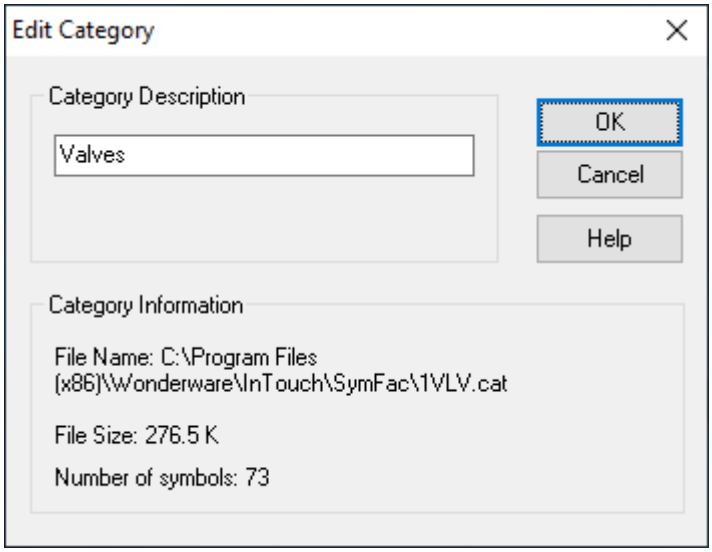
您可以查看类别路径、文件大小以及向导个数。

查看类别的属性

1. 在 **Symbol Factory** 对话框中，使用鼠标右键单击**类别**列表中的类别。

   此时出现**编辑类别**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**类别描述**框中，**输入类别**的新描述，然后**选择确定**。描述的最大**长**度是 40 个字符。
3. 在**类别信息**区域中，**查**看属性。

类别信息	**描述**
文件名	**类别** (.cat) 文件路径。缺省条件下，此路径是 C:\Program Files (x86)\Wonderware\InTouch\SymFac。
文件大小	**类别**文件的大小，以 KB **为单**位。
符号个数	**类别**中包含的向**导总**数。最多**为** 32767。

4. **选择确定**。

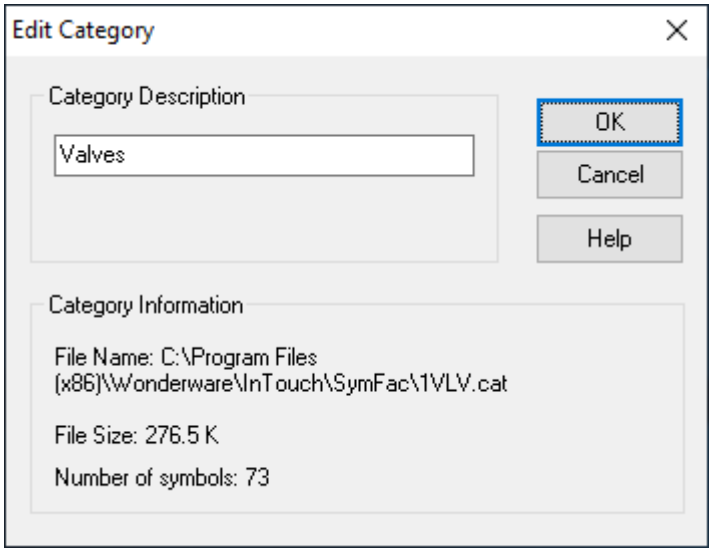
编辑现有的类别

您只能**编辑类别**名。

编辑现有的类别

1. 在 Symbol Factory **对话**框中，使用鼠**标右键单击类别**列表中的**类别**。
   此**时**出现**编辑类别**对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**类别描述**框中，**输入类别**的新描述，然后**选择确定**。描述的最大长度是 40 个字符。
3. **选择确定**。

删除类别

使用 Windows 的“资源管理器”，可以通过指定类别的文件名来删除类别 (.cat) 文件。

提示：您可以在”编辑类别“对话框中验证该类别的文件名。

配置 Symbol Factory

配置 Symbol Factory 时，可以指定：

- **选择**向导时是否显示工具提示。
- 在 **Reichard Software 出品的 Symbol Factory** 对话框中是否应该显示额外选项。缺省条件下，要编辑类别和向导描述，必须使用鼠标右键单击该项目。
- 类别 (.cat) 文件的位置。每个类别中的所有向导的全部数据都存储在一个类别文件中。由于性能方面的原因，此路径应该仅包含 .cat 文件。要与其他开发人员共享向导，请将此路径设置为网络文件夹。请参阅在网络上共享一个类别的符号。

警示：请勿将类别文件放置在本地 InTouch 应用程序文件夹中。相反，要将类别文件保存到：C:\Program Files\Wonderware\intouch\symfac。

配置 Symbol Factory

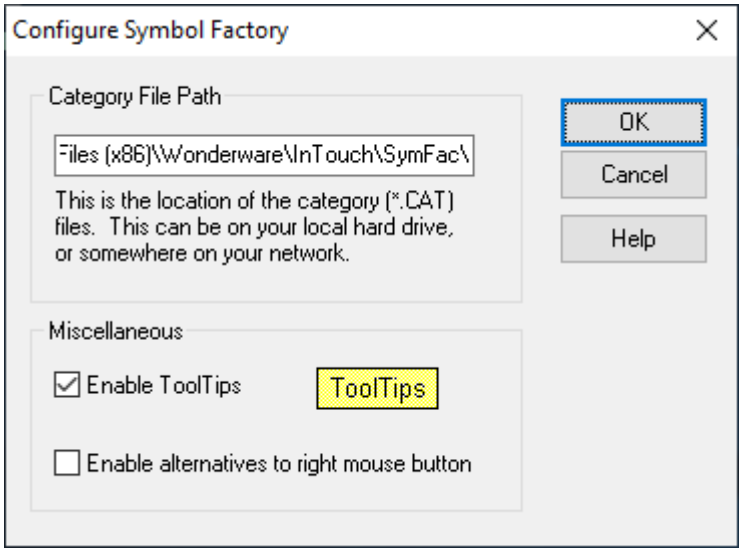
1. 在 Symbol Factory 对话框中，**选择选项**。

   此时出现符号**选项**对话框。

2. **选择配置**。

   此时出现**配置 Symbol Factory** 对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



3. 配置 Symbol Factory。执行以下操作：
   - 在**类别文件路径**框中，输入要用于保存 Symbol Factory 类别 (.cat) 文件的位置。
   - 如果要在 **Reichard Software 出品的 Symbol Factory** 对话框中显示向导的工具提示，请选中**启用工具提示复选**框。
   - 如果要将按钮添加到 **Reichard Software 出品的 Symbol Factory** 对话框中，请选中**启用鼠标右键的替代按钮复选**框。您可以使用以下按钮而不是鼠标右键来编辑类别和向导：

   编辑类别

   显示所选类别的“编辑类别”对话框。

   编辑符号

   显示所选向导的“编辑符号”对话框。
4. 选择确定。

疑难排解

如果不小心卸载了 Symbol Factory 向导，您需要重新安装。如需有关安装向导的详细信息，请参阅《*AVEVA ™ InTouch HMI 应用程序开发指南*》中的向导。

如果不小心删除了向导并希望恢复它，则必须检索它。

从类别中检索已删除的向导

1. 将文件 ~cat.bak 重命名为 temp.cat。
2. 运行 Symbol Factory 并检查已删除的向导是否已经恢复。将它移到其原始类别中，然后删除 temp.cat 文件。
3. 如果前一个步骤不起作用，请在按住 CTRL 键的同时使用鼠标右键单击包含已删除的向导的类别。这会压缩类别文件，并创建一个新的备份 ~cat.bak。

执行前面的步骤，直到找到已删除的向导。

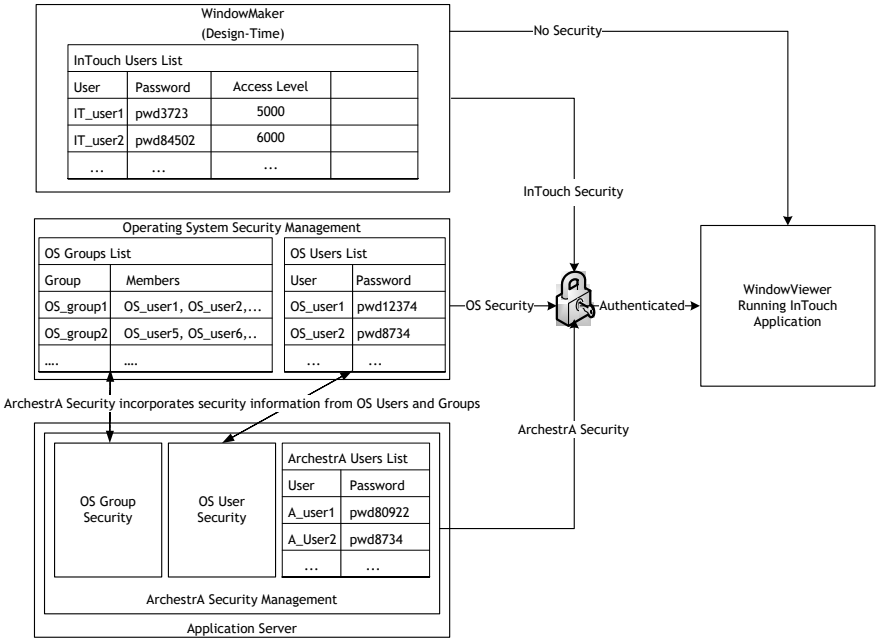
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

保护 InTouch 安全

您可以使用以下机制保护 InTouch 应用程序：

- 传统的基于 InTouch 的安全性
- 基于操作系统的安全性
- 基于 ArchestrA 的安全性

下图显示三种安全性之间的关系。



InTouch 安全性功能

要在运行时保护 InTouch 应用程序，可以：

- 设置不活动超时时段
- 锁定键
- 隐藏菜单

配置不活动超时

您可以将 WindowViewer 配置成从 InTouch 应用程序自动注销不活动的操作员。因为不活动而注销操作员之后，操作员必须再次登录。通过设置自动不活动注销时段，可以在操作员离开工作站而导致其无人照管时，防止他人未经授权访问您的 InTouch 应用程序。

有一个计时器测量操作员未与正在运行的 InTouch 应用程序进行交互的时段。每次操作员使用鼠标或任何其它输入设备输入数据时，该计时器会重置。如果计时器超时，用户会自动注销。

注意：不活动计时器不会因 Active-X 控件和 OLE 自动化控件而重置。

自动注销操作员的过程分为两个步骤：

1. 操作员的不活动时段超过指定的警告时段时，WindowViewer 将 $InactivityWarning 系统标记设置为 1。您可以在条件 QuickScript 中使用 $InactivityWarning 标记显示一个窗口，以警告操作员因不活动而将要

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注销的情况。操作员通过在指定的超时时段发生之前作出响应以保持登录。操作员采取某种操作时，$InactivityWarning 标记与不活动计时器重置为零。

2. 如果操作员在不活动警告之后未能作出响应，到达超时时段时，$InactivityTimeout 系统标记设置为 1。$InactivityTimeout 为 1 时，WindowViewer 将登录的操作员姓名设置为保留名 None，并将 $AccessLevel 安全性标记设置为 0。用户会自动注销。

您可以在警告功能之外独立使用超时功能。

要配置不活动超时

1. 打开 WindowMaker。
2. 在文件菜单上，指向配置，然后选择 **WindowViewer**。

   此时出现 **WindowViewer** 配置屏幕。
3. 在不活动（秒）区域中，配置警告与超时值。执行以下操作：
   - 在警告框中，输入在 $InactivityWarning 标记设置为 1 之前可以经过的秒数。
   - 在“超时”框中，输入在 $InactivityTimeout 标记设置为 1 且用户自动注销之前可以经过的秒数。

4. 选择确定。
5. 要在不活动警告时间过去之后显示名为“警告 - 暂停注销”的窗口，请创建以 "$InactivityWarning" 为条件并使用以下脚本主体的条件脚本：

```
Show "暂停注销";
```

6. 要在不活动超时之后显示“已注销”窗口，请创建以“$InactivityTimeout”为条件并使用以下脚本主体的条件脚本：

```
Show "已注销";
```

$InactivityTimeout 系统标记

指出为不活动配置的时间已经过去。

类别

安全性

用法

$InactivityTimeout

附注

不活动计时器的时间过去时设置为 1。如需有关设置注销时间的详细信息，请参阅配置不活动超时。

注意：不活动计时器不会因 ActiveX 控件、OLE 和自动化控件而重置。

数据类型

离散（只读）

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

$InactivityWarning

示例

下例是一个“为真时”条件脚本：

```
If $InactivityTimeout == 1 THEN
   Show "已注销";
ENDIF
```

另请参阅

$InactivityWarning

$InactivityWarning 系统标记

指出为警告用户注销即将发生所配置的时间已过去。

类别

安全性

用法

$InactivityWarning

附注

不活动警告的时间过去时设置为 1。不活动计时器仅通过鼠标或键盘活动进行重置。如需有关设置注销警告的详细信息，请参阅配置不活动超时。

注意：不活动计时器不会因 ActiveX 控件、OLE 自动化控件及 SPC 向导而重置。

数据类型

离散（只读）

示例

下例是一个“为真时”条件脚本。

```
If $InactivityWarning == 1 THEN
   Show "暂停注销";
ENDIF;
```

另请参阅

$InactivityTimeOut

锁定系统键

通过在运行 InTouch 应用程序的计算机上禁用系统键，可以限制操作员访问 Windows 标准函数。例如，您可以阻止操作员使用 Windows CTRL+ALT+DEL 组合键来显示任务管理器对话框。禁用系统键可以阻止操作员从 InTouch HMI 切换到其它 Windows 应用程序。

WindowViewer 有一些键过滤器选项，可以设置系统键在 InTouch 应用程序启动时的缺省状态。键过滤器在活动时禁用系统键。

系统键的禁用取决于您希望各种 InTouch 用户去完成哪些任务。大多数功能键应该对操作员禁用。管理员仍然需要这些功能键来执行自己的 InTouch 任务。

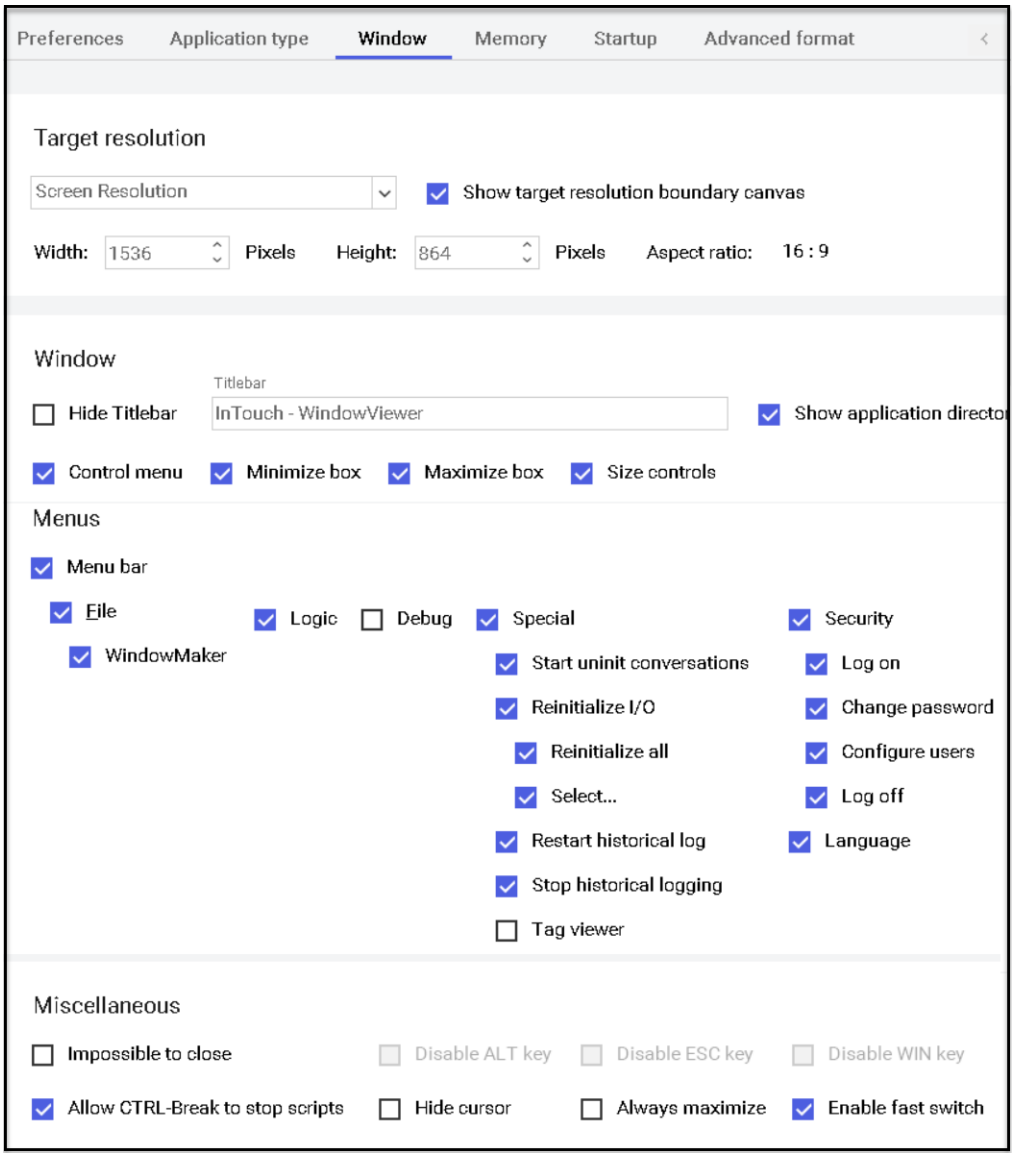
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

您可以编写一个脚本，根据登录到 WindowViewer 的用户的访问级别来启用或禁用系统键。您可以在脚本中使用 EnableDisableKeys() 函数，以便有选择性地启用或禁用一些 Windows 功能键。

启用键过滤器

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。
3. 选择窗口选项卡。



1. 在其它区域中，禁用 WindowViewer 系统键。执行以下操作：
   - 清除启用快速切换复选框，以便从 WindowViewer 中删除可以将用户切换到 WindowMaker 的开发按钮。
   - 选中禁用 **ALT** 键复选框，以在运行 InTouch 应用程序的计算机上禁用 ALT 键。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 选中禁用 **WIN 键**复选框，以在运行 InTouch 应用程序的计算机上禁用 WIN 键。
- 选中禁用 **ESC 键**复选框，以在运行 InTouch 应用程序的计算机上禁用 ESC 键。

2. 选择确定。
3. 编写一个在 WindowViewer 开始运行 InTouch 应用程序时运行的脚本。

   此脚本应该包含一些语句，根据登录到 WindowViewer 的用户的访问级别来动态执行键的锁定或释放。

   在脚本中包含 EnableDisableKeys() 函数以启用/禁用 ALT、ESC 及 WIN 键。EnableDisableKeys() 函数会根据离散参数值来启用或禁用系统键：

   ```
   EnableDisableKeys(AltKey,EscKey,WinKey);
   ```

   参数值为 1 时，将启用键过滤器来禁用该键。

EnableDisableKeys() 函数

启用/禁用 Alt、Escape 及 Windows 键的键过滤器。

类别

查看

语法

```
EnableDisableKeys(AltKey, EscKey, WinKey);
```

参数

AltKey
启用或禁用 Alt 键的键过滤器的整数：

1 = 启用过滤器（禁用 Alt 键）

0 = 禁用过滤器（启用 Alt 键）

EscKey
启用或禁用 Escape 键的键过滤器的整数：

1 = 启用过滤器（禁用 Esc 键）

0 = 禁用过滤器（启用 Esc 键）

WinKey
启用或禁用 Windows 键的键过滤器的整数：

1 = 启用过滤器（禁用 Win 键）

0 = 禁用过滤器（启用 Win 键）

附注

禁用 Alt 键时也会禁用 Win+L 组合键（锁定 Windows 桌面）。Win+L 是涉及 Alt 键的另一个组合键的快捷键。因此，禁用 Alt 键时也会禁用锁定 Windows 桌面的快捷键。

禁用 Esc 键会为所有操作禁用该键。

示例

```
EnableDisableKeys(0,0,0); // 启用所有三个键
EnableDisableKeys(1,1,1); // 禁用所有三个键
EnableDisableKeys(0,0,1); // 启用 Alt 和 Esc 键，禁用 Windows 键。
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

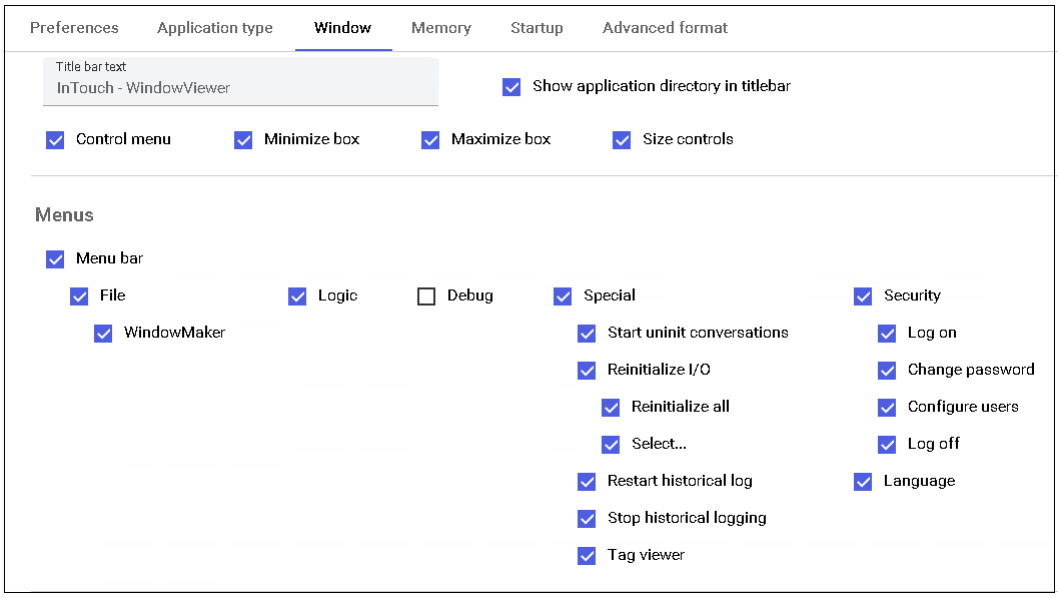
在运行时隐藏菜单项

您可以在运行 InTouch 应用程序时，通过隐藏 WindowViewer 菜单与命令来限制操作员访问它们。

在运行时隐藏菜单项

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择 **WindowViewer**。

   此时出现 WindowViewer 配置屏幕。
3. 选择窗口选项卡。



4. 在菜单区域中，选择希望对操作员隐藏的 WindowViewer 菜单与命令。执行以下操作：
   - 清除“WindowMaker”复选框，使 WindowMaker 命令在 WindowViewer 文件菜单中不可用。清除此选项不影响快速切换到 WindowMaker。
   - 清除“逻辑”复选框，以隐藏包含启动与停止 QuickScript 的命令的 WindowViewer **逻辑**菜单。

   注意：您可以使用 $LogicRunning 系统标记使操作员能够启动与停止所有的 QuickScript。
   如果选择允许 **CTRL-Break** 停止脚本选项，则无论是否显示**逻辑**菜单，操作员都可以停止运行所有的 QuickScript。
   当前正在执行的异步 QuickFunction 无法停止。不过，您可以阻止操作员启动新的异步 QuickFunction。

   - 如果正在测试应用程序，请选中“调试”复选框。否则清除**调试**复选框，以便在运行时隐藏**调试**菜单。
   - 清除**特别**菜单项以阻止操作员停止正在进行的 InTouch 功能，如记录与 I/O 连接。
   - 清除**安全性**复选框，以阻止操作员更改与安全性相关的选项。
5. 在窗口区域中，选择希望在 WindowViewer 中提供给操作员的窗口控件。这些选项会影响运行 InTouch 应用程序的窗口。执行以下操作：
   - 清除**控制菜单**复选框，以隐藏可以关闭、最小化、最大化窗口及调整窗口大小的这些控件。
   - 清除**最小化框**复选框，以阻止操作员最小化窗口。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

- 清除**最大化框复选**框，以阻止操作**员**最大化窗口。
- 清除**大小控制复选**框，以阻止操作**员调**整窗口大小。

6. 在**标题栏**区域中，配置运行 InTouch **应**用程序的窗口的**标题栏**。**执**行以下操作：
   - 在**标题栏文本**框中，**输**入要在 WindowViewer **标题栏**中**显**示的**标题**。
   - **选**中**显示应用程序目录复选**框，以在**标题栏**中包含 InTouch **应**用程序文件**夹**的路径。
   - **选**中**隐藏标题栏复选**框，以**隐**藏窗口的**标题栏**。
7. 在**其它**区域中，**执**行以下操作：
   - **选**中**无法关闭复选**框，以阻止操作**员关闭**正在运行 InTouch **应**用程序的 WindowViewer 窗口。**选择**此**选项时**禁用窗口的**关闭**按**钮**。

     如果希望**隐**藏**关闭**按**钮**，**请**清除**窗口**区域中的**控制菜单复选**框。
   - 清除**允许 CTRL-Break 停止脚本复选**框，以禁用能**够让**操作**员**停止 QuickScript 的 CTRL + BREAK **组**合**键**。

注意：当前正在**执**行的异**步** QuickFunction 无法停止。不**过**，可以阻止**执**行新的异**步** QuickFunction。

   - **选**中“**隐**藏光**标**”**复选**框，以在运行**时隐**藏鼠**标**光**标**。如果在**设计**触摸屏**应**用程序，**这**会很有用。
   - **选**中“**总**是最大化”**复选**框，以**让**运行 InTouch **应**用程序的窗口保持最大化**显**示在操作**员**屏幕上。
8. **选择确定**。
9. 重新启**动** WindowViewer 以**应**用更改。

基于身份**验证**与授**权**的安全性

InTouch 安全性是一个分**为**两个**步骤**的**过**程，第一**步**确定**尝试**使用**应**用程序的人**员**是否**为**有效用**户**。第二**步**确定通**过**身份**验证**的用**户拥**有哪些 InTouch **权**限。

比**较**身份**验证**与授**权**

身份**验证**是**验证**用**户**身份的**过**程。通常，操作**员**会在使用 InTouch **应**用程序之前**输**入用**户**名与口令**进**行**验证**。作**为**身份**验证过**程的一部分，所有三**种类**型的安全性都会在登**录过**程中**验证**用**户**的凭**证**。

授**权**是确定通**过**身份**验证**的用**户**是否可以**访问**所**请**求的**资**源的**过**程。通常，**对** InTouch 功能的**访问权**限是基于用**户**在**组**中的成**员关**系或是基于所指定的**访问级别**来授予的。

不同的身份**验证**安全模式

所有**类**型的 InTouch 安全性都在登**录过**程中使用用**户**名与口令来**验证**用**户**身份。**每种类**型的安全性都在身份**验证过**程中提供一**种**不同的机制来**验证**用**户**名与口令。

使用基于 InTouch 的安全性

您可以**为应**用程序**应**用安全机制。缺省条件下，InTouch **应**用程序不采用安全措施。不**过**，通**过**将功能**链**接到内部**标记**，可以**对**允**许**操作**员执**行哪些功能作出限制。此外，在**应**用程序上**设**立安全机制**时**，可以**创**建**审**核跟踪**线**索，将**报**警与事件同登**录**到 InTouch HMI 的操作**员关联**起来。

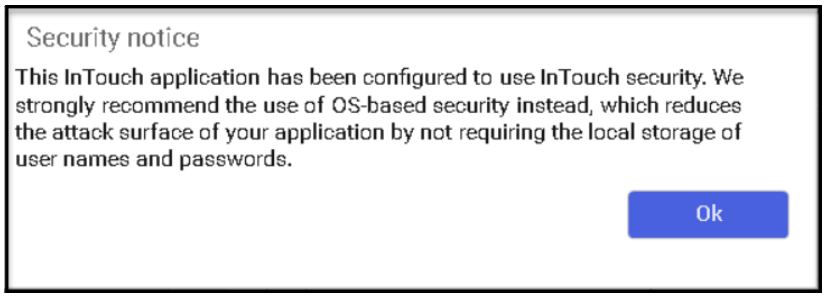
要将“安全性**类**型”设置为 InTouch

1. 打**开** WindowMaker。
2. 在**文件**菜**单**上，**选择配置**，然后**选择安全性**。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有**权**利。

此时出现安全性配置屏幕。

3. 将安全性类型选择为 **InTouch**。

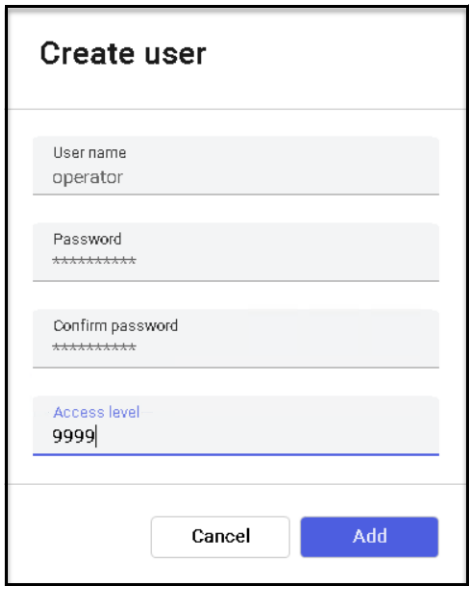


您将安全性类型添加为 InTouch 时，会启用配置用户部分。安全性基于操作员通过输入用户名与密码登录到 InTouch 应用程序来进行验证。您必须给每个操作员指定用户名、密码及访问级别。

要配置用户

1. 选择“添加”图标（Alt+A）。

   此时出现创建用户对话框。



2. 在用户名框中，输入要指定给操作员的名称。
3. 在密码框中，输入操作员密码，最多可以包含 29 个字符。
4. 在确认密码框中，再次输入同一密码。
5. 在访问级别框中，输入操作员的访问级别（最低 = 0 到最高 = 9999）。

将新用户名添加到安全性列表并重新启动 WindowMaker 或 WindowViewer 之后，缺省用户名自动重置为 None，访问级别为 0，这会阻止访问 WindowMaker 与 WindowViewer 中的配置用户命令。不过，Administrator 帐户与密码保持不变，并且仍可以使用。

操作员登录到应用程序之后，访问任何受保护的功能时，系统都会根据为链接到该功能的内部安全性标记所指定的值来验证操作员的密码和访问级别，以确定是否授予访问权限。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

对于独立应用程序，只允许具有管理员权限的用户在 InTouch WindowMaker 中打开和编辑应用程序。如果没有管理员权限的用户试图启动 InTouch WindowMaker，将显示一个错误对话框，通知用户需要管理权限才能继续。没有管理权限的用户可以通过用于托管应用程序的 IDE 启动 WindowMaker。

使用基于操作系统的安全性

基于操作系统的身份验证方法从 Windows 操作系统中继承了某些帐户策略的实行方式，而其它一些策略则是在 InTouch HMI 中实行。密码策略（如最大与最小密码寿命，以及最小密码长度）则是由操作系统实行的。

在安装期间使用的用户名会作为操作系统的一部分发挥作用。Windows 域必须设置所需的帐户策略，以实行这些标准。InTouch HMI 实行不活动超时时段策略。

在基于操作系统身份验证的方法中，可以从与 Windows“网络域”或“工作组”关联的用户列表中选择用户名。每个用户名都指定有一个访问级别，确定该用户对给定活动所拥有的权限。操作系统在内部管理密码，因此 InTouch HMI 不在存放应用程序的节点上存储密码。

基于操作系统的安全性使用 InTouch AddPermission() 脚本函数来定义与维护一份用户及其相应访问级别的列表。此列表在执行 Addpermission() 调用之后创建，并写入磁盘。包含身份验证详细信息的文件不会复制到 NAD 客户端节点上。

通过执行 WindowViewer 特别菜单中安全性下的登录菜单命令（如果显示特别菜单），操作员可以登录到应用程序；或者，您也可以使用链接到内部安全性标记的触控输入对象来创建自定义的登录窗口。

用于在应用程序上设立安全机制的命令位于 WindowMaker 与 WindowViewer 中安全性下。这些安全性命令用于登录与注销应用程序、更改密码，以及配置有效用户名、密码、访问级别的列表。

例如，通过指定登录的操作员的访问级别必须大于 2000，可以控制对某个窗口的访问、某个对象的可见性，等等。

使用基于 ArchestrA 的安全性

将某个节点配置成使用 ArchestrA 安全性时，InTouch HMI 使用 Application Server 中的方法与对话框来执行登录与注销操作。用户在 Application Server 的 Galaxy“储备库”节点上配置。如需详细信息，请参阅 Application Server 文档。

ArchestrA 安全性使您可以轻松地定义用户并指定允许执行的操作。这些安全权限是使用自动化对象基于用户可以执行的操作来定义的。基本方法包含以下步骤：

1. 定义安全性模型。
2. 根据安全模型组织整理要保护的自动化对象。
3. 根据安全性模型定义用户。

系统管理员通过创建相应的用户配置文件来定义系统用户。然后，通过从安全性模型中预定义的用户角色列表中进行选择，系统管理员可以将一个或多个角色指定给每个用户。

如果在基于 ArchestrA 的安全性模式下使用 InTouch，口令的最大字符数是 31。

InTouchView 用户通过基于口令的登录帐户进行正常的身份验证。

使用智能卡进行身份验证

智能卡是一种包含嵌入式集成电路的袖珍卡。该卡具有用于数据的安全存储，包括私钥和公钥证书。卡持有人通过个人标识号 (PIN) 进行身份验证，可以有权仅访问卡上的特定数据。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

可以配置 InTouch 应用程序以支持将智能卡用于用户身份验证。可以将智能卡证书和关联 PIN 号用于身份验证，而不是使用需要提供用户名、口令和域的应用程序。还可以选择使用名称、口令和域而不是智能卡进行登录。

需要用户身份验证的操作（如登录或安全/验证写入）也可以利用智能卡身份验证。如需有关详细信息，请参阅使用安全和验证写入。

设置智能卡身份验证

必须执行以下操作来设置智能卡身份验证：

- 将 InTouch 应用程序配置成使用 InTouch OS 或 ArchestrA OS 安全性。ArchestrA 安全性可以基于用户，也可以基于组。可使用 System Platform IDE 配置 ArchestrA 安全性。如需有关详细信息，请参阅 IDE 文档。
- 针对您的网络将 WindowViewer 计算机加入正确的域中。
- 在 WindowMaker 中，为 InTouch 应用程序启用智能卡身份验证。如需有关详细信息，请参阅在 WindowMaker 中启用智能卡身份验证。
- 为您将在其中使用智能卡的域配置智能卡。
- 在 WindowViewer 计算机上安装卡驱动程序。智能卡及其驱动程序是专门用于硬件的。如需有关安装与设置智能卡读取器的信息，请参阅针对您的特定读取器的文档。
- 将智能卡读取器连接到 WindowViewer 计算机的合适端口。如需有关说明，请参阅智能卡附带的文档。
  - 需要多个智能卡来执行验证写入函数（这些功能涉及多个用户）。
  - 要将智能卡用于“终端服务器”和 RDP 客户端，那么必须将智能卡读取器附加到客户端系统中以启用智能卡身份验证。要将智能卡读取器连接到使用 RDP 的“终端服务器”，您需要确保 RDP 客户端连接设置启用了本地设备和资源下的智能卡选项。

在 WindowMaker 中启用智能卡身份验证

您必须在 WindowMaker 中启用智能卡身份验证，然后才能使用智能卡读取器进行身份验证。

配置“智能卡”选项

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择安全性。
3. 从可用选项中选择一种安全性类型：**None**、 **InTouch**、**OS** 和 **ArchestrA**。
   - 如果选择 **ArchestrA**，请确保已使用 IDE 配置 ArchestrA OS 安全性（基于操作系统用户或基于操作系统用户组）。
   - 如果您选择 **OS** 或 **ArchestrA** 安全性类型，则会启用智能卡身份验证复选框。选中该复选框以启用智能卡身份验证。

通过智能卡登录

您可以使用智能卡登录 InTouch WindowViewer。要使用智能卡登录 InTouch 应用程序，您必须拥有一个已启用智能卡身份验证的应用程序。

您的智能卡必须至少拥有一个在您的域中配置的证书。必须将智能卡读取器附加到运行 WindowViewer 的计算机。您需要输入所使用的智能卡的 PIN。

如果在您尝试登录时未在读取器中检测到智能卡，则会提示您改为使用您的用户名和密码进行身份验证。

可以使用智能卡对安全和验证数据写入进行身份验证。如需有关详细信息，请参阅使用安全和验证写入。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

通过智能卡登录

1. 运行 WindowViewer。
2. 插入您的智能卡（如果尚未插入）。
3. 在特别菜单上，指向安全性，然后单击登录。

   此时会出现“登录”对话框。

   如果您插入了智能卡，则您的登录证书（域和用户名）会显示在对话框中。

   如果从 WindowViewer 中的脚本执行 LogonCurrentUser() 或 PostLogonDialog() 函数，则也会出现智能卡登录对话框。这些函数仅用于 InTouch 脚本，无法用于 ArchestrA 客户端脚本。

4. 在 **PIN** 框中输入所使用的智能卡的 PIN。

   如果智能卡不可用，则系统会提示您使用您的用户 ID 和密码登录。

5. 选择确定。这样，您就登录到 WindowViewer 了。

注意：在以智能卡用户身份登录后，您必须将智能卡留在智能卡读取器中。如果您取出智能卡，那么系统会将您注销。

使用安全和验证写入

可以配置 InTouch 应用程序，以便操作员可以向使用特定安全性分类配置的 Galaxy 属性写入数据：

- “安全写入”分类要求运行时操作员输入其凭证才能完成写入操作。
- “验证写入”分类需要两个签名。如果提供了合适凭证，则操作员可以写入数据，但是还需要附加验证者授权才能完成写入操作。

安全和验证写入具有以下要求：

- 必须为 Galaxy 启用安全性。
- 必须为 InTouch 应用程序启用 ArchestrA 安全性。
- 运行时操作员必须拥有在 Galaxy 中配置的合适操作权限：
  - 操作员必须拥有“可以修改操作属性”操作权限才能执行安全写入或验证写入。
  - 验证者必须拥有“可以验证写入”操作权限才能确认验证写入。

无论是谁当前作为 InTouch 应用程序的运行时用户登录，安全或验证写入都始终需要用户身份验证。即使您不是已登录的用户，您也可以修改使用安全或验证写入安全性分类配置的属性。这不会影响已登录用户的会话。

注意：对于启用了安全性并迁移到 Application Server 3.5 版的 Galaxy，则“可以修改操作属性”操作权限设置会复制到“可以验证写入”属性。从 Application Server 3.5 开始，Galaxy 在缺省条件下会禁用“可以验证写入”操作权限设置。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

在 InTouch Tag Viewer 中，运行时用户只能写入引用 Application Server 属性的间接标记。

可以使用智能卡对安全和验证数据写入进行身份验证。如需有关详细信息，请参阅使用安全和验证写入。

执行安全写入

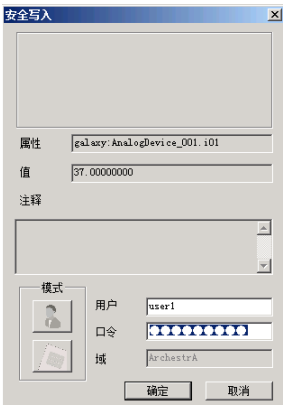
如果您尝试为使用安全写入安全性分类配置的 Application Server Galaxy 属性修改值，则必须使用有效安全性帐户（域名、用户名和密码）或智能卡对自己进行身份验证。仅当智能卡读取器附加到 WindowViewer 计算机时，您才能使用“智能卡”选项。

您必须拥有 Galaxy 中的“可以修改操作属性”操作权限才能执行安全写入。

您针对安全写入的身份验证不会影响当前已登录用户的会话。如果您之前使用智能卡登录，则必须重新对自己进行身份验证。

执行安全写入

1. 尝试修改使用安全写入安全性分类配置的属性的值。此时出现安全写入对话框。如果无智能卡可用，则会禁用模式按钮。



2. 通过从预定义注释列表中进行选择或在注释文本框中输入注释，来为写入操作添加注释。注释限制为 200 个字符。

   可以使用 SignedWrite() 脚本函数预定义注释列表，也可以在注释文本框中输入新注释。仅当使用 SignedWrite() 脚本函数时，才能访问预定义注释列表。

3. 如果您使用网络用户帐户进行身份验证，则会显示该用户帐户选项。

   执行以下操作。

   a. 在用户名框中，输入您的用户名。缺省情况下会显示当前已登录用户的名称。如果当前还没有任何用户登录，则该框为空。

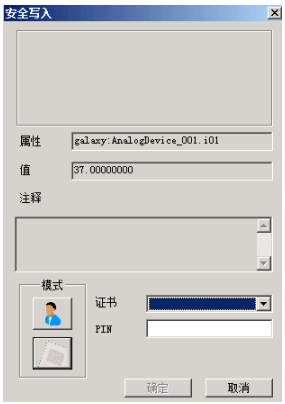
   b. 在密码框中，输入与用户名关联的密码。

   c. 在域框中，键入域名。

   d. 选择确定。

4. 如果您使用智能卡进行身份验证，则会出现“智能卡”选项。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



执行以下操作来使用智能卡进行身份验证。

a. 在**证书**列表中，选择您的智能卡证书。证书列表会以“域名/用户名”的形式显示。要使证书出现在该列表中，智能卡当前必须插入到附加到计算机的读取器中。缺省情况下会显示当前已登录用户的证书。如果您在**安全写入对话**框打开时插入或移除卡，则证书列表会自动更新。

b. 在 **PIN** 框中输入所使用的智能卡的 PIN。

c. 选择 **确定**。

当智能卡存在时，如果您要改为使用您的名称、密码和域进行身份验证，请选择“模式”按钮。转到第 3 步。

执行验证写入

如果您尝试为使用验证写入安全性分类配置的 Application Server Galaxy 属性修改值，则必须使用有效安全性帐户（域名、用户名和密码）或智能卡对自己进行身份验证。该写入还必须由另一个人员进行验证。

- 操作员必须拥有“可以修改操作属性”操作权限才能执行验证写入。
- 验证者还必须拥有“可以验证写入”操作权限才能确认验证写入。

您针对验证写入的身份验证不会影响当前已登录用户的会话。

仅当智能卡读取器附加到 WindowViewer 计算机时，才能使用智能卡选项。可以使用智能卡以操作员或验证者的身份或两者进行登录，但是操作员和验证者必须是两个不同的人员。如果您之前使用智能卡登录，则必须重新对自己进行身份验证。

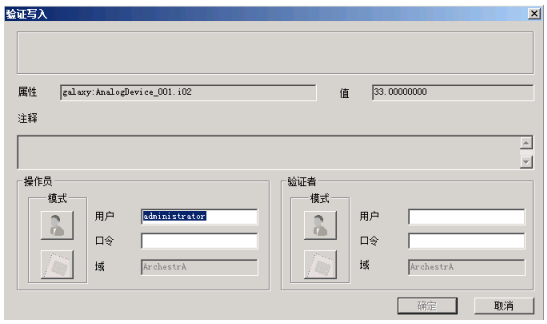
您具有以下选项：

- 可以使用两个智能卡读取器和两个智能卡。
- 如果您只有一个智能卡读取器可用，则可以将一个智能卡读取器及一个智能卡用于操作员或验证者。当操作员使用证书号和 PIN 登录时，验证者需要使用用户名和密码登录，反之亦然。
- 可以对操作员和验证者都使用用户名和密码身份验证。

执行验证写入

1. 尝试修改使用**验证写入**安全性分类配置的属性的值。

   此时出现**验证写入对话**框。如果无智能卡可用，则会禁用**模式**按钮。



2. 通过从预定义注释列表中进行选择或在注释框中输入自己的注释，来为写入操作添加注释。注释限制为 200 个字符。

   仅当使用 SignedWrite() 脚本函数时，才能访问预定义注释列表。

3. 如果您使用网络用户帐户进行身份验证，则会显示该用户帐户选项。

   执行以下操作。

   a. 在用户名框中，输入您的用户名。缺省情况下会显示当前已登录用户的名称。如果当前还没有任何用户登录，则该框为空。

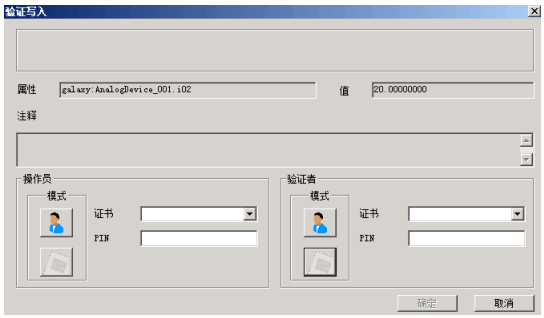
   b. 在密码框中，输入与用户名关联的密码。

   c. 在域框中，键入域名。

   d. 选择确定。

   e. 要改为使用智能卡进行身份验证，请选择证书按钮。转到第 4 步。

4. 如果您使用智能卡进行身份验证，则会显示智能卡选项。



   执行以下操作来使用智能卡进行身份验证。

   a. 在证书列表中，选择您的智能卡证书。证书列表会以“域名/用户名”的形式显示。要使证书出现在该列表中，智能卡当前必须插入到附加到计算机的读取器中。缺省条件下会显示当前已登录用户的证书（如果用户使用智能卡登录）。如果您在安全写入对话框打开时插入或移除卡，则证书列表会自动更新。

   b. 在 PIN 框中键入所使用的智能卡的 PIN。

   c. 选择确定。

   d. 当智能卡存在时，如果您要改为使用您的名称、密码和域进行身份验证，请选择“模式”按钮。转到第 3 步。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

自定义“安全写入”/“验证写入”对话框

可以使用 SignedWrite() 脚本函数在安全写入或验证写入对话框中配置以下内容：

- 显示原因消息
- 填充预定义注释列表

- 允许在注释文本框中进行编辑

如需有关 SignedWrite() 函数以及如何使用该函数的信息（包括语法、参数和详细示例），请参阅《工业图形编辑器用户指南》。

在运行时使用 SignedWrite() 函数

可以使用 SignedWrite() 函数直接向需要安全或验证写入签名的属性指定值。

当使用安全或验证写入安全性分类配置某个值和修改该值时，会出现安全写入或验证写入对话框。出现在安全写入或验证写入对话框中的内容因修改值的方式而异。

- 如果使用 SignedWrite() 函数修改值，则安全写入或验证写入对话框会基于函数中的参数设置显示选项。
- 如果通过用户操作修改值，则原因消息区域会显示域属性说明（如果存在）。如果属性不是域属性或没有说明，则原因消息区域会显示属性所述的 ApplicationObject 的说明。预定义注释列表不可用。

当您尝试在 InTouch WindowViewer 中修改属性的值时，您可以查看安全写入或验证写入对话框中的原因消息。对话框会显示属性名称以及要向属性写入的新值。

注意：原因说明以及预定义注释列表和框仅显示在 InTouch WindowViewer 中（而不在标记查看器中）的安全写入或验证写入对话框中。

管理用户并设置其授权级别

要给需要使用 InTouch HMI 的一组用户设施安全机制，必须：

- 给每个用户指定用户名与口令身份验证凭证。
- 给每个用户指定 InTouch 授权级别（访问级别）。

配置 InTouch 安全性身份验证与授权

对于每个操作员，您需要指定用户名、密码以及访问级别。

None 与 **Administrator** 名称是保留名，只有 Administrator 的密码可以更改（缺省值是 wonderware）。给应用程序配置用户名之后，更改 Administrator 密码。Administrator 的缺省访问级别 (9999) 是最高的，允许访问所有的 InTouch 功能，包括配置用户命令。

您也可以将一个“用户输入 - 离散”按钮链接到 $ConfigureUsers 标记，以允许经过授权且访问级别大于或等于 9000 的操作员访问配置用户对话框来编辑安全性用户名列表。操作员选择该按钮时，$ConfigureUsers 标记的值设置为 1，此时出现配置用户对话框。操作员关闭对话框时，系统会将值重置为 0。这是一个只用于写入操作的系统离散标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

注意：$ConfigureUsers 标记仅在安全性类型设置为 InTouch 时起作用。对于基于 ArchestrA 与基于操作系统的安全性，它不起作用。

为应用程序操作员配置安全性

1. 打开 WindowMaker。
2. 在文件菜单上，选择配置，然后选择安全性。

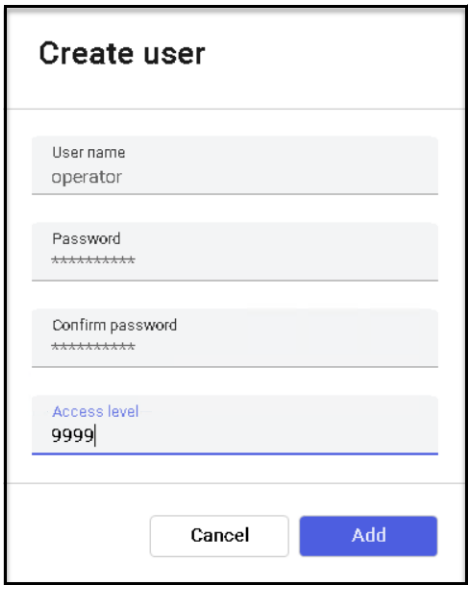
   此时出现“安全性配置”屏幕。
3. 将安全性类型选择为 **InTouch**。

   此时出现“登录”对话框。
4. 输入用户名与密码。

   注意：只有具有管理员帐户或访问级别高于 9000 的用户才能为 InTouch 配置操作员。

   此时会启用配置用户部分。
5. 选择”添加“图标。

   此时出现创建用户对话框。



6. 要添加安全性帐户，请执行以下操作：
   a. 在用户名框中，输入要指定给操作员的名称。
   b. 在密码框中，输入操作员密码，最多可以包含 29 个字符。
   c. 在访问级别框中，输入操作员的访问级别（最低 = 0 到最高 = 9999）。
   d. 选择添加将用户名添加到 InTouch 安全性列表。
7. 要更改用户名，请从配置用户网格选择条目。
8. 执行所需的更改，然后选择更新。
9. 要删除用户名，请从配置用户网格选择条目，然后选择删除。
10. 选择确定。

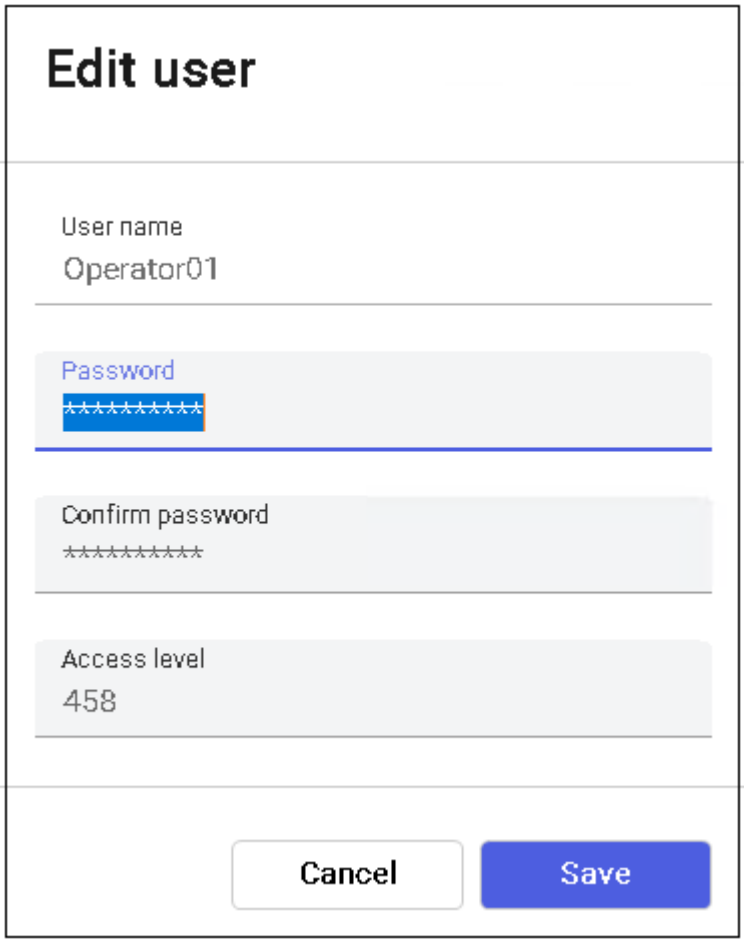
© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

更改 InTouch 操作员密码

操作员可以更改其 WindowMaker 或 WindowViewer 的密码。

更改 WindowMaker 中的操作员密码

1. 在文件菜单上，选择配置，然后选择安全性。
   此时出现安全性配置屏幕。
2. 选择用户的相关安全性类型。
   系统可能会要求您登录。这是为了确保只有管理员或访问级别高于 9000 的用户帐户才能更改密码。
3. 输入用户名与密码。
4. 从配置用户列表中，选择需要更改密码的用户帐户，然后选择编辑。
   此时出现编辑用户屏幕。



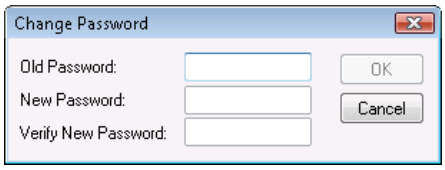
5. 在密码字段中，输入密码。
6. 在确认密码字段中，再次输入密码进行确认。
7. 选择保存。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

更改 WindowViewer 中的操作员密码

1. 启动 WindowViewer。
2. 在特别菜单上，选择安全性，然后选择更改密码。

   此时出现更改密码对话框。



3. 配置密码。执行以下操作：
   - 在旧密码框中，输入旧密码。
   - 在新密码框中，输入新密码。
   - 在校验新密码框中，再次输入新密码。
4. 选择确定。

使用离散按钮更改操作员密码

如果不打算在 WindowViewer 中显示特别菜单，则可以创建一个离散按钮，并将它链接到 $ChangePassword 内部标记。$ChangePassword 标记的值设置为 1 时，出现更改密码对话框。操作员然后可以更改自己的密码。操作员关闭对话框时，系统将 $ChangePassword 的值重置为 0。这是一个只用于写入操作的系统离散标记。

设置基于操作系统的身份验证与授权

基于操作系统的安全性根据经过授权的 Windows 用户组的列表来验证 InTouch 用户。您可以在本地计算机或 Active Directory 服务器上创建 Windows 用户组。您必须通过将 Windows 用户添加到特定的组来将它们与这些组关联起来。如需有关创建用户组的详细信息，请参阅 Windows 操作系统文档。

然后通过在脚本中使用 AddPermission() 函数，您可以将 InTouch 访问级别指定给 Windows 组。AddPermission() 函数通常在应用程序启动时调用，这样在用户准备登录时，WindowViewer 便可以识别所有经授权的用户组。

通常在创建 InTouch 应用程序之后，立即指定基于操作系统的安全性。

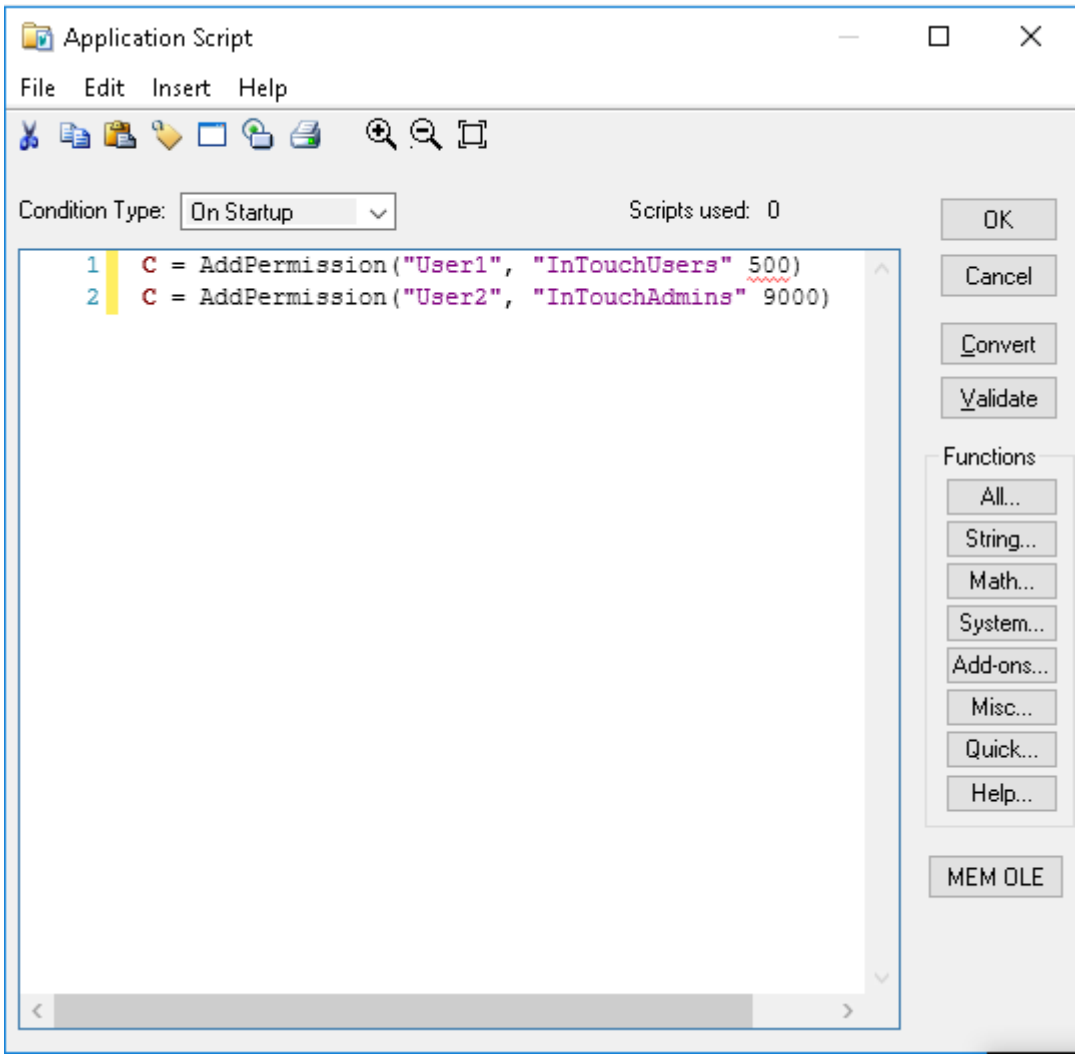
将 InTouch 应用程序配置成使用操作系统身份验证和 InTouch 内部授权之后，“特别”菜单的“安全性”下的“更改密码”、“登录”、“配置用户”和“注销”等命令会变为不可用。

设置基于操作系统的安全性并配置访问级别

1. 在脚本窗格上，选择应用程序。

   此时出现应用程序脚本对话框。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



2. 在**条件类型**列表中，**单击启动时**。
3. 使用 AddPermission() 函数指定**组**名与相**应**的**访问级别**。AddPermission() 的参数是操作系**统**（或域）、**组**名以及**访问级别**。
4. **选择确定**。

设置基于 ArchestrA 的安全性

ArchestrA 安全性系**统**是一个全局功能，它**应**用于 Galaxy 数据**库**中的**每**个**对**象。它是用**户**同 Galaxy 中的**对**象与函数之**间**的一个基于**关**系的系**统**。此系**统**是基于安全性角色（配置、系**统**管理以及运行**时权**限）与安全性**组**，以确定特定安全性角色的**对**象**级**运行**时权**限。安全性系**统**的配置在“集成**开发环**境”（Integrated Development Environment，**简**称 IDE）中完成，并通**过**其自身的**编辑**器**应**用于**每**个**对**象。

将 InTouch **应**用程序配置成使用 ArchestrA 身份**验证**之后，WindowMaker 中的**更改密码**、**登录**、**配置用户**和**注销**命令会**变为**不可用。

设置基于 ArchestrA 的安全性

1. 在 WindowMaker 中打**开**窗口。
2. 在**文件**菜**单**上，**选择配置**，然后**选择安全性**。

此时出现安全性配置屏幕。

3. 从安全性类型选项中选择 **ArchestrA**。

AddPermission() 函数

将特定的 InTouch 访问级别指定给本地系统或域上给定的用户组。在调用 AddPermission() 函数之后，属于该组的用户登录到 InTouch HMI 时，该用户获得指定的访问级别。

类别

安全性

语法

```
DiscreteTag=AddPermission( "Domain", "Group", AccessLevel);
```

参数

域
组所在的域或本地计算机的名称。

组
Windows 用户组。

AccessLevel
您希望与给定的组关联的 InTouch 访问级别。

附注

仅对于操作系统安全性有效。调用此函数时，将在指定的域或工作组中检查是否存在指定的组。如果成功，则返回 TRUE，指定的“访问级别”与组关联，以便在随后的用户登录操作中使用。在所有其它情况下（即，如果为任何一个参数指定了无效的值），则返回 FALSE。

此函数通常配置成在应用程序启动时运行。它不影响当前登录的用户。只有在成功调用 AddPermission() 函数之后登录的用户，才会获得与自己的组关联的访问级别。

示例

```
DiscreteTag=AddPermission( "corporate_hq", "InTouchAdmins", 9000);
DiscreteTag=AddPermission( "johns01", "InTouchUsers", 5000);
```

Operations Control 连接体验

AddPermission() 方法在 Operations Control 连接体验中仅接受两个参数：

- CONNECT 组
- 访问级别

Operations Control 连接体验中 AddPermission() 的脚本函数：

```
DiscreteTag=AddPermission("", "CONNECT group", AccessLevel);
```

如果运行时用户是 CONNECT 中多个组的成员，则访问级别将由具有最高访问级别的组确定。

另请参阅

PostLogonDialog(), InvisibleVerifyCredentials(), IsAssignedRole(), AttemptInvisibleLogon(), QueryGroupMembership()

ChangePassword() 函数

显示改变口令对话框，允许登录的操作员更改自己的口令。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

安全性

语法

```
[Result=]ChangePassword();
```

返回值

返回以下整数值之一：

0 = 已按过“取消”。

1 = 已按过“确定”。

附注

如果操作员使用触摸屏，则操作员可以使用字母数字键盘输入新口令。

示例

以下脚本可以放到按钮上，或是从条件脚本或数据改变脚本中调用。

```
Errmsg=ChangePassword();
```

$AccessLevel 系统标记

定义当前登录的用户的访问级别。

类别

安全性

用法

$AccessLevel

附注

此标记的值由 InTouch HMI 中指定给当前登录的用户的安全性配置文件的访问级别确定。在 WindowViewer 中，此配置文件可以使用配置用户菜单来访问。

$AccessLevel 的实际数值对于 WindowViewer 没有任何意义，除非是 9000 或更大的值，这代表管理员权限，并且会在 WindowViewer 中启用安全性菜单。$AccessLevel 系统标记可用于在系统中进一步自定义安全性。

数据类型

整型（只读）

有效值

0 到 9999

示例

以下语句用在一个可视化链接中，根据登录的用户的访问级别来确定是否让某个对象（如按钮）变为可见：

```
$AccessLevel >= 2000;
(对象可以有一个关联的“禁用”链接和基于 $AccessLevel 的表达式。)
$AccessLevel < 5411;
IF $AccessLevel <=500 THEN
Show "Access Denied"; {popup window denying access}
ELSE
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

```
Show "Access Granted"; {popup window granting access}
ENDIF;
```

另请参阅

$Operator、$OperatorEntered、$PasswordEntered、$ConfigureUsers

$ChangePassword 系统标记

显示改变口令对话框。

类别

安全性

用法

```
$ChangePassword
```

附注

将此值设置为 1 以显示改变口令对话框。对话框关闭时, $ChangePassword 系统标记的值重置为 0。如果不将此系统标记的值设置为 1，则结果未定义。

数据类型

离散（只写）。

有效值

1

示例

您可以创建一个打开改变口令对话框的离散按钮。给该按钮指定一个选择了“置位”选项的离散按钮链接。按下该按钮时，$ChangePassword 系统标记设置为 1，并打开改变口令对话框。

另请参阅

$AccessLevel、$OperatorEntered、$PasswordEntered、$Operator、$ConfigureUsers

$ConfigureUsers 系统标记

显示配置用户对话框。

类别

安全性

用法

```
$ConfigureUsers
```

附注

此函数仅对于 InTouch 安全性有效。

将值设置为 1 以打开配置用户对话框。

对话框关闭时，此系统标记的值重置为 0。如果不将此系统标记的值设置为 1，则结果未定义。

用户必须有大于 9000 的 $AccessLevel 才可以显示此对话框。

数据类型

离散（只写）。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

有效值

1

示例

您可以创建一个打开配置用户对话框的离散按钮。给该按钮指定一个选择了“置位”选项的离散按钮链接。按下按钮时，$ConfigureUsers 系统标记设置为 1，配置用户对话框打开。

另请参阅

$Operator、$OperatorEntered、$ChangePassword、$PasswordEntered、$AccessLevel

登录与注销

根据用于保护应用程序的安全性的类型，从 InTouch 应用程序进行的登录与注销也存在着差异。

登录到采用 InTouch 安全机制的应用程序

如果登录信息不正确或是无效，则出现一条消息，指出登录尝试不成功。

如果登录成功，则 $AccessLevel 系统标记设置为与用户（在 InTouch 安全性用户列表中）关联的预定义值。

注意：您也可以使用 PostLogonDialog() 函数显示登录对话框。如需详细信息，请参阅 PostLogonDialog() 函数。

登录到应用程序

1. 在文件菜单上，选择配置，然后选择安全性。
2. 选择相关安全性类型。

   此时出现登录对话框。
3. 在名称框中，输入您的用户名。
4. 在密码框中，输入您的密码。
5. 选择登录。

登录到采用操作系统安全机制的应用程序

用户登录到 InTouch 应用程序时，出现一个要求提供以下信息的对话框：

- 用户名
- 密码
- 域或本地计算机名

域/用户名组合传递给操作系统以验证用户的凭证。此时，不管是否启用操作系统缓冲区，都会尝试一次登录。如果没有缓冲区（例如，由于网络中断），因而用户无法登录，但先前使用缓冲区时该用户曾通过身份验证，则会从本地的 InTouch 缓冲区获取该用户的全名与访问级别。

如果成功通过所有安全性检查，该用户会被视为已登录到 InTouch HMI，相关的数据结构（如 $Operator）会进行更新。否则显示一条错误消息。

如果操作员从未成功登录过，并且域不可用，则登录尝试会失败。此时 InTouch HMI 将一个系统事件记录到错误日志中。

如果口令过期，则显示一条错误消息。在操作员单击确定之后，出现更改过期的口令对话框，这样操作员便可以更改口令，并可以使用新口令再次尝试登录。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

登录到采用 ArchestrA 安全机制的应用程序

用户通常会通过输入有效的用户名与口令来登录和注销采用 ArchestrA 安全机制的 InTouch 应用程序。

如果 InTouch 应用程序已配置成 ArchestrA 安全性 "None"，则使用缺省用户的登录凭证，并且不会提示操作员登录。以下过程假设系统已经配置成 ArchestrA 身份验证模式，如"Galaxy"、"OS User based"（基于操作系统用户）、"OS Group based"（基于操作系统用户组）。

登录到采用 ArchestrA 安全机制的应用程序

1. 启动采用 ArchestrA 安全机制的 InTouch 应用程序。此时出现登录对话框。
2. 输入有效的用户名与口令。如果系统无法验证您的身份，则会再次提示您登录。

在系统验证您的登录凭证之后，会根据安全性模型中同您关联的角色/权限来授予对将来所有操作的访问权限。

从 InTouch 应用程序中注销

操作员在完成工作之后从 InTouch 应用程序中注销。您也可以将应用程序配置成在一定时间内未有任何动作时自动注销操作员。如需详细信息，请参阅配置不活动超时。

从应用程序中注销

1. 在文件菜单上，选择凭据。
2. 在 InTouch HMI WindowMaker“用户”部分，选择注销。

注意：如果您已在配置器的许可模式选项卡中选择 **Operations Control 连接体验**，那么凭据选项不可用。

创建自定义的登录窗口

如果 WindowViewer 中未显示特别菜单，则可以创建自定义的登录窗口让操作员登录到应用程序。

创建自定义的登录窗口

- 将 $OperatorEntered、$PasswordEntered、$OperatorDomainEntered 系统标记链接到用户输入对象，或在脚本中使用它们以设置用户名、口令及域名。这些标记是仅用于写入操作的内部消息型标记。

只有在安全模式基于操作系统时，才要求使用 $OperatorDomainEntered 标记。否则会忽略此标记。如果安全模式基于操作系统，并且 $OperatorDomainEntered 为空，则视为指向本地计算机。

将某个值写入 $PasswordEntered 系统标记时，会发生使用 $OperatorEntered、$PasswordEntered 及 $OperatorDomainEntered 系统标记值进行的登录尝试。如果仅将值写入 $OperatorEntered 或 $OperatorDomainEntered 系统标记，则不会发生登录。

如果输入有效，$AccessLevel 与 $Operator 内部标记会设置为它们的预定义值（在安全性用户列表中配置）。

您也可以将一个“用户输入 - 离散”按钮链接到 $ConfigureUsers 标记，以允许经过授权且访问级别大于或等于 9000 的操作员访问配置用户对话框来编辑安全性用户名列表。操作员选择该按钮时，$ConfigureUsers 标记的值设置为 1，此时出现配置用户对话框。操作员关闭对话框时，系统会将值重置为 0。（这是一个只用于写入操作的系统离散标记。）

注意：$ConfigureUsers 标记仅在安全性类型设置为 InTouch 时起作用。对基于 ArchestrA 的安全性，它不起作用。

PostLogonDialog() 函数

显示 InTouch 登录对话框并返回 TRUE。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

安全性

语法

```
DiscreteTag=PostLogonDialog();
```

示例

```
DiscreteTag=PostLogonDialog();
```

另请参阅

InvisibleVerifyCredentials()、AttemptInvisibleLogon()、IsAssignedRole()、QueryGroupMembership()、AddPermission()

LogonCurrentUser() 函数

使用当前登录到 Windows 操作系统的用户帐户登录 InTouch。

- 配置了操作系统安全性的 InTouch：用户登录到 WindowViewer。
- 配置了 ArchestrA 安全性的 InTouch：用户必须为基于 ArchestrA 操作系统用户安全性或基于 ArchestrA 操作系统组安全性的成员。
- 配置了基于 ArchestrA 操作系统用户安全性或 ArchestrA 操作系统组安全性的 InTouch 且该用户帐户已配置智能卡凭证：用户使用智能卡凭证登录。如果将智能卡从读取器中取出，则会注销该用户。

类别

安全性

语法

```
IntegerResult = LogonCurrentUser();
```

返回值

如果登录失败，则返回 -1，并且不更改指定给 $Operator、$OperatorName、$OperatorDomain 及 $AccessLevel 的值。

附注

此功能仅用于 InTouch 脚本，无法用于 ArchestrA 客户端脚本。

示例

```
IntegerResult = LogonCurrentUser();
```

另请参阅

AddPermission() 函数

AttemptInvisibleLogon() 函数

InvisibleVerifyCredentials() 函数

IsAssignedRole() 函数

PostLogonDialog() 函数

QueryGroupMembership() 函数

Logoff() 函数

从 InTouch 应用程序中注销用户。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

安全性（只写）

语法

```
DiscreteTag = LogOff();
```

附注

注销当前登录的用户，并且将当前用户状态设置为缺省的 none 操作员。

示例

```
DiscreteTag = LogOff();
```

另请参阅

PostLogonDialog()、InvisibleVerifyCredentials()、IsAssignedRole()、AttemptInvisibleLogon()、QueryGroupMembership()、AddPermission()

AttemptInvisibleLogon() 函数

AttemptInvisibleLogon() 函数可以用在脚本中，使用所提供的凭证将用户登录到 InTouch。此时不要求该用户输入口令或用户 ID。

类别

安全性

语法

```
 DiscreteTag=AttemptInvisibleLogon( "UserId", "Password", "Domain" );
```

参数

UserId
有效的用户帐户名。

密码
用户的口令。

域
用户所属的本地计算机、工作组或域的名称。只有在当前安全性类型是基于操作系统时，此参数才适用。

返回值

如果身份验证成功，则返回 TRUE。否则返回 FALSE。

附注

尝试使用所提供的凭证登录到 InTouch HMI。

- 如果登录尝试成功，则返回 TRUE，并且相应地更新 $OperatorDomain、$OperatorName、$AccessLevel 及 $Operator 等系统标记。
- 如果登录尝试失败，则返回 FALSE，并且当前登录的用户（如果存在）将继续是当前用户。

Domain 参数仅对基于操作系统的安全性是有效的。如果在使用 ArchestrA 安全模式, 并且如果 ArchestrA 安全性又在使用基于操作系统的安全性，则 *UserId* 参数应该包含带域名或计算机名的全限定用户名。

示例

安全性基于操作系统时：

```
DiscreteTag=AttemptInvisibleLogon("UserId", "Password", "Domain" );
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

安全性基于 InTouch 或基于 ArchestrA 时：

```
DiscreteTag=AttemptInvisibleLogon("UserId", "Password", "" );
```

另请参阅

AddPermission() 函数

InvisibleVerifyCredentials() 函数

IsAssignedRole() 函数

PostLogonDialog() 函数

QueryGroupMembership() 函数

AttemptInvisibleLogonEx() 函数

AttemptInvisibleLogonEx() 函数可以用在脚本中，使用“凭据管理器”中存储的凭据将用户登录到 InTouch。此时不要求该用户输入口令或用户 ID。

类别

安全性

语法

```
DiscreteTag=AttemptInvisibleLogonEx("Credential Name");
```

参数

凭证名称
凭据管理器中存储的凭据的名称。对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。

返回值

如果身份验证成功，则返回 TRUE。否则返回 FALSE。

附注

尝试使用“凭据管理器”中保存的凭据登录到 InTouch HMI。对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。

- 如果登录尝试成功，则返回 TRUE，并且相应地更新 $OperatorDomain、$OperatorName、$AccessLevel 及 $Operator 等系统标记。
- 如果登录尝试失败，则返回 FALSE，并且当前登录的用户（如果存在）将继续是当前用户。

Domain 参数仅对基于操作系统的安全性是有效的。如果在使用 ArchestrA 安全模式，并且如果 ArchestrA 安全性又在使用基于操作系统的安全性，则 *UserId* 参数应该包含带域名或计算机名的全限定用户名。

示例

安全性基于操作系统时：

```
DiscreteTag=AttemptInvisibleLogonEx("TestCredentialName01");
```

另请参阅

PostLogonDialog()、InvisibleVerifyCredentials()、IsAssignedRole()、QueryGroupMembership()、AddPermission()

$OperatorEntered 系统标记

用于输入有效的用户名。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

安全性

用法

```
$OperatorEntered
```

附注

您可以使用此标记名来创建自定义的登录窗口。您可以将触控输入对象与/或 QuickScript 链接到此标记，以设置登录帐户的用户名。

注意：$OperatorEntered 系统标记有效时，$AccessLevel 与 $Operator 系统标记设置为它们的预定义值。

数据类型

消息（只写）。

另请参阅

$AccessLevel、$Operator、$PasswordEntered、$ChangePassword、$ConfigureUsers

$PasswordEntered 系统标记

用于输入有效的口令。

类别

安全性

用法

```
$PasswordEntered
```

附注

$PasswordEntered 系统标记总是读为空字符串。绑定到此系统标记的显示链接总是为空。因为此标记总是返回一个空字符串，所以数据改变脚本从不触发此标记。您可以使用此标记名来创建自定义的登录窗口。您可以将触控输入对象与/或脚本链接到此标记，以设置用户的口令。

注意：$PasswordEntered 有效时，$AccessLevel 与 $Operator 系统标记设置为它们的预定义值。

数据类型

消息（只写）。

另请参阅

$AccessLevel、$Operator、$OperatorEntered、$ChangePassword 和 $ConfigureUsers

$OperatorDomainEntered 系统标记

域名由操作员输入。

类别

安全性

附注

只要 $PasswordEntered 标记发生更改，便会尝试进行登录，而不显示任何对话框。登录尝试将 $*Entered 标记用于输入用户名，将 $OperatorDomainEntered 的字符串值用作域名（只有在当前模式是基于操作系统的安全性时才使用）。如果模式并非基于操作系统，则忽略此标记。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

字符串型

示例

```
$OperatorEntered == "john";
$OperatorDomainEntered == "Corporate_HQ";
$PasswordEntered == "password";
```

另请参阅

$Operator

基于操作员或访问级别启用与禁用功能

在给应用程序采用安全机制之后，可以将 $AccessLevel 与 $Operator 安全性标记用在按钮、动画链接表达式或 QuickScript 中，以控制是否允许登录的操作员执行特定的应用程序功能。

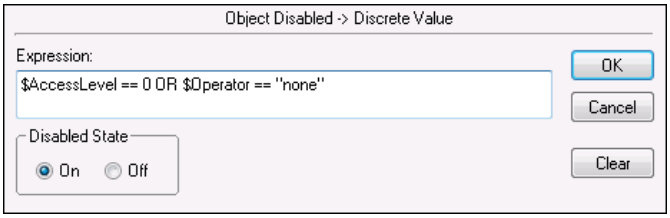
例如，要根据登录的用户的访问级别来显示对象，则可以在可见性动画链接表达式中使用以下语句：

```
$AccessLevel >= 2000;
```

或者也可以使用 IF 语句来限定脚本：

```
IF $Operator == "DayShift" THEN
   Show "Control Panel Window";
   {and other lines that only execute for the DayShift Operator}
ENDIF;
```

通过使用失效动画链接，也可以基于内部安全性标记的值来控制对象的触动功能。例如：



通过使用此表达式，如果无人登录，则该对象或按钮会被禁用，以防意外操作。

InvisibleVerifyCredentials() 函数

InvisibleVerifyCredentials() 函数可以用在同步 QuickScript 中，以验证给定用户的凭证，而不必将用户登录到 InTouch HMI。

类别

安全性

语法

```
AnalogTag=InvisibleVerifyCredentials( "UserId", "Password", "Domain" );
```

参数

UserId
作为本地计算机、工作组或域的一部分的 Windows 操作系统用户帐户名。

密码
帐户口令。

域

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

帐户的 Windows 域。

附注

如果提供的用户、口令及域组合有效，则以整数的形式返回与用户关联的相应访问级别。否则返回 -1。

注意：InvisibleVerifyCredentials() 函数必须从同步 QuickScript 中运行。如果从异步 QuickScript 中运行，此函数总是返回 -1。

此函数不更改当前登录的用户。Domain 参数仅对基于操作系统的安全性是有效的。如果使用 ArchestrA 安全模式，并且如果 ArchestrA 安全性又使用基于操作系统的安全性，则 UserId 参数应该包含带域名或计算机名的全限定用户名。

示例

```
AnalogTag=InvisibleVerifyCredentials( "john", "Password", "corporate_hq" );
```

另请参阅

AddPermission() 函数

AttemptInvisibleLogon() 函数

IsAssignedRole() 函数

PostLogonDialog() 函数

QueryGroupMembership() 函数

InvisibleVerifyCredentialsEx() 函数

InvisibleVerifyCredentialsEx() 函数可以用在同步 QuickScript 中，以验证给定用户的凭据，而不必将用户登录到 InTouch HMI。

类别

安全性

语法

```
AnalogTag=InvisibleVerifyCredentialsEx("Credential Name");
```

参数

凭证名称
凭据管理器中存储的凭据的名称。对于独立 InTouch 应用程序，从“应用程序管理器”中检索凭据。对于托管 InTouch 应用程序，从 Application Server 的“凭据管理器”中检索凭据。

附注

如果提供的用户、口令及域组合有效，则以整数的形式返回与用户关联的相应访问级别。否则返回 -1。

注意：InvisibleVerifyCredentialsEx() 函数必须从同步 QuickScript 中运行。如果从异步 QuickScript 中运行，此函数总是返回 -1。

此函数不更改当前登录的用户。Domain 参数仅对基于操作系统的安全性是有效的。如果使用 ArchestrA 安全模式，并且如果 ArchestrA 安全性又使用基于操作系统的安全性，则 UserId 参数应该包含带域名或计算机名的全限定用户名。

示例

```
AnalogTag=InvisibleVerifyCredentialsEx("john", "Password", "corporate_hq");
```

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

另请参阅

PostLogonDialog()、AttemptInvisibleLogon()、IsAssignedRole()、QueryGroupMembership()、AddPermission()

检索当前登录的操作员的有关信息

对于任何安全性系统，跟踪都是一项主要功能。您可以使用一组安全性系统标记来确定登录到 InTouch 应用程序的用户、用户登录的域，以及尝试登录的时间。

GetAccountStatus() 函数

返回用户口令过期之前的天数。

类别

安全性

语法

```
Result=GetAccountStatus(Domain, UserID);
```

参数

域
用户帐户所在的域或本地计算机的名称。

UserID
作为本地计算机、工作组或域的一部分的 Windows 用户帐户名。

返回值

此函数也返回以下值：

结果	描述
-1	帐户口令已过期
-2	帐户口令永不过期
-3	帐户已锁定
-4	帐户已禁用
-5	帐户信息无效

附注

此脚本函数用于基于操作系统的安全性。不要将此函数用于 ArchestrA 安全模式。

如果 GetAccountStatus() 函数用于 ArchestrA 安全性，则脚本会尝试直接从域控制器检索帐户信息。只要 ArchestrA Galaxy<1>“</1>储备库<2>”</2>使用相同的域的操作系统安全性<3>，</3>此函数便可以起作用<4>。</4>

示例

```
Status = GetAccountStatus("Corporate_HQ","Operator");
```

IsAssignedRole() 函数

确定当前登录的用户是不是特定用户角色的成员。仅适用于 ArchestrA 安全性。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

类别

安全性

语法

```
DiscreteTag=IsAssignedRole( "RoleName" );
```

参数

RoleName
与 Application Server 用户关联的角色。

附注

仅对 ArchestrA 安全模式有效，适用于当前登录的用户。如果用户当前已登录，并且拥有在 Galaxy IDE 中指定的 *RoleName* 角色，则返回 TRUE。否则返回 FALSE。

示例

```
DiscreteTag=IsAssignedRole( "Administrators" );
```

另请参阅

AttemptInvisibleLogon()、PostLogonDialog()、InvisibleVerifyCredentials()、QueryGroupMembership()、AddPermission()

QueryGroupMembership() 函数

确定当前登录的用户是不是特定用户组的成员。仅适用于操作系统安全性。

类别

安全性

语法

```
DiscreteTag=QueryGroupMembership( "Domain", "Group" );
```

参数

域
组所在的域或本地计算机的名称

组
组的名称。

附注

仅对操作系统安全模式有效，适用于当前登录的用户。如果用户当前已登录，并且是域上某个组的成员，则返回 TRUE。否则返回 FALSE。

QueryGroupMembership() 函数在基于操作系统的安全性中可以正常使用；对于基于 ArchestrA 的安全性，仅当 ArchestrA 安全性设置为基于操作系统的安全性时，它才能正常使用。

示例

```
DiscreteTag=QueryGroupMembership( "corporate_hq", "InTouchAdmins" );
DiscreteTag=QueryGroupMembership( "JohnS01", "InTouchUsers" );
```

另请参阅

PostLogonDialog()、InvisibleVerifyCredentials()、IsAssignedRole()、AttemptInvisibleLogon()、AddPermission()

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

$OperatorName 系统标记

如果使用基于操作系统或 ArchestrA 的身份验证并且某人登录后未注销，则包含操作员的全名。否则此标记包含登录的用户的名称（与 $Operator 标记的内容相同）。

类别

安全性

数据类型

字符串（只读）

示例

```
IF $OperatorName <> "" THEN
   {Configure some defaults}
ENDIF;
```

另请参阅

$Operator

$OperatorDomain 系统标记

根据使用的安全性，包含不同的值：

- 如果选择基于操作系统的安全性，并且操作员已成功登录，则 $OperatorDomain 标记包含登录时指定的域或节点名。
- 如果选择 ArchestrA 安全性，并且用户已登录，则 $OperatorDomain 包含 "ArchestrA"。
- 如果选择 InTouch 安全性，则 $OperatorDomain 标记包含字符串 "InTouch"。
- 如果选择 "None"，则它是一个空字符串 ("")。

类别

安全性

数据类型

字符串型

示例

```
IF $OperatorDomain == "PRODUCTION" THEN
   {Allow change to setpoint}
ELSE
   {Change denied}
ENDIF;
```

另请参阅

$Operator

$Operator 系统标记

包含登录的用户的登录名。

类别

安全性

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

数据类型

字符串型

$VerifiedUserName 系统标记

如果成功调用 InvisibleVerifyCredentials() 函数，并且如果安全性模式设置为基于操作系统或基于 ArchestrA Application Server 的安全性，则包含通过验证的用户的全名。如果调用失败，则此系统标记设置为空。

类别

安全性

用法

```
$VerifiedUserName
```

附注

$VerifiedUserName 系统标记更改时（调用 InvisibleVerifyCredentials() 函数时），生成一个事件。

数据类型

消息（只读）。

有效值

用户的全名。

示例

```
Tag = InvisibleVerifyCredentials( "john","password", "Plant_Floor");{ 如果调用成功，则
$VerifiedUserName 设置为 “John Smith” 并生成运算符事件。If the above call is not successful,
$VerifiedUserName is set to "".}
```

另请参阅

InvisibleVerifyCredentials()、$OperatorName、$Operator

安全系统标记与函数的摘要

下表显示在各种安全模式下您可以使用的安全系统标记与函数：

	InTouch 安全性	操作系统安全性	ArchestrA 安全性
$AccessLevel	是	是	是
$ChangePassword	是	是	是
$ConfigureUsers	是	否	否
$InactivityTimeout	是	是	是
$InactivityWarning	是	是	是
$Operator	是	是	是
$OperatorDomain	否	是	是*
$OperatorDomainEntered	否	是	是*

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

	InTouch 安全性	操作系统安全性	ArchestrA 安全性
$OperatorEntered	是	是	是
$OperatorName	是	是	是
$PasswordEntered	是	是	是
$VerifiedUserName	否	是	是
AddPermission()	否	是	否
AttemptInvisibleLogon()	是	是	是
AttemptInvisibleLogonEx()	是	是	是
ChangePassword()	是	否	否
EnableDisableKeys()	是	是	是
GetAccountStatus()	否	是	是*
InvisibleVerifyCredentials()	否	是	是*
InvisibleVerifyCredentialsEx()	否	是	是*
IsAssignedRole()	否	否	是
Logoff()	是	是	是
LogonCurrentUser()	否	是	是*
PostLogonDialog()	是	是	是
QueryGroupMembership()	否	是	是*

* ArchestrA 安全性基于操作系统用户或用户组时

允许非管理员用户使用的应用程序管理器操作

允许非管理员用户使用的操作是允许管理员使用的操作的一部分。限制非管理员用户对关键操作的访问将会防止发生安全威胁。下表列出了 InTouch HMI 应用程序管理器的每个选项卡下面的操作：

InTouch 选项卡

操作	非管理员
新建应用程序	否
启动 WindowMaker	否
启动 WindowViewer	是
DBLoad	否
DBDump	否

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

操作	非管理员
删除应用程序	否
重命名应用程序	否
应用程序属性	否
导出为模板	否
导入应用程序	否
OPC UA 服务器设置	是
查找应用程序	是
为 IoT 导出	是
上传到 CONNECT	是
为边缘设备配置用户	否
将标记数据发布为 AVEVA Insight 数据源	是
节点属性	
应用程序开发 > 在 WindowViewer 中启动以下应用程序作为服务	否
应用程序开发 > 允许网络应用程序开发	是
解决方法	是
内存设置	否
性能	否
刷新	是
视图	是
CONNECT 登录	是
更改缩略图	是
打开应用程序文件夹	是

Web 客户端选项卡

操作	非管理员
启用 Web 客户端	否
启动 Web 客户端	是
Web 客户端设置	
图形刷新率	否

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

操作	非管理员
报警刷新率	否
Web 客户端网站名称	否
显示标题	否
启用导航栏	否
允许匿名访问	否
AIM 注册设置	
使用 AIM 服务器作为身份验证服务器	否
标识服务器	不适用
用户名	否
Password	否
安全网关	否
允许将工业图形嵌入任何网站	否

将基于角色的安全性应用于应用程序文件夹

将基于角色的安全性应用于应用程序文件夹可确保只有授权用户才能访问应用程序文件夹中的资源并编辑应用程序。具有基于角色的安全性的应用程序以下称为安全应用程序。不具有基于角色的安全性的应用程序被称为不安全应用程序。即使是在导入或修改现有应用程序时，您也可以保护应用程序文件夹。

默认情况下，InTouch 应用程序会从其创建位置的父文件夹继承权限，并且不会应用其他权限。要限制对独立应用程序的应用程序文件夹的访问，请启用 Application Manager 的工具 > 节点属性 > 安全性下限制 **InTouchDevelopers** 和 **InTouchOperators** 组中的用户对独立 **InTouch** 应用程序的访问选项。在节点上启用安全功能后，将对 InTouch 视图应用程序应用严格的读写权限。

定义安全用户角色

安装 InTouch HMI 后，以下 InTouch 用户组将自动添加为本地用户组：

- InTouch 开发人员 - 属于该组的用户可以完全控制 InTouch 应用程序根文件夹。他们可以编辑和管理应用程序，并对整个应用程序具有读/写权限。
- InTouch 操作员 - 属于该组的用户对 InTouch 应用程序根文件夹具有有限的控制权。他们可以运行应用程序。他们对大多数文件具有只读访问权限，并且需要对少数文件具有读/写权限。

您可以限制仅“InTouchDevelopers”和“InTouchOperators”组中的用户才能访问 InTouch 应用程序。

分配域组

您可以将域组分配给“InTouchDevelopers”或“InTouchOperators”组。属于该域组的用户将具有开发人员或操作员级别的访问权限。

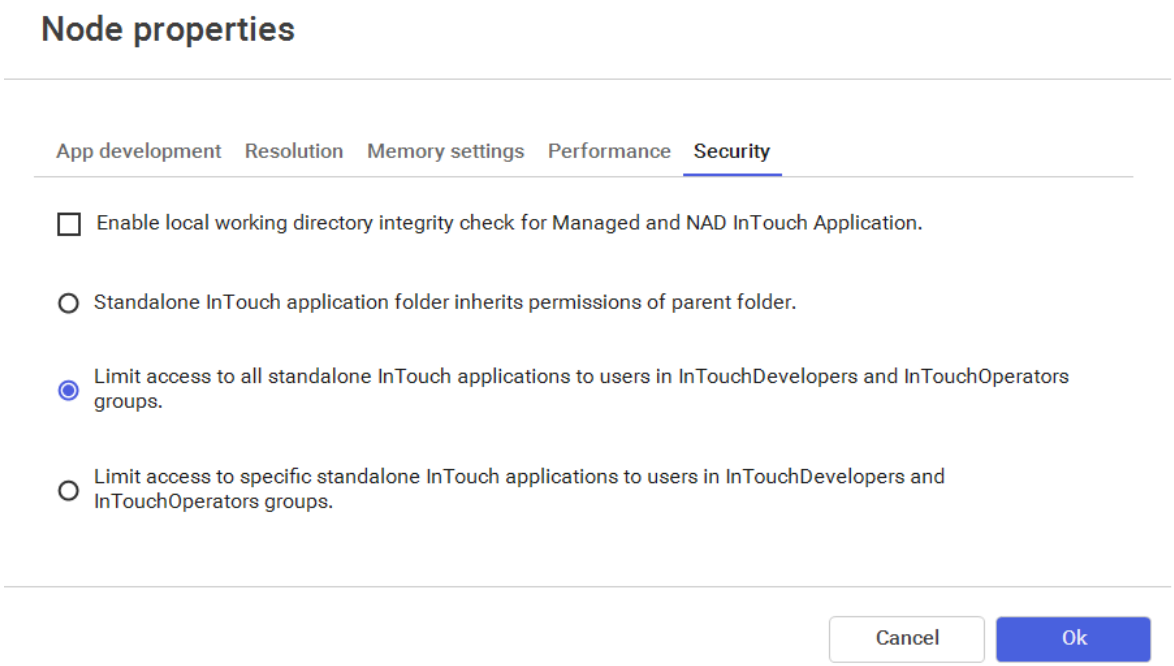
不支持将本地组分配给“InTouchDevelopers”或“InTouchOperators”组。建议将用户添加到“InTouchDevelopers”或“InTouchOperators”组。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

要启用 InTouch 应用程序文件夹的安全性

1. 以管理员身份启动“应用程序管理器”。
2. 在工具菜单中的工具选项卡上，单击安全性。

   此时出现安全性屏幕。
3. 选择限制 InTouchDevelopers 和 InTouchOperators 组中的用户对独立 InTouch 应用程序的访问为所有应用程序文件夹启用安全性，或选择限制 InTouchDevelopers 和 InTouchOperators 组中的用户对特定独立 InTouch 应用程序的访问仅为选定应用程序文件夹启用安全性。



4. 选择确定。

附加信息：

- 在此节点属性对话框中选择“确定”后，如果选择了按计算机或按应用程序进行保护，将对应用程序文件夹应用访问控制列表 (ACL) 。
  - 如果选择按计算机，将更新每个应用程序（包含其文件夹）的 ACL。
  - 如果选择按应用程序，则仅更新用户选择的应用程序。
- 不包括通用命名约定 (UNC) 路径或映射网络驱动器上的任何应用程序。
- 执行“查找应用程序”或“导入应用程序”时，如果在计算机级别启用了应用程序文件夹安全，将向新添加的应用程序文件夹应用安全措施。
- 可以在以下路径的文件中使用条目“SecureApplicationFolder”找到此配置的设置。

  C:\ProgramData\wonderware\InTouch\CONFIGURATION.ini

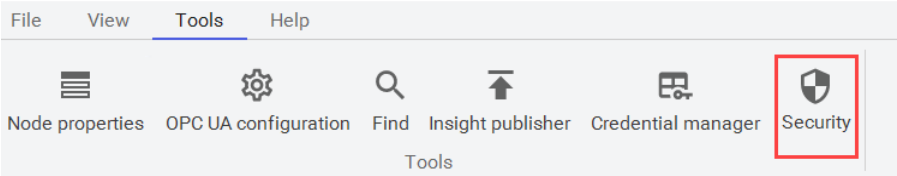
配置用户和组

安全性对话框允许将用户和组配置为 InTouchDevelopers 或 InTouchOperators 角色的成员。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

添加用户和组

1. 以管理员身份启动“应用程序管理器”。
2. 在工具菜单中的工具选项卡上，单击安全性。



此时出现安全性屏幕。

3. 单击 + 域用户和组或 + 本地用户。

将分别出现选择用户或组或选择用户 对话框。

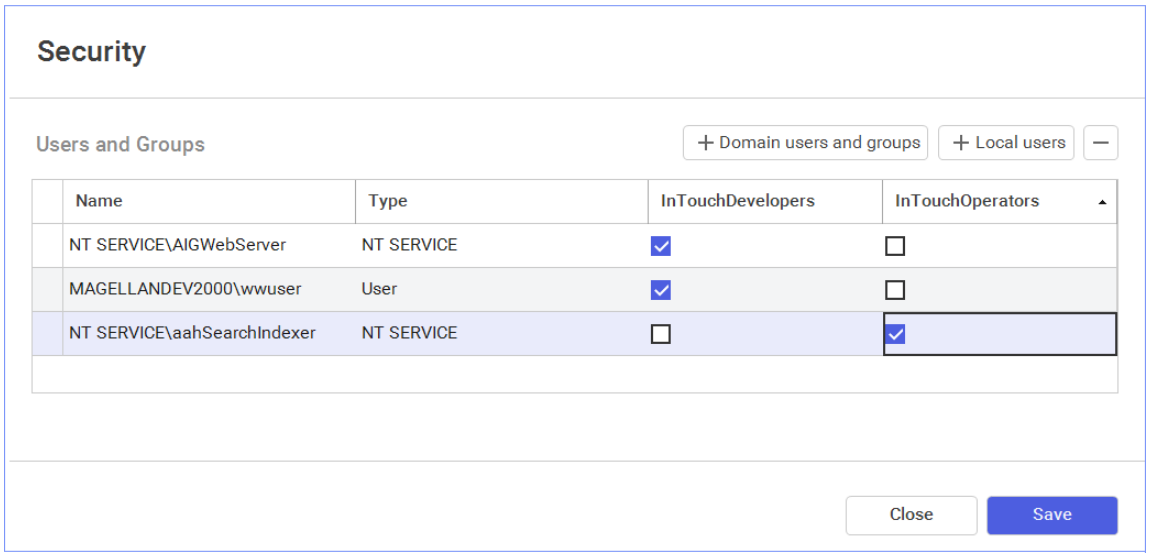
注意：支持 Windows Active Directory (AD) 用户和组，以及 Windows 本地用户。不支持且无法添加 Windows 本地组。

4. 输入用户名或组名，然后单击确定。

注意：您可以按对象名称、类型或位置搜索用户和组。单击检查名称，以确保所选对象解析为 AD 名称或组。

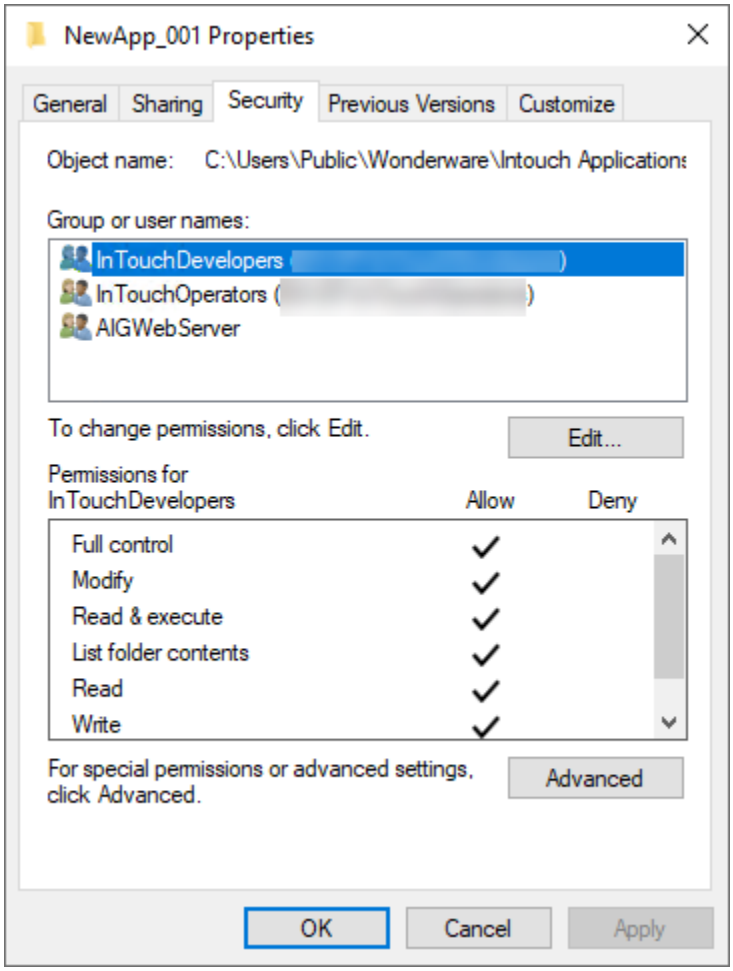
输入的用户或组显示在用户和组网格中。

5. 要将用户/组指定为 InTouch 开发人员，请选中 **InTouchDevelopers** 列中的复选框。
6. 要将用户/组指定为 InTouch 操作员，请选中 **InTouchOperators** 列中的复选框。
7. 单击保存。



默认情况下，NT SERVICE\AIGWebServer 和当前登录的用户会添加为成员，NT SERVICE\aahSearchIndexer 会添加到 InTouchOperators 组。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。



要删除用户和组：

1. 在用户和组网格中，选择要删除的条目。
2. 单击 - 图标。

   此时将删除所选用户/组。

附加信息

- InTouchDevelopers 和 InTouchOperators 保留为计算机上的本地组。
- InTouchDevelopers 和 InTouchOperators 安全权限不适用于 NAD 应用程序。网络路径或映射网络驱动器上的任何应用程序都被视为 NAD 应用程序。
- 上述在“安全性”对话框中添加的任何用户都将添加到相应的组。
- 这些组将有权访问独立 InTouch 应用程序的每个应用程序文件夹。
- 如果在启用了该功能的情况下新建一个应用程序，将使用 InTouchDevelopers 和 InTouchOperators 的相应访问权限更新文件夹的访问控制列表。
- 对于迁移的应用程序，将在迁移过程中对其应用安全权限。
- 如果打开应用程序文件夹并查看文件夹属性，可以打开“安全性”选项卡来查看所应用的访问权限。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

托管和 NAD InTouch 应用程序的本地工作目录完整性检查

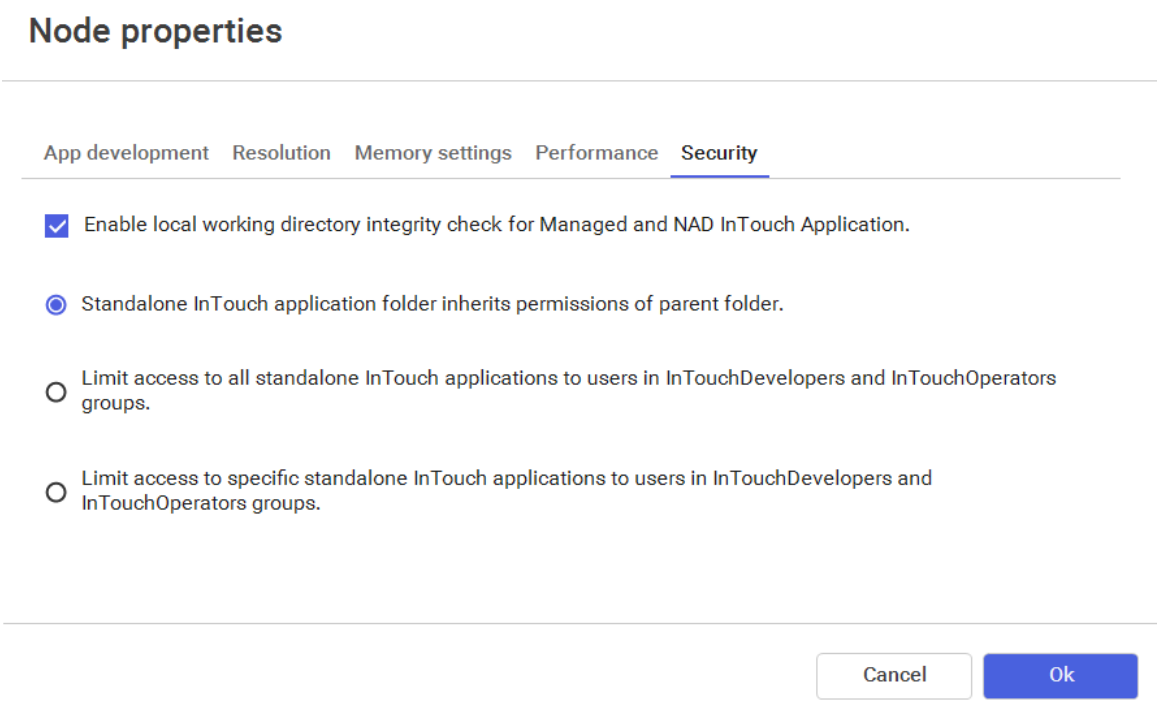
此功能可用于比较已部署的应用程序和本地工作目录中的应用程序。当为托管 InTouch 应用程序和 NAD 应用程序启动 WindowViewer 时，将对已部署的应用程序和本地工作目录中的应用程序执行文件时间戳和文件内容哈希比较。比较仅限于特定文件类型，包括 .dlls、.exe 和 .vedef。如果文件不匹配，WindowViewer 会将已部署的应用程序复制到本地工作目录并运行该应用程序。

启用完整性检查

1. 在应用程序管理器中，转到工具 > 节点属性 > 安全性。
2. 选中为托管和 NAD InTouch 应用程序启用本地工作目录完整性检查复选框。

   注意：此选项独立于限制“InTouchDevelopers”和“InTouchOperators”组中的用户对独立 InTouch 应用程序的访问选项。您可以在同一节点上同时启用它们，也可以单独启用它们。

3. 选择确定。



注意：复制过程可能会导致延迟启动 WindowViewer。如果在节点上启用了“网络应用程序开发” (NAD)，安全模式选项将被禁用。

托管 InTouch 应用程序的行为

如果比较后发现文件不匹配，则表示：

- 已经部署一个较新的应用程序。
- 本地工作目录中的应用程序文件的内容已被修改。

在这两种情况中，WindowViewer 会将已部署的应用程序复制到本地工作目录并运行该应用程序。这将导致启动时间延迟。

应用程序的 InTouch.ini 文件有两个额外的设置：

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

1. 名称：NewDefaultLocalWorkingDirectory
   - 对于 2023R2 或更高版本的新应用程序，值为 1。
   - 对于 2023R2 或更高版本的迁移应用程序，值为 0。
2. 名称：VIEWEDMANAGEDDIALOG

   当您导航到文件 > 配置 > **WindowViewer** > 托管应用程序时，值设为 1。这是因为本地工作目录已更新的通知只显示一次。

通知：

对于新的托管应用程序，通知中说明默认本地工作目录已更新为 %LOCALAPPDATA%\ArchestrA\ManagedApp。

对于迁移后的应用程序，通知中说明建议的本地工作目录为 %LOCALAPPDATA%\MigratestrA\ManagedApp。

要恢复默认工作目录，请执行以下操作：

在托管应用程序的 WindowViewer 中，转到文件 > 配置 > **WindowViewer** > 托管应用程序，并选择页面底部的恢复缺省值。

本地工作目录字段将更新为：

%LOCALAPPDATA%\ArchestrA\ManagedApp

NAD InTouch 应用程序的行为

如果比较后发现文件不匹配，则表示：

- NAD 储备库应用程序已更新。
- 本地工作目录中的应用程序文件的内容已被修改。

在这两种情况中，WindowViewer 会将 NAD 储备库应用程序复制到本地工作目录并运行该应用程序。

如果先前在 SP 2023 R2 之前的节点上启用了 NAD，然后升级到 SP 2023 R2 或更高版本，将不会修改本地工作目录路径。将显示一条建议使用新文件路径的通知。

对于新安装的 System Platform 2023 R2 或更高版本上的 NAD，默认本地工作目录路径为：

C:\Users*USER*\AppData\Local\NAD

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

附录 AL 管理 InTouch HMI 的安全性

本节介绍 InTouch HMI 的不同安全准则和安全配置。有关详细信息，请参阅《网络安全部署指南》。

InTouch HMI 的安全配置

下表列出了需要为 InTouch HMI 配置的安全区域以及本指南中提供相应说明的部分的详细信息。

安全区域	本指南中的主题	摘要
保护主机上的应用程序和内容	配置不活动超时	将 WindowViewer 配置成从 InTouch 应用程序自动注销不活动的操作员。 启用连接体验后不支持非活动超时。
通过身份验证和授权保护系统安全	基于身份验证与授权的安全性	用户需要在使用 InTouch 应用程序前进行身份验证。InTouch 会验证经过身份验证的用户是否有权使用特定功能。
	使用虚拟帐户	在访问报警函数时，使用虚拟帐户提供了一层额外的安全性。请参阅《InTouch HMI 报警与事件指南》中的使用虚拟帐户。
保护主机上的应用程序和内容	锁定系统键	通过在运行 InTouch 应用程序的计算机上禁用系统键，限制操作员访问 Windows 标准功能。
通过身份验证和授权保护系统安全		
通过 ICS 软件管理用户和组	使用基于 InTouch 的安全性	通过将功能链接到内部标记，对操作员可以执行的功能进行限制。
通过 Windows 管理用户和组	使用基于操作系统的安全性	从 Windows 操作系统继承一些用户/组帐户策略。
保护应用程序数据文件夹安全	将基于角色的安全性应用于应用程序文件夹	用户可以定义用户角色和用户组，以管理对应用程序数据文件夹的访问。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。

© 2015-2026 AVEVA Group Limited 或其子公司。保留所有权利。