



AVEVA™ Operations Management Interface

2023 R2 SP1

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

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

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

U.S. GOVERNMENT RIGHTS

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

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

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

Contents

Welcome to AVEVA Operations Management Interface (OMI)	34
Legal Information	34
Contact information	35
Supported browsers	35
Supported languages	35
Supported client operating systems	35
Release notes for AVEVA Operations Management Interface (OMI)	37
2024 Releases	37
October 2024	37
Resolved issues - AVEVA OMI 2023 R2 SP1	37
Known issues - AVEVA OMI 2023 R2 SP1	38
June 2024	41
Resolved issues - AVEVA OMI 2023 R2 Patch 01	41
Known issues - AVEVA OMI 2023 R2 Patch 01	44
About Operations Management Interface (OMI)	48
Tutorial videos	49
Create a Galaxy	50
Navigate the Object Wizard interface - part 1	50
Navigate the Object Wizard interface - part 2	51
Add an Object Wizard to a template	52
Configure an instance	53
Configure a new asset and graphic	54
Create a graphic	55
Create a screen profile	57
Configure a screen profile	57
Create a layout	58
Add a pane to a layout	59
Create a ViewApp namespace	60
Deploy a ViewApp	60
Launch a ViewApp	61
Add an external content item	62
Create and configure a responsive layout	63
Design standards	65
Configure components: Galaxies	65
Create a Galaxy	66
About Galaxies	68

Connect to a Galaxy	68
View the configuration file	70
Set single sign-on for OMI ViewApps	70
Turn off multithreaded processing for OMI ViewAPP	70
About the integrated development environment (IDE)	71
Install and activate your license	71
Manage security certificate verification	71
View the Galaxy	71
Template folder	73
Visualization folder	73
CONNECT	74
IDE application views	74
Application object icons	74
Perform a search in a view	77
Model view	77
Deployment view	79
Derivation view	80
IO devices view	81
Mapping view	82
Operations view	82
Customize your workspace	83
Dock views	84
Float views	84
Hide views	84
Reset the workspace	85
Synchronize views	85
Configure user preferences	86
Use the simplified layout	88
Log on with security enabled	88
Change users	88
Search for content	89
About the Industrial Graphic Editor	90
Platform common services	92
Configure and deploy PCS services	92
Create an instance of a service	93
Assign the service instance to a node	93
Deploy the instance	93
Configure service TCP ports	93
Configure the ASBGRBrowsing service	95
Configure the ASBMxDataProvider service	96
Configure the ASBAuthentication service	96
IOMBLSService	96
Configure and deploy the OPC UA service	97
Client access rules and galaxy security	100
Automation objects	101
View object properties	102
About templates and instances	103
Instances	104
Templates	104

Propagation	105
About base templates	105
Application templates	106
Device integration templates	107
System templates	107
WinPlatform object	107
AppEngine object	107
Area object	107
ViewEngine object	108
WebViewEngine object	108
InTouchViewApp object	108
ViewApp object	109
About derived templates	109
Work with Objects	109
Manage folders	109
Create folders	111
Create child folders	111
Delete folders	112
About using CONNECT	112
Log into CONNECT	112
Navigate between shared drives	113
Upload/download graphics to the cloud	113
Upload graphics to CONNECT	114
Download graphics to the local visualization folder	114
Upload/download support for various cloud content	115
Manage graphics in CONNECT	115
Manage graphics with multiple users	116
About cloud graphic versions	116
Overwrite graphic contents in the cloud	117
Cloud graphic versions - scenarios and examples	118
Create derived templates	119
Derive templates from another derived template	120
Create contained templates	120
ApplicationObject containment	121
Use contained names	124
Containment examples	125
View containment relationships	127
Rename contained objects	128
Edit objects	129
Object editor user assistance	129
Help file structure	130
Object editor layout	130
Lock and unlock template attributes	131
Set object security	134
Group lock and security controls	135
About the attributes page	136
Scripts page	140
Object information page	142
Customize help	143

Locate the help folders	143
Use the galaxy browser to reference objects.....	143
Browse for attributes.....	143
View attribute details in the galaxy browser	145
Browse for graphics.....	147
Browse for element properties	149
Create a filter for the galaxy browser.....	149
Change how information is shown in the galaxy browser	151
Manage objects.....	152
Check out objects	152
Check in objects.....	153
Validate objects	153
Validate scripts and other external components.....	154
Validate an object manually	154
Create an instance.....	155
Object names.....	155
Delete an object	156
Export objects	157
Protect objects on export	158
About protecting objects on export	158
Export objects as protected	159
Export objects with I/O auto assignment.....	159
Export script function libraries	159
Import objects	160
Import protected objects	162
Import objects with I/O auto assignment	163
Import a SQLData object	163
Import script function libraries	164
Import client controls	165
Import an app from the AVEVA digital exchange	165
Import AVEVA apps and client controls	165
Add a control from an AVEVA app to a ViewApp	166
Configurable properties in AVEVA apps	167
AVEVA app development guidelines	169
AVEVA app limitations and restrictions	170
Troubleshoot AVEVA or WPF apps that fail to import	171
After you import	171
Configure Objects	172
Add attributes to an object	172
Configure an attribute	173
Attribute name conventions	174
Attributes and scripting	176
Add features to attributes	177
Features inheritance	177
The I/O Feature	178
Configure I/O as read-only	181
Configure I/O as read/write	182
When objects are on scan	183
Use Read/Write I/O in Scripts	183

Configure I/O as write-only	184
Work with Outputs	185
I/O data quality	186
ATS-configure I/O for use with AVEVA Telemetry Server Communication Drivers	186
The History Feature	187
The Limit Alarms Feature	192
The Rate of Change (ROC) Alarms Feature	194
The Deviation Alarms Feature	196
The State Alarm Feature	198
The Bad Value Alarms Feature	200
The Statistics Feature	201
Using the Log Change Feature	202
I/O auto assignment	203
Assign areas and objects to scan groups	204
Rename application, system, and device integration objects	206
Rename scan groups	206
Edit and validate I/O assignments	206
How to use the mapping view for I/O auto assignment	207
Validate I/O references	209
Override I/O auto assignments	209
Batch overrides of I/O assignments	210
Upload runtime configuration changes for auto-assigned objects	211
I/O auto-assignment workflow example	211
I/O auto-assignment with OPC clients	212
Override scenario 1 — Siemens S7 controllers with an OPCClient object	213
Override scenario 2 — Allen-Bradley CIP controllers with an OPCClient object	214
Write and edit scripts	215
About object scripts	216
Script execution	217
Script locks and change propagation	220
Create and work with content	221
Add a graphic to an object	222
Errors and warnings in an owned graphic	222
Edit a graphic	223
Set graphics properties with a galaxy style library	223
Use element styles to modify graphics	223
Use quality and status indicators to provide runtime feedback	223
Use UI themes to set the appearance of WPF controls	224
Symbol Wizards	225
Create Symbol Wizards	228
Embed a Symbol Wizard into an application	228
Rename a graphic	229
Delete a graphic	229
Rearrange objects	230
Rearrange objects: drag and drop	230
About rearranged objects	230
Assign alias name to an object	231
Steps to Assign or Remove an Alias Name	233
About alias names during operations	234

Object Wizards	235
About Object Wizards	236
Object Wizard features	238
Create a basic Object Wizard	238
Object wizard configuration best practices	240
Object wizard GUI panes	241
Object wizard user interface details	244
Object wizard components	249
Add content for use with an object wizard	250
Add a new object-owned symbol	251
Link to shared content in the visualization folder	252
Associate content with a choice or option	253
Disassociate content from a choice or option	254
Multiple content item associations	254
Map attribute and graphic configurations to a choice or option	254
Attribute and graphic overrides	255
Delete content from an object	255
Delete an attribute or content from a template	256
Example1: Map a symbol wizard to an object wizard	256
Example2: Configure a linked graphic associated with multiple choices	260
Conditional visibility expressions for choice groups, choices, and options	262
Add a basic visibility expression	263
Add a visibility expression with logical operators	263
Operators, rules, and behavior for visibility expressions	264
Object wizard visibility settings in a derived template	265
Object wizard trimming	266
Enable trimming	267
Attribute and graphic overrides	268
Enter overrides and set visibility for attribute and graphic values	270
Attribute and graphic overrides	271
Configurable settings for attribute and graphic values	272
Object wizard derivation	273
Set visibility of choices and options in a derived template	273
Reasons to derive a template from an object wizard	274
Changes to object wizards in derived templates	274
Behavior of object wizard components in a derived template	275
Object wizard test mode	275
Verify behavior of visibility expressions	276
About object wizard behavior in test mode	277
Propagation of object wizard changes to derived objects	277
Object wizard behavior when a default choice is deleted	278
Propagation of attribute and graphic settings	278
Configure instances	278
Configure an instance with the IDE	279
Guidelines for configuring instances with an object wizard	280
Override default settings	280
Create a new instance graphically	281
VisualBuild Object options	283
VisualBuild graphic options	284

Add a configured asset to an overview graphic	285
Symbol Wizard Editor	286
Create Symbol Wizards	289
Embed a Symbol Wizard into an application	289
Integrate Symbol Wizards with Object Wizards	290
About the galaxy style library	291
Open the galaxy style library	292
Manage galaxy style libraries	292
Duplicate a galaxy style library	293
Set the default galaxy style library	293
Rename a galaxy style library	294
Export a galaxy style library	294
Export all galaxy style libraries	295
Import a galaxy style library	296
About the quality and status tab	297
Show data status	298
Show quality and status with the status element	298
Show quality and status by overriding	299
Quality and status styles	299
Use status elements	299
Draw the Status Element on the Canvas	300
Configure the Status Element	300
Set the Appearance of a Status Element	300
Override Element Appearance in Accordance with the Quality and Status of its Attributes	300
Override the Text Appearance of Elements to Indicate Non-Good Status or Quality	301
Override the Fill Appearance of Elements to Indicate Non-Good Status or Quality	301
Override the Line Appearance of Elements to Indicate Non-Good Status or Quality	301
Add Outlines to Elements to Indicate Non-Good Status or Quality	301
Preview all Status Appearances	302
Reset an Override Appearance to its Default	302
About element styles	303
Element styles details	303
Format Styles tab	304
Configure a global numeric format	305
Format list	306
User-Defined Global Number Styles	307
Rename User-defined Global Number Styles	307
Import and Export User-Defined Global Number Styles	308
Switch to another galaxy style library during runtime	308
Build applications	310
About screen profiles	311
About the Screen Profile Editor	311
Create a screen profile	313
Create a screen profile from the visualization folder	314
Configure a screen profile	315
Add a screen to a screen profile	317
Delete a screen from a screen profile	317

Rearrange screens in a screen profile	318
Configure a screen	319
About screen properties	319
Resize a screen	321
Set a primary screen	322
Set additional screens	323
Manage a screen profile	325
Delete a screen profile	325
Rename a screen profile	326
Duplicate a screen profile	327
Export a screen profile	327
Import a screen profile	328
AppFramework layouts	329
About the Layout Editor	330
About fixed vs. responsive layouts	335
About responsive layouts	337
Design-time editor actions: fixed layout	338
Configure the Layout Editor	340
Create a layout 1	341
Create a layout 2	341
Manage a layout	343
Delete a layout	344
Duplicate a layout	345
Rename a layout	346
Export a layout	346
Import a layout	347
Configure a layout	348
Add panes to layout	349
Configure a layout: slide-in panes	350
Configure a layout: add slide-in panes	351
Resize slide-in panes	352
Show slide-in panes at runtime	353
Rename a pane	354
Remove a pane from a layout	355
Extend panes in a layout	356
Extend panes in a layout	356
Set the default pane of a layout	358
Set the pane presentation style	358
Set layout properties	359
Set layout properties values	369
Select a screen profile layout property	370
Select a pane layout property	371
Set pane properties 1	372
Set pane properties 2	377
Set pane properties: content types	378
Set pane properties: display mode	379
Set pane properties: hide when empty	380
Set pane properties: auto-fill	381
Set layout divider properties	381

Add content to panes	382
Browse for content to add to layout	383
Search for content to add to layout	384
About adding content to an empty pane	386
Add content to multi-content panes	387
Move content to another pane	388
About the responsive layout mode	389
Build a responsive layout	390
About responsive layout methods	395
About responsive layout propagation	396
Responsive Layout: Mostly Fluid	398
Mostly fluid responsive layout example	399
Configure a column drop responsive layout	400
Configure a layout shifter responsive layout	405
Configure a layout shifter responsive layout example	406
Responsive layout limitations	408
Build layout scripts	409
Types of layout scripts	410
About built-In layout scripts	411
About named scripts	411
About event-handler scripts	411
Namespaces for use with layout scripts	412
About Built-in scripts	412
Configure built-in script	413
Tips to ensure proper On Show script execution	414
Add a named script	414
About the ShowContent() function	416
Configure the Show/Hide Content script functions	417
About the best pane match algorithm	418
Content display rules	420
Layout event handler scripts	421
Add an event-handler script to a layout from the Build tab	421
Add an event-handler script to a layout from the Script tab	423
Configure an event handler script	425
Layout event handler parameters	427
Layout execution order	427
Data binding in a layout	428
About the MyContent namespace	428
Add property binding	429
Set custom property binding type	430
Use a layout as content	431
Effects of autofill on a layout linked to an asset	431
About ViewApp namespaces	431
About the ViewApp namespace editor	433
Create a ViewApp namespace	436
Add and configure custom ViewApp namespace attributes	437
Export a ViewApp namespace	439
Import a ViewApp namespace	439
Create references to ViewApp namespace attributes	440

About retentive attributes	441
About alarms	441
Event definition	442
Types of events	442
Alarm definition	443
Types of Alarms	445
Configure State Alarms	445
Configure Limit Alarms	448
Configure Target Deviation Alarms	449
Configure Rate of Change Alarms	450
Statistical alarms	450
Set alarm state with object attributes	451
AlarmModeCmd attribute	451
AlarmInhibit attribute	451
AlarmMode attribute	451
AlarmModeEnum attribute	452
Alarms and buffered data	452
Alarm detection	452
Set the alarm state for individual alarms	452
Enable, silence, and disable alarms	453
Enable alarms	454
Silence alarms	454
Disable alarms	454
Shelve alarms	455
Enable alarm shelving	456
Shelve alarms at runtime	457
Throttle alarms	458
Propagate timestamps with alarms and events	458
Alarms or events become active	458
Alarms become disabled	459
Alarms revert to normal	459
Alarm acknowledgement	459
Acknowledge alarms when signature is required	459
Configure alarms	460
Configure alarms for system objects	460
Configure WinPlatform object alarms	461
Configure alarm priority for communication failure	461
Register using "Galaxy_<GalaxyName>" instead of "Galaxy" option	462
Configure alarms for an AppEngine object	465
Configure alarms and events for application objects	466
Set alarms on the attributes page	469
Distribute alarms and events	471
Subscribe to alarms and events from a client	472
Using InTouch HMI as the Alarm and Event Client	472
Syntax of alarm queries	472
Alarm query syntax when 'Register Using Galaxy_<GalaxyName>' is enabled	473
Alarm query examples	473
Alarm requirements for InTouch client applications	474
Alarms and event implemenation in application server and runtime applications	474

Configure plant state-based alarms.....	476
Mapping alarm modes to plant states	477
Configure the state-based alarm mode on an area object.....	478
View plant state-based alarms at runtime	479
User access and security at runtime.....	480
Configure priority ranges for alarm historization, mapping, and shelving.....	480
Configure historization for alarms and events	481
Monitor alarm severities at runtime	482
Alarm ranking at runtime	482
Aggregated alarm information.....	483
Configure alarm state aggregation.....	483
Aggregated alarm state information	484
AlarmCntsBySeverity attribute	486
AggregatedAlarmStatesIncluded attribute	486
Local alarm display in the Object Viewer	487
Aggregated alarm display in OMI ViewApp navigation	489
Alarm adorner attributes.....	491
Monitor alarm state information at runtime.....	492
About graphic elements	493
About the Graphic Editor	494
About the graphic element property commonalities	496
About graphic-specific and element-specific properties	497
About zoom percentage	497
Set the zoom percentage	498
About Layout pane property effects on zoom behavior	499
Set zoom percentage attributes	500
Minimal set of scripts and animations to enable programmatic zooming of graphics	502
Monitor and show quality and status	506
Use status elements.....	506
Draw the Status Element on the Canvas	507
Configure the Status Element	507
Set the Appearance of a Status Element	507
Override Element Appearance in Accordance with the Quality and Status of its Attributes	507
Override the Text Appearance of Elements to Indicate Non-Good Status or Quality	508
Override the Fill Appearance of Elements to Indicate Non-Good Status or Quality	508
Override the Line Appearance of Elements to Indicate Non-Good Status or Quality	508
Add Outlines to Elements to Indicate Non-Good Status or Quality	509
Preview all Status Appearances	509
Reset an Override Appearance to its Default	509
Change images at runtime	510
Use animation.....	511
Create and manage graphics	512
Create a graphic	512
Create graphics in the visualization folder	513
Create graphics in AutomationObject templates	513
Create graphics in AutomationObject instances	514
Organize graphics	514
Create folders in the visualization folder	515
Use graphic thumbnails	515

Move graphics between folders	516
Rename graphics	516
Copy graphics	516
Delete a graphic	517
View a graphic in read-only mode	517
Rename graphic folders	517
Delete graphic folders	518
Move graphic folders	518
Customize graphic folders	518
Create multiple configurations of a graphic	519
Understand visual and functional graphic configurations	519
Visual graphic configurations	519
Functional graphic configurations	519
About graphic scaling	520
About the Industrial Graphic Editor	520
Create a fixed-size graphic for a ViewApp pane with a known size	522
Create a fixed-size graphic for a ViewApp pane size	523
About fixed-size graphics when pane is resized	524
About fixed-size graphics during runtime	525
Create faceplate graphics	525
About faceplate graphics creation	526
About graphics visibility during runtime	527
About settings attributes	529
About custom properties for graphics	533
Use custom properties to show historical summary data	534
Analog statistical summary data	535
State statistical summary data	536
Historical summary period	537
Show statistical summary data	537
Use binding in custom properties	539
Change the expression or reference of a custom property at runtime	541
About using an External Content item	542
Create an External Content item	542
About the External Content editor	543
Configure an External Content item	545
Specify the URL of External Content	547
Specify media types for External Content	547
Link an External Content item to an object	548
Associate External Content to a ViewApp	548
Manage external content	550
Create an app for external content	550
Script Editor	551
Add a script to a graphic	551
Working with the SignedWrite() function for secured and verified writes	552
SignedWrite() runtime behavior	552
SignedWrite() script execution sequence at runtime	553
SignedWrite() scripting tips	553
Examples of using the attribute parameter in the SignedWrite() function	553
Configure the SignedWrite() script function	555

How to write to an attribute configured for Secured Write or Verified Write security classification	556
Example: how to change element properties using scripts	557
ViewApp editor	557
About the ViewApp Editor	558
Configure the ViewApp Editor	562
About ViewApp and derived templates	564
Create a ViewApp	564
Manage a ViewApp	565
Delete a ViewApp	565
Rename a ViewApp	566
Export a ViewApp	566
Import a ViewApp	567
Configure a ViewApp	567
Manage screen profiles in a ViewApp	568
Manage layouts in a ViewApp	569
Set additional properties for a ViewApp	570
Add content to a ViewApp	570
Browse content to add to a ViewApp	572
Search for content	572
About adding content to an empty pane	574
Add content to multi-content panes	575
Show or hide ViewApp navigation items	576
About navigation item security	577
Configure navigation item security	578
User permissions and navigation item security	579
About previewing a ViewApp	579
Preview a ViewApp	580
About ViewApp previews at different stages	582
Switch ViewApp language at runtime	585
Define and configure galaxy languages	586
Graphics language switching	586
About localization workflow	586
Alarm app language switching	587
Configure languages for a Galaxy	588
Add a language to a Galaxy	588
Remove a language from a Galaxy	590
Modify the font for a language	591
Change the default language for a Galaxy	591
Export files for offline translation	592
Types of language dictionary files	593
Export language data for all Industrial Graphics in a Galaxy	594
Export language data for specific objects	595
Export graphic language data for a managed InTouch application	597
Export graphic language data for a published InTouch application	598
Export text to an existing dictionary file	598
Translate exported graphic language files	599
Translate exported graphic text dictionary files	599
Import translated language files	600

Import translated graphic dictionary files	600
Examples of graphic or object mismatch handling during language imports	602
Language data handling for galaxy operations	604
Export alarm comments for offline translation	605
Guidelines and recommendations	605
Organize your export	605
Translation file formatting and editing	605
Reimport	605
About the alarm comments language file	606
Export alarm comments from very large galaxies	606
Export all galaxy alarm comments	606
Export alarm comments by area	607
About file names	607
Export objects not assigned to an area	607
Re-export alarm comments	608
Export new untranslated alarm comments	608
Export modified existing alarm comments	609
Translate exported alarm comment language files	609
Import translated alarm comment language files	612
Test the language switching functionality at run time	613
About predefined language attributes	613
About attribute setting	613
About ViewApp navigation	614
About the Layout Editor and navigation	616
About the Layout Editor and navigation components	617
About ViewApp hierarchical navigation	620
About ViewApp hierarchical navigation	621
Trigger hierarchical navigation actions	622
About auto-fill navigation	623
About content types and navigation	625
Assign a content type	625
Action sequencing when content or other types are applied in ViewApp	626
About ViewApp custom navigation	627
Add custom navigation to a ViewApp	627
Configure properties for custom navigation items	628
ViewApp custom navigation rules	630
Hide assets in ViewApp navigation during runtime	631
Hide assets from a ViewApp navigation model	631
About ViewApp navigation hierarchical display	632
About display of controls in the ViewApp navigation hierarchy	633
About predefined navigation attributes	635
Predefined navigation attribute rules	636
Use navigation attributes in action scripts	643
About client controls embedded into a graphic	646
Import client controls: options	647
Import client controls	648
Example of installing the TagPicker control	648
Import previously exported client controls	648
Organize client controls	649

View and change the properties of client controls	649
Example of changing a property of the TagPicker control	650
Bind client control properties to attributes or element references	650
Example of data binding in the TagPicker control	652
Configure client control event scripts	652
Example of configuring an event script for the TagPicker control	653
Animate client controls	654
Export client controls	655
Secure client controls	655
Include dynamically loaded assemblies with the client control	655
Requirements for both inclusion methods	656
Sample XML for a dynamically loaded assembly list	656
XML schema for the dynamically loaded assembly list	656
Embed the XML manifest resource in the primary assembly	657
Include the XML manifest resource in an external configuration file	657
Prevent dynamically loaded assembly import issues	657
View additional client control information	658
View client control assemblies	658
View class name, vendor, and version of a client control	659
View objects and graphics referencing client controls	660
Ensure dynamically loaded assemblies for client controls are imported	660
Dynamically loaded assemblies rules	660
XML sample for listing the file location of dynamically loaded libraries	660
XML schema for the dynamically loaded assembly list	661
Embed the XML manifest resource in the primary assembly	661
Include the XML manifest resource in an external configuration file	662
Tips on dynamically loaded assemblies	662
Embed client controls	662
Embed the TagPicker client control	663
About OMI Apps	663
AlarmApp	664
Get started with the AlarmApp	664
Configure runtime properties of the AlarmApp	665
Follow Current Context property	675
Alarm queries	677
Alarm query syntax	678
Client Mode property	678
Show Grid property	679
Show Headers property	680
Show Status property	680
Show Context Menu property	680
Auto Scroll property	681
Flash UnAck Alarms property	681
Ack Signature property	681
Ack Min Priority property	681
Ack Max Priority property	681
Show Command Bar property	681
Set AlarmApp properties	682
Show current alarms and events	683

Show historical alarms and events	685
Set alarm colors	687
Set event record colors	688
Set return to normal alarm record colors	689
Set heading, grid, and window color	690
Set alarm shelving colors	690
Set priority ranges and column options for alarm records	691
Set colors for acknowledged alarms	691
Set colors for unacknowledged alarms	692
Set unacknowledged alarms to flash	692
Rename, resize, and reorder columns	693
Rename columns	695
Resize columns	695
Change column order	696
Sort alarms	697
Filter alarms	698
Wildcards in queries	698
Use an existing query	699
Add a new query filter	699
Construct filters	700
Modify a query filter	701
Delete a query filter favorite	702
Export query filter favorites	702
Import query filter favorites	702
Set time zone and format	703
Set the time zone	704
Set a time format	705
Set a .NET time format	706
Configure runtime behavior	707
Show heading, grid, or status bar	708
Automatic query for alarms on startup	708
Scroll automatically to new alarms	709
Hide errors, warnings, and status messages	709
Restrict user access to rows and columns	710
Retain hiding when changing the alarm query filter	710
Override a frozen grid	711
Customize the no records message	711
Configure the AlarmApp to require an ACK signature	711
Configure the AlarmApp to require a SHELVING signature	712
Configure the runtime shortcut menu	713
Use the AlarmApp at runtime	715
Refresh the AlarmApp grid	715
View status bar information	715
View status bar information of current modes	715
View status bar information of historical modes	716
Acknowledge alarms	717
Acknowledge alarms with signature	719
Acknowledge alarms using a Smart Card	720
Dismiss alarms at runtime	721

Shelve and unshelve alarms at runtime	722
Shelve alarms	722
Show shelved alarms	723
Unshelve alarms	724
Sort alarms at runtime	725
Filter alarms at runtime	726
Use an existing query	726
Add a new query filter at runtime	726
Modify a query filter	727
Save runtime modifications to queries and filters	727
Delete a query filter	728
Import query filter favorites	728
Export query filter favorites	728
Filter alarms with client-based filtering	729
Reset the grid	731
Hide alarms	731
Freeze and unfreeze the alarm grid	732
Show alarm statistics	733
Set font size for the alarm grid	733
AlarmApp scripts	734
Write a named script for the AlarmApp	735
Alarm client control pass-through properties	736
AlarmApp scriptable properties	736
Write an event handler script	737
Write an event handler script from the Properties grid	737
Write an event handler script from the Layout script editor	739
ContentPresenterApp	741
ContentPresenterApp runtime examples	745
Configure the ContentPresenterApp	749
More information about ContentPresenterApp properties	750
ContentType property	761
Layout properties	763
Padding property	764
Size Mode property	765
Display Columns property	765
Display Rows property	766
Stretch property	766
Search Levels property	766
Contextualization	767
Navigation Start Node property	768
ContentPresenterApp scripting	768
ContentPresenterApp property validation	770
Graphic displays during runtime	770
Upgrade and migration of the ContentPresenterApp	781
DocViewerApp	782
Change the zoom level of a document	783
Touch support	783
Optimize the DocViewerApp for a touch device	783
Compatibility of the DocViewerApp	784

Configure the DocViewerApp to show Word documents	784
Before you begin	785
DocViewerApp error message	786
Create an External Content item	786
Configure an External Content item	786
Link an External Content item to an object or template	788
Associate an External Content item with a ViewApp	789
Details of the External Content editor	790
More information about media types	791
More information about specifying a URI	792
AVEVA GraphicRepeaterApp	792
Configure the GraphicRepeaterApp	796
GraphicRepeaterApp example	797
Configure GraphicRepeaterApp properties	801
Details of background properties	808
Details of comma-delimited data	809
Details of the Data property	810
Calculate the number of repeatable graphics	810
Example layout script with the GraphicRepeaterApp	811
Comparison of GraphicRepeaterApp properties and scripting attributes	811
Details of sorting properties	812
Details of the Live Sorting property	813
Details of the Fill style property	813
Details of the Padding property	813
Details of the DisplayColumns property	813
Details of the Display Rows property	814
Validity checking during runtime	814
HamburgerApp	815
Configure the HamburgerApp	815
AVEVA Historian Client Trend	817
HistoricalTrendApp properties	817
Configure HistoricalTrendApp properties	818
Upgrading HistoricalTrend App Runtime Properties	821
Follow Current Context Property	822
Configure a trend	823
Creating a New Server Connection	823
Add a source tag to a trend	825
Tag Picker	826
Servers pane	828
Tags pane	828
Filter pane	829
Time picker	830
Configure trend application options	831
Configure retrieval options	831
Configure color options	832
Configure time zone options	833
Configure miscellaneous options	834
Configure other options	836
Configure trend file properties	837

Configure general properties	837
Configure color properties	838
Configure axis properties	840
Configure limit properties	841
Configure annotation properties	842
Configure target region properties	843
Configure a trend to use a summary tag	844
Absolute or relative times	845
Absolute time	845
Relative time	846
Switch between absolute and relative time: example	847
Time offset formats	847
Use time offsets to compare data in absolute time	848
Replicated tags	851
Add a source tag or replicated tag	852
Find a source tag or replicated tag	852
Replace a source tag or replicated tag	853
Work with runtime trend charts	854
View data in the trend chart	854
Refresh the trend chart	855
Delete a tag	856
Configure trend options for a tag	856
Configure display options	856
Define a target region for a tag	858
Configure retrieval options for a tag	862
Scroll through tags in a trend	864
Highlight a tag	864
Show a single tag in the trend	864
Stack traces	865
Show live data	865
Show historical data in replay mode	866
Configure predictive data retrieval	867
Tag scale	867
Show no scales on the value axis	868
Show a single scale on the value axis	868
Show multiple scales on the value axis	870
Show cursor values on the value axis	872
Scale tags up or down	872
Scale tags automatically	874
Return tags to their original scale	875
Move tags up or down in the chart	875
Rubber band scaling	876
Pan in the trend chart	877
Axis cursors	877
Move a cursor	878
Show or hide the axis cursors	878
Show or hide the cursor difference	879
Zoom in or out	879
Show or hide the chart grid	879

View tag definition information	880
View the hierarchical name in a trend	881
View trend data in a table format.....	882
View the data log in a narrow format.....	882
View the data log in a wide format.....	883
View statistics	884
Work with scatter plots	886
View data in a scatter plot	887
Scale tags in a scatter plot	888
Configuring axes in a scatter plot.....	889
How are value pairs matched?	889
Quality calculation for data points	889
Pan and zoom in a scatter plot	890
Define a target region for a scatter plot	890
Examples of target regions in scatter plots.....	890
Configure scatter plot properties	893
Other scatter plot considerations	895
Annotations	895
Add an annotation	896
View the annotation list	897
Edit an annotation	897
Delete an annotation	898
Save the annotations list as a .CSV file	898
Print annotations	898
Work with trend files	900
Create a trend	900
Configure default settings for a trend file	900
Open an existing trend	901
Save a trend	901
Close a trend	902
Undo or redo actions	902
Output trend data	902
Print trend data	902
Print trend sets	903
Print a trend set	903
Save trend data to a .CSV file	903
Save the trend chart as an image file	903
E-mail a trend file	904
Copy a trend chart to the Windows clipboard	904
Historical trend pen scripts	904
Write an HistoricalTrendApp named script	905
Write an event handler script	906
Write an event handler script from the properties grid	907
Write an event handler script from the Layout script editor	909
Data retrieval options	910
Understanding retrieval modes	911
Cyclic retrieval	911
Cyclic retrieval - how it works	912
Cyclic retrieval - supported value parameters	912

Cyclic retrieval - query example	913
Cyclic retrieval - initial values	913
Cyclic retrieval - handling NULL values	913
Delta retrieval	913
Delta retrieval - how it works	913
Delta retrieval - supported value parameters	914
Delta retrieval - query example	914
Both leading and trailing edge detection for discrete tags	915
Delta retrieval - initial values	915
Delta retrieval - handling NULL values	915
Full retrieval	917
Full retrieval - how it works	917
Full retrieval - supported value parameters	918
Full retrieval - query example	918
Full retrieval - initial values	918
Interpolated retrieval	918
Interpolated retrieval - how it works	918
Interpolated retrieval - supported value parameters	919
Interpolated retrieval - query example	920
Interpolated retrieval - initial and final values	920
Interpolated retrieval - handling NULL values	920
"Best fit" retrieval	920
Best fit retrieval - how it works	922
Best fit retrieval - supported value parameters	923
Best fit retrieval - query example	923
Best fit retrieval - initial and final values	924
Best fit retrieval - handling NULL values	924
Average retrieval	924
Average retrieval - how it works	925
Average retrieval - supported value parameters	927
Average retrieval - query examples	927
Average retrieval - initial and final values	929
Average retrieval - handling NULL values	929
Minimum retrieval	929
Minimum retrieval - how it works	929
Minimum retrieval - supported value parameters	930
Minimum retrieval - query examples	931
Minimum retrieval - initial and final values	931
Minimum retrieval - handling NULL values and incomplete cycles	931
Maximum retrieval	933
Maximum retrieval - how it works	933
Maximum retrieval - supported value parameters	934
Maximum retrieval - query examples	934
Maximum retrieval - initial and final values	934
Maximum retrieval - handling NULL values and incomplete cycles	935
Integral retrieval	936
Integral retrieval - how it works	936
Integral retrieval - supported value parameters	937
Integral retrieval - query example	937

Integral retrieval - initial and final values	937
Integral retrieval - handling NULL values	937
Slope retrieval	937
Slope retrieval - how it works	938
Slope retrieval - supported value parameters	939
Slope retrieval - query example.....	939
Slope retrieval - initial and final values	939
Slope retrieval - handling NULL values	939
Counter retrieval	939
Counter retrieval - how it works	940
Calculations for a manually reset counter	941
Counter retrieval - supported value parameters	941
Counter retrieval - initial and final values	942
Counter retrieval - handling NULL values	942
Counter retrieval - handling illegal values.....	942
Counter retrieval - query example.....	942
ValueState retrieval	943
ValueState retrieval - how it works	943
ValueState retrieval - supported value parameters	945
ValueState retrieval - query examples.....	945
ValueState retrieval - initial and final values.....	945
ValueState retrieval - handling NULL values	946
RoundTrip retrieval	946
RoundTrip retrieval - how it works	946
RoundTrip retrieval - supported value parameters	947
RoundTrip retrieval - query examples	947
RoundTrip retrieval - initial and final values	949
RoundTrip retrieval - handling NULL values	949
Predictive retrieval (wwFilter)	949
Bounding value retrieval	949
Bounding value retrieval - how it works	950
Bounding value retrieval - query examples	950
Understanding retrieval options	951
Which options apply to which retrieval modes?	951
Using retrieval options in a Transact-SQL statement	953
Cycle count (X values over equal time intervals) (wwCycleCount)	954
Resolution (values spaced every X ms) (wwResolution)	955
Resolution - query examples	956
About "phantom" cycles	957
Time deadband (wwTimeDeadband)	959
Time deadband - query examples	959
Value deadband (wwValueDeadband)	959
Value deadband - query examples	960
History version (wwVersion)	960
History version - query example	961
Interpolation type (wwInterpolationType)	962
Timestamp rule (wwTimestampRule)	964
Time zone (wwTimeZone)	965
Quality rule (wwQualityRule)	966

Quality rule - query examples	967
State calculation (wwStateCalc)	974
Analog value filtering (wwFilter)	975
Statistically removing outliers (SigmaLimit)	976
Converting analog values to discrete values (ToDiscrete)	977
"Zeroing" around a base value (SnapTo)	979
Selecting values for analog summary tags (wwValueSelector)	981
Edge detection for events (wwEdgeDetection)	983
Edge detection for analog tags	984
Leading edge detection for analog tags	986
Trailing edge detection for analog tags	986
Both leading and trailing edge detection for analog tags	987
Edge detection for discrete tags	987
Leading edge detection for discrete tags	988
Trailing edge detection for discrete tags	988
Retrieval styles for trend	988
Working with retrieval styles	988
Location and structure of retrieval styles	989
Structure of the retrieval styles file	989
Creating and editing retrieval styles	991
Retrieval style XML elements	991
styleCollection XML element	991
retrievalStyle XML element	992
duration XML element	992
retrieval XML element	993
Using the standard retrieval styles	994
Retrieval styles, application settings, and tag settings	996
ImageViewerApp	997
Change the orientation of an image	997
Change the size of an image	999
Scroll an image	999
Touch support	999
Optimize the ImageViewerApp for a touch device	1000
Compatibility of the ImageViewerApp	1000
Configure the ImageViewerApp	1000
Before you begin	1001
ImageViewerApp error message	1002
Create an External Content item	1002
Configure an External Content item	1002
Link an External Content item to an object or template	1004
Adding an External Content item to a ViewApp	1005
Details of the External Content editor	1006
More information about media types	1007
More information about specifying a URI	1008
InSightApp	1008
Steps to configure the InSightApp	1009
InsightApp security requirements	1011
Favor trusted connections	1012
Require trusted connections	1013

Using a self-signed certificate	1013
MapApp	1014
Configuring the MapApp: overview	1015
Before configuring the MapApp	1016
Create user credentials	1016
Set up a map server	1017
Configure global MapApp properties	1020
Map source properties	1023
Create a custom layer ZIP file	1026
Zoom layers	1031
Location assets	1032
Configure a MapControl	1033
More details on bounding box properties	1036
Graphic limitations of graphics in the MapApp	1038
Properties supported by all graphic elements with some limitations	1038
Supported graphic elements and additional properties	1039
Supported animations	1042
Major unsupported industrial graphic features	1045
MapApp enhancements	1046
Enhanced MapApp scripting support	1046
Map App zoom level enhancements	1046
Current limitations of the MapApp widget	1047
Export and import map source data	1047
NavigationApp	1048
About ViewApp navigation hierarchical display	1049
About display of controls in the ViewApp navigation hierarchy	1050
Configure the Breadcrumb control	1052
How to configure the NavBreadcrumbControl	1053
PDFViewerApp	1055
Change the view of a PDF document	1057
Search in a PDF document	1057
Navigate in a PDF document	1058
Change the orientation of a PDF document	1059
Copy and paste content from a PDF document	1059
Change the page view of a PDF document	1061
Touch support	1062
Optimize the PDFViewerApp for a touch device	1062
Configure the PDFViewerApp	1063
Before you begin	1064
PDFViewerApp error message	1064
Create an External Content item	1064
Configure the External Content item for a PDF document	1065
Link an External Content item to an object or template	1066
Add an External Content item to a ViewApp	1067
Details of the External Content editor	1068
More information about media types	1070
More information about specifying a URI	1070
PLCViewerApp introduction	1071
PLCViewerApp requirements	1072

PLCViewerApp configuration	1074
PLCViewerApp required and optional information	1075
Configure PLC data sources	1075
Create an asset mapping file	1078
Create a translation file	1079
Integrate information from a custom translation file	1080
Configure PLCViewerApp properties	1081
PLCViewerApp runtime version check	1083
PLC ViewerApp during runtime	1084
Runtime enhancements of the PLCViewerApp	1085
See more information from the PLCViewerApp	1085
Set view context	1086
Navigation from a ViewApp	1087
SpreadsheetViewerApp	1089
Change the zoom level of a spreadsheet	1090
Touch support	1090
Optimize the SpreadsheetViewerApp for a touch device	1090
Compatibility of the SpreadsheetViewerApp	1091
Configure the SpreadsheetViewerApp	1091
Before you begin	1092
SpreadsheetViewerApp error message	1093
Create an External Content item	1093
Configure an External Content item for the SpreadsheetViewerApp	1093
Link an External Content item to an object or template	1095
Add an External Content item to a ViewApp	1096
Details of the External Content editor	1097
More information about media types	1099
More information about specifying a URI	1099
TitleBarApp	1100
Configure a title bar	1100
TitleBarApp properties	1101
WWWebAppControls	1105
Configure a web control	1106
Using AuthenticationProviders mode with WebAppControls	1108
Import OMI Apps (WebHelp)	1108
Troubleshoot AVEVA or WPF apps that fail to import	1109
Add a control from an AVEVA app to a ViewApp	1109
Configurable properties in AVEVA apps	1110
AVEVA app development guidelines	1112
AVEVA app limitations and restrictions	1113
Expose underlying controls of a custom app in layout scripting	1114
About industrial widgets	1116
Web widgets	1117
Configure a widget for a web-enabled AVEVA OMI ViewApp	1118
Configure a widget for other web-enabled AVEVA products	1120
Configurable properties in widgets	1120
Widget properties	1121
Carousel widget	1121
DataGrid widget	1122

Configure the Data Grid widget	1126
Configure Data Grid widget properties	1126
Data Grid properties	1127
Prepare source data for the Data Grid widget	1129
Map_App widget	1129
QRCode_Scanner widget	1131
Teamwork widget	1132
Web_Browser widget	1133
OMI Web Client	1133
Considerations when creating a ViewApp for the OMI web client	1133
Prerequisites for using the OMI web client	1134
Install a security certificate on Windows	1135
HistoricalTrendApp and TrendPen configuration	1137
Security considerations for the OMI web client	1138
General OMI web client limitations	1139
Place the web client in an iFrame	1140
Known issues in the OMI web client	1140
Industrial Graphics limitations in the OMI web client	1141
Graphic element limitations	1141
Animation limitations	1142
Embedded widget limitations	1143
Script limitations	1143
HideGraphic() limitations	1144
ShowContent() limitations	1144
ShowGraphic() limitations	1144
Font limitations	1144
OMI App limitations in the OMI web client	1145
AlarmApp limitations	1146
ContentPresenterApp limitations	1146
GraphicRepeaterApp limitations	1146
HistoricalTrendApp and TrendPen limitations	1146
MapApp limitations	1147
TitleBarApp limitations	1150
WebBrowserApp limitations	1150
Widget limitations in the OMI web client	1150
Layout script limitations in the OMI web client	1150
ViewApp namespace attribute limitations	1150
OMI web client responsive layout limitations	1151
About use of touch screens in ViewApp	1151
Build a ViewApp that supports touch gestures during runtime	1151
About ViewApp touch interactions	1152
About touch interactions supported during runtime	1154
Runtime touch interaction limitations	1155
About InTouch Access Anywhere	1156
About InTouch Access Anywhere architecture	1157
Verify an InTouch Access Anywhere installation	1157
Configure InTouch Access Anywhere	1158
Configure InTouch Access Anywhere server	1158
About configuring a firewall port exception	1158

Use a command prompt to configure a firewall port exception	1158
Use the Windows GUI to configure a firewall port exception	1159
Configure a firewall program exception	1160
Use the Server Configuration console for settings	1160
Use the Server Configuration console General tab	1160
Use the Server Configuration console Performance tab	1161
Use the Server Configuration console Communication tab	1161
Use the Server Configuration console Acceleration tab	1162
Use the Server Configuration console Security tab	1163
Use the Server Configuration console Logging tab	1164
About advanced server configuration (For Administrator Use Only)	1164
About the InTouch Access Anywhere web component	1164
About installation of the InTouch Access Anywhere Server	1164
Modify the InTouch Access Anywhere interface	1164
Modify the InTouch Access Anywhere connection name	1165
About InTouch Access Anywhere secure connections	1165
About secured WebSocket communication to remote desktops	1165
Secured WebSocket connections using InTouch Access Anywhere secure gateway	1167
About Using a trusted certificate	1167
About InTouch Access Anywhere on MobileDevices	1167
Supported Browsers for InTouch Access Anywhere	1167
About HTTPS mode for InTouch Access Anywhere in the browser	1168
Configure InTouch Access Anywhere	1168
Modify the InTouch Access Anywhere Interface	1168
About static configuration of the config.js file	1169
Define InTouch Access Anywhere configuration groups	1172
About settings precedence for InTouch Access Anywhere	1172
About form posts in InTouch Access Anywhere	1173
Embed InTouch Access Anywhere in an iframe	1174
Hide InTouch Applications from InTouch Access Anywhere	1174
Configure gestures for touch devices	1175
About HTTPS mode for InTouch Access Anywhere in the browser	1176
InTouch Access Anywhere server limitations	1177
Networking limitations	1177
Browser limitations	1177
Navigational limitations	1178
Network application development limitations	1178
Build an InTouch Access Anywhere Secure Gateway	1179
Connect to an InTouch Access Anywhere Server through the Secure Gateway	1180
Connect to an InTouch Access Anywhere server through a secure gateway	1180
Configure a secure gateway node to host a single InTouch Access Anywhere server	1183
Configure a secure gateway node to show a list of InTouch applications	1184
About the InTouch Access Anywhere Secure Gateway Configuration Portal	1184
Secure Gateway configuration dashboard	1185
Configure Secure Gateway e-mail alerts	1186
InTouch Access Anywhere HTML5 client configuration	1186
Configure the Access Anywhere server to work with a secure gateway	1187
About security allowlists	1188
Configure the origin header parameter for allowlist security	1190

Configure session cookie timeout	1190
Advanced secure gateway configuration	1190
Provide high availability of a secure gateway	1191
Restrict access to and from a secure gateway	1191
About a secure gateway built-in authentication server	1191
Disable authentication server when access is through a broker	1193
About InTouch Access Anywhere Secure Gateway ports and SSL certificates	1193
Add a certificate thumbprint to the EricomSecureGateway.Config file	1197
Configure a secured port and SSL certificate	1198
Configure failover gateways	1199
Configure the built-In InTouch Access Anywhere web server	1199
About the InTouch Access Anywhere Secure Gateway web server proxy	1200
Connect to an InTouch Access Anywhere server through the Secure Gateway web server	1200
Enable non-secured port for HTTPS auto-redirect	1200
Disable HTTP/HTTPS filtering	1201
Advanced configuration using the EricomSecureGateway.Config file	1201
Prevent access to non-listed folders	1202
InTouch Access Anywhere Secure Gateway known limitations	1202
About InTouch Access Anywhere certificate error messages	1202
About InTouch Access Anywhere Secure Gateway log files	1203
About using InTouch Access Anywhere	1204
About InTouch Access Anywhere Supported Browsers	1204
About InTouch Access Anywhere and ViewApps	1204
View an application with InTouch Access Anywhere	1205
About information needed to sign on to InTouch Access Anywhere	1205
Sign on to InTouch Access Anywhere	1205
Sign on directly to InTouch Access Anywhere	1205
Sign on through a Secure Gateway	1207
Sign on with single sign-on	1209
Troubleshoot sign on problems	1209
Advanced connection options	1210
Advanced connection settings: connection page	1211
Advanced connection settings: security page	1212
Advanced connection settings: language and audio page	1213
Advanced connection settings: display page	1214
View an InTouch Access Anywhere session	1217
Resize an application window	1218
Use the Access Anywhere toolbar	1220
Clipboard support	1221
Copy text from a remote InTouch application	1221
Copy text to a remote InTouch session	1222
Use finger gestures	1222
Change an InTouch application shown by WindowViewer	1224
End an InTouch Access Anywhere session	1224
Session auto-logoff	1224
Automatic reconnect	1225
About device support for InTouch Access Anywhere	1225
About running InTouch Access Anywhere on different operating systems	1225

Running InTouch Access Anywhere on Android	1225
Running InTouch Access Anywhere on Google Chrome	1227
About Chromebook keyboard commands and InTouch Access Anywhere	1227
Running InTouch Access Anywhere on iOS	1228
Running InTouch Access Anywhere on Windows RT	1229
Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations	1231
Running InTouch Access Anywhere: common behaviors	1235
About deployment	1239
Prepare to deploy a ViewApp	1239
Prepare to deploy a ViewApp: security	1239
Prepare to deploy a ViewApp: galaxy status	1240
Assign an engine to a ViewApp	1240
Prepare to deploy a ViewApp: assign a ViewApp to an engine	1241
Assign a ViewApp to an engine	1241
About deploying a ViewApp	1241
Deploy a ViewApp	1242
Determine deployment status	1245
Deploy an updated ViewApp	1245
About undeploying a ViewApp	1246
Undeploy a ViewApp	1246
Run a ViewApp	1248
View deployed ViewApps	1248
Launch a ViewApp	1249
About the ViewApps Application Manager	1250
Display a web ViewApp on a mobile device	1251
About the InTouch web client	1253
Launch a ViewApp in a web browser	1253
ViewApp behavior in the web client	1254
Troubleshoot OMI web client connection issues	1255
Launch ViewApp in the OMI Web Client using connected experience	1256
Log in with AVEVA Identity Manager	1257
Use Azure Active Directory credentials to sign on	1258
Use Azure Active Directory credentials for secured write authentication	1259
Use Azure Active Directory credentials for verified write authentication	1260
Open a ViewApp	1262
Search a ViewApp NavTree	1262
Enable runtime navigation model searches	1263
About search results	1264
Process late data at runtime	1266
Enable historical playback	1267
About playback attributes	1268
Manually deploy the playback service	1271

Manage OMI applications	1273
Manage product licensing and user authorization.....	1273
Connected experience and operations control.....	1273
Configure operations control and connected experience	1274
Authentication and entitlement.....	1279
Authenticate with the AVEVA Identity Manager	1280
Behavior if connection is lost	1284
Single Sign-On (SSO)	1285
License mode	1286
Start and run the IDE	1286
Connect to a galaxy	1287
Connect to a galaxy with no security configured	1287
Connect to a galaxy not using Operations Control connected experience (configuration mismatch) ..	1288
Sign out of the IDE	1288
Start Application Manager	1289
Launch a ViewApp using the connected experience	1289
Configure user permissions for connected experience	1290
Launch ViewApp in the OMI Web Client using connected experience	1291
Sign out from an OMI ViewApp	1292
Switch OMI ViewApp users	1293
Secured and verified writes	1293
Deploy scenarios	1293
Change Operations Control connected experience mode	1294
Using the OCMC with connected experience	1295
Galaxy migration to support connected experience	1295
Signing on using the connected experience	1296
Signing out using connected experience	1297
Licensing and the OMI web client	1298
About ViewApp security	1298
About Galaxy security	1300
Configure Galaxy security	1301
About Galaxy security groups	1301
Manage security groups	1302
Manage security roles	1302
Create a security role	1302
About the default security role	1303
About the Credentials tab	1303
Create a new credential	1304
Associate an OS group with a named credential	1305
Configure credentials	1307
Apply log in security	1308
Apply log in security example	1308
About security attributes	1310
Configure security attributes	1311
About security attribute data types	1312
Manage industrial graphics	1314
Import and export graphics	1314

Import graphics from .aaPKG files	1315
Export graphics	1315
Programmatically import and export industrial graphics.....	1316
Implement and use the GraphicAccess API.....	1317
Overridden text string import and export.....	1317
Export overridden text strings	1317
Import overridden text strings	1319
Configure security for graphics	1320

Welcome to AVEVA Operations Management Interface (OMI)

Welcome to AVEVA Operations Management Interface (OMI) 2023 R2 SP1.

Please review the release notes sections beginning at [Release notes for AVEVA Operations Management Interface \(OMI\)](#) The topics there include issues fixed in this release.

Note: Enhancements and fixed issues sometimes include specific Architectural and Development changes that we want you to be aware of; we have identified these with an "Important" note. For example, an Important note might be included when certain actions or special configuration must take place during an upgrade, or when there are changes to APIs or system behavior.

Finally, we really want your feedback, because you are helping us to make this a better product. We will work closely with you to understand what you want out of this application so that we can improve future versions. If you experience any difficulty using AVEVA OMI, please let us know.

Thank you,

The AVEVA OMI Team

Legal Information

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

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

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

U.S. GOVERNMENT RIGHTS

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

12-212, FAR 52.227-19, or their successors, as applicable.

Publication date: Friday, December 13, 2024

Publication ID: 1495798

Contact information

AVEVA Group Limited

High Cross

Madingley Road

Cambridge

CB3 0HB. UK

<https://sw.aveva.com/>

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

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

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

Supported browsers

Important: Cookies must be enabled in ALL browsers, even while in private browsing or incognito mode.

Supported browsers are used for configuration in AVEVA CONNECT, for downloading the AVEVA Situational Awareness Library (SAL) client, and for viewing user documentation.

AVEVA SAL supports the following browsers:

- Google Chrome
- Microsoft Edge, version 79 or newer
- Mozilla Firefox

AVEVA SAL no longer supports the following browsers:

- Microsoft Internet Explorer 11
- Microsoft Edge, version 78 or older

Supported languages

We are pleased to provide examples in the following additional languages: French, Spanish, simplified Chinese. They can be enabled and updated by an administrator.

Supported client operating systems

The following table outlines the software requirements for the server/computer on which the AVEVA Situational Awareness Library (SAL) client resides.

Component	Specification
Microsoft Windows	Windows 10 Enterprise 2021 LTSC Windows 10 Enterprise 22H2
Microsoft Windows Server	Microsoft Windows Server Standard 2022 with Desktop Experience Microsoft Windows Server Standard 2019 with Desktop Experience
Microsoft .NET Framework	4.8
Infragistics	22.2

Release notes for AVEVA Operations Management Interface (OMI)

The release notes describe the new and enhanced functionality available in AVEVA Operations Management Interface (OMI), providing an overview of the most significant changes.

Note: Enhancements and Fixed Issues sometimes include specific architectural and development changes that we want you to be aware of; we have identified these with an "Important" note. For example, an Important note might be included when certain actions or special configuration must take place during an upgrade or when there are changes to APIs or system behavior.

Release notes are available for the releases in the following years:

- [2024 Releases](#)

2024 Releases

Release notes are available for the releases in the following months:

- [October 2024](#)
- [June 2024](#)

October 2024

The following sections list resolved and known issues applicable to AVEVA OMI 2023 R2 SP1.

- Resolved issues - 2023 R2 SP1
- Known issues - 2023 R2 P01

Resolved issues - AVEVA OMI 2023 R2 SP1

This table describes important resolved issues in AVEVA OMI. Many of these issues were also included in Patch Releases 01 though 04 of System Platform 2023. Some issues that have been fixed in this release may not be listed.

Originating Issue	Related Issues	Case/SR Number	Description
IMS 3266509		960403458	Some graphics and associated text intermittently displayed incorrectly upon initial loading.

IMS 3299530		960408823	When using SQL Server Reporting Services (SSRS), the web browser in OMI flipped back to the original date setting instead of using the updated date setting.
IMS 3425706		960437228	After the import of a translation file, OMI did not start properly on request.
IMS 3476034		960395585	Model Drive Manufacturing Execution System (MES) process selection was unavailable when it should have been.
IMS 3479556		960438667	The OMI application could not be properly started in read-only mode.
IMS 3289435		960410447	For popup windows in the OMI application, the "HideSelf" button functioned incorrectly by instantly reopening popup windows that the user had closed.

Known issues - AVEVA OMI 2023 R2 SP1

This section describes known issues that remain in Application Server 2023 R2 SP1 and AVEVA OMI 2023 R2 SP1.

Issue Number	Description
Security mismatch	<p>Some connected experience configuration scenarios can result in unexpected behavior.</p> <p>Scenario 1: Ability to access security tabs when an unsupported authentication mode has been configured.</p> <p>If a galaxy using Galaxy security has been created without connected experience, and then connected experience is configured, if you:</p> <ol style="list-style-type: none"> 1. Connect to the galaxy and change the

	<p>authentication mode to Authentication Providers.</p> <p>2. Add a new group and cancel the changes.</p> <p>You will be able to access the security tabs in an unsupported authentication mode.</p> <p>Scenario 2: Cancelling read-only message box in the security editor closes the IDE.</p> <p>If a galaxy using Galaxy security has been created without connected experience, then the IDE is closed, and then you configure connected experience and login to the IDE with an AVEVA Connect user, when you:</p> <ol style="list-style-type: none">1. Connect to the galaxy and click Yes when the dialog appears.2. Launch a second instance of the IDE and connect to the same galaxy.3. Navigate to the security page in the second IDE. A dialog opens and asks if you want to open in read-only mode.4. Click No. <p>The IDE closes (the IDE should not close in this scenario).</p> <p>Scenario 3: The Authentication Providers security mode can be enabled without configuring any roles.</p> <p>If you connect to a galaxy and select Authentication Provider security, and then navigate to the Roles tab, but:</p> <ol style="list-style-type: none">1. You do not add any roles.2. Click Save. <p>The IDE will shut down. Note that changes should not be allowed in this scenario since no role is configured and there is no Administrator login support under Operations Control mode.</p>
IMS 2897808	The SignedAlarmAck feature is not currently supported for Application Server and AVEVA OMI when AVEVA Operations Control connected experience is configured. Attempting to sign an alarm acknowledgement fails, and an error message is generated that warns of incorrect user credentials.
IMS 1869732	If the domain controller goes down and at the same

	<p>time, an IDE user attempts to perform a change user operation, there is a slight possibility that Windows will display a crash dialog for the IDE. Note: This is a very rare scenario.</p> <p>Workaround: Restart the IDE to proceed.</p>
IMS 1861501	<p>Endpoints on remote nodes, including redundant SSO nodes, are not discovered when the Global Discovery Server is not available. As a result, logging in to a Galaxy using Azure AD credentials when the security mode is Authentication Provider is not successful.</p> <p>Workaround: For discovery of any endpoints that are not on the local node, then a GDS (global discovery) server MUST be available for discovery. You can configure a secondary GDS to avoid non-discovery, which can occur when the redundant SSO is on a different node. You can configure secondary discovery servers in the System Platform IDE. See the Application Server help for instructions on configuring multi-galaxy communication. You can also configure a secondary discovery server through the PCS Configuration utility.</p> <ol style="list-style-type: none">1. Open the PCS Services Configuration utility C:\Program Files (x86)\Common Files\ArchestrA\ Services\aaServiceConfiguration.exe2. Open the Configure link (near the top of the utility window), the select Discovery Service from the menu.3. Enter a secondary node name for both the Local and Cross solution. "Local solution" is equivalent to Local Galaxy server in the System Platform IDE, and "Cross solution" is equivalent to Multi-Galaxy server.
IMS 1846869	<p>When System Platform is configured with a standalone System Management Server (SMS) and Authentication Provider, other nodes pointing to the SMS node may not be able to reach the Authentication Provider on the SMS node for user authentication.</p> <p>Workaround: Use one of the recommended SMS system configurations described in the <i>System Platform Installation Guide</i>.</p>
IMS 1842703	If, after configuring a node to connect to an existing System Management Server, you choose to make the node a redundant SSO server, the Configurator does

	not indicate that the Authentication Provider configuration parameters must be configured (green checkmark is still shown and the Configure button is grayed out). Workaround: Select the Authentication Provider configuration option and configure the parameters as described in the <i>System Platform Installation Guide</i> .
CR L00138837	AppEngine can crash due to calls made to API dlls even when using try/catch. This issue is specific to Nullable data types. The recommendation is to use wrapped API DLLs in Application Server scripting.
TFS 1404918	If a network failure occurs on the active engine node and Warm Redundancy is enabled, the last value is shown as NULL for slow-changing Tags. Workaround: Perform a SQL Insert or CSV Insert for the tag in question on the Historian side. This will restore the value in the Historian trend.

June 2024

The following sections detail new information applicable to the June 2024 release:

- Resolved issues - AVEVA OMI 2023 R2 Patch 01
- Known issues - AVEVA OMI 2023 R2 Patch 01

Resolved issues - AVEVA OMI 2023 R2 Patch 01

This table lists AVEVA OMI issues resolved by Patch 01. Some issues that have been fixed in this release may not be listed.

Original Issue	Related Issues	Case/SR Number	Description
IMS 839617		960372244	If you performed these steps: 1. Created a layout and exported it to a package (.aaPKG) file. 2. Imported the package file to a galaxy that used a different regional setting--for example, if the

Original Issue	Related Issues	Case/SR Number	Description
			<p>original GR node was set to Dutch(Belgium) and the new GR node was set to English.</p> <p>3. Tried to open the layout in the IDE. The layout would not open and you would see a message similar to this in the log: <i>System: 221; 0; 0; 0 is not a valid value for Int32.</i></p>
IMS 2507960		None	<p>Faceplate popup graphics opened using ShowGraphic appeared at the top-right corner of the window instead of in the center.</p>
IMS 2623094		960329675	<p>Certain OMI ViewApps would sometimes freeze for five to ten seconds while a user was browsing object attributes.</p>
IMS 2865278		960346739	<p>When using Multi-Galaxy Communication, running an OMI ViewApp that retrieved data from another galaxy failed if Remote Desktop Services was installed and enabled on the node where the ViewApp was running. A warning was written to the log.</p>
IMS 3023269		960368607	<p>If more than 30 users were connected to OMI using RDP sessions, the following warning message was logged when a user closed an</p>

Original Issue	Related Issues	Case/SR Number	Description
			application: <i>Release of global semaphore for the view application <app_name> failed. Deploy/undeploy operation may be blocked.</i>
IMS 3031945		960379257	Addressed issues with tooltip text wrapping and combo box size when viewing OMI ViewApps in the OMI web client.
IMS 3032032		960379261	In the OMI web client, when interaction animation was applied to both an outer object and an inner object, the animation for the outer object overlapped the animation for the inner object. This could make it impossible to select the inner object.
IMS 3037177		960383529	Changing the background color for the ScreenCaptureAndPrint App at design time did not change the color when the ViewApp was opened in runtime.
IMS 3040067		960383669	When opening a faceplate graphic in a popup in the OMI web client, the popup window included a minimize button.
IMS 3046276		None	Changes made to graphics contained in a ViewApp were not always reflected at runtime after the ViewApp was redeployed.
IMS 3060440		None	After migrating a galaxy from 2023 Patch 02 to 2023R2, an exception

Original Issue	Related Issues	Case/SR Number	Description
			could occur when previewing a ViewApp. A warning was written to the logger:
IMS 3060445		960385565	Content displayed through the ShowContent script in a tabbed multi-content layout pane with autofill enabled was not cleared when you changed the navigation node.
IMS 3061706		960386804	When a custom property was assigned with System.DateTime.Now and the SetCustomPropertyValue script was simultaneously called on the same custom property, it caused a memory leak. Eventually, this caused the viewer to crash with a <i>SystemOutOfMemoryException</i> error.
IMS 3199437		960392885	Switching between navigation items in the NavTree app in a ViewApp could cause the following message in the log: <i>System.ArgumentNullException: Value cannot be null.</i>

Known issues - AVEVA OMI 2023 R2 Patch 01

This section describes known issues that remain in Application Server and AVEVA OMI 2023 R2 Patch 01.

Issue Number	Description
IMS 2918856	<p>Note: This is an update to information found in the System Platform 2023R2 Readme file.</p> <p>If the Trend Pen graphic element and the</p>

Issue Number	Description
	<p>HistoricalTrendApp OMI ViewApp cannot retrieve historical data in the OMI web client, the reason may be that the Historian Server node name and port are not set in the AppSettings.json file on the node hosting the WebViewEngine. Unless the Historian Server is on the same node as the WebViewEngine, you must define the Historian Server system name and port number in AppSettings.json. Follow the procedure below.</p> <p>IMPORTANT: You must follow these steps even if you already did so before installing the patch. Because we have added support for multiple Historian Servers to the web client, the format of the AppSettings.json file is changed in Patch 01. An AppSettings.json file from the 2023R2 release will not work with 2023R2 Patch 01.</p> <p>For your reference, the patch installation makes a copy of the existing AppSettings.json file with a .bak extension. Do not just rename or copy this file for use with Patch 01. Because of the file format change, it will not work.</p> <ol style="list-style-type: none"> 1. On the node hosting the WebViewEngine, make a copy of the file C:\Program Files (x86)\Common Files\Archestra\Services\proxyserver\appsettings.json for safekeeping. Then, open the original file in a text editor. 2. Find this section in the file: <pre>//"history-data-{serverName}": { // "clusterId": "history- data-{serverName}", // "CorsPolicy": "customPolicy", // "Match": { // "Path": "/history- data/{serverName}/{**catch-all}" // }, // "Transforms": [// { // "PathRemovePrefix": // "/history-data/{serverName}" // } //], //}, // "history-data-{serverName}": { // "HttpClient": { // "BaseAddress": "http://127.0.0.1:5000" // } // } }</pre>

Issue Number	Description
	<pre data-bbox="915 297 1514 846"> "DangerousAcceptAnyServerCertificate": "true" // }, // "HttpRequest": { // "Version": "1.1", // "VersionPolicy": // "RequestVersionExact" // }, // "Destinations": { // "destination1": { // "Address": // "[https://https://{serverName}:32573/" // } // }, // "UseDefaultCredentials": // "true" //}, </pre> <p data-bbox="878 868 1514 1368"> 3. Make two changes to this section: a. Uncomment the section by removing the // at the beginning of each line. b. Change each instance of {servername} to the server name of the Historian Server node. There are six, shown in bold in the sample above. 4. If you want to define more than one Historian Server so that users can choose the one they want at runtime, copy this section and change the server name to the one for the additional Historian. Repeat this as needed. 5. Save your changes and close the file. </p> <p data-bbox="915 1389 1514 1664">Note: This is only one possible cause of failure for the HistoricalTrendApp and the Trend Pen element in the OMI web client. It may also be due to a security certificate issue. See the description of IMS 2918856 in the "AVEVA OMI Issues" section of the <i>System Platform 2023 R2 Readme</i> file for more information and the workaround steps for the security certificate issue.</p>
IMS 3057574	<p data-bbox="878 1712 1514 1896">Changes made to an element style (for example, adding an animation or changing the line color) are not reflected when an OMI ViewApp is opened with the web client, even if the ViewApp is redeployed. The changes do appear in the desktop client.</p> <p data-bbox="878 1907 1421 1945">Workaround: Undeploy and redeploy the OMI</p>

Issue Number	Description
	VlewApp.

About Operations Management Interface (OMI)

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

Select a workflow group below for more information about using AVEVA OMI to create a View Application that can monitor and manage your production processes.

Design Standards



Build Applications



Deploy Applications



Run Applications



If you want to watch videos that demonstrate how to complete important workflows within each workflow group, see the [Tutorial videos](#)

Additional information about creating content for your ViewApps is divided by function: System Platform IDE, Application Server Scripting, and the Graphic Editor.

System Platform IDE



Application Server Scripting



Industrial Graphic Editor



Tutorial videos

This collection of short videos demonstrates some of the important workflows to create and run a ViewApp. The videos are organized into the typical workflow groups to design, build, deploy, and run a ViewApp.

Double-click on a running video to expand it to full screen. Press the Esc key to return the video to its original size.

To view these videos on a Windows Server operating system, you may need to enable "Desktop Experience". For more information about enabling Desktop Experience, see the documentation for your Windows Server version.

Galaxy Infrastructure	Create a Galaxy			
Standards	Configure a new asset and graphic	Configure an instance	Create a graphic	
Object Wizards	Navigate the Object Wizard interface - part 1	Navigate the Object Wizard interface - part 2	Add an Object Wizard to a template	
Screen Profiles	Create a screen profile	Configure a screen profile		
Layouts	Create a layout	Layout Editing Modes	Configure a Layout	Add a pane to a layout
			Create and configure a responsive layout	
ViewApp Namespaces	Create a ViewApp namespace	Configure ViewApp Namespace Attributes		
Apps	Add an external content item			
ViewApps				
ViewApp Deployment	Deploy a ViewApp			
Run ViewApps	Launch a ViewApp			

Note: For easier viewing, double-click on running videos to expand them to the full size of your screen. Select your Esc key to return the video to its original size.

Create a Galaxy

Video Tutorial: Create a galaxy in AVEVA OMI

https://player.vimeo.com/video/992308785?badge=0&autoplay=0&player_id=0&app_id=58479

Navigate the Object Wizard interface - part 1

Video Tutorial: Navigate the object wizard interface in AVEVA OMI part 1

https://player.vimeo.com/video/992306518?badge=0&autoplay=0&player_id=0&app_id=58479

Navigate the Object Wizard interface - part 2

Video Tutorial: Navigate the object wizard interface in AVEVA OMI part 2

https://player.vimeo.com/video/992306382?badge=0&autoplay=0&player_id=0&app_id=58479

Add an Object Wizard to a template

Video Tutorial: Add an ObjectWizard to a template in AVEVA OMI

https://player.vimeo.com/video/992308285?badge=0&autoplay=0&player_id=0&app_id=58479

Configure an instance

Video Tutorial: Configure an instance in AVEVA OMI

https://player.vimeo.com/video/992308542?badge=0&autoplay=0&player_id=0&app_id=58479

Configure a new asset and graphic

Video Tutorial: Create a new asset and graphic in AVEVA OMI

https://player.vimeo.com/video/992309822?badge=0&autoplay=0&player_id=0&app_id=58479

Create a graphic

Video Tutorial: Create a graphic (symbol) in AVEVA OMI

https://player.vimeo.com/video/992310065?badge=0&autoplay=0&player_id=0&app_id=58479

Create a screen profile

Video Tutorial: Create a screen profile in AVEVA OMI

https://player.vimeo.com/video/992309954?badge=0&autoplay=0&player_id=0&app_id=58479

Configure a screen profile

Video Tutorial: Configure a screen profile in AVEVA OMI

https://player.vimeo.com/video/992308648?badge=0&autoplay=0&player_id=0&app_id=58479

Create a layout

Video Tutorial: Create a layout in AVEVA OMI

https://player.vimeo.com/video/992308927?badge=0&autoplay=0&player_id=0&app_id=58479

Add a pane to a layout

Video Tutorial: Add a pane to a layout in AVEVA OMI

https://player.vimeo.com/video/992308228?badge=0&autoplay=0&player_id=0&app_id=58479

Create a ViewApp namespace

Video Tutorial: Create a ViewApp namespace in AVEVA OMI

https://player.vimeo.com/video/992306078?badge=0&autoplay=0&player_id=0&app_id=58479

Deploy a ViewApp

Video Tutorial: Deploy a ViewApp in AVEVA OMI

https://player.vimeo.com/video/992311905?badge=0&autoplay=0&player_id=0&app_id=58479

Launch a ViewApp

Video Tutorial: Launch a ViewApp in AVEVA OMI

https://player.vimeo.com/video/992306707?badge=0&autoplay=0&player_id=0&app_id=58479

Add an external content item

Video Tutorial: Add external content in AVEVA OMI

https://player.vimeo.com/video/992306989?badge=0&autoplay=0&player_id=0&app_id=58479

Create and configure a responsive layout

Video Tutorial: Introduction to responsive layouts in AVEVA OMI

https://player.vimeo.com/video/992306239?badge=0&autoplay=0&player_id=0&app_id=58479

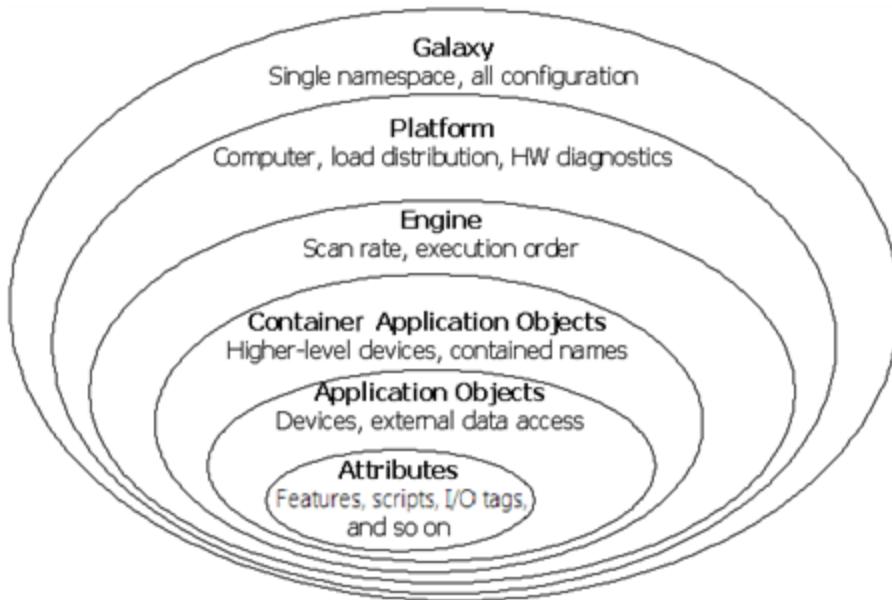
Design standards

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

Galaxy Infrastructure 	Configure components: Galaxies	About the integrated development environment (IDE)	Platform common services
Standards 	Automation objects	Object Wizards	Symbol Wizard Editor
Visual Styles 	About the galaxy style library	About element styles	Format Styles tab About the quality and status tab

Configure components: Galaxies

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



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

Create a Galaxy

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

Guidelines for creating a new galaxy

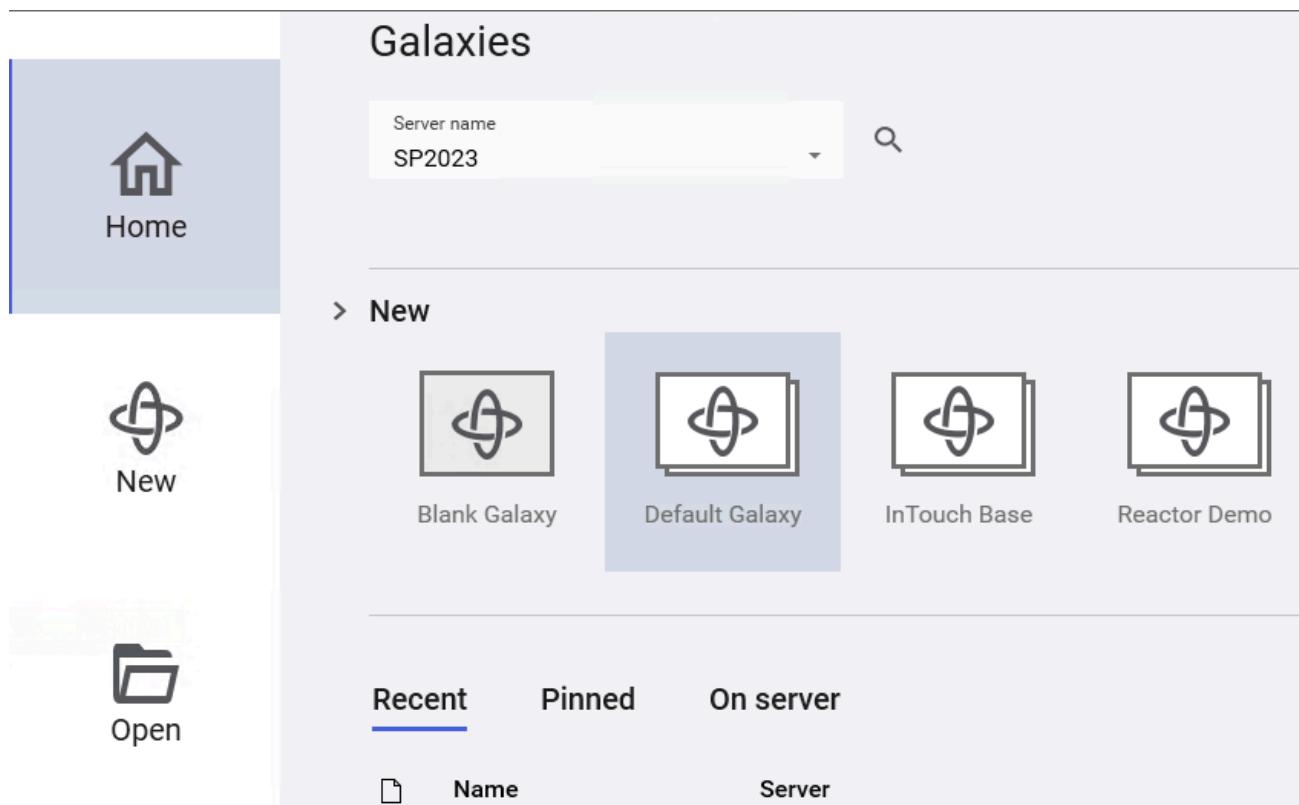
- Activate a valid development license before opening the IDE.
- You can only create a new Galaxy on a computer with the Bootstrap and the Galaxy Repository software installed. This is referred to as the GR node.
- The computer name of the GR node, as well as the name of the domain on which the computer exists, should comply with NETBIOS and DNS guidelines. Names should include only English language alphanumeric characters. The use of special characters or characters from other language character sets that are not also included in the English language character set can result in a failure to create the new Galaxy.
- To be able to create a new Galaxy, TCP/IP must be enabled on the computer hosting the SQL Server database.

The TCP/IP protocol setting can be verified with **SQL Server Configuration Manager**. See the *System Platform Installation Guide* for instructions on enabling the TCP/IP protocol for SQL Server.

- New Galaxies are created without security. To learn more about setting security for your Galaxy, see Working with Security.

To create a new galaxy

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



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

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

The **New Galaxy** dialog box opens.

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

Note: You cannot use the following reserved names as Galaxy names: Me, MyContainer, MyArea, MyHost, MyPlatform, MyEngine and System. You cannot use a name that conflicts with an existing object in the template.

5. To open the galaxy after it is created, make sure **Connect to this Galaxy** is selected.
6. Click **Create**. The **Create Galaxy** dialog box opens and shows the progress of the Galaxy database being

created.

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

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

About Galaxies

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

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

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

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

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

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

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

Connect to a Galaxy

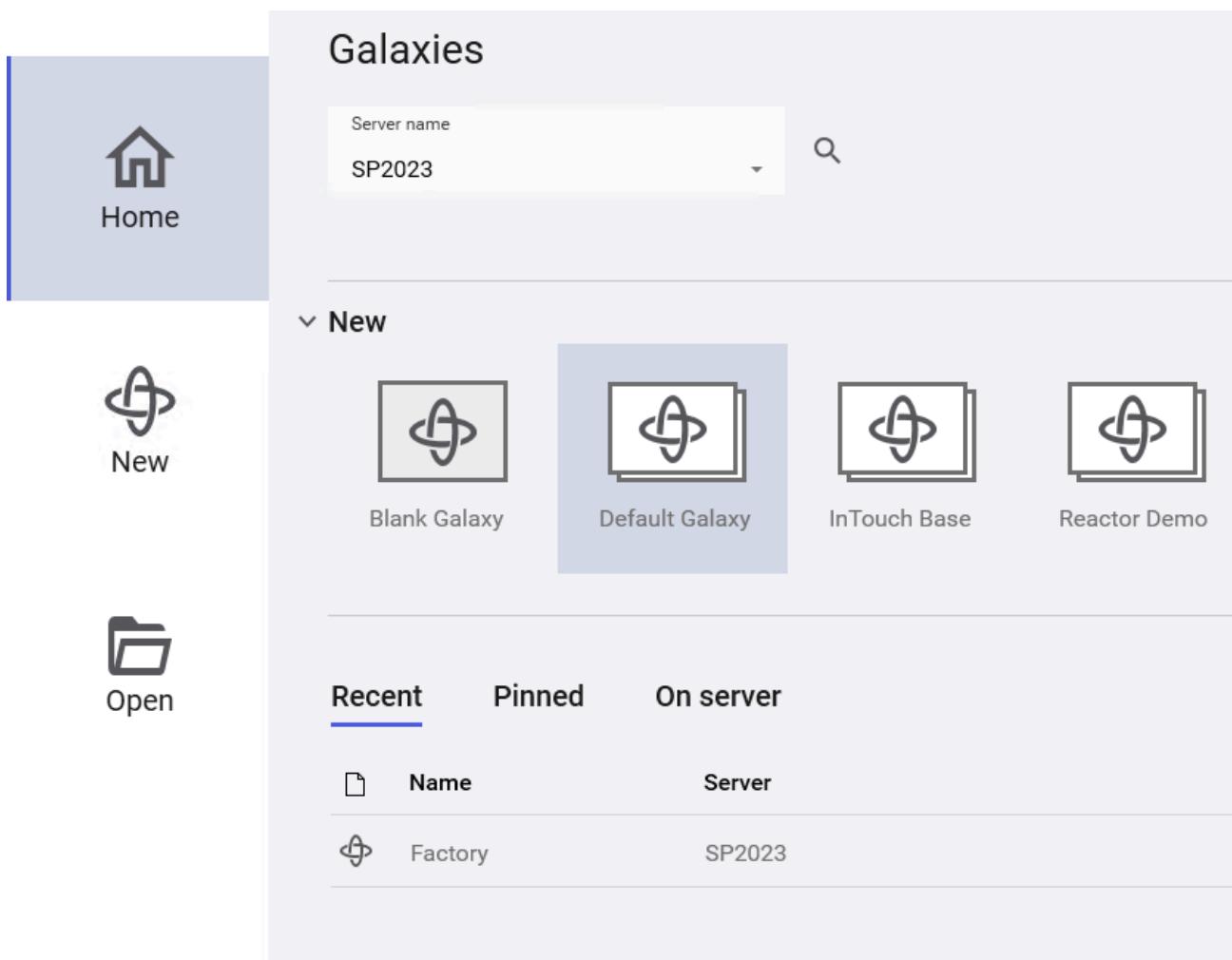
You must connect to a Galaxy before you can perform work in it. To connect to a Galaxy, you must be a member of the **aaConfigTools** OS group *on the GR node*. To add a user to the **aaConfigTools** group, use the Windows Control Panel. If you are not a member of **aaConfigTools**, you can instead start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

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

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

To connect to a galaxy

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



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

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

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

To log in to a galaxy

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

Once the Galaxy opens in the IDE you can start working with your Galaxy. The IDE provides different ways to view the structure of the Galaxy and how it will be deployed to runtime. See [IDE application views](#) for more information.

View the configuration file

A configuration file, `view.exe.config`, is used define various settings for use by AVEVA OMI ViewApp applications at runtime. Most of these settings are used to load different runtime dependent assemblies, and do not require any changes. The location of `view.exe.config` will vary, depending on the selected installation directory.

The default location for the file is:

`C:\Program Files (x86)\Archestra\Framework\Bin\ViewAppFramework`

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

Set single sign-on for OMI ViewApps

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

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

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

Turn off multithreaded processing for OMI ViewAPP

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

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

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

About the integrated development environment (IDE)

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

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

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

Install and activate your license

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

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

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

Manage security certificate verification

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

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

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

View the Galaxy

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

- **Template folder:** this contains different object templates that you use to build a digital twin of your physical plant. The digital twin is a set of objects that interfaces with PLCs, RTUs, or other controllers. Expand the top level folders to see the different templates in the folders. For more information, see [Template folder](#). For a complete discussion of template and instance objects, see [About templates and instances](#).

- **Visualization folder:** this contains different graphic models that you use to construct the runtime graphical representation of your physical plant, including gauges, equipment, and valves. The graphic models in the folder include Industrial Graphics and Situational Awareness Library graphics. For more information, see [Visualization folder](#).

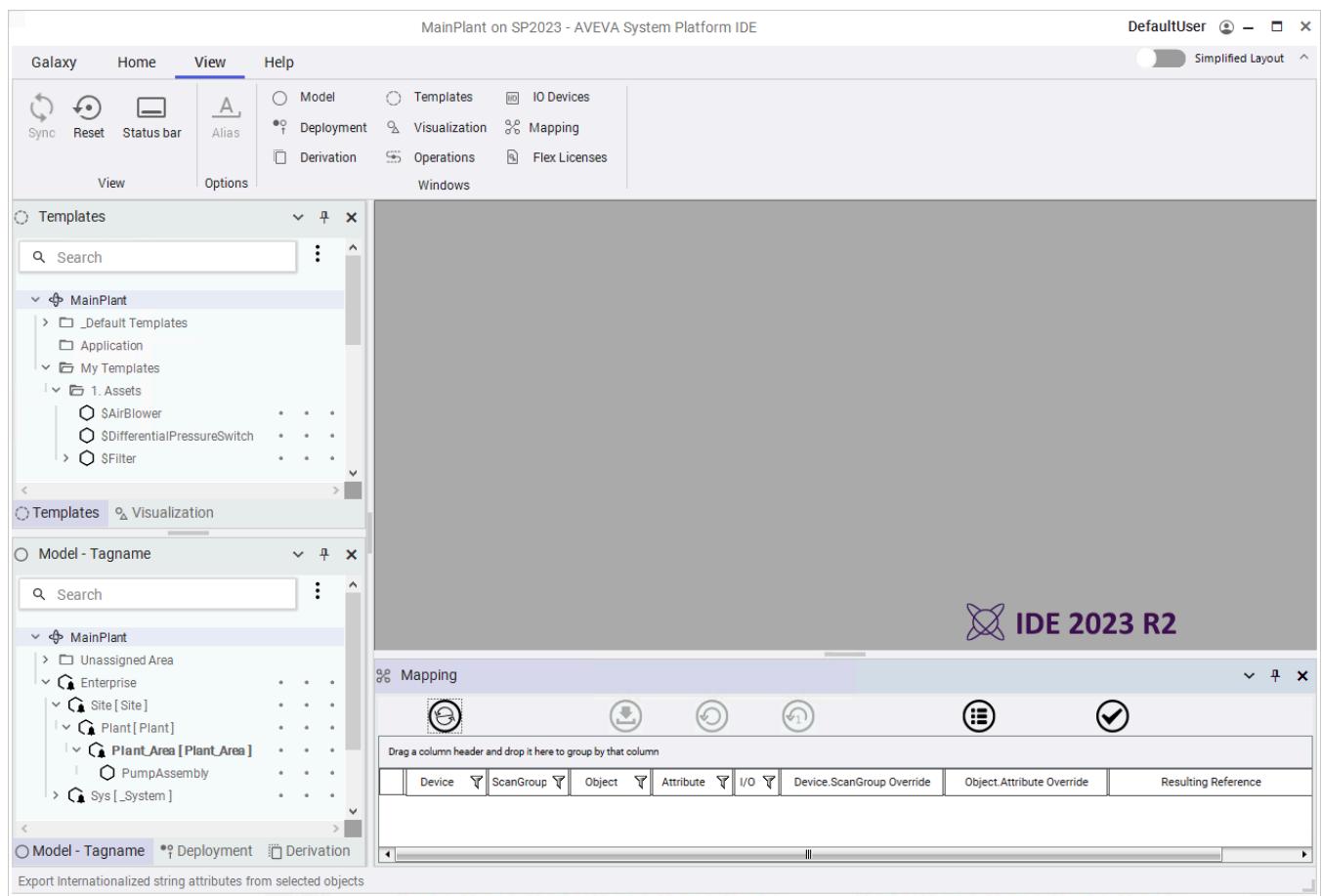
When the Galaxy opens, the application views are shown by default in the lower left pane. See [IDE application views](#) for more information.

- **Model view:** this shows the object relationship to the automation scheme layout. The objects are organized into Areas that typically represent the physical plant layout.
- **Deployment view:** this shows the object relationship to the computers that comprise the deployed system that the objects run on.
- **Derivation view:** this shows the derivation path from base template to the instances. This view allows a user to see all object instances that were based on a given template. All templates and instances appear in this view.

You can open additional application views:

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

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



Template folder

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

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

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

Visualization folder

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

- The Default Galaxy includes several graphic libraries, including the Industrial Graphics Library, the Situational

Awareness Library, and an icon library. The Default Galaxy also includes various Screen Profiles, Layouts, Apps and Controls.

- The Blank galaxy includes only a basic set of Apps and Controls.

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

CONNECT

CONNECT can be configured within the Visualization folder to provide centralized storage and access for reusable Industrial Graphics. You can use these graphics as templates to simplify building applications. For more information, see [About using CONNECT](#).

IDE application views

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

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

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

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

View	Shortcut
Model View	Ctrl+Shift+M
Deployment View	Ctrl+Shift+D
Derivation View	Ctrl+Shift+R
Template Folder	Ctrl+Shift+T
Visualization Folder	Ctrl+Shift+V
Operations View	Ctrl+Shift+O
IO Devices View	Ctrl+Shift+I
Mapping view	Ctrl+Shift+G

Application object icons

All IDE views (Templates, Visualization, Model, Deployment, Derivation, and IO Devices) allow you to choose whether to show objects in tree (hierarchical) view or in a list (flat) view. By default, objects are shown in a tree.

To change to the view selection, click the three vertical dots at the top right of the view window. Default objects are shown in bold.

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

System object icons



Galaxy (project)



GRPlatform



WinPlatform



Cloud Platform



AppEngine



Primary (1) Redundant AppEngine



Backup (2) Redundant AppEngine



Area



Application Object (asset)

View engine icons



View Engine



Webview Engine

ViewApp icons



OMI ViewApp



InTouch ViewApp

Device integration icons



DDE SuiteLink Client

InTouch Proxy

OPC Client



Redundant DI



Scan group

Visualization icons

	Screen profile
	Layout
	Symbol
	External content
	Namespace
	Client control
	OMI app
	Widget

Status Icons

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

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

Validation Status	Editing Status	Deployment Status

Redundant AppEngine Validation Status

See [AppEngine Redundancy](#) for more information about working with redundant AppEngines.

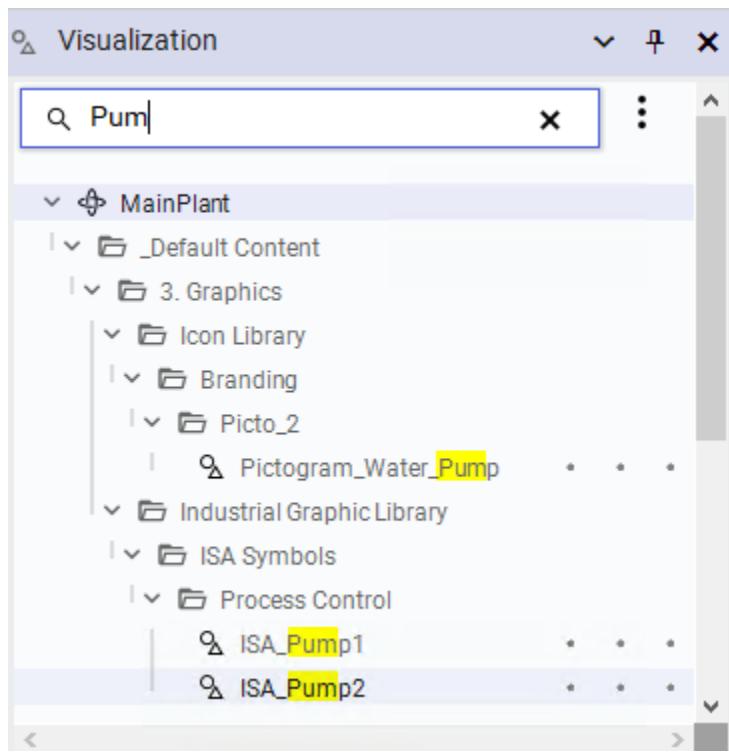
	Both redundant AppEngines are not deployed
	This AppEngine is not deployed, but its redundant partner is deployed.
	This AppEngine is deployed, but its redundant partner is not deployed.

	Deployed, but the redundant AppEngines have a pending configuration update.
	This AppEngine has a pending configuration update, but its redundant partner is up-to-date.
	Both redundant AppEngines have pending software updates.
	This AppEngine has a pending software update, but its redundant partner is up-to-date.
	This AppEngine is up-to-date, but its redundant partner has a pending software update.

Perform a search in a view

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

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



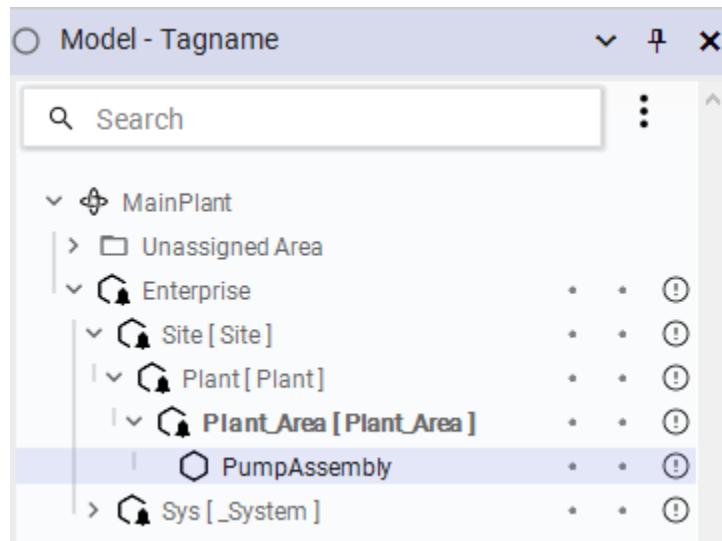
Note: In Model view, the search includes either object tagnames or aliases, whichever is currently shown.

Model view

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

represents an application perspective of the processes that users are emulating: for instance, specific process areas, tanks, valves, pumps and their relationships based on containment. For more information about containment, see [Create contained templates](#).

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



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

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

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

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

Some objects' hierarchical, or contained, names are truncated if you have multiple levels shown. To view the entire hierarchical name, select the object and select **Properties** on the **Home** ribbon. The entire hierarchical name is shown in the **Properties** dialog box. For more information about hierarchical names, see [Use contained names](#).

- Objects that currently do not belong to an Area are listed under Unassigned Area. Containment relationships between parent and child objects are shown there.
- In each branch of the tree, objects are listed in alphabetical order. Default objects are shown in bold.



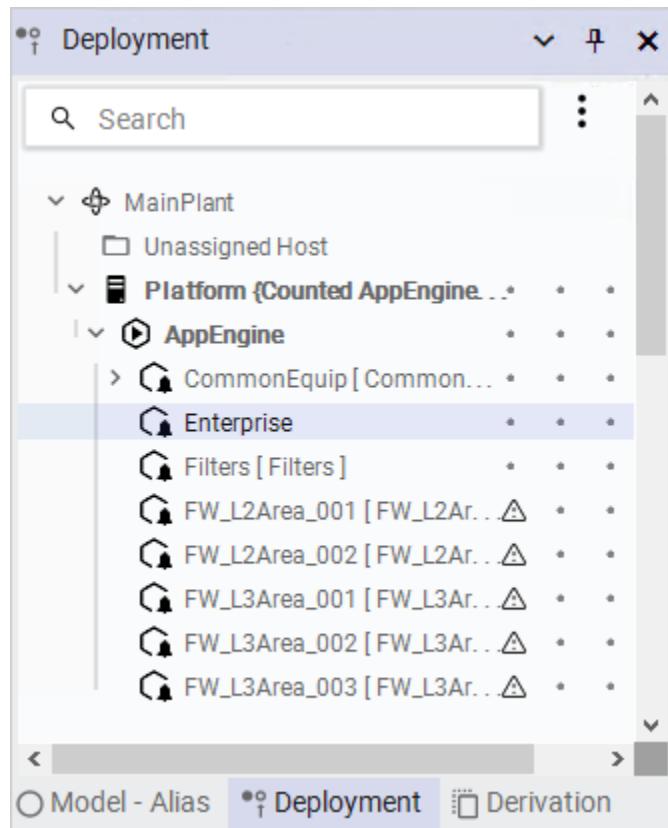
To assign an object to another, drag it onto the host object. If the object is an inappropriate assignment match, the international Not symbol appears. To unassign an object, drag it to the **Unassigned Host**.

folder.

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

Deployment view

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



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

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

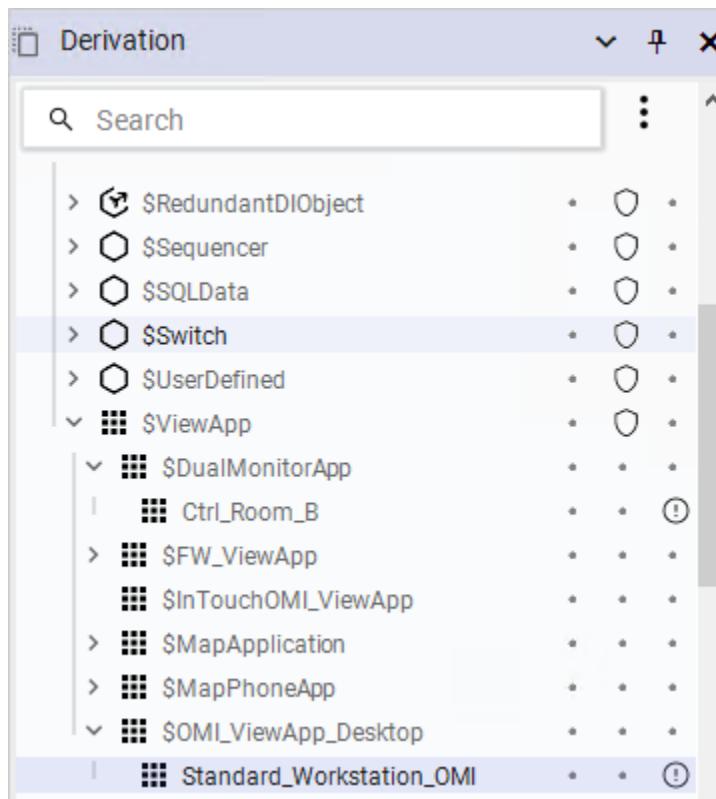
- The top of the tree is the Galaxy.
- WinPlatforms are shown under the Galaxy.
- Under each WinPlatform, assigned AppEngines are listed.
- Under each AppEngine, assigned Areas and objects configured to communicate with PLCs, RTUs, etc. such as OPCClient objects, are listed.

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

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

Derivation view

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



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

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

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

Objects with names that start with a "\$" are templates, either base or derived. Under each branch of the tree,

child objects are listed in alphabetical order.

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

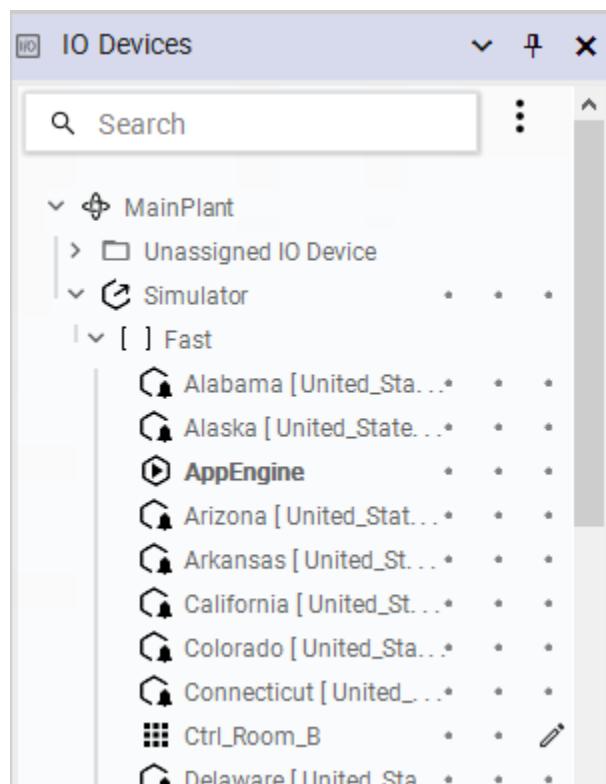
IO devices view

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

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

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

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



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

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

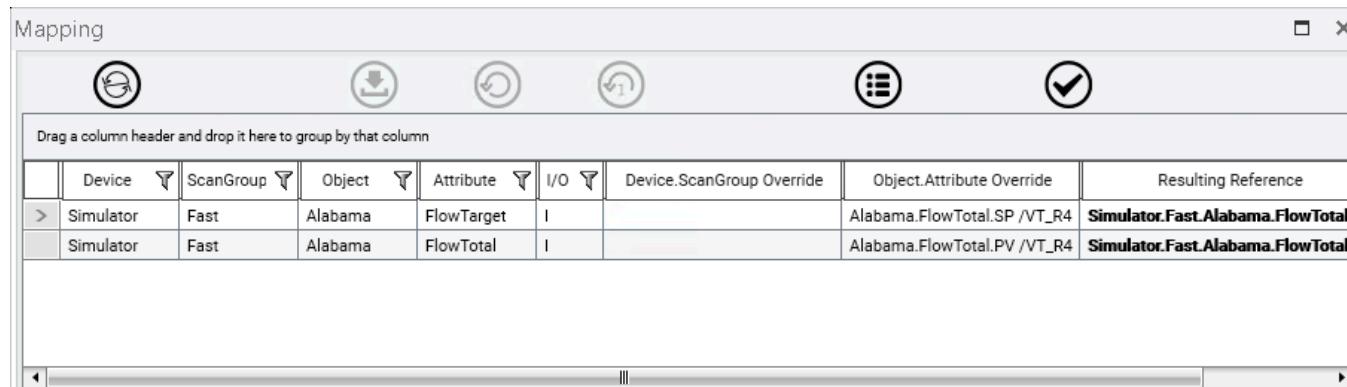
Device linkage, which is established by assigning a system or application object to a scan group, ensures that all of the object's attributes that have been prepared for automatic I/O assignment will match up with a corresponding input source or output destination in the scan group. The I/O references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

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

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

Mapping view

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



	Device	ScanGroup	Object	Attribute	I/O	Device.ScanGroup.Override	Object.Attribute.Override	Resulting Reference
>	Simulator	Fast	Alabama	FlowTarget	I		Alabama.FlowTotal.SP /VT_R4	Simulator.Fast.Alabama.FlowTotal.
	Simulator	Fast	Alabama	FlowTotal	I		Alabama.FlowTotal.PV /VT_R4	Simulator.Fast.Alabama.FlowTotal.

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

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

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

Operations view

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

Operations				
	Name	State	Status	Command result
	■■■ \$InTouchOMI_ViewApp	○ ✓ • •	Good	Succeeded - Validation completed
	■■■ \$OMI_Control_Room_3	○ ✓ • •	Good	Succeeded - Validation completed
	■■■ OMI_Control_Room_3S	○ ✓ • ⓘ	Good	Succeeded - Validation completed

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

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

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

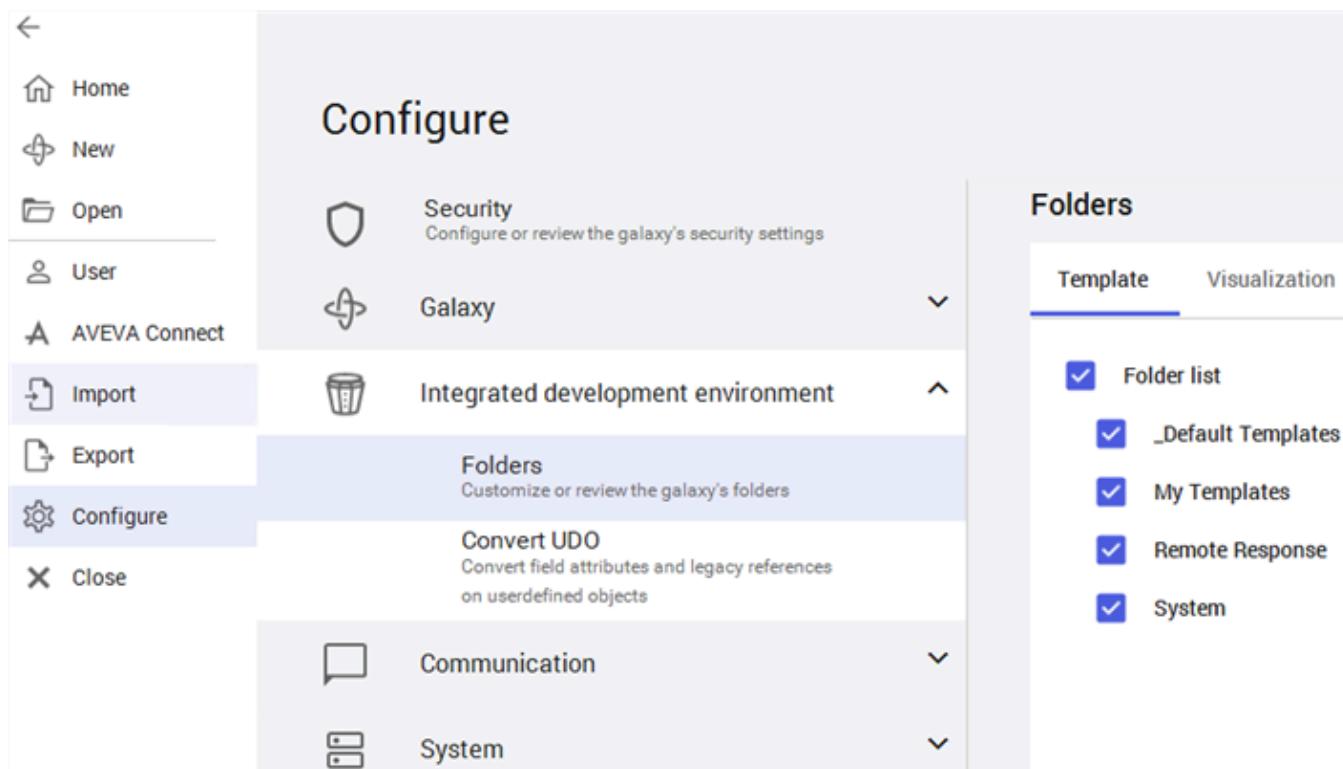
When validation is complete, the **Command Result** column shows a "Succeeded" or "Failed" message and additional information about the validation results. For more information about validating objects, see [Validate objects](#).

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

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

Customize your workspace

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



Dock views

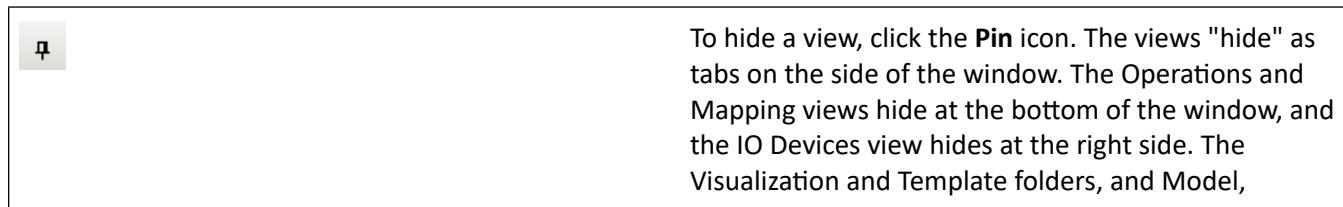
To dock a view to a new location:

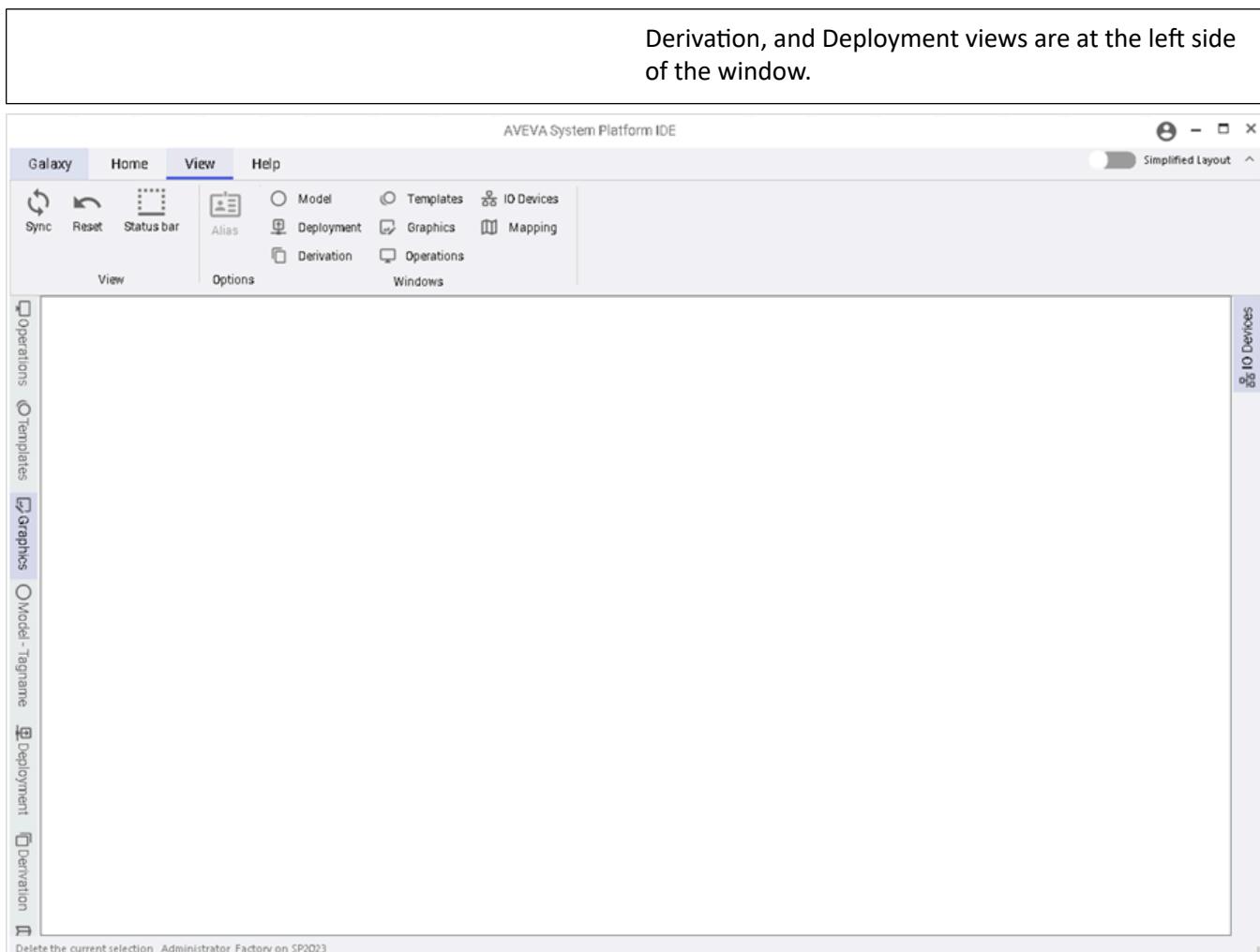
1. Place the cursor on the title bar of the view (or folder) and hold down the left mouse button.
2. As you drag the view from its position, blue directional arrows appear on the screen. While continuing to hold down the mouse button, position the cursor on one of the arrows. The position on the screen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

Float views

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

Hide views





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

Reset the workspace

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

To return the views to the default docking

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

Synchronize views

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

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

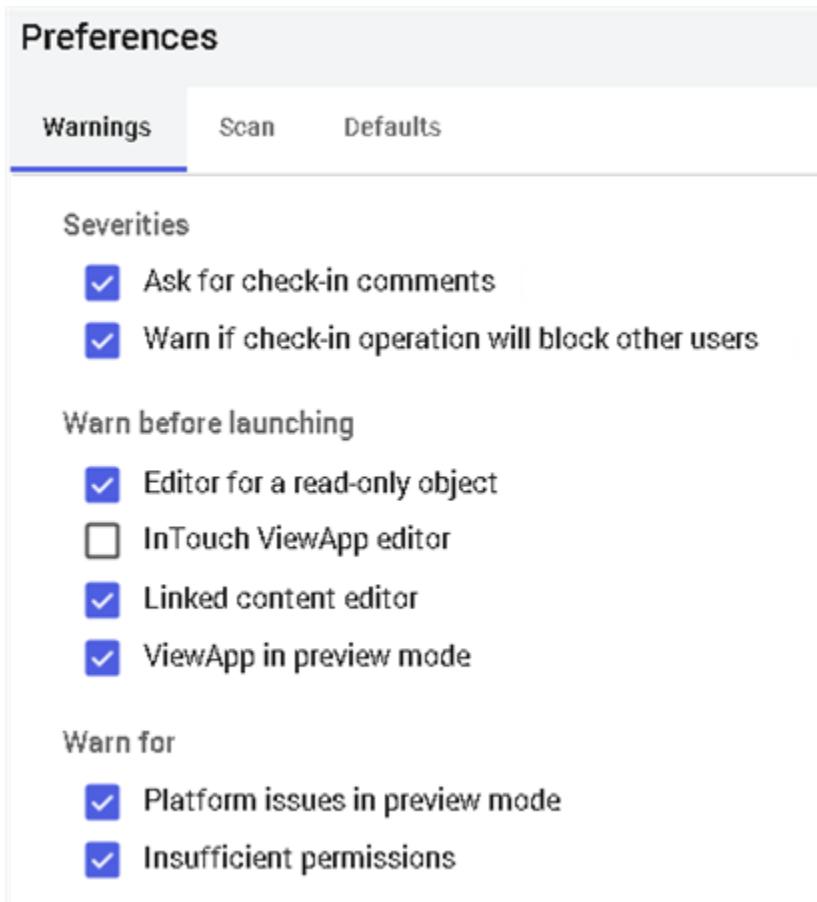
Configure user preferences

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

If you specify a security group, that security group must already exist. For more information about security and security groups, see [Work with security](#).

To configure user information

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



- On the **Warnings** page, select the check box for each option you want:
 - Ask for check-in comments** to be prompted to type comments when checking in templates and objects.
 - Warn if check-in operation will block other users** if you want to be warned if this occurs.
 - Editor for a read-only object** to see a prompt that informs you if you are opening an instance or template as read-only. The prompt warns you if you open an instance or template while someone else is working on it. If someone else is working in the instance or template, you cannot make changes.
 - InTouchViewApp editor** to see a prompt each time you attempt to edit an InTouchView application instance. You can edit the associated template or cancel the operation. If you do not select this check box, the request to edit the InTouchViewApp instance is automatically redirected to the associated template.

- **Linked content editor** to see a prompt each time you attempt to edit a Visualization folder item, such as a graphic or layout, that is linked to an object. You can edit the linked content or cancel the operation. If you do not select this check box, the linked content opens in its editor (graphics in the Graphic Editor, layouts in the Layout Editor, etc.) without a warning, and you will be able to make changes to the content. If you then save the content, the item in the Visualization folder is updated with your edits, and the changes propagate to all objects that link to the content.
- **ViewApp in preview mode** to restore the first warning message that appears after requesting a preview of a ViewApp. You need to select this prompt to display the message again after you select the check box on the warning message to hide the message.
- **Platform issues in preview mode** to restore the warning message that appears after requesting a preview without the WinPlatform deployed. You need to select this prompt to display the message again after you select the check box on the warning message to hide the message.
- Select the **Warn for insufficient permissions** to see a prompt that tells you if you have permission to create or modify InTouchView applications. These permissions authorize or prevent you from creating and modifying InTouchView Applications.

3. On the **Scan** page:

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

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

- Enable or disable **Auto context selection**. This setting is enabled by default:
 - When **Auto context select** is enabled (checked), an object selected in the **Model**, **Derivation**, or **Deployment** view remains selected when you switch between views.
 - When **Auto context select** is disabled (unchecked), the **Model**, **Derivation**, and **Deployment** views do not keep the same object selected as you switch between views, so a different object can be retained in each view.
- In the **Field Attributes** section, select the **Show Field Attributes** check box to display the **Field Attributes** tab in the User Defined Object editor. By default, the **Field Attributes** tab is displayed only if field attributes are already defined for the object. You can override the default by selecting this option. For more information about working with legacy field attributes, see [Field attributes, UDAs, and extensions pages](#) in the Application Server help file

4. On the **Defaults** page, type the object names of the Framework objects you want to be the defaults with respect to assignment relationships.

- **Platform** name.
- **Application engine** name.
- **Area** name.
- **View engine** name.
- Your **Security group** name, if any.

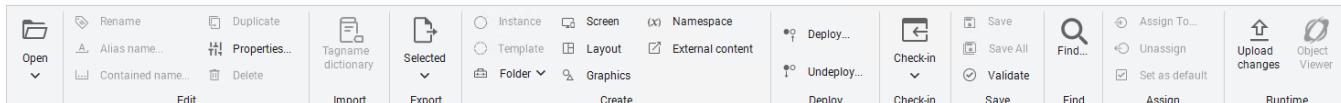
Leave a field blank if you do not want a default for that type of object.

Your changes are saved automatically.

- When you are done, click **OK**.

Use the simplified layout

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



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



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

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

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

Log on with security enabled

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

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

For information about setting up security in the System Platform environment, see [Work with security](#).

To log on to a Galaxy

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

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

Change users

You can change users in a Galaxy at any time. Any security restrictions associated with a user change when the user logs on or logs off from the Galaxy. For more information about users and security, see [Work with security](#).

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

To change users

- From the ribbon, select **Galaxy**, then select **User**.

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

Search for content

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

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

Asset Searches

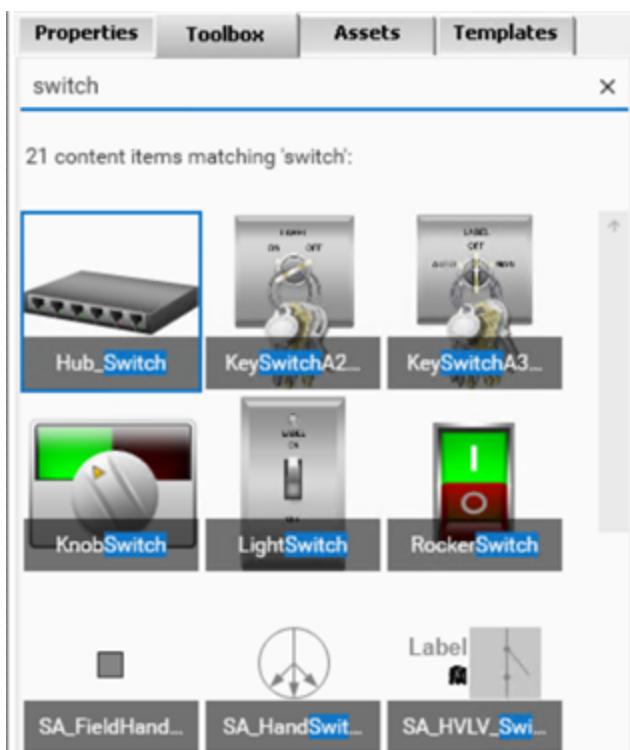
An Assets search is applied across the entire Galaxy. The search results show the first batch of Areas and Assets containing the search term in a multi-column list displayed in ascending alphabetical order. A message beneath the **Search** field indicates the number of items found in the search that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

Assets	WaterPlant
Filter	WaterPlant ▶ SystemArea ▶ Simulation
Filter	WaterPlant ▶ SystemArea ▶ Simulation
Filter	WaterPlant ▶ SystemArea ▶ Simulation
Filter	WaterPlant ▶ SystemArea ▶ Simulation
Filter	WaterPlant ▶ SystemArea ▶ Simulation

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

Toolbox Searches

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



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

About the Industrial Graphic Editor

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



- **Properties grid**

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

- **Toolbox Tab**

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

- _Default Content contains numbered subfolders for default Screen Profiles, Layouts, Graphics, Apps, and ViewApp Namespaces (the content of ViewApp Namespaces is not applicable to the Industrial Graphics Editor and thus is not displayed).
- MyContent contains a similar set of (empty) numbered subfolders. These are provided as a ready-to-use repository for any new content that you add or derive from the default content.
- Any folders you create also appear in the **Toolbox** view. A breadcrumb trail above the folder structure shows the current selected folder.

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

- **Assets Tab**

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

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

- **Templates Tab**

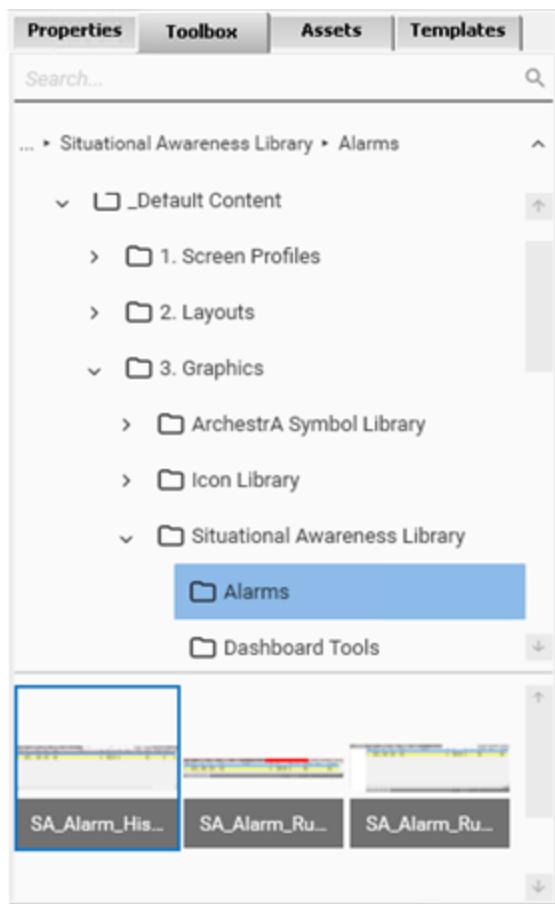
The **Templates** tab shows the folder tree structure of the Galaxy's Template folder consisting of folders and object templates. As in the **Toolbox** tab, the **Templates** view organizes the templates under two main folders, "_Default Content" and "My Content."

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

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

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

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



Platform common services

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

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

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

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

Configure and deploy PCS services

The configuration tool and workflow are the same for all user-configurable PCS services. To configure a PCS service, you will need to perform the following tasks:

1. Create an instance of a PCS service

2. Assign the service instance to a node
3. Deploy the instance

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

Create an instance of a service

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

Assign the service instance to a node

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

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

Deploy the instance

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

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

- [Configure the ASBGRBrowsing service](#)
- [Configure the ASBMxDataProvider service](#)
- [Configure the ASBAuthentication service](#)

Configure service TCP ports

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

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

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

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

services to function.

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

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

7502, and 7503 to be open. Three galaxies would require ports 7500-7505 to be open.

Configure the ASBGRBrowsing service

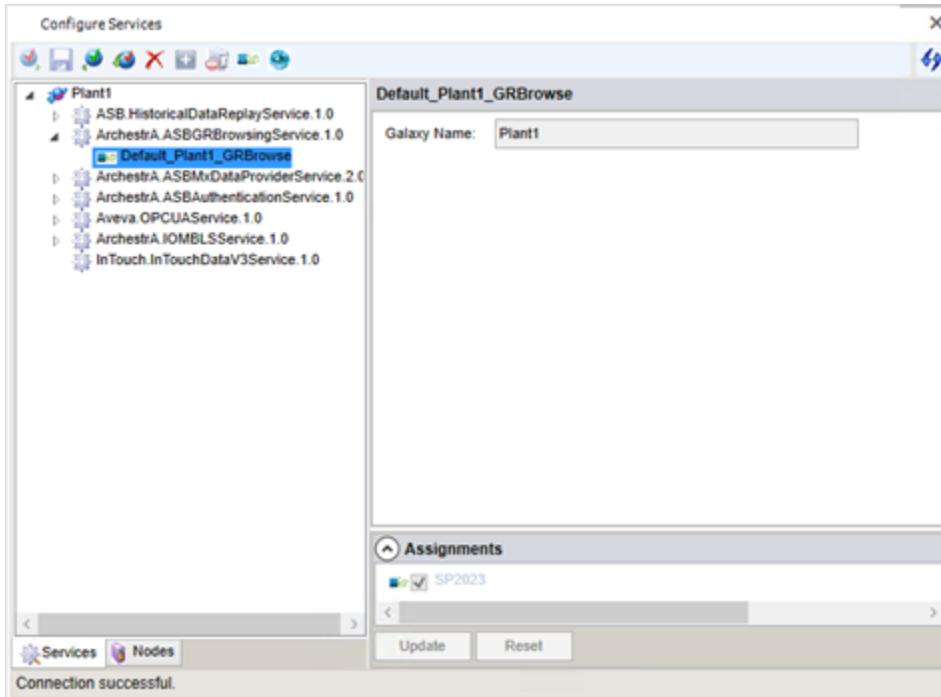
The ASBGRBrowsingService enables you to browse Application Server namespaces.

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

To configure the ASBGRBrowsing Service

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

The **Configure Services** utility opens.



3. Expand the tree view. Right-click **ASBGRBrowsingService**, and then select **Create**. You can also select **ASBGRBrowsingService**, then type **CTRL+N**. The new instance appears in the tree structure.
4. Right-click the instance name and select **Check-out**
5. Enter the Galaxy name in the **Galaxy Name** box.
6. In the **Assignments** area, select the node you want to assign to the instance, and then select **Update**.
Note: The **Update** button is enabled only when you select a node. You cannot delete a node that is already assigned to a service instance. To delete a node, you must first unassign the node from the service instance and then delete it.
7. On the left-pane, right-click the instance, and then select **Check-in**.
8. On the left-pane, right-click the instance, and then select **Deploy**. A message appears indicating whether the

node is successfully deployed.

Configure the ASBMxDataProvider service

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

To configure the ASBMxDataProvider Service

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

Configure the ASBAuthentication service

The ASBAuthenticationService provides user authentication.

To configure the ASBAuthentication Service

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

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

IOMBLSService

The IOMBLSService provides services for browsing, configuring, and deploying objects and visual elements. All the services provided by IOMBLSService are hosted in a single executable (IOMBL.Sexe). IOMBLService is not

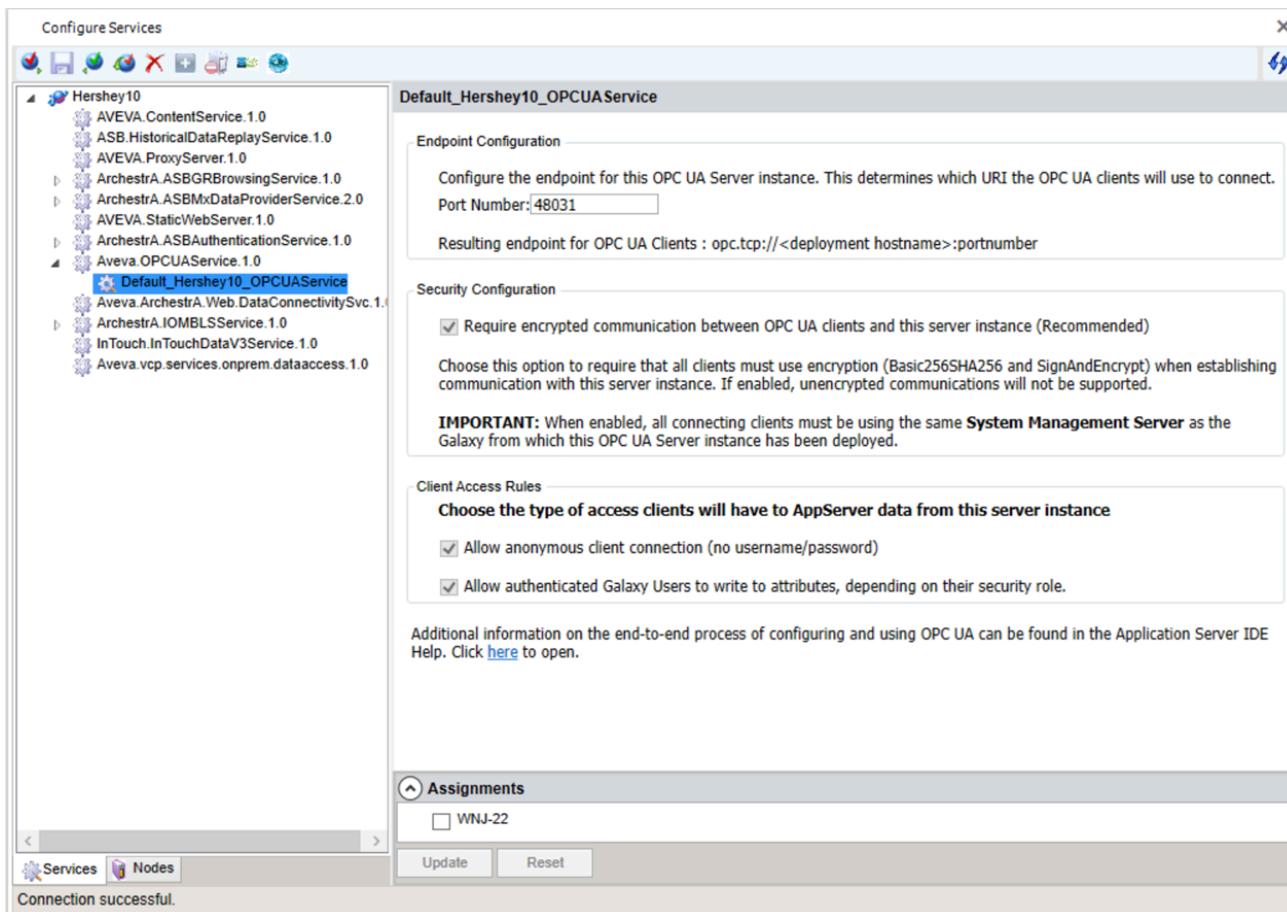
user-configurable, and runs in the background. Do not delete, duplicate, close, or modify this service in any way.

Configure and deploy the OPC UA service

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

To configure and deploy the OPC UA Server Service

1. On the IDE ribbon, select **Galaxy** then select **Configure**
2. Select **System**, then select **Services**.
3. The **Configure Services** utility opens.



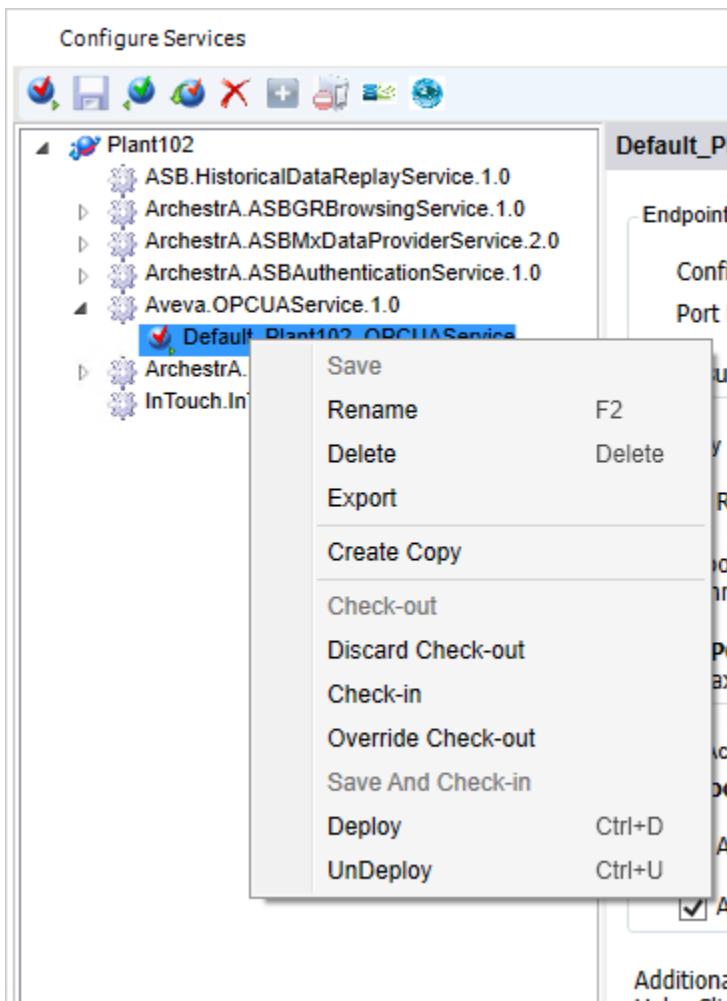
4. Expand the items in the left-hand pane as needed. Right-click the OPC UA service name and select **Check-out** from the context menu.
5. Edit the **port number** for the OPC UA instance if necessary. The default port is **48031**.
6. **Security configuration**

- **Require encrypted communication:** It is strongly recommended that you enable this option, as this will encrypt the payloads across the connection. Note that the client must match this configuration.

Important! An OPC UA connection cannot be established if you do not enable this option while OS security is enabled, even if the option "Allow authenticated Galaxy Users to write attributes" (under Step 7) is enabled.

7. Client access rules

- **Anonymous client connection:** Allowing anonymous client connections is recommended ONLY for initial setup configurations and testing. Anonymous client connections should be disabled for production environments. Refer to [Client access rules and galaxy security](#) to see the effect that this option has on data access.
 - **Allow authenticated Galaxy Users to write attributes:** Enabling this setting will only take effect when encrypted communication is also enabled (see Step 6, above). The OPC UA client must match this configuration.
8. The **Assignments** section (below the right pane) shows both the local node and deployed remote runtime nodes to which the OPC UA service can be assigned. Select the checkmark next to the runtime node where you want to deploy the OPC Server service, and then click **Update**.
9. In the left pane, right-click the OPC UA service instance name and select **Check-in** from the context menu.



10. Right-click the instance again to open the context menu and select **Deploy**, or press CTRL+D. A message appears indicating whether the service has been successfully deployed to the OPC UA client node. If deployment is successful, the icon next to the instance name changes to indicate that the instance has deployed.
11. If you have deployed the service to a remote node, you can verify functionality through the Operations Control Management Console.
- a. Under Operations Integration Server Manager, select the remote node name.

- b. Expand Operations Integration Supervisory Servers and select the OPCUA Client (under OI.GATEWAY.3). Check the following settings:
 - a. Server Node: localhost
 - b. OPCUA Server: opc.tcp://localhost:48031

To add additional OPC UA services

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

1. Right-click **AVEVA.OPCService**, and then select **Create** from the context menu, or press CTRL+N. The new instance appears in the tree structure.
Note: Each instance must have a unique port number. Enter the port number in the **Base Address** field. The default port number is **48031**. See [Configure service TCP ports](#) for a list of port numbers used by ASB services.
2. Rename the OPC UA service as needed. Right-click on the service name and select **Rename** from the context menu, or press F2. Then, enter the new name.
3. Repeat the steps above for configuring and deploying each additional OPC UA service.

To change a deployed OPC UA service

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

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

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

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

Galaxy Security settings do not have any affect on these behaviors. See [Working with Security](#) for more information.

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

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

role allows them to do so.

See [Client access rules and galaxy security](#) for more information about user permissions for each setting combination.

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

Client access rules and galaxy security

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

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

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

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

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

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

None	Authenticated	Disabled	Disabled	NO	N/A	N/A
None	Anonymous	Enabled	Enabled	YES	YES	NO
None	Anonymous	Enabled	Disabled	YES	YES	NO
None	Anonymous	Disabled	Enabled	NO	N/A	N/A
None	Anonymous	Disabled	Disabled	NO	N/A	N/A

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

Automation objects

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

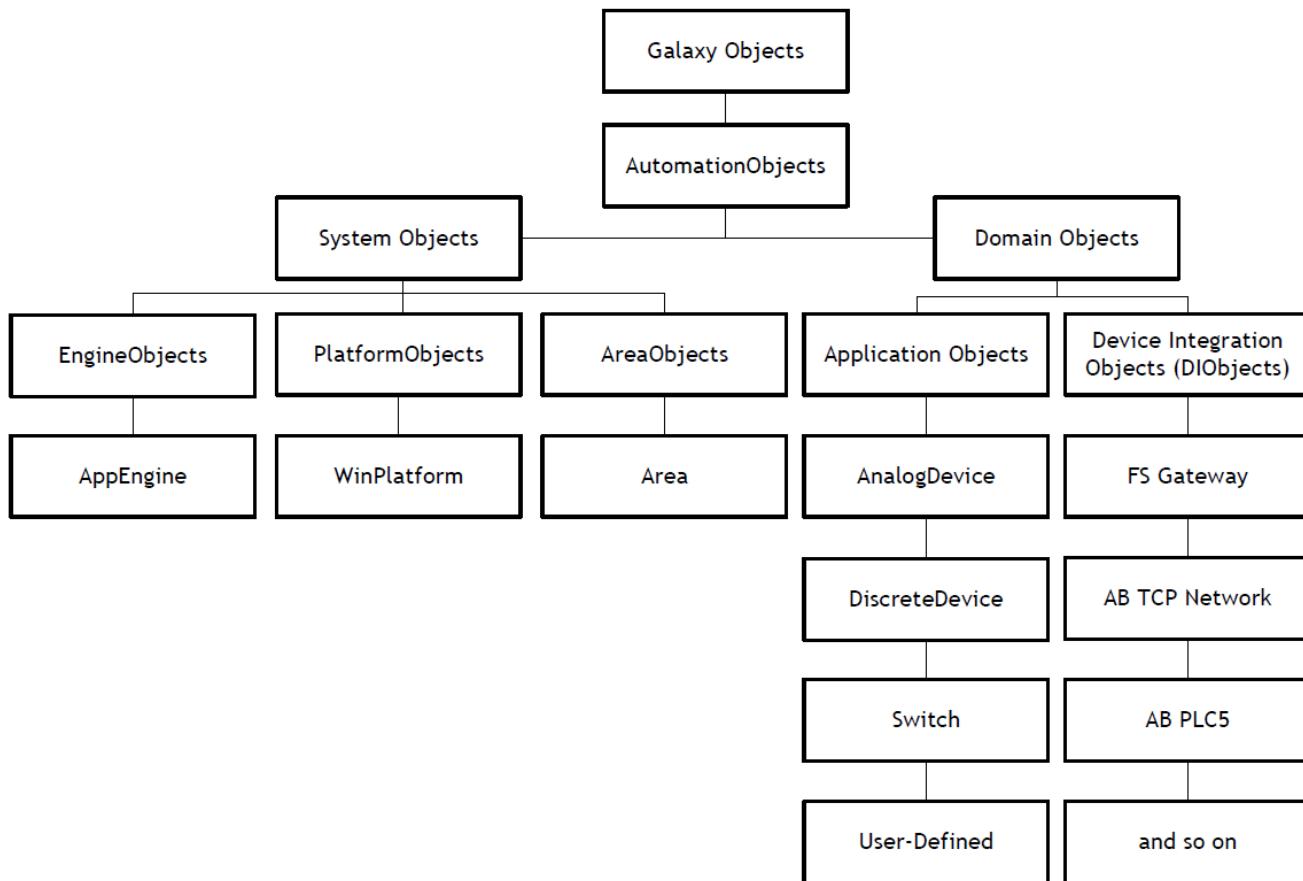
Templates are objects whose properties can be configured to represent a variety of process equipment types in a ViewApp.

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

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

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

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



View object properties

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

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

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

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

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

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

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

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

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

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

For more information about specifying the properties of objects, see [Work with Objects](#).

About templates and instances

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

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

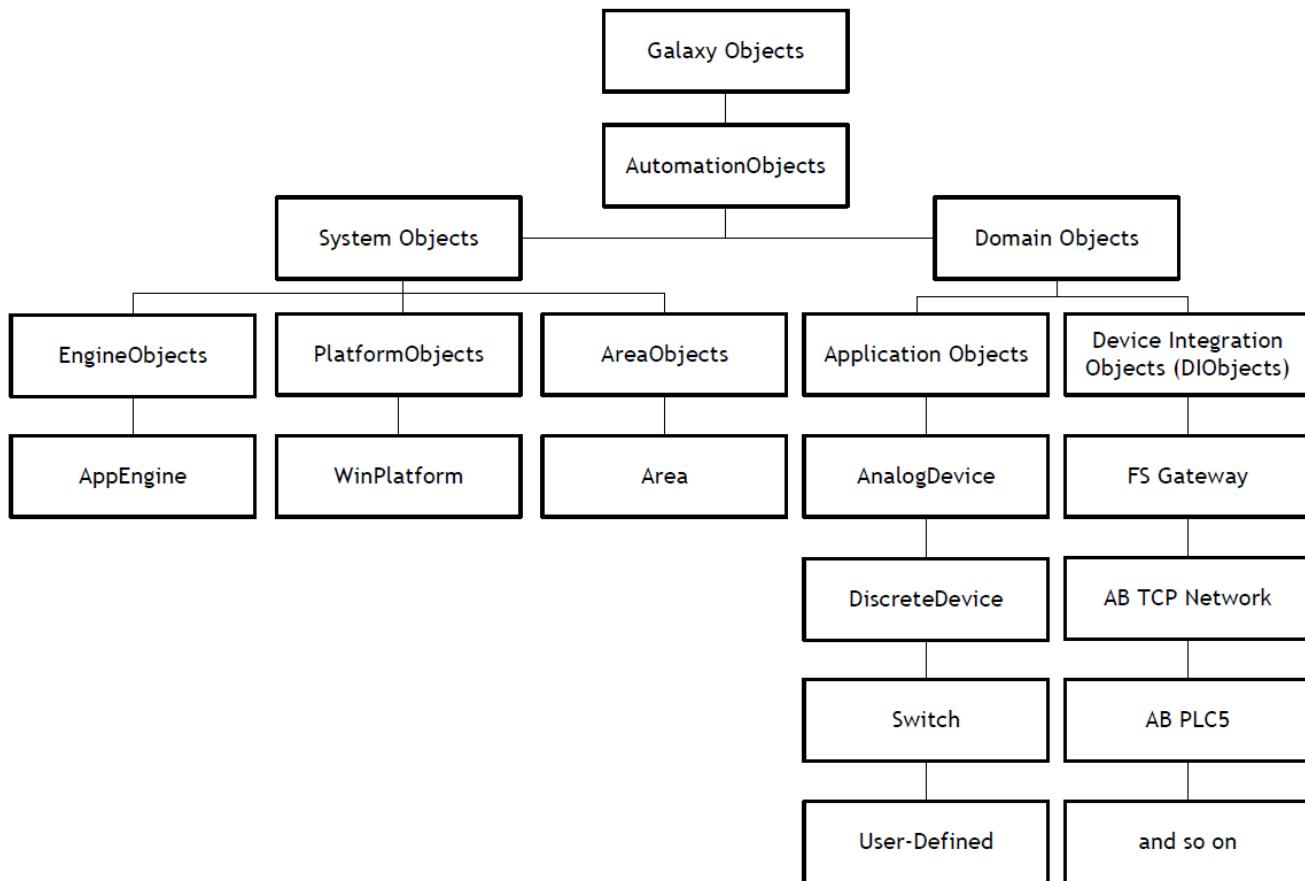
For example, you might create a template for valves. You configure the template with all the unique attributes for valves. You use that template to make object instances of valves. You can further configure and customize each object instance to represent a specific valve.

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

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

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

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



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

Instances

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

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

Templates

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

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

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

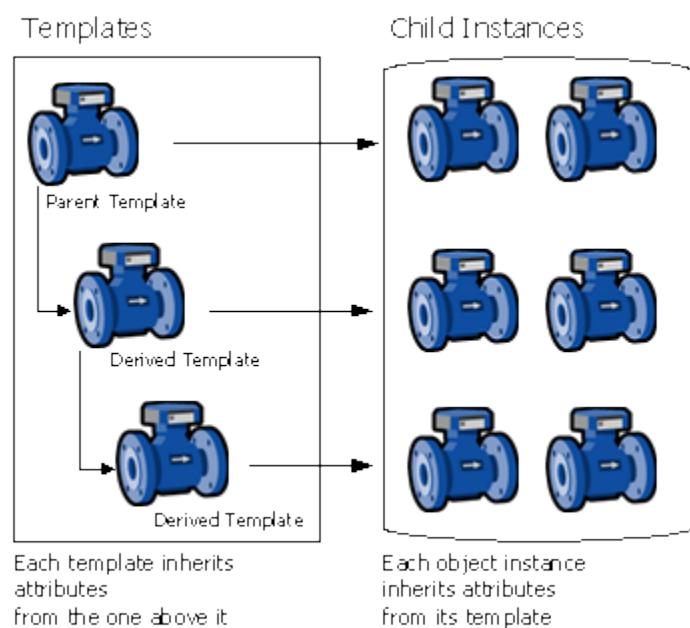
A template can also define an area of your environment. You can extend and customize a template by adding attributes, scripts, or features to meet the specific needs of your environment. Objects inherit attributes, scripts and features from their parents.

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

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

Templates only exist in the development environment.

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



Propagation

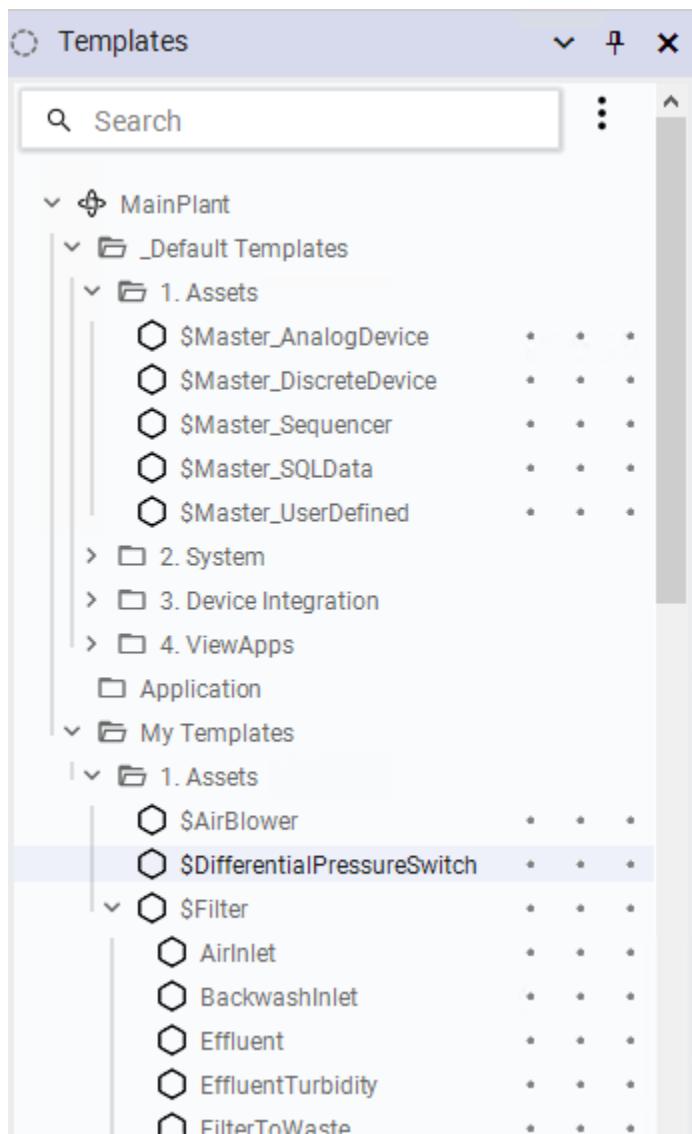
If you need to change something about all diaphragm valves, you can change the template for the Diaphragm valve and all diaphragm valves in your application inherit the changes, assuming the attributes are locked in the parent template. This makes it easy to maintain and update your application.

About base templates

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

You cannot modify base templates, but you can use them to create derived templates, which are copies of the base templates. We recommend that you use the first-level derived templates that are provided under the **_Default Templates** folder to create additional templates. Most of these first-level derived templates have the prefix **Master_**. You can modify derived templates and use them to create instances of them for your

applications.



The Template folder includes:

- Asset templates
 - Use these templates to represent real devices in your Galaxy, such as pumps and valves.
- Device Integration templates
 - Use these templates to create instances that communicate with external devices, such as PLCs.
- System templates
 - Use these templates to define system instances, like other computers.

Application templates

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

Device integration templates

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

For example:

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

System templates

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

WinPlatform object

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

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

AppEngine object

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

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

Area object

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

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

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

ViewEngine object

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

- Hosts ViewApp objects which will be used through the desktop client.
- Hosts InTouchViewApp objects.
- Contains the logic to set up and initialize objects when they are deployed.
- Contains the logic to remove objects when they are undeployed.
- Determines the scan time which all objects within that particular engine run.
- Determines if the ViewApp objects that it hosts will be read/write or read-only.

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

WebViewEngine object

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

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

You can set a WebViewEngine object so the ViewApps that it hosts are either read-write (the default) or read-only, as described in Set a WebViewEngine to read-only.

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

InTouchViewApp object

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

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

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

ViewApp object

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

- Provides visualization at runtime for an AVEVA OMI application.
- Incorporates a Screen Profile and the content defined in an associated Layout.
- Provides runtime access to your production environment.
- A ViewApp can be read/write or read-only, depending on how the ViewEngine or WebViewEngine hosting the ViewApp object is configured. For more information, see [Set a WebViewEngine to read-only](#).
- The OMI Web client supports being saved as a Progressive Web Application (PWA). When an OMI Web client PWA is launched, it displays a landing page listing all the deployed ViewApps. For an OMI Web client PWA to work, the machine with the deployed WebViewEngine must be reachable.

About derived templates

All templates you create within the IDE are derived templates.

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

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

After you create a new derived template, you can customize it. For more information about customizing and extending templates, see [Create derived templates](#).

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

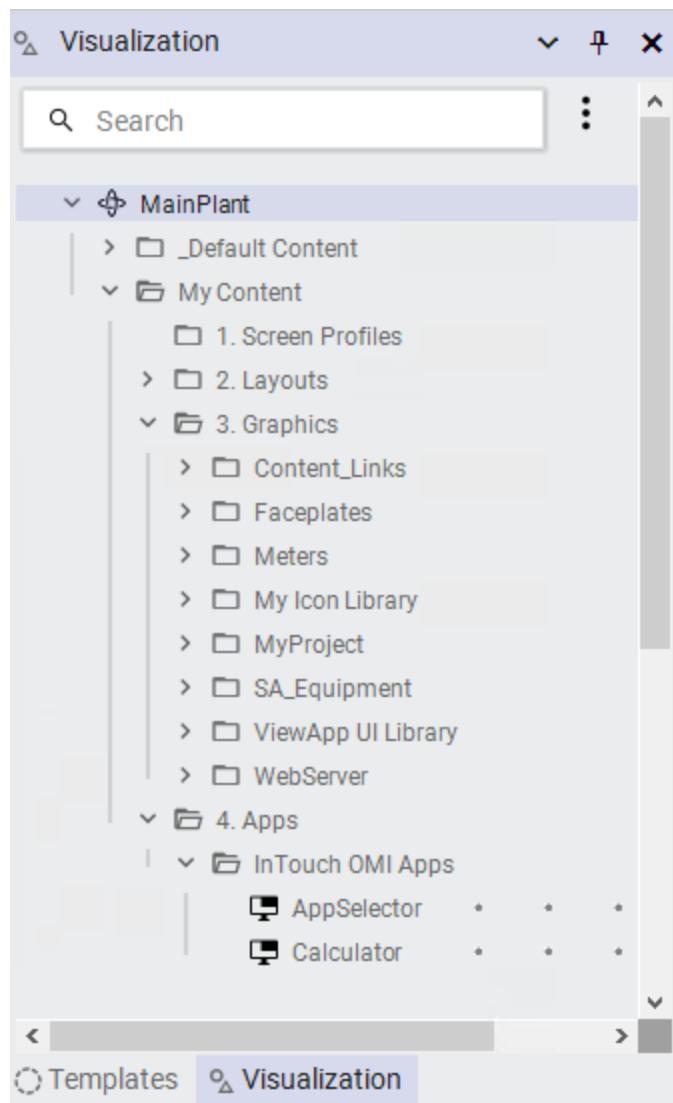
For more information about customizing instances, see [Edit objects](#).

Work with Objects

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

Manage folders

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



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

The screenshot shows the AVEVA Operations Management Interface. On the left, there's a navigation pane with several sections: 'Security' (with a shield icon), 'Galaxy' (with a gear icon), 'Styles' (under Galaxy), 'Alarms and events' (under Galaxy), 'Languages' (under Galaxy), 'Integrated development environment' (with a trash bin icon), 'Folders' (selected, with a folder icon), 'Convert UDO' (under IDE), and 'Communication' (with a communication icon). On the right, there's a 'Folders' panel with tabs for 'Template' and 'Visualization'. The 'Template' tab is selected, showing a list of templates: 'Folder list' (unchecked), '_Default Templates' (checked), 'My Templates' (checked), 'NewFolder_001' (unchecked, highlighted in grey), and 'System' (checked).

Create folders

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

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

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

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

Create child folders

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

To create a child folder

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

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

You can now drag templates into the new folder.

Delete folders

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

To delete a folder

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

About using CONNECT

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

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

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

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

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

Log into CONNECT

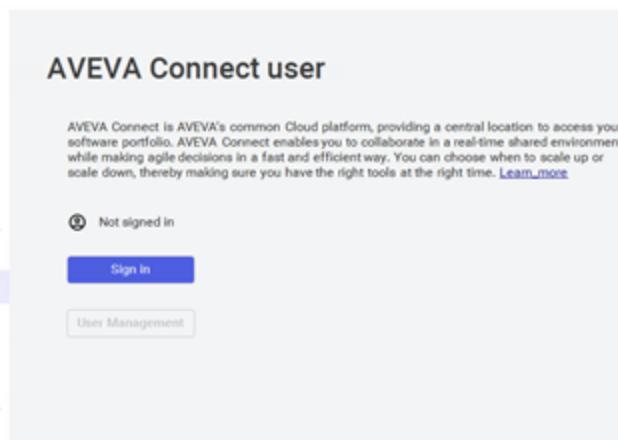
You must have a CONNECT account and authorization from your administration to access CONNECT.

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

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

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

The CONNECT user configuration screen appears.



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

If the credentials are verified, you are signed in.

Note: You must have the Content Contributor role in order to access the CONNECT Industrial Graphics drive. This is required for both Operations Control connected experience and non-connected experience use.

5. Select the **AVEVA Cloud Integration Studio solution**. For example: Pym Technologies.
6. The AVEVA Drive appears as a separate repository within the Visualization folder.

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

Navigate between shared drives

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

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

The Visualization folder will refresh to display the selected drive.

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

Upload/download graphics to the cloud

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

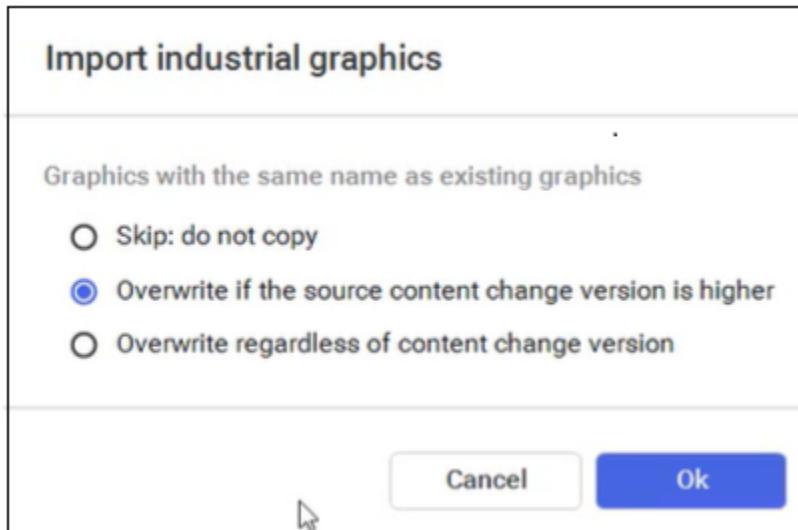
- If the contents already exist in the destination drive, the contents will remain in the same toolset location. Otherwise, the contents will be added in the same toolset path as the source

For the selected embedded contents

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

Upload graphics to CONNECT

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



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

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

Download graphics to the local visualization folder

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

1. Drag the graphic(s) or folder from the AVEVA Drive and drop it to the local Visualization folder.
The **Download Content** dialog box requests confirmation of the download.



- Select **Yes** to continue with the download.
 - Select **No** to cancel the operation.
 - Select **View Details** to view the list of controls and namespaces used. Any errors during downloading are also displayed.
2. If the graphic already exists, the Import industrial graphic dialog box appears.
 3. Select the appropriate option.
 4. The graphic(s) are downloaded to the local Visualization folder. Embedded graphics are also copied.

Upload/download support for various cloud content

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

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

Manage graphics in CONNECT

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

- Right-click the AVEVA Drive or any of the folders to create new graphic or toolset.
- Double-click to edit graphics using the graphic editor.

- Right-click and select the corresponding options to rename, duplicate or delete the graphic, and update thumbnails.

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

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

Manage graphics with multiple users

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

- A lock icon against the graphic icon in the folder indicates that the graphic is checked out by the logged in user.
- An edit icon against the graphic icon in the folder indicates that the graphic is being edited by another user.

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

About cloud graphic versions

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

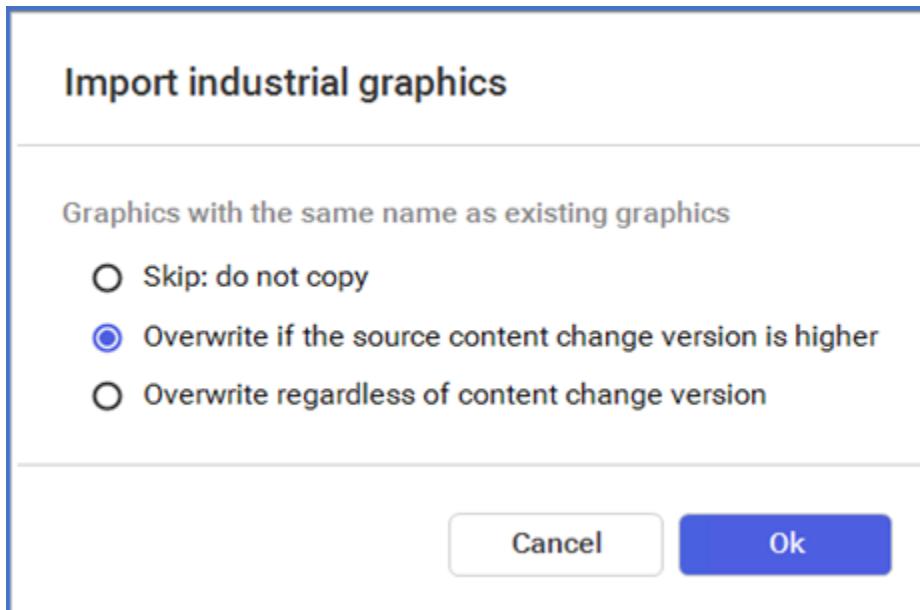
Key concepts of cloud graphic version support

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

- Just like the local Visualization folder, graphic names must be unique within any single Cloud repository. This holds true even if the graphics are stored by products with different versions of the Industrial Graphics subsystem.

Overwrite graphic contents in the cloud

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



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

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

Overwrite custom client controls and widgets in the cloud

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

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

Cloud graphic versions - scenarios and examples

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

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

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

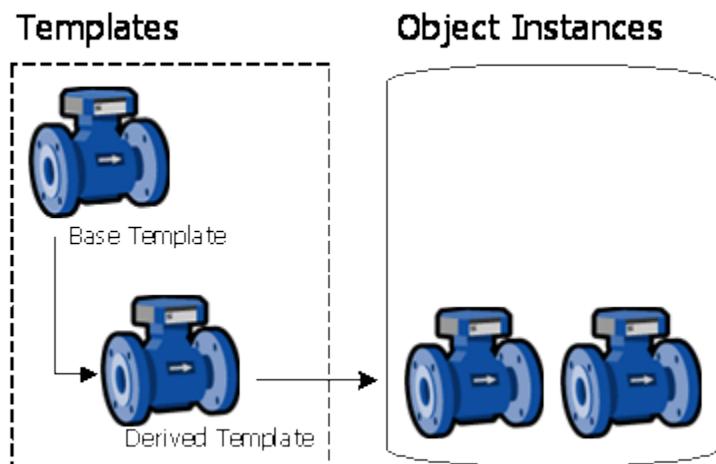
Scenario / Steps	Expectation
Scenario 1: P2_New creates S3. P1_Old attempts to edit S3.	S3 is saved in newer graphic version and hence it cannot be edit, embed, or download in older version. P1_Old receives a message stating that S3 is saved in newer version and cannot be opened.
Scenario 2: P2_New opens S2. P2_New saves S2 in Cloud. Then, P1_Old attempts to open S2.	S2 created in older graphic version can be opened and saved in the newer graphic version. S2 is saved in newer graphic version and hence it cannot be edit, embed, or download in older version. P1_Old receives a message stating that S3 is saved in newer version and cannot be opened.
Scenario 3: P1_Old saves S1 and S2, where S2 is embedded in S1 P2_New saves and updates only S2. P1_Old attempts to edit S1.	S1 is opened because it is still the same version in P1_Old. The embedded S2 will fail to load.
Scenario 4: P2_New creates a new S4 in Cloud. P2_New embeds S1, S2, and S3 in S4.	S1 is in older graphic version but still can be loaded in S4 (newer version). S2 and S3 can be embedded because they are the same graphic version as S4.
Scenario 5: P1_Old creates a new S5 in Cloud. P1_Old embeds S1, S2, and S3 in S5.	S1 can be embedded because they are the same graphic version. S2 and S3 cannot be embedded due to newer version.
Scenario 6: P1_Old download S3.	S3 cannot be downloaded because it is in newer graphic version.
Scenario 7: Cloud repository has a graphic S3 which is in newer graphic version. P1_Old attempts to upload a graphic with same name	By selecting Option 3, the graphic can be overwritten.

S3.	
Scenario 8: P2_New has a graphic S8. P1_Old creates locally S8, S9, and embed S8 in S9. P1_Old uploads S9 to Cloud.	S9 is uploaded to Cloud. S8 is prevented upload because a newer version exists in the Cloud. By selecting Option 3, the content can be overwritten.

Create derived templates

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

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



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

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

To derive a template from another template

1. In the **Template folder** or **Derivation** view pane, select the template to use as the parent template
2. On the **Home** ribbon, in the **Create** area, select **Template**. A derived template is created in the same folder as its parent and placed in name edit mode. The default name is the same as the parent template followed by a numeric sequence.
3. Rename the derived template, if needed. Template names can be up to 128 alphanumeric or special characters, including the required \$ as the first character. The second character cannot be \$ and the name must include at least one letter. Template names cannot contain spaces.

Note: WinPlatform template names are restricted to 127 characters.

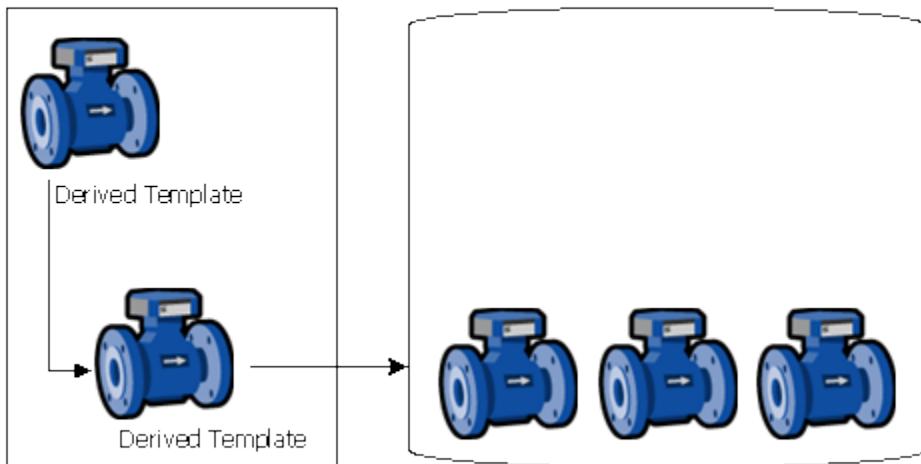
Note: The following reserved names cannot be used as template names: Me, MyContainer, MyArea, MyHost, MyPlatform, MyEngine and System.

4. You are ready to customize your new template. For more information, see [Edit objects](#).

Derive templates from another derived template

You can create derived templates from other derived templates. The child template inherits attributes from all parent templates. Any changed attributes in the immediate parent overrides attributes in grandparent levels. If you change locked attributes in the parent template, the locked attributes propagate to the derived template.

Templates Object Instances



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

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

To create a derived template from a derived template

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

Template names can be up to 128 alphanumeric or special characters, including the required \$ as the first character. The second character cannot be \$ and the name must include at least one letter. Template names cannot contain spaces.

Note: WinPlatform template names are restricted to 127 characters.

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

3. You can create another derived template by repeating the previous steps, or you can customize your new derived template. For more information about customizing your template, see [Edit objects](#).

Create contained templates

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

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

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

Name	Description
Tag Name	The unique name of the individual object. For example, Valve1.
Contained Name	The name of the object within the context of its container object. For example, the object whose Tag name is Valve1 may also be referred to as Tank1.Outlet, if Tank1 contains it and it has the contained name "Outlet".
Hierarchical Name	<p>Hierarchical names that are fully-qualified names of a contained object include the name of the objects that contain it.</p> <p>Because the object that contains it may also be contained, there are potentially multiple hierarchical names that refer to the same object.</p> <p>For example, if:</p> <ul style="list-style-type: none"> "Reactor1" contains Tank1 (also known within Reactor1 by its contained name "SurgeTank"). "Tank1" contains Valve1 (also known within Tank1 by its contained name "Outlet"). <p>Valve1 could be referred to as:</p> <ul style="list-style-type: none"> "Valve1" "Tank1.Outlet" "Reactor1.SurgeTank.Outlet".

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

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

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

ApplicationObject containment

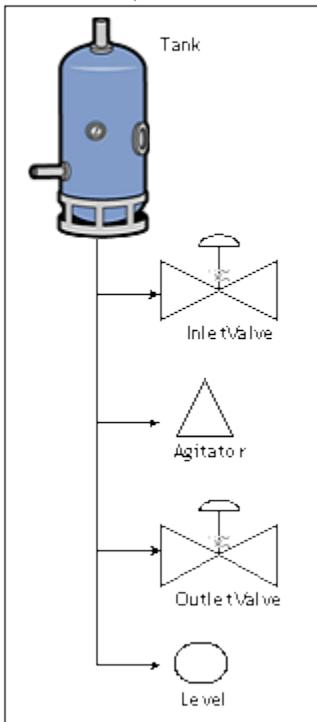
ApplicationObjects can be contained by other ApplicationObjects. This provides context for the contained object and a naming hierarchy that provides a powerful tool for referencing objects.

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

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

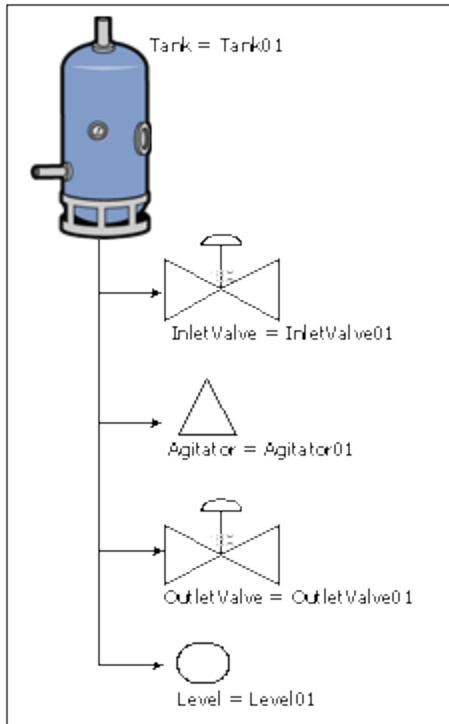
- Inlet Valve
- Agitator
- Outlet Valve
- Level

Tank Template



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

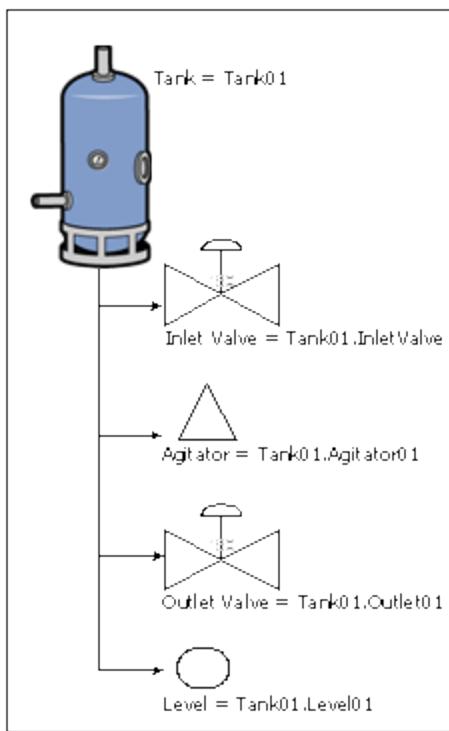
- Inlet Valve = InletValve01
- Agitator = Agitator01
- Outlet Valve = OutletValve01
- Level = Level01



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

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

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



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

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

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

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

Use contained names

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

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

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

Name	Description
Hierarchical name	<p>Hierarchical names that are fully-qualified names of a contained object include the names of the objects that contain it.</p> <p>Because the object that contains it may also be contained, there are potentially multiple hierarchical names that refer to the same object.</p> <p>For example, if:</p> <ul style="list-style-type: none"> "Reactor1" contains Tank1 (also known within Reactor1 by its contained name "SurgeTank"). "Tank1" contains Valve1 (also known within Tank1 by its contained name "Outlet"). <p>Valve1 could be referred to as:</p> <ul style="list-style-type: none"> "Valve1" "Tank1.Outlet" "Reactor1.SurgeTank.Outlet".

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

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

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

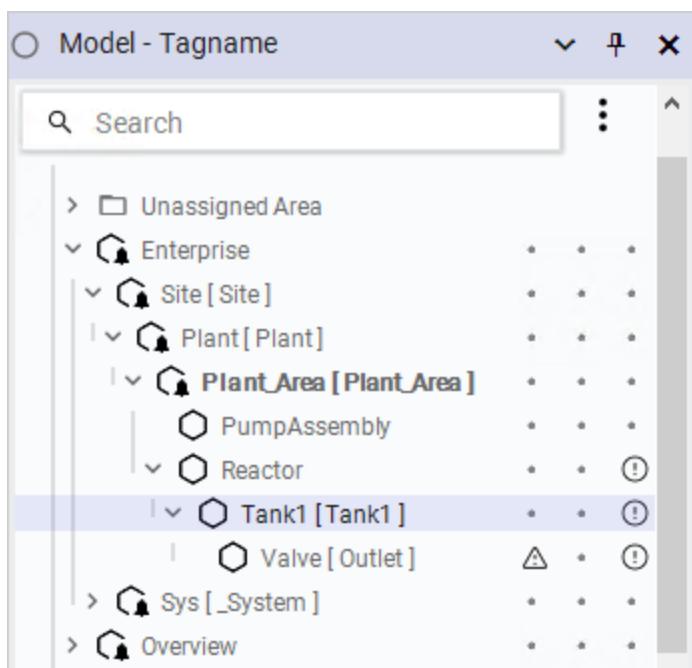
Containment examples

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

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

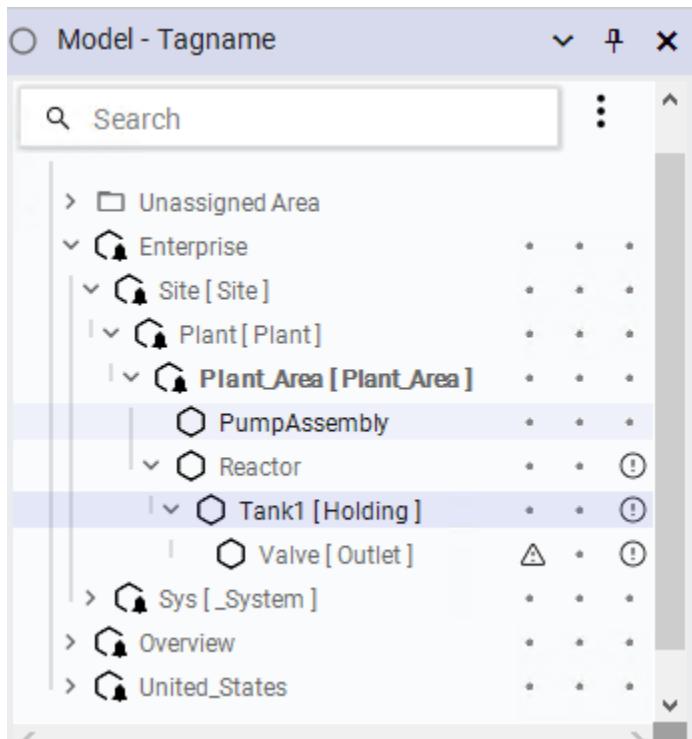
To implement containment

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



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

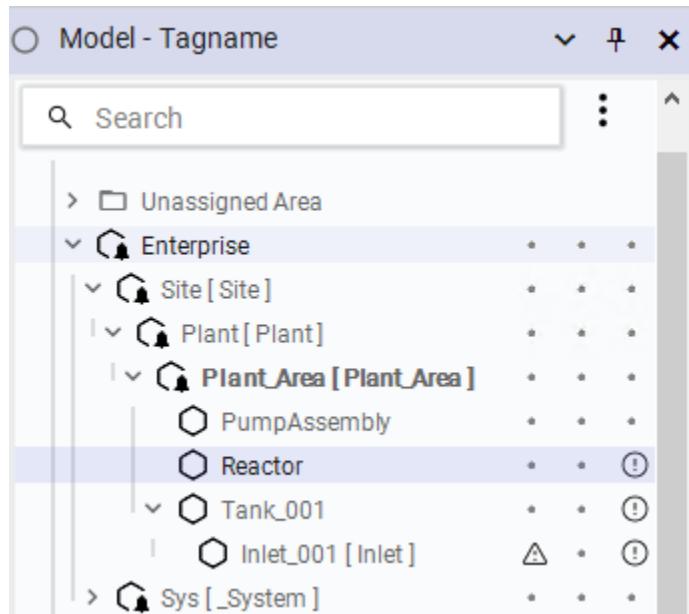
For the three objects in this example (Reactor1 containing Tank1 containing Valve), the following naming hierarchy exists:



To implement template-level containment

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

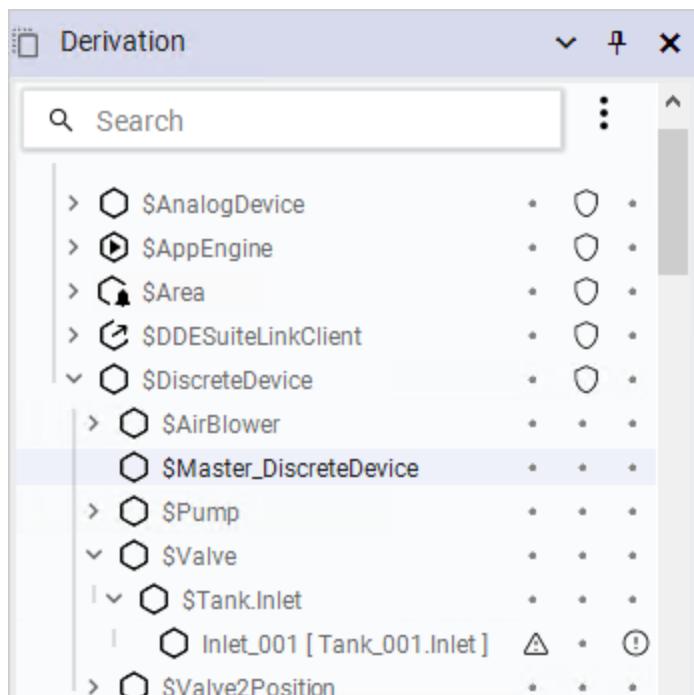
1. Create the following derived templates: \$Tank from \$UserDefined and \$Valve from \$DiscreteDevice.
2. Derive \$Inlet from \$Valve.
3. In the **Template folder**, drag \$Inlet on to \$Tank. If \$Tank already contains a template named Inlet, the Galaxy generates a unique tagname for the new contained template, such as Inlet_001.
The contained template now has a hierarchical name \$Tank.Inlet.
4. Create an instance (Tank001) of \$Tank.
5. The **Model and Deployment views** show an instance Tank001 that contains an instance called Inlet.



View containment relationships

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

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



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

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

Rename contained objects

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

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

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

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

To rename an object's contained name

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

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

Edit objects

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

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

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

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

To edit an object in the Object Editor

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

Note: If you are adding I/O attributes to an application object or system object, such as an area object, the preferred way to add and edit I/O references is through the **IO Devices** view and **Mapping** view. Editing I/O references set for automatic assignment in the Object Editor disables automatic assignment. For more information about I/O auto assignment, see [I/O auto assignment](#).

3. Make your changes. For more information about locking attributes, see [About the attributes page](#). For more information about setting security, see [Set object security](#).
4. To save your changes and keep the object open for more editing, on the **Home** ribbon, in the **Save** area, select **Save**.
5. When you are finished editing the object, click the **Save and Close** icon at the top left of the Object Editor.



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

Object editor user assistance

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

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

Help file structure

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

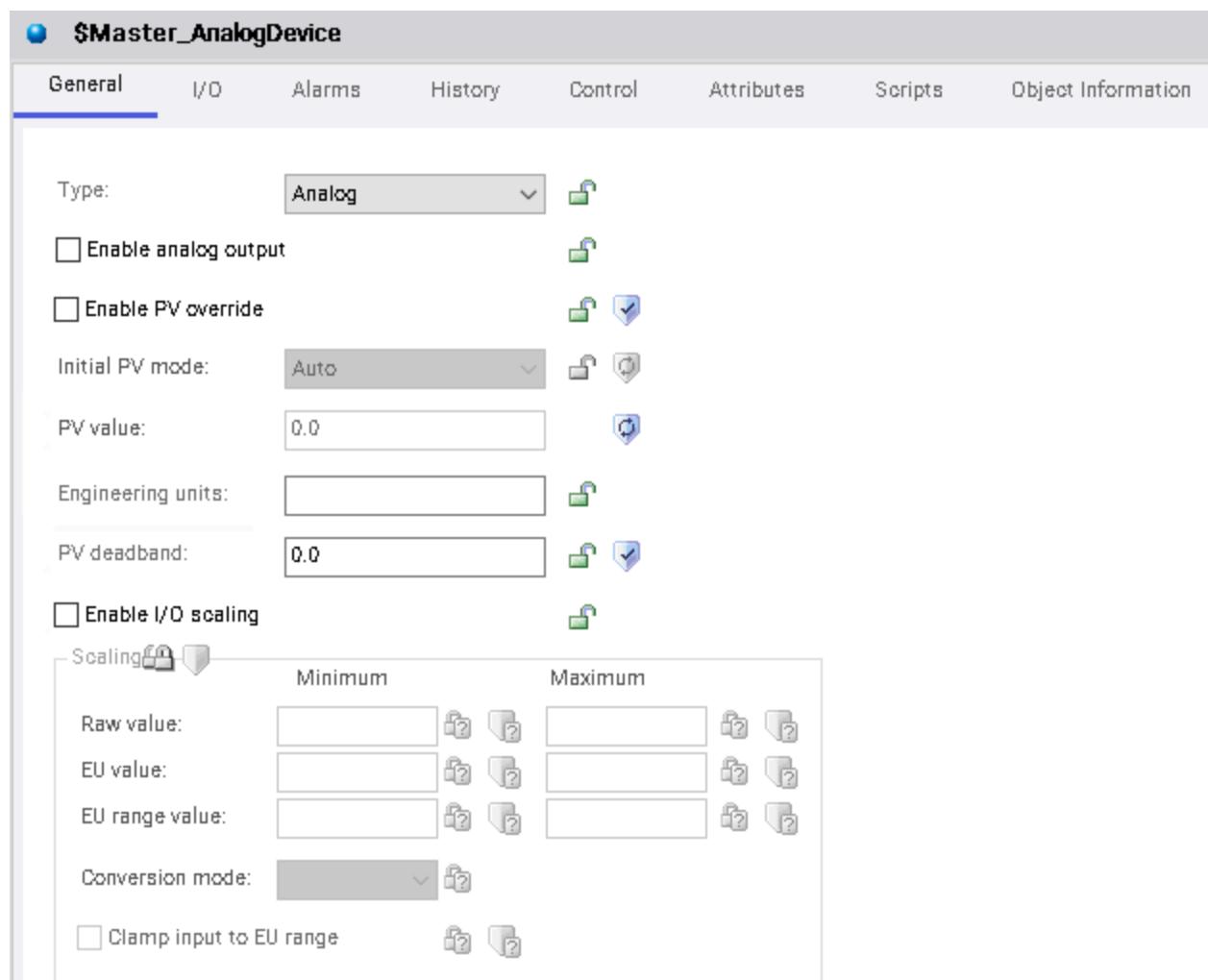
Tag name	The object's name.
Contained Name	The object's contained name. For more information, see Create contained templates .
Description	A short summary of what the object is for.
Code Base	The code version of the object.
Derived From	The immediate parent template for the object.
Object Version	The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

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

Object editor layout

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

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



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



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



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

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

Lock and unlock template attributes

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

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

Note: When you lock values and settings for an attribute, they are not visible when you use Object Wizards, and

thus the locked values and settings will be inherited by all derived objects.



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

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

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

See [Group lock and security controls](#) for information about locking or unlocking all of a feature's attributes with a single click.

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

Icon	Name	Description
A small icon showing a document with a lock symbol, indicating a local lock.	Locked (in me)	The associated attribute is locked (in me) and enabled. Only templates can have this kind of lock. The attribute value is read/write. Derived templates and instances do not have a unique copy of this attribute. Child objects share the locked attribute of the parent. Changing the value of a locked attribute in the parent template updates the value of that attribute in all derived templates and instances.
Icon for locked in parent	Locked (in parent)	The associated attribute is locked in the parent object and cannot be unlocked or modified by the child object. Both templates and instances can have these. The attribute is read-only. The templates and objects do not

Icon	Name	Description
		have a unique copy of this attribute, but instead use the attribute value in the parent where the attribute is locked.
	Unlocked	The associated attribute is unlocked and enabled. Both templates and instances can have this kind of lock. The attribute is read/write. The object has its own copy of the attribute value and the value is not shared by derived objects.
	Indeterminate	Refers to a specified group of options. An indeterminate state indicates different lock states for individual options in the group.
	Undefined	The associated attribute doesn't exist. This indicates that another attribute must be enabled before the associated attribute is created and before its lock status can be determined.

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

Example: Lock an attribute

1. Create a derived template from the \$DiscreteDevice base template. Name the derived template \$Valve.
2. Open the \$Valve template and on the **States** page, lock one of the attributes by clicking the **Lock** icon.
3. Save \$Valve.
4. Create a derived template from \$Valve. Name it \$BigValve.
5. Create an instance from \$Valve named Valve1.

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

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

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

Example: Unlock an attribute

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

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

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

Set object security

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



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

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

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

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

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

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

Group lock and security controls

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



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

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

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

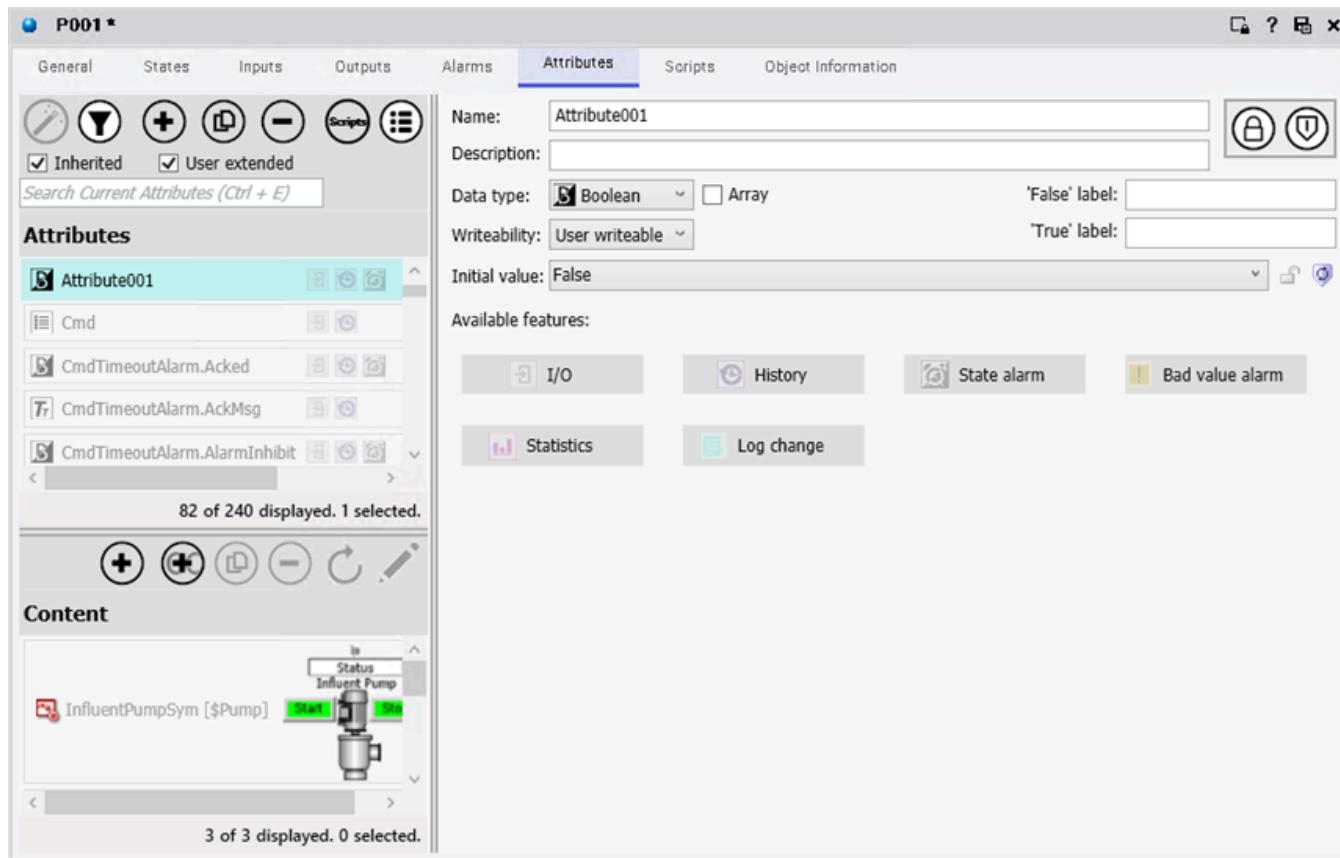
If an attribute is locked in the template, you can change the value in that template, but not in the derived children. If you change the value in the parent template, the change propagates to all child objects. For more information, see [Configure Objects](#).

About the attributes page

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

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

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



Opens the **Object Wizard** editor.



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



Adds a new attribute.



Duplicates the selected attributes.



Removes the selected attributes.



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



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

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

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

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

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

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

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

Property	Description
Name	Attribute name (for example, Attribute001).
Description	User defined; a description of the attribute.

Property	Description
Data type	<p>Can be set to Boolean, Integer, Float, Double, String, Time, ElapsedTime, or InternationalizedString.</p> <p>The data type determines which features can be added to an attribute. For example, when configuring a Boolean data type, only two alarm features are available: State Alarm and Bad Value Alarm. When configuring an Integer data type, there are four available alarm features: Limit Alarms, ROC Alarms, Deviation Alarms and Bad Value Alarms.</p> <p>To create an array, check the Array option and specify the array's length in the # of elements box. You can create an array for each data type except InternationalizedString.</p>
Writeability	<p>Can be set to Calculated, Calculated retentive, Object writeable, or User writeable. See the <i>Object Viewer User Guide</i> for additional information about writeability categories.</p> <ul style="list-style-type: none"> • Calculated: Permits only scripts within the same object to write to the attribute. Calculated attributes are not saved across restarts. If you select Calculated for an attribute, only scripts running on the same object can write to the attribute. • Calculated Retentive: Permits only scripts within the same object to write to the attribute. Calculated Retentive attributes are saved across engine restarts. • Object Writeable: Permits other objects to write to this attribute in addition to being set by scripts within this object. Object Writeable attributes are saved across restarts. This category is not user writeable. • User Writeable: Permits other users to write to this attribute in addition to being set by scripts and objects throughout the system. User Writeable attributes are saved across restarts. They can be locked at configuration time. If locked, they are read-only.
Initial value	<p>Specifies the initial value for the attribute when the object is deployed. Enter value data for each data type. In the case of a non-arrayed Boolean, select True or False in the list box. For an arrayed Boolean, select</p>

Property	Description
	the appropriate checkbox.

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

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

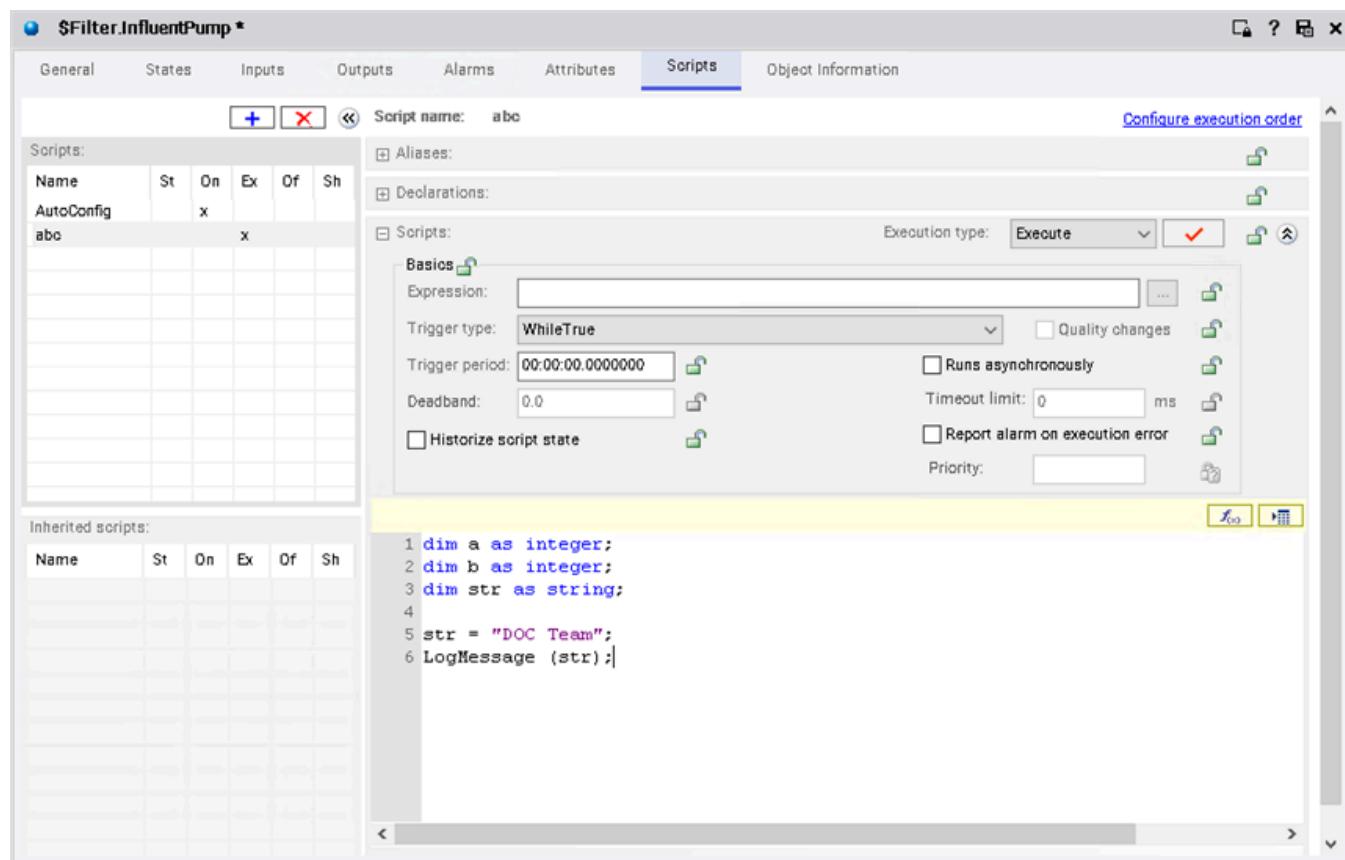
- At the bottom of the right side, fields for configuring activated features are displayed. Available features include:

Feature	Description
 I/O	<p>The I/O feature is used to activate the attribute for I/O. You can select Read (input), Read/Write (input and output), or Write (output).</p> <p>When you select the I/O feature for application objects and system objects, such as areas, the "Read from / Write to attribute text box contains the placeholder reference <IODevice>. [HierarchicalName]. [AttributeName]. InputSource or <IODevice>. [HierarchicalName]. [AttributeName]. OutputDest.</p> <p>You should not edit these placeholders in the Object Editor, but instead use the IO Devices view to assign the object to a scan group. This will allow the placeholder to automatically resolve to the correct I/O reference. See I/O auto assignment for additional information.</p>
 History	<p>The History feature enables historization to the Historian.</p> <p>Note: The AppEngine hosting the object must be configured for historization.</p>
 State alarm	<p>The State Alarm feature is available for Boolean data types. You can set alarm priorities, the alarm state, an alarm message, and time deadband.</p>

Feature	Description
 Limit alarms	The Limit Alarms feature is available for integer, float, and double data types. With it, you can set value alarms (Hi, HiHi, Lo, LoLo), and the limits, priority, and messages to apply to each alarm limit.
 ROC alarms	The Rate of Change (ROC) Alarm feature is available for integer, float, and double data types. With it, you can set alarms if changes within the specified time period exceed set values. Configurable parameters include change limits, change direction, alarm priority, the alarm message, how often to evaluate changes to the value, and time period.
 Deviation alarms	The Deviation Alarm feature is available for integer, float, and double data types. With it, you can set deviation alarms (major and minor), and the tolerance, priority, messages, target value, deadband, and settling period.
 Bad value alarm	The Bad Value Alarm feature adds an alarm if the value returned from the attribute is determined to be bad quality.
 Statistics	The Statistics feature monitors statistics associated with the object.
 Log change	The Log Change feature will cause the attribute to generate an event each time the attribute value changes.

Scripts page

The **Scripts** page is divided into several areas. To learn more about using scripts, see [Write and edit scripts](#).



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

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

Object information page

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

P001 *

General	States	Inputs	Outputs	Alarms	Attributes	Script
Description:	<input type="text" value="Single Feedback Backwash Pump"/>					
Hierarchical name:	<input type="text" value="P001"/>					
Container:	<input type="text" value="N/A"/>					
Codebase:	ArchestrA.DiscreteDevice.3					
Derived from:	<input type="text" value="\$Pump"/>					
Host:	CommonEquip					
Area:	CommonEquip					
Security group:	Default					
Execution order						
Process order:	<input type="text" value="None"/>					
Relative object:	<input type="text"/>					
Click the button to add help for this object. <input type="button" value="Add Object Help"/>						

This page includes the following fields:

- **Description:** A short summary of the object's purpose.
- **Hierarchical Name:** The fully qualified name of a contained object, including the container object's TagName.
- **Container:** The name of the other object that contains this object, if applicable.
- **Codebase:** The code version of the object.
- **Derived From:** The immediate parent template of the object, either a base or derived template.
- **Host:** Another object to which the object is assigned (for example, a WinPlatform hosts an AppEngine). An object's host determines where an object will run when it is deployed.
- **Area:** An object that represents a logical grouping to which this object belongs. An object's area mostly

affects the way in which its alarms are reported to alarm clients.

- **Security group:** The security group the object is associated with.
- **Execution order:** If you want this object to run before or after another object within its engine's scan, select from the **Process order** list. Click the **Browse** button to specify the **Relative object** in the **Attribute Browser**. For more information about the **Attribute Browser**, see [Use the galaxy browser to reference objects](#).
- **Add Object Help:** Opens a copy of the HTML help page for the template this object is derived from. You can edit this information. This allows you to create Help about the object you are currently configuring for downstream users. This Help appears when you select an object in a view and then click **Object Help** on the **Help** menu. See [Customize help](#) for more information.

Customize help

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

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

Locate the help folders

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

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

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

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

Use the galaxy browser to reference objects

Use the Galaxy Browser to browse for:

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

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

Browse for attributes

You use the Galaxy Browser to browse for:

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

- Attributes of templates.

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

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

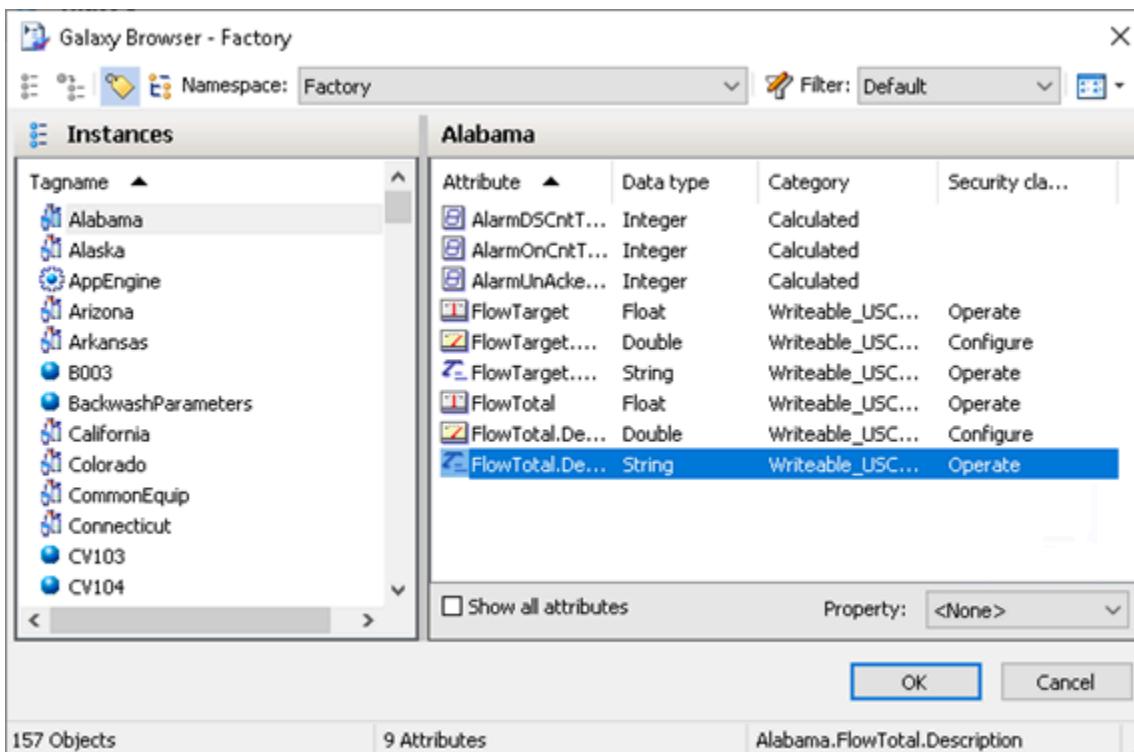
The Galaxy Browser shows objects in the left pane and the attributes associated with the current selection in the right pane. Only attributes that can be referenced at runtime are shown. You can browse "Me." references for alarm messages only.

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

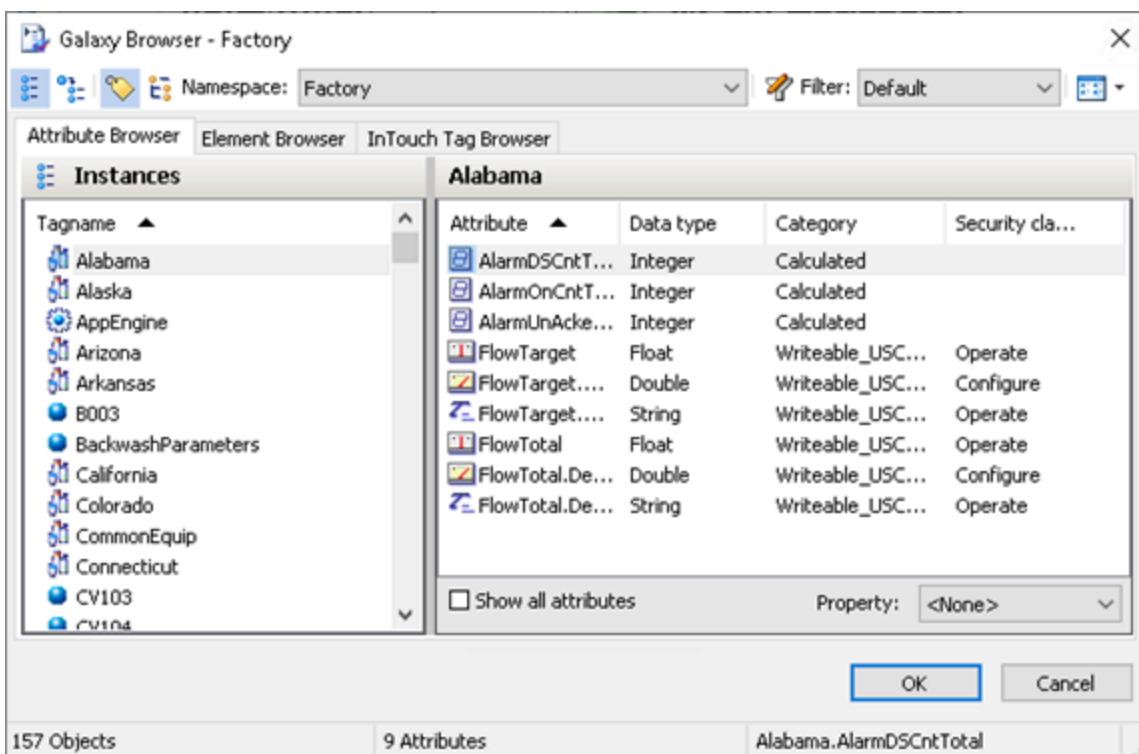
The status bar displays the attribute property name and the graphic element attribute name and description.

To browse for attributes

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



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



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



4. You do not have to explicitly make a selection in the **Property** list. If you only select an attribute (leaving **Property** set to <none>), the property of the attribute defaults to **Value**.
If the option in the Object Editor is already configured with an object reference, the **Attribute Browser** shows it or expands to the nearest matching object/attribute/property currently configured in the Galaxy.
If you selected text in the script editor, that text is used as the initial reference string and the browser finds the nearest attribute reference to the selected text
5. When you are done selecting the attribute/property, click **OK** to place the reference into the Object Editor and close the **Galaxy Browser**.
 - The fully-qualified reference string appears in the editor option.
 - If you are working in the script editor, the selected reference appears in the script at the current cursor position and replaces text that was selected.

View attribute details in the galaxy browser

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

Important: The Galaxy Browser shows only the Primary AppEngine and its attributes of a redundant pair. Any

Backup AppEngine is not shown. See [AppEngine Redundancy](#) in the Application Server help for more information.

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

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

Data Type	<p>The data type of the attribute:</p> <ul style="list-style-type: none">• Integer• Boolean• Float• Double• String• Internationalized String• Time• ElapsedTime• ReferenceType• CategorizedStatusType• DataTypeEnum• SecurityClassificationEnum• DataQualityType• CustomEnum• CustomStruct <p>For information about each of these, see the help for the object.</p>
Value	<p>The primary value of the attribute.</p> <p>Sometimes, a list of numbers is included in the Property list. Those numbers map to single bits in an integer attribute's Value property. Valid bit field specifiers are:</p> <p>.00 (least significant bit) .01 .02 .03 .04 .05 .06 .07 .08 .09 .10 .11 .12 .13 .14 .15 .16 .17 .18 .19 .20 .21 .22 .23 .24 .25 .26 .27 .28 .29 .30 .31 (most significant bit)</p>

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

Browse for graphics

You use the Galaxy Browser to browse for graphics from:

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

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

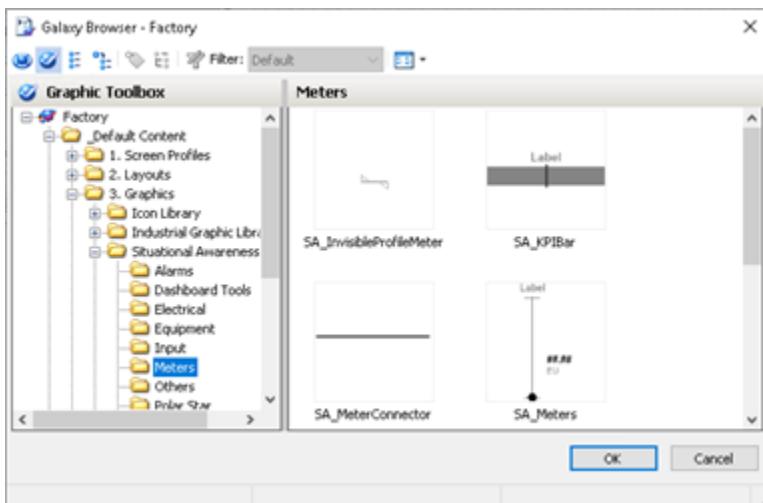
Within the same user session, the Galaxy Browser remembers the last browsed location for graphics and presents it whenever called as the starting location so that context is kept. Initial default location for graphic browsing is the Visualization folder with the root selected (Galaxy node).

You can use the Galaxy Browser to browse graphics from:

	The Visualization folder. The browser shows the Visualization folder organization on the tree in the left pane and a list of the graphics contained in the currently selected node (Galaxy node or folder node) in the right pane.
	AutomationObject templates. The browser shows the Template folder in the left pane and the graphics associated with the currently selected template in the right pane.
	AutomationObject instances. The browser shows a flat list of existing instances of objects in the left pane. The right pane shows the graphics associated with the currently selected instance. You create or apply filters to reduce the scope of the instances shown in the left pane.
	Relative references. This is possible only when you edit a graphic belonging to an object and browse for graphics from that specific object.

To browse for graphics from the graphic editor

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



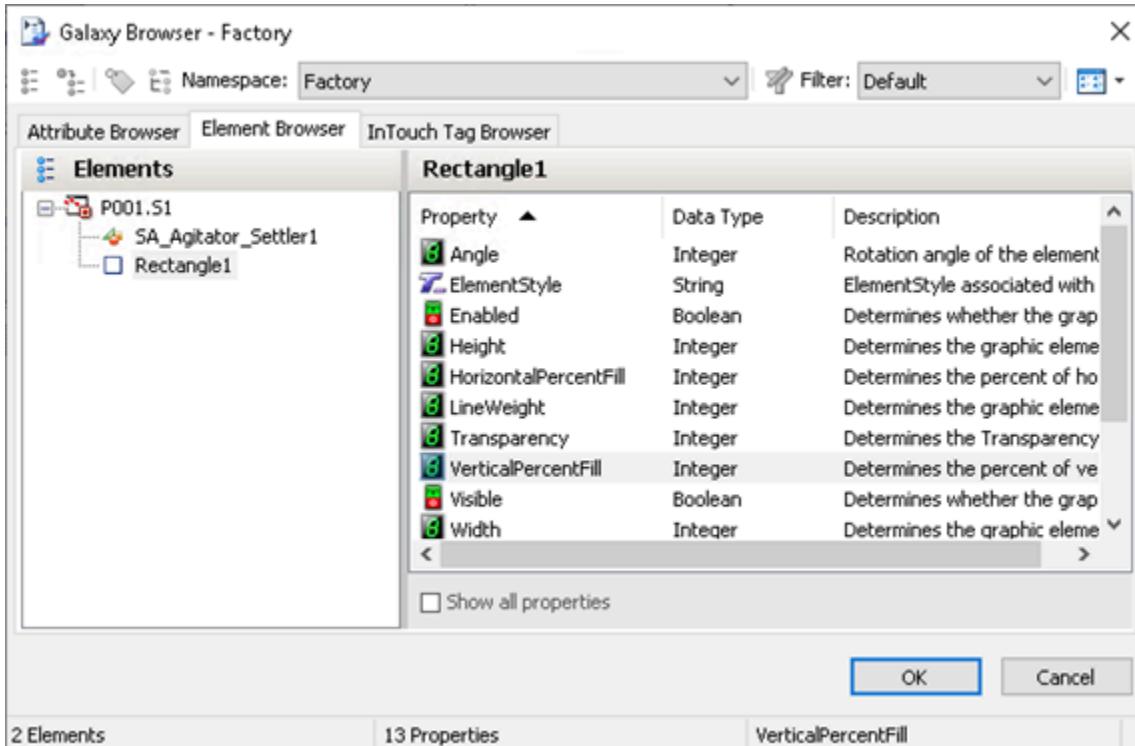
2. Select a graphic from the list and then click **OK**.
3. Click in the canvas to place the graphic.

Browse for element properties

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

To browse for element properties

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



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

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

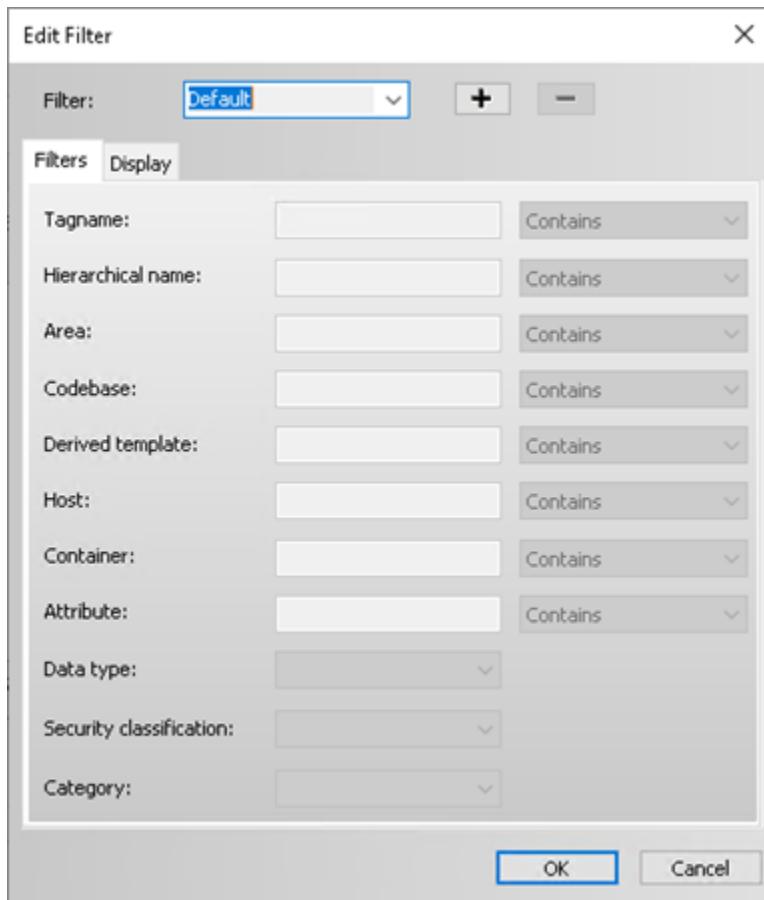
Create a filter for the galaxy browser

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

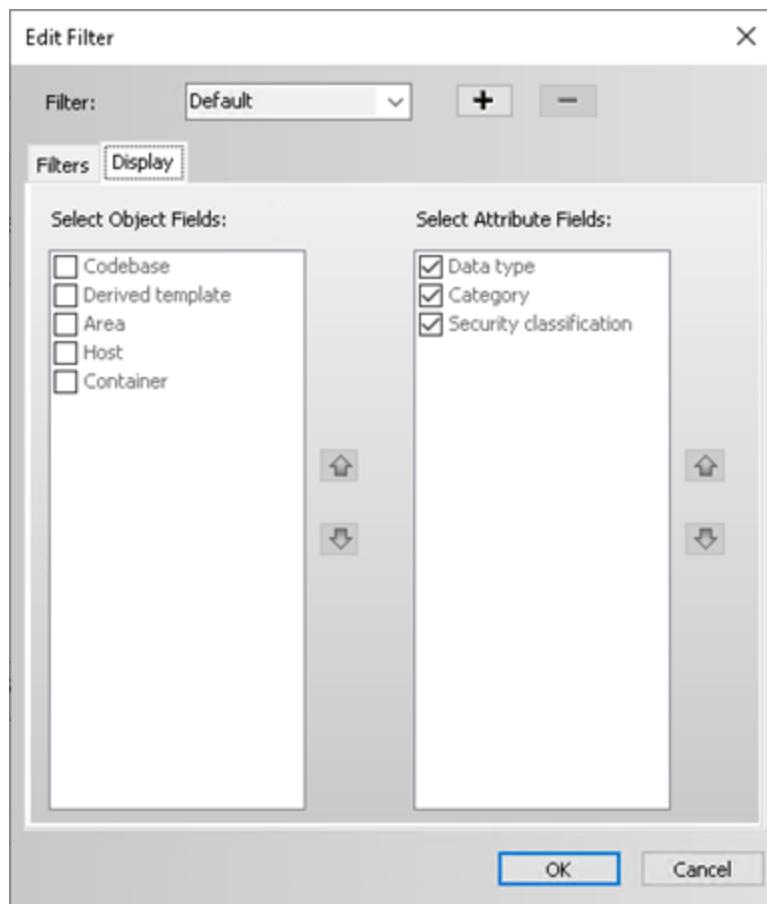
The Default filter provides an unfiltered list of objects and attributes. It cannot be edited.

To create a filter

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



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



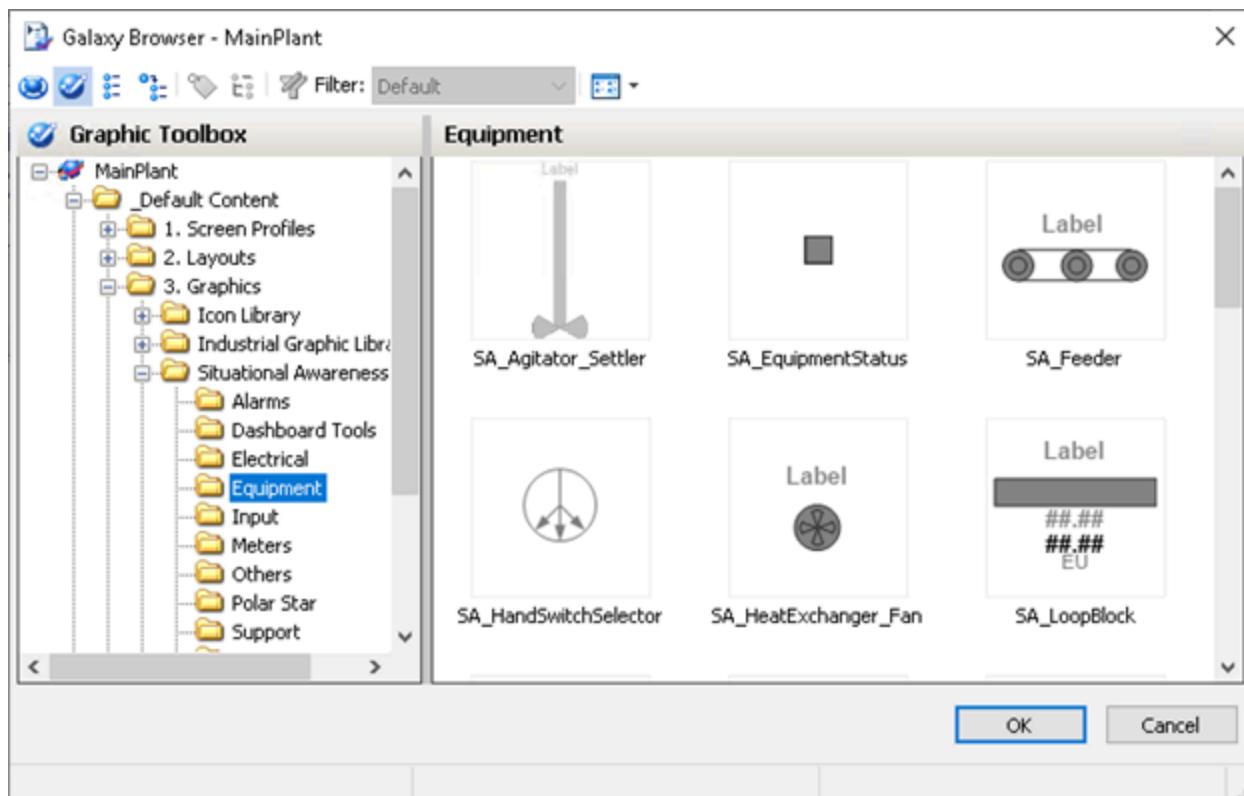
6. Configure the columns to show in the right pane of the **Galaxy Browser**. Use the up and down arrows to set the order for the columns.
7. Click **OK**. The new filter appears in the **Filter** list.

Change how information is shown in the galaxy browser

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

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

The following figure shows graphic thumbnails.



Manage objects

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

Check out objects

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

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

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

To check objects out

1. In the **Template folder** or **Application views**, select the objects you want to work with.
2. Do one of the following:
 - To check out the object without opening the Object Editor, on the **Home** ribbon, select **Check-in**, then select **Check Out**.

- To check out and edit the object, open it in the Object Editor. An object is automatically checked out to you when you open its editor.

Check in objects

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

You can turn this dialog box off if you do not want to enter information about your changes. For more information, see [Customize your workspace](#).

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

To check in an object to the Galaxy database

If the object is open in the Object Editor:

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

If the object is not open in the Object Editor:

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

Validate objects

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

Validating an object's configuration includes:

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

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

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

To open the Validation area

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

Validate scripts and other external components

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

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

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

For example:

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

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

Two kinds of indicators are shown in the object icons:

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

Validate an object manually

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

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

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

You can select more than one object for validation.

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

Note: You cannot cancel validation operations.

To manually validate one or more objects

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

Name	State	Status	Command result
Inlet_001	△ • ⓘ	Warning	Succeeded - Validation completed
Reactor	⊖ • ⓘ	Good	Succeeded - Validation completed
Tank_001	⊖ • ⓘ	Good	Succeeded - Validation completed

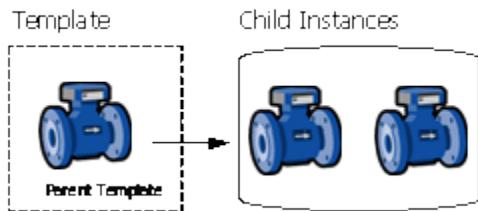
3. Continue using the IDE to perform other operations, if needed, while validation is going on, including work on other objects in the Galaxy. If you are validating a Galaxy, then you must leave the Galaxy alone until the validation process is complete.
4. When the validation process is complete, you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

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

Create an instance

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

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



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

To create an instance

1. Select the template you want to use for the instance. For example, to create a valve instance, select a valve template.
 2. On the **Home** ribbon, in the **Create** area, select **Instance**. An instance is created.
 3. Type the name for the instance. Instance names can be up to 128 alphanumeric characters. You cannot use \$ as the first character. The name must include at least one letter. Instance names cannot include spaces.
 4. To move the new instance in the **Deployment** view or **Model** view, drag the instance to the new location.
- You are ready to configure the instance, if needed. For more information, see [Edit objects](#).

Object names

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

- Template names can be up to 32 alphanumeric characters, including the required \$ as the first character. The second character cannot be \$ and the name must include at least one letter. You cannot use spaces in an object name.

- Instance names can be up to 128 alphanumeric characters. You cannot use \$ as the first character. The name must include at least one letter. You cannot use spaces.
- The object name extension "_New" cannot be used if another object in the Galaxy uses the same name without the "_New" extension. For example, if you have an object named "Pump," you cannot have a second object named "Pump_New." You could, however, use "Pump_Old" and "Pump_New."

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

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

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

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

Delete an object

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

- Deployed instances

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

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

To delete an object from the Galaxy

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

Export objects

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

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

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

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

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

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

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

To export an object

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

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

Protect objects on export

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

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

About protecting objects on export

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

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

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



Export objects as protected

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

To protect objects on export

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

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

Export objects with I/O auto assignment

Objects configured for I/O auto assignment can be exported the same way as other objects. See [Configure Objects](#) for additional information.

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

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

Note: The I/O mapping references are permanently deleted if matching DI objects and scan groups are not found. The I/O references will not reconstitute, even if you later add the scan groups and DI objects to which the application objects were formerly mapped.

- When exporting DI objects, there are no special considerations related to I/O assignments. I/O references for application objects will not change when the DI objects are imported into another Galaxy.

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

Export script function libraries

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

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

export the script function libraries separately.

To export a script function library

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

Import objects

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

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

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

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

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

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

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

Note: Libraries, Client Controls, and OMI Apps compiled with any version of .NET Core (including .NET 5, 6, 7, or 8) are not supported in System Platform. While the system may indicate a successful import, these components will not appear in the IDE or be usable in design time or at runtime.

Some .NET standard libraries may work, but compatibility is not assured. For the best results, we recommend only using libraries compiled with .NET Framework 4.8.1.

To import objects

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

Import preferences

Objects with same Tagname and Codebase as an existing object

- Skip: Do not import**
- Overwrite objects if the imported configuration version is higher**
- Overwrite objects regardless of configuration version**

Base Templates with a different revision number in the Codebase or a different minor version

- Skip: Do not migrate**
- Migrate**

Objects with same Tagname but with a different Codebase

- Skip: Do not import**
- Rename object in Galaxy**
- Rename importing object**

Append to object name

Template Protection Change Management

- Never overwrite an unprotected object with a protected object**

Cancel

Import

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

- **Migrate** - objects with an older codebase are migrated when a newer the replacement object.
For more information about migrating, see [After you import](#).
6. In the **Objects with same Tagname but with a different Codebase** area, select one of the following:
 - **Skip: Do not import** leaves the existing object unchanged.
 - **Rename object in Galaxy** imports an object with a matching tagname but a different codebase from the existing one. The existing object is not overwritten but is renamed.
 - **Rename importing object** imports an object with a matching tagname but a different codebase from the existing one. The existing object is not overwritten. The imported object is renamed.
 7. In the **Template Protection Change Management** area, select the checkbox to prohibit overwriting an unprotected object with a protected object (default). Clear the checkbox to allow overwrites.
 8. Click **Import**. The import process starts.
 9. When the import process is complete, you can start using the objects you imported.

Import protected objects

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

Import Preferences	Result for Protected Objects
Objects with same Tagname and Codebase as an existing object: <ul style="list-style-type: none"> • Skip: Do not import • Overwrite objects if the imported configuration version is higher (default) • Overwrite objects regardless of configuration version 	Overwrite preferences are governed by the Template Protection Change Management option, "Never overwrite an unprotected object with a protected object" (default). <ul style="list-style-type: none"> • Overwrites can occur if both the packaged and galaxy objects have the same protection status. • A protected object can overwrite an unprotected object only if specifically permitted by clearing the "Never overwrite ..." option. • When overwriting a protected parent object, protection is not cascaded to child objects. Unprotected child objects remain unprotected.
Base Templates with newer or older Codebases (or minor version updates): <ul style="list-style-type: none"> • Skip: Do not migrate • Migrate 	Import and migration of base templates is unchanged by protected object import functionality.

Import Preferences	Result for Protected Objects
<p>Objects with same Tagname but with a different Codebase:</p> <ul style="list-style-type: none"> • Skip: Do not import • Rename object in Galaxy • Rename importing object 	<p>Object renaming on import can occur only if the object to be renamed is not protected.</p> <ul style="list-style-type: none"> • If the object in the Galaxy is protected, and the importing object is not, and Rename importing object is selected, the importing object is renamed to have the suffix "_new". • If the object in the Galaxy is unprotected, and the importing object is protected, and Rename object in Galaxy is selected, and the Never overwrite an unprotected object with a protected object option is not selected, the object in the Galaxy is renamed with the suffix "_old".

Import objects with I/O auto assignment

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

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

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

Import a SQLData object

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

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

- Windows Integrated Security
- Windows Account
- SQL Server Authentication

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

Import script function libraries

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

Note: Libraries, Client Controls, and OMI Apps compiled with any version of .NET Core (including .NET 5, 6, 7, or 8) are not supported in System Platform. While the system may indicate a successful import, these components will not appear in the IDE or be usable in design time or at runtime.

Some .NET Standard libraries may work, but compatibility is not assured. For the best results, we recommend only using libraries compiled with .NET Framework 4.8.1.

To import a script function library

1. From the ribbon, select **Galaxy**, then **Import**, then **Scripts**.
2. Browse to the directory that contains the script function library you want to import.
3. Select the script function library to be imported, and then select **Open** to start importing the selected library. Acceptable file types and file name extensions for script function libraries are:
 - Script Library Files (.aaSLIB file extension)
 - .NET or .COM files (.dll, .tlb, .olb, or .exe file extensions)
 - InTouch Script Extension Files (.wdf file extensions)An information window opens when the import successfully completes.
4. Validate any graphics that use the custom script functions contained in the imported library. See [Validate objects](#) for additional information.
5. Redeploy any InTouchViewApp instances containing graphics that use the script functions in the imported library.

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

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

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

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

- Install it directly on the target platform and register it.

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

Import client controls

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

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

Note: Libraries, Client Controls, and OMI Apps compiled with any version of .NET Core, including .NET 5, 6, 7, or 8) are not supported in System Platform. While the system may indicate a successful import, these components will not appear in the IDE or be usable in design time or at runtime.

Some .NET Standard libraries may work, but compatibility is not assured. For the best results, we recommend only using libraries compiled with .NET Framework 4.8.1.

Import an app from the AVEVA digital exchange

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

To access the Digital Exchange

1. From the ribbon, select **Galaxy**, then **Import**.
2. Select **Objects**, then **Explore product hub**.
Your browser will open and connect to the AVEVA Digital Exchange.
3. Log in, or sign up for new account.
4. Click on the link for the **System Platform Developers Community** and browse for apps. Select the app you want and follow the online instructions.

Import AVEVA apps and client controls

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

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

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

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

Note: Libraries, Client Controls, and OMI Apps compiled with any version of .NET Core, including .NET 5, 6, 7, or

8) are not supported in System Platform. While the system may indicate a successful import, these components will not appear in the IDE or be usable in design time or at runtime.

Some .NET Standard libraries may work, but compatibility is not assured. For the best results, we recommend only using libraries compiled with .NET Framework 4.8.1.

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

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

To import an AVEVA App

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

The contained App, along with all dependent files, including dependent files contained in subfolders, are imported when you select **OK**.

If the App does not import successfully, see [Troubleshoot AVEVA or WPF apps that fail to import](#).

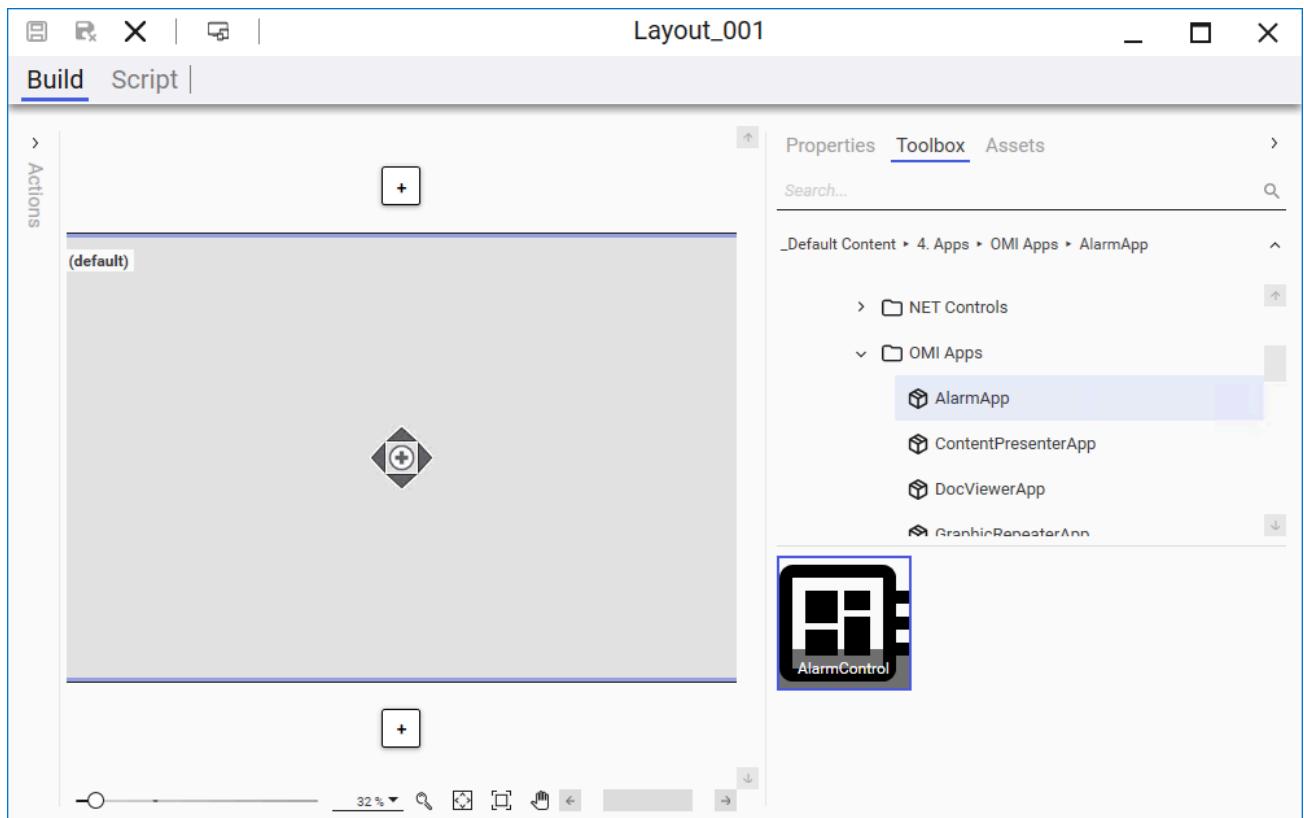
After the import process completes successfully, a new object appears in the Visualization folder named `DisplayModule_00n`, where *n* is 1 or the next available integer. The new object can be renamed, deleted, or exported as an .aaPKG file for use in another Galaxy. See [Export objects](#) for additional information.

Add a control from an AVEVA app to a ViewApp

Controls from the imported App appear in the **Toolbox** tab of the **Layout** or **ViewApp** editors.

To assign a control from an App to a ViewApp

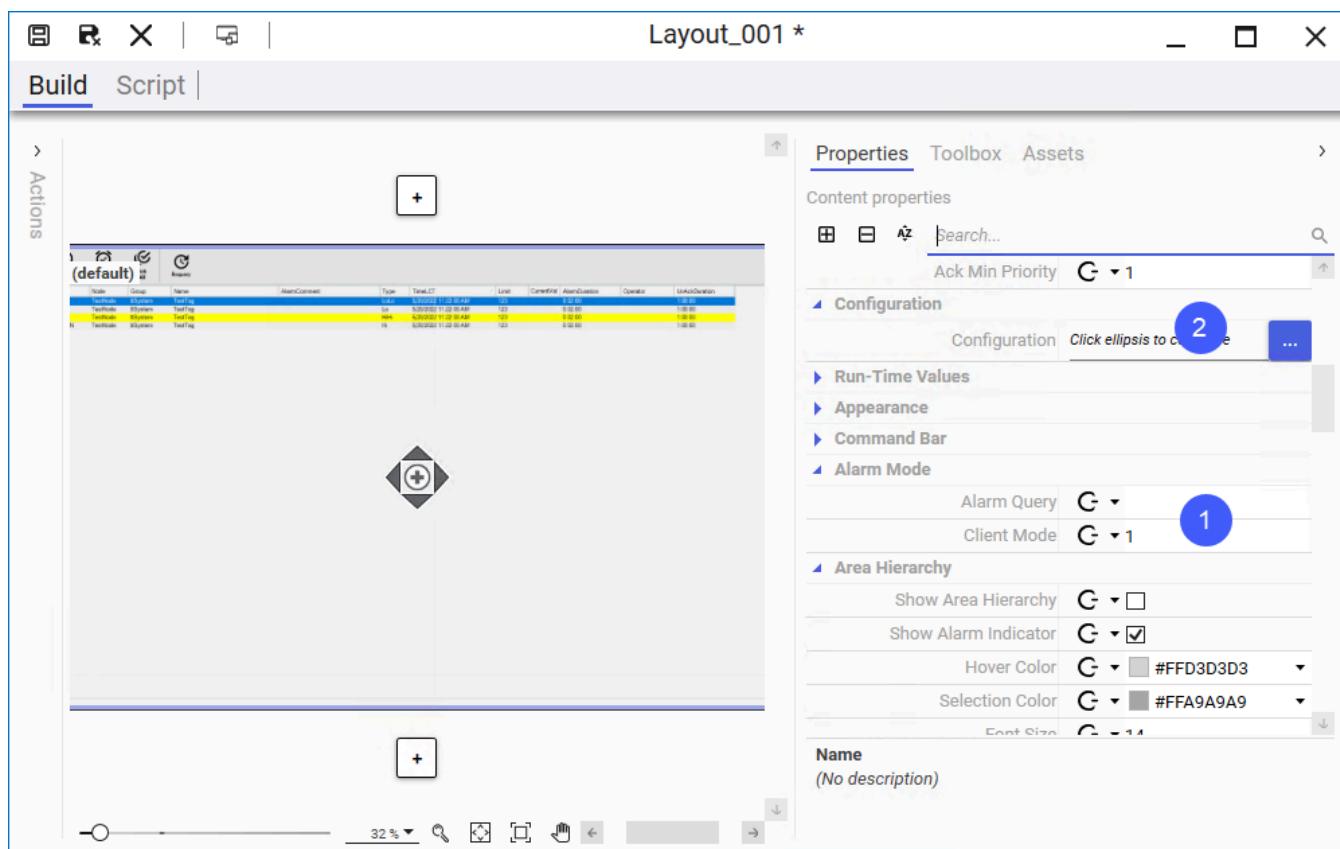
1. Open a layout in the **Layout** Editor and select the **Build** tab at the left of the editor.
2. Select the **Toolbox** tab on the right side of the editor and select a folder that contains the AVEVA Apps you want to add.
3. Select the AVEVA App to be added. The controls available in the App are shown.
4. Select a control and drag it onto a pane.
5. To view or override the control's public properties, select the **Properties** grid. Enter overrides as needed.
6. Save the layout. You can then use the layout that contains the embedded control in a ViewApp.



Configurable properties in AVEVA apps

AVEVA apps can contain two types of configurable properties:

- ① — Dependency properties
- ② — CLR properties (.NET framework properties)



Configurable Property Data Types

- System.Boolean
- System.Byte
- System.Char
- System.Decimal
- System.Double
- System.Int16
- System.Int32
- System.Int64
- System.SByte
- System.Single
- System.String
- System.UInt16
- System.UInt32
- System.UInt64
- System.DateTime
- System.Drawing.Color
- System.Windows.Media.Color

- System.Windows.Media.Brush
- Any property that can be converted to or from string

CLR properties can only be set at configuration (design) time, while dependency properties can be bound to any attribute and can be set at runtime. If a dependency property is changed at runtime, the new value is propagated to the attribute. Dependency properties are data-bindable at design time, and you can set the binding direction to in (read), out (write), or both. To change the binding direction, click on the arrow to toggle the binding direction.

For dependency properties, you must specify the attribute to which you are binding the property, as well as the direction of the binding (in (read only), out (write only) or both. The attribute naming convention is the same you use, for example, when configuring an animation in the Graphic Editor. Note that the ViewApp Editor does not provide syntax validation for attribute names, but the Logger will list any configuration errors.

Binding is only supported for the following basic data types.

DataBindable Properties

- System.Boolean
- System.Byte
- System.Char
- System.Decimal
- System.Double
- System.Int16
- System.Int32
- System.Int64
- System.SByte
- System.Single
- System.String
- System.UInt16
- System.UInt32
- System.UInt64

Any data type can be converted to and from the System.String type property.

AVEVA app development guidelines

The following types of properties are available for configuring WPF-based displays:

- Public writable CLR properties on the display that are not hidden using the browsable attribute
- Dependency Properties
- Nested Properties which satisfy the criteria of configurable properties
- Enum and Flag properties
- Any property that can be converted to or from string

The following types of properties are NOT available for configuring WPF-displays:

- Any property defined by the System.Windows.UIElement class
- Any property defined by the System.Windows.FrameworkElement class
 - The Tooltip property is an exception to this and can be configured
- The following properties defined by the System.Windows.Controls.Control class
 - IsTabStop
 - TabIndex
 - Template
- Any property defined by the System.Windows.Controls.ContentControl class

Optional Filter File: AppManifest.xml

You can include an optional file called **AppManifest.xml** in your imported AVEVA App. See the *AVEVA OMI SDK Help*, located in the **AVEVA Documentation** folder, for additional information about AppManifest.xml. This file will allow you to filter properties when you are configuring the app. The basic file structure is:

- DLL name
- Control name
 - Property

The following is a sample AppManifest.xml file, from the Hamburger App. The files specifies controls and properties that are exposed to System Platform IDE users as they configure a layout or ViewApp.

```
<?xml version="1.0"?>
<AppManifest AppVersion="1.0">
    <Filters>
        <!--List all the Controls that need to be exposed to the System Platform IDE
        user to utilize them in a View Application.
        Only the controls listed here will be available for the IDE users to place
        them on panes within a Layout or a ViewApp.-->
        <Control AssemblyName="AVEVA.Apps.HamburgerApp"
ControlFullName="AVEVA.Apps.HamburgerApp.HamburgerButton">
            <!--List the public properties on this control that need to be exposed
            when user is configuring this control in a Layout or
            a ViewApp. The Property Editor within the Layout Editor and ViewApp
            Editor includes these properties when this control is
            selected and allows the user configure new default values.-->
            <Property Name="HamburgerColor" />
            <Property Name="SlideInPanePosition" />
            <Property Name="Background" />
        </Control>
    </Filters>
</AppManifest>
```

AVEVA app limitations and restrictions

- Events and scripting are not supported
- Only basic datatype properties configuration

- Apps cannot be imported from a network share location
- Collections and arrays are not supported
- Upgrading Apps is not supported
- Nested properties can be viewed and configured, but they cannot be configured with data binding
- To host the .NET controls, use the WPF element **WindowsFormHost** to wrap the control into WPF.

```
<UserControl x:Class="AVEVA.Visualization.TrendApp.TrendControl"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="300">
    <Grid>
        <WindowsFormsHost x:Name="FormsHost" />
    </Grid>
</UserControl>
```

Troubleshoot AVEVA or WPF apps that fail to import

There are two major factors that may stop an AVEVA or WPF app from successfully importing.

Expose controls and properties: Make sure that you have included an AppManifest.xml file with the app, and that all controls and properties that are needed in your app are explicitly exposed in it. For additional information about the AppManifest.xml file, see [AVEVA app development guidelines](#) and the [AVEVA OMI SDK Online Help](#).

Block business logic from executing during design time: If you have business logic executing in the control constructor, or in static properties or static constructors, the import operation will not be successful. The control constructor is invoked when the control is instantiated. Therefore, do not include business logic within the control or static constructors, or in static properties.

- If you cannot remove business logic from a control or static constructor, or a static property, make sure the property is not invoked or that the property does not execute while in design mode.
- To block execution of logic within a constructor while in design time, add the following line to the constructor:

```
!DesignerProperties.GetsInDesignMode
```

After you import

Imported templates are listed in the proper folder in the Template folder as defined in the object. Imported instances are shown in the Application views.

The following post-import rules apply:

- If a folder does not exist, it is created.
- If the object belongs to a security group that does not exist, it is associated with the Default security group.
- If the object belongs to an area that does not exist, it is associated with the Unassigned Area.
- If the host to which the object is assigned does not exist, it is assigned to the Unassigned Host.

- If you selected **Migrate** from the **Import Preferences** dialog box, the migrated objects are marked with "software upgrade required" if they are deployed. These objects will be upgraded when the objects are redeployed.
- If you import a new version of an existing instance, the new version is marked as requiring deployment if the existing object is already deployed.

Configure Objects

Enhance and extend derived templates and instances by adding and configuring features, attributes, scripts, and graphics. These elements can be associated with an Object Wizard in a derived template to simplify the task of creating instances.

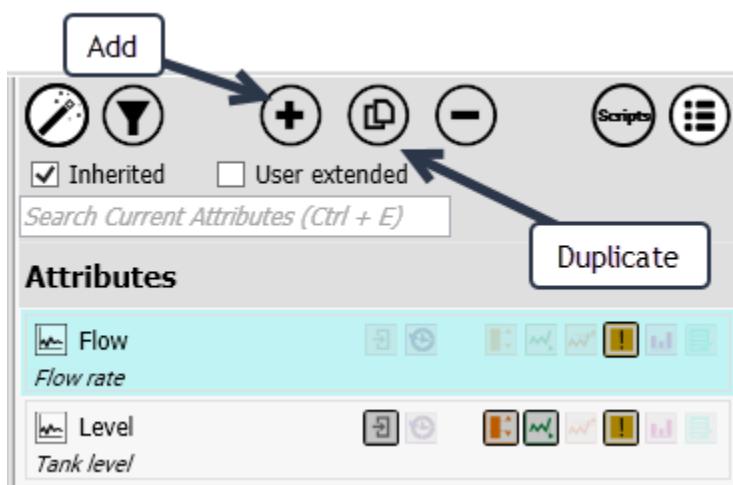
The **Object Editor** includes a separate page for adding scripts. All other elements (graphics, attributes, Object Wizards, and external content) are added and configured from the Object Editor's main page. For an overview of the **Attributes** page, see [About the attributes page](#).

Add attributes to an object

There are two ways to add attributes to an object. These are:

1. Select the **Add Attribute** button. The first attribute you add to an object has the default name "Attribute001" and the default data type "Boolean".
Subsequent attributes added with the **Add Attribute** button use the previous attribute's name and increments the number at the end of the name, or appends "1" at the end of the name if the previous name did not end in a number. It also retains the previous attribute's description, Data type, and Writeability.
2. To duplicate an existing attribute, select the **Duplicate Attribute** button. Selecting **Duplicate Attribute** retains any features that were configured for the selected attribute, in addition to name, description, Data type and Writeability.

Note: When you add an attribute to a derived template, the attribute, its data type, and writeability are automatically locked in derived objects.



If attribute parameters such as initial values and security classifications are locked in the template, they cannot be changed in child instances. If these parameters are unlocked in the template, the initial value and security are

editable and lockable in derived templates. When unlocked in either the base or derived template, the value is editable in instances.

After you add an attribute to an instance, it appears in the **Attribute Browser** list for use with the scripting and attribute configuration functions. For more information about using the **Attribute Browser**, see [Use the galaxy browser to reference objects](#).

In the **Attributes** page, you can define the following initial information and parameters for the attribute:

- Add a new attribute to an object.
- Name the attribute and provide a description.
- Configure its data type.

For a Boolean data type, you can specify different text strings for the '**False**' label and '**True**' label. For example, if a Boolean attribute is associated with the status of a motor, you can specify the states as "**Stopped**" and "**Running**". Text boxes appear for you to enter these strings when you select a Boolean data type.

These labels are also shown in the **Value** and **Limit** columns of the Alarm and Event database and InTouch AlarmView control.

- Specify the attribute writeability.
- Set initial values if the attribute is user writeable.
- Enable and set locks and security on the new attribute.
- Set whether the new attribute is an array and how many elements are in the array.

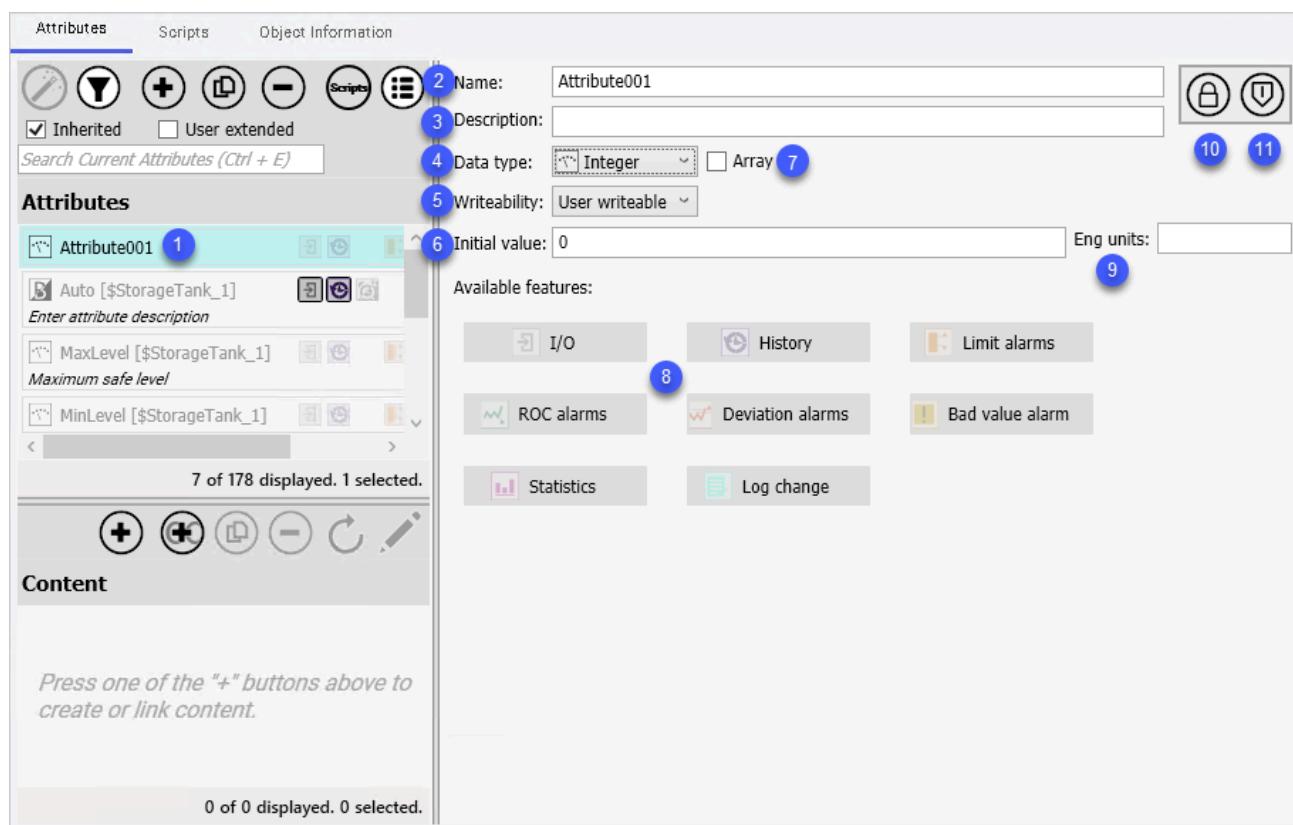
You can then add Features to the attribute. For more information, see [Add features to attributes](#).

Configure an attribute

To configure attributes

1. Highlight the attribute you want to configure.
2. Edit the **Name** of the attribute.
3. Enter a **Description**.
4. Select a **Data type**. See Data type for additional information.
 - Boolean: Enter False and True labels (if not False and True).
 - Integer: Enter Initial value and Engineering units.
 - Float: Enter Initial value and Engineering units.
 - Double: Enter Initial value and Engineering units.
 - String: Enter Initial value.
 - Time: Enter Initial value.
 - ElapsedTime: Enter Initial value.
 - InternationalizedString: Enter Initial value.
 - BigString: Enter Initial value.
5. Select Writeability type:
 - Calculated

- Calculated retentive
 - Object writeable
 - User writeable
6. Enter the Initial value for the attribute.
 7. Select the checkbox if the attribute is part of an array.
 8. Enter the Engineering unit type (numeric data types only).
 9. Select Features and configure them. See [Add features to attributes](#) for additional information.
 10. Lock the Initial value, if you do not want users to be able to modify the value in derived objects. You can show or hide the lock icon by toggling the **Show/Hide Lock icons** button. See [Lock and unlock template attributes](#) for additional information.
 11. Configure the attribute's security state. You can show or hide the security icon by toggling the **Show/Hide Security icons** button. See [Set object security](#) for additional information.



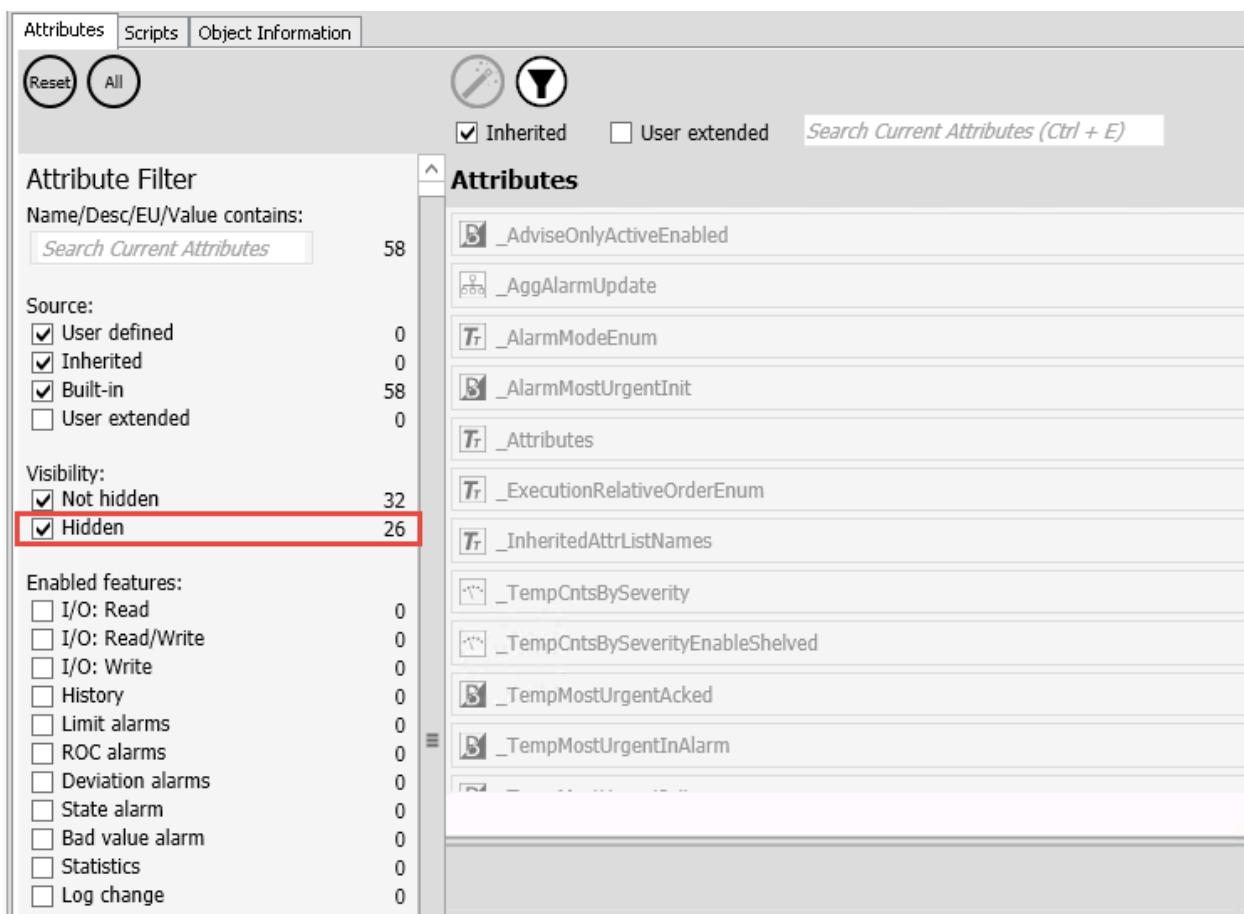
Attribute name conventions

Attribute names can use alphanumeric characters and the following special characters: . (period), _ (underscore), \$, and #. The attribute name can be up to 329 characters. The 329 character limit includes extensions to an attribute that result from the addition of features, such as historization, alarms, and I/O that can be associated with an attribute. These additions are categorized as "User extended" attributes. Note that separator periods in the attribute string also count against the 329 character limit. See [Add features to attributes](#) for more information about user extended attributes.

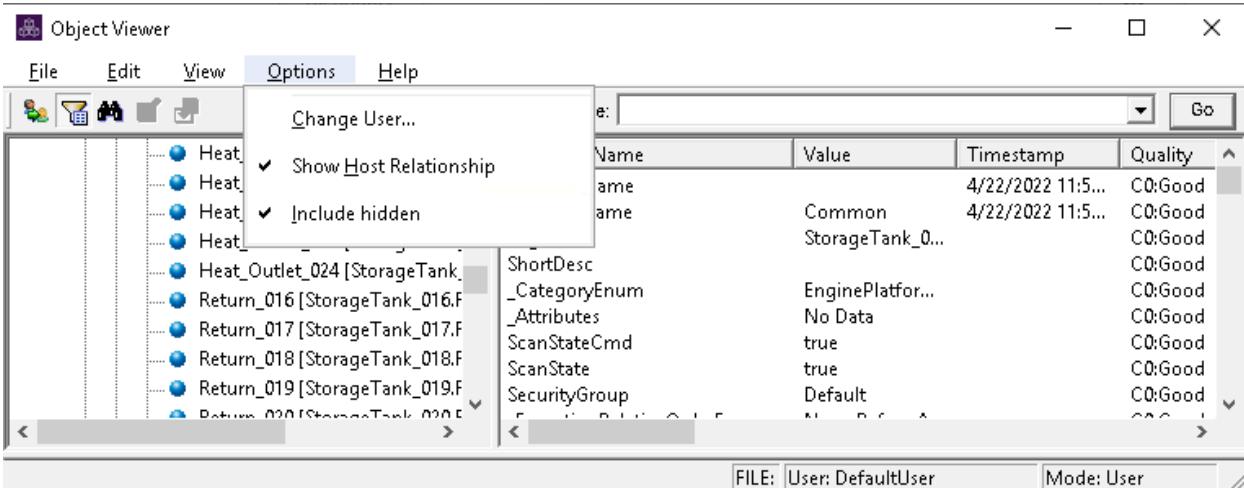
- An attribute name that starts with an underscore (_) as the first character of the name is a hidden attribute.

By default, hidden attributes are not shown in the **Attribute Editor**, the **Properties>Attributes** dialog box, or the **Object Viewer**.

- To view hidden attributes in the **Attribute Editor**, select the **Hidden** filter option.



- To view hidden attributes in the **Properties** dialog box, select **Include Hidden**.
- To view hidden attributes in the **Object Viewer**, select **Include Hidden**.



Note: We recommend that you do not use hidden attributes in scripts.

- Using the word "quality" as an attribute name is not supported. "Quality" is used by InTouch HMI in a set of

dotfields to show the reliability of data values assigned to an I/O tag. An attribute named "Quality" cannot be accessed through FS Gateway or InTouch due to a naming conflict.

Important: After creating an attribute, it is available for adding Features or additional attributes. If you extend an attribute you have added to an object or another attribute, and then delete or rename that attribute, all associated Features and additional attributes added to the object are lost.

To create and associate a attribute with an object

1. On the **Attributes** page of the Object Editor, click the **Add (+)** button. An attribute is added to the **Attributes** list.
2. Type the new attribute name and add an optional description.
3. In the **Data type** list, select the **Data type** for the new attribute. The available features and basic options change depending on your selection in the **Data type** list.
4. Set the remaining parameters as needed.

Note: For detailed information about each item on the **Attributes** page, see [About the attributes page](#).

5. Lock the values, if needed. The lock is available only when you are working with a template. If you are working with an instance, it shows the lock status for the value in the parent object.
6. Set any security you need. For more information about setting security, see [Set object security](#).
7. Save and close the Object Editor when you are done.

Attributes and scripting

Follow these guidelines and best practices when using attributes in scripts:

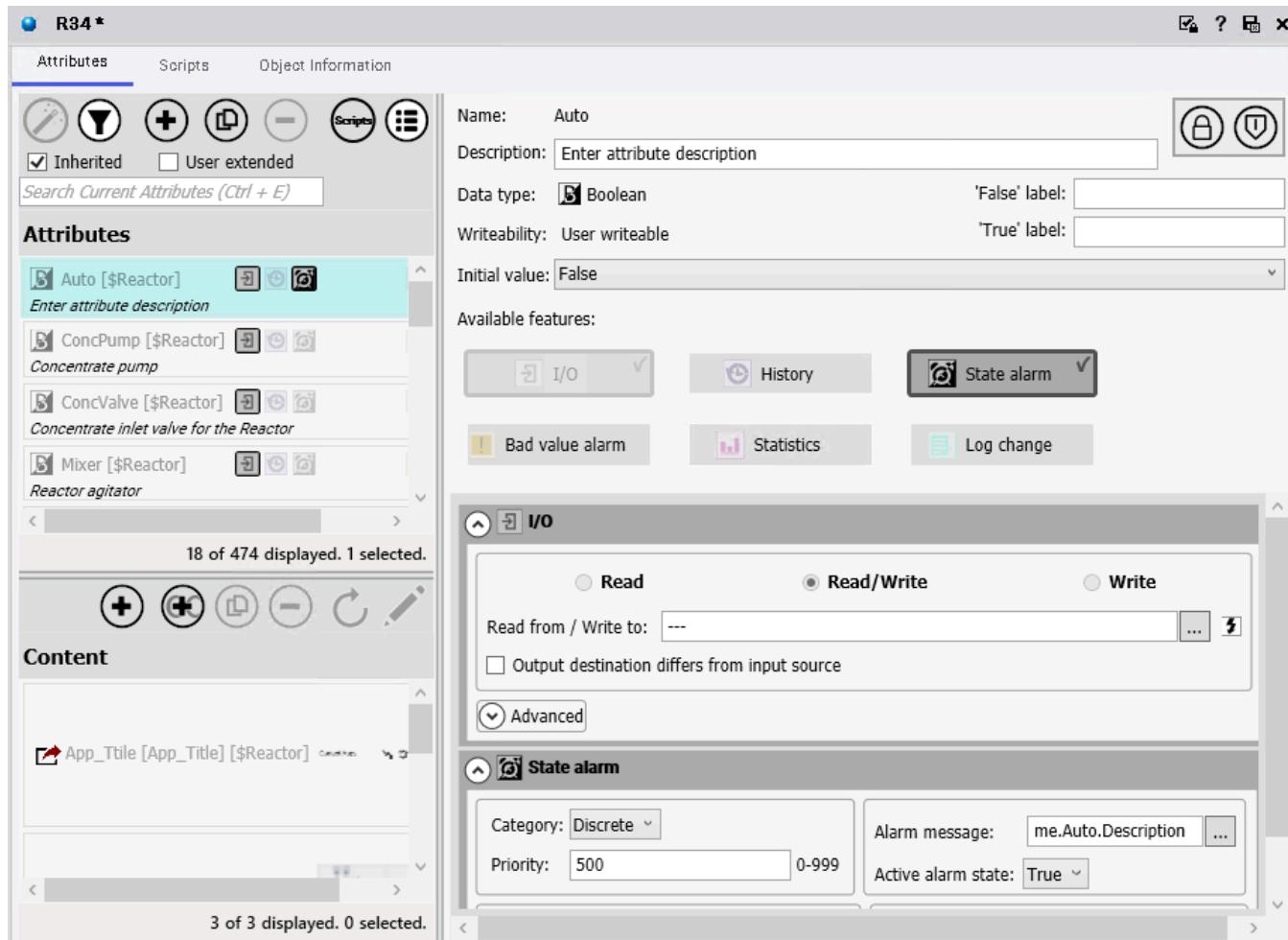
- If you use **Calculated** and **Calculated retentive** attributes as counters, they must be manually initialized. For example, if you use `me.Attribute=me.Attribute+1` as a counter in a script, you must also initialize the attribute with something like `me.Attribute=1` or `me.Attribute=<some attribute value>`.
- **Calculated** attributes can be initialized in scripts with **Execution type** triggers of **On Scan** and **Execute**, but not initialized in **Startup** scripts.
- If you use an attribute **array** in a **startup script**, value and quality may become bad when the array is initialized in the startup script. To prevent this, set the dimension after initializing the array as shown below. This will ensure that the value is set and the attribute quality is good.

```
Me.ArrayAttribute[]="0,0,0,0,0,0,0,0,0,0";
Me.ArrayAttribute.Dimension1 = 10;
```

- You must initialize **Calculated retentive** attributes in **Startup** scripts and you can initialize these attributes in **On Scan** and **Execute** scripts. A **Calculated retentive** attribute retains the attribute's current value after a computer restart, redundancy-related failover, or similar situation in which valid checkpoint data is present. Your **Startup** script should contain a statement testing the Boolean value of the `Engine.StartingFromCheckpoint` attribute on the object's `AppEngine`. If the value is `TRUE`, do not initialize the attribute. If the value is `FALSE`, initialize the attribute. For more information about `Engine.StartingFromCheckpoint`, see the help for the `AppEngine` object, available from the `AppEngine` object editor.
- We recommend that you **do not use hidden attributes** (attribute names that begin with an underscore) in scripts.

Add features to attributes

The **Attributes** page allows you to configure an existing attribute for I/O, alarms, and history functionality not embedded in the original object.



Features inheritance

You can add Features to attributes that are in either derived templates or in instances. You cannot add Features to attributes in Base templates. The following parent-child object characteristics also apply to Features added to objects:

- If you add a Feature to an attribute in a derived template that has objects derived from it, all child objects inherit the Feature.
- You cannot add a Feature to attributes on derived objects that duplicate parent object Features in name and type.
- You cannot add a Feature with the same name as an existing Feature.
- Renaming a Feature on an attribute in the template to which it was originally added renames all other objects derived from the template. This change happens when the template is checked in.
- You can check in a template with an attribute configured with a new Feature with the same name as an

existing Feature on an attribute in a derived object. The template definition of the Feature overrides the Feature in the derived object.

- If you remove a Feature on an attribute from a template, that Feature is removed from any child object. You see the change when you check in the template.

To create and associate a Feature with an object

1. On the **Attributes** page, select an attribute from the **Attributes List**. The available Features change to the Features allowed for the attribute.
2. Click the button for the Feature you want to apply to the selected attribute. The associated parameters for each kind of Feature become available. For detailed information about each item on the **Attributes** page, see [About the attributes page](#).
3. Select the parameters for the Feature you have added. Available Features, which vary depending on the attribute's Data Type and Writeability, are the following:
 - **I/O:** For information about adding the I/O Feature, see [The I/O Feature](#).
 - **History:** For information about adding the History Feature, see [The History Feature](#).
 - **Limit alarms:** For information about adding a Limit alarm Feature, see [The Limit Alarms Feature](#).
 - **ROC alarms:** For information about adding a Rate of Change (ROC) alarm Feature, see [The Rate of Change \(ROC\) Alarms Feature](#).
 - **Deviation alarms:** For information about adding a Deviation alarm Feature, see [The Deviation Alarms Feature](#).
 - **State alarm:** For information about adding a State alarm, see [The State Alarm Feature](#).
 - **Bad Value alarm:** For information about adding a Bad Value alarm, see [The Bad Value Alarms Feature](#).
 - **Statistics:** For information about adding a Statistics Feature, see [The Statistics Feature](#).
 - **Log change:** For information about adding a Log Change Feature, see [Using the Log Change Feature](#).
4. Lock the values, if needed. The lock symbol is available only when you are working with a template. If you are working with an instance, it shows the lock condition of the value in the parent object.
5. Set any security for the attribute. For more information about setting security, see [Set object security](#).
6. Save and close the Object Editor to include the new Features in the configured object.

The I/O Feature

The I/O Feature allows you to configure all aspects of data input and output for an attribute.

You can configure I/O type and you can specify input sources and output destinations. The I/O types you can specify are:

- Read (Input): See [Configure I/O as read-only](#).
- Read/Write (InputOutput): See [Configure I/O as read/write](#).
- Write (Output): See [Configure I/O as write-only](#).

You can also configure advanced properties, described in the following table. The attribute's data type and I/O type determine what Advanced I/O properties are available.

I/O Feature Advanced Property	Description
Buffered	<p>Enable buffered data to propagate to data subscribers the entire subset of values accumulated within a single scan cycle.</p> <p>Buffering data ensures that if a given attribute changes its value several times during a single scan cycle, there is no folding of data, which occurs when an accumulation of values of multiple data changes within a single scan are overwritten and only the latest value is stored.</p> <p>Note: If Galaxy-wide buffering is enabled, you cannot enable (or disable) buffering for individual attributes. With Galaxy-wide buffering, all attributes are buffered. See Auto-Configure Data Buffering for additional information.</p>
Deadband	<p>Specify the minimum amount by which a value must vary in order for the attribute to register a change, for example, by triggering an alarm, historizing an alarm or event, or triggering a script.</p> <p>The Deadband property is not available for string data types.</p>
Reflect input to output	<p>Enable to propagate an input value to an output destination. Enabling automatically disables the Output destination differs from input source option.</p> <p>Typically, set this option when you want to read an input from one source, manipulate its value in a script, and send the manipulated value to a different destination address, all during a single scan of an object. For more information, see Use Read/Write I/O in Scripts.</p> <p>Available only when the I/O type is Read/Write. You must set separate Read from and Write to properties.</p>
Output every scan	<p>Available only when Reflect input to output is selected. Write to the specified output destination occurs even when there has been no state or data change since the previous scan.</p> <p>The timestamp is not updated if there has been no state or data change since the previous scan.</p>

I/O Feature Advanced Property	Description
Enable I/O scaling	<p>Enables scaling between the raw value and the Engineering Units (EU) value. Scaling is the process of taking raw data from a device and presenting it as an appropriate value for your application.</p> <p>Application Server supports two types of data scaling: linear and square root, described in this table.</p> <p>Available only for integer, float, and double data types.</p>
Maximum	<p>Available only when I/O scaling is enabled.</p> <p>Raw: The raw input maximum to be used in the scaling equation.</p> <p>EU value: The maximum value in engineering units to be used in the scaling equation.</p> <p>Extended EU range: The highest value allowed for the attribute before it is clamped, if clamping is enabled, or set to NaN. This value must be greater than or equal to the specified maximum EU value.</p>
Minimum	<p>Available only when I/O scaling is enabled.</p> <p>Raw: The raw input minimum to be used in the scaling equation.</p> <p>EU value: The minimum value in engineering units to be used in the scaling equation.</p> <p>Extended EU range: The lowest value allowed for the attribute before it is clamped, if clamping is enabled, or set to NaN. This value must be less than or equal to the specified minimum EU value.</p>
Conversion mode	<p>Available only when I/O scaling is enabled.</p> <p>Select Linear or Square Root conversion mode from the list.</p> <p>Linear: Typically converts directly from one value scale and type (raw) to another value scale and type (EU).</p> $\text{ScaledValue} = ((\text{RawValue}-\text{RawMin})/(\text{RawMax}-\text{RawMin}) * (\text{EngUnitsMax}-\text{EngUnitsMin})) + \text{EngUnitsMin}$ <p>Square Root: Provides more precise scaling, typically used when the raw value is too large to be usefully scaled to an EU.</p> $\text{ScaledValue} = \sqrt{((\text{RawValue}-\text{RawMin})/(\text{RawMax}-\text{RawMin}))}$ <p>If $\text{RawValue} < \text{RawMin}$, then $\text{ScaledValue} =$</p>

I/O Feature Advanced Property	Description
	$(sqrt(RawMin) * (EngUnitsMax-EngUnitsMin)) + EngUnitsMin$
Clamp input to EU range	<p>Available only when I/O scaling is enabled.</p> <p>If enabled, the scaling calculation result is clamped at either the maximum EU range value or the minimum EU range value, and the attribute quality is set to Uncertain.</p> <p>If not enabled, and the attribute value exceeds the EU range, then the value will continue to scale out of range, and the quality is set to Bad.</p>

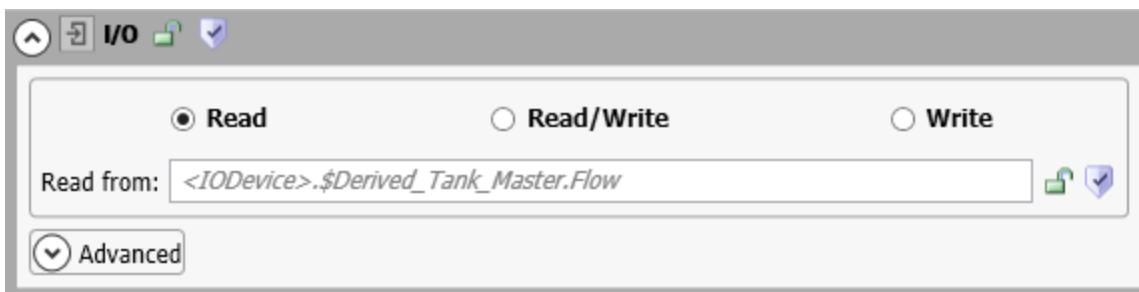
Configure I/O as read-only

Select **Read** in the I/O parameters area. Define an input **source** by using I/O auto-assignment, by typing in the reference string, or by clicking the **Attribute Browser** button at the right.

- Use I/O auto-assignment to prepare the input source for automatic assignment to a configured Device Integration (DI) object or other data source. For information about using I/O auto-assignment, see [I/O auto assignment](#).

Important: If the InputSource attribute is locked in the parent template, the attribute cannot be updated with the resolved reference when the object is deployed, and the runtime value will be "---Auto---".

- Use the **Attribute Browser** to select an attribute and automatically insert the correct reference string for that attribute. For more information about using the Attribute Browser, see [Use the galaxy browser to reference objects](#) .
- **Telemetry Server:** See [ATS-configure I/O for use with AVEVA Telemetry Server Communication Drivers](#) for additional information.



You can add multiple Read (input) I/O Features to an object. However, you cannot add a Read/Write (InputOutput) I/O Feature to an attribute that already has either a Read or a Write (output) I/O Feature. Arrays are not supported.

Note: Lockable attributes can be configured with a Read I/O Feature, but they only function correctly during runtime if the configured attribute is unlocked.

If the data types of the attribute and its I/O source attributes are the same, they are set to equal values according to the object's execution rate. If the two attributes are different data types, coercion rules are applied.

If coercion fails or the input value is out of the attribute's range, quality for the configured attribute is set to Bad. Otherwise, the configured attribute quality matches the **source** attribute. When the object is Off Scan, quality is always Bad and user sets are accepted.

Attributes configured with a Read I/O Feature are not protected by their security classification. The only enforced security specifies if an IDE user can edit or add features to the object. A Read I/O Feature can be added to a template or instance. If added to a template, the existence of the Read I/O Feature is automatically locked in derived objects.

Configure I/O as read/write

Select **Read/Write** in the I/O parameters area. Define an input **source** by using I/O auto-assignment, by typing in the reference string, or by clicking the **Attribute Browser** button at the right.

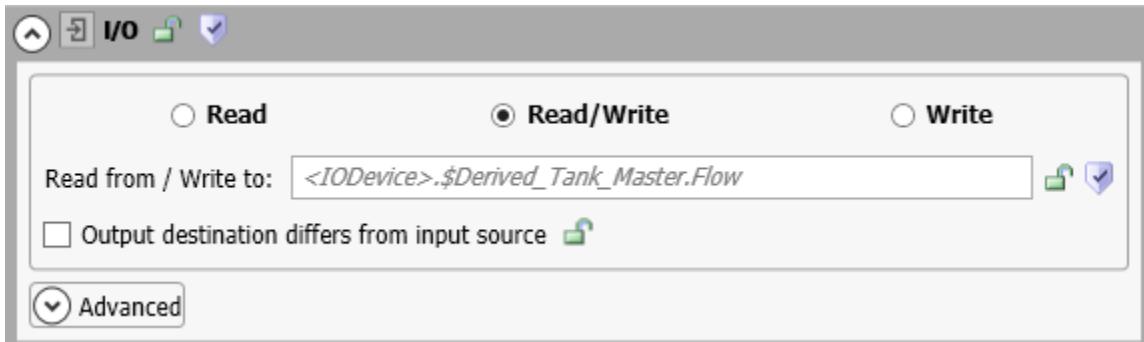
- Use I/O auto-assignment to prepare the input source for automatic assignment to a configured Device Integration object or other data source. For information about using I/O auto-assignment, see [I/O auto assignment](#).
- If InputSource or OutputDest attributes, or both, are locked in the parent template, the attributes cannot be updated with the resolved reference when the object is deployed, and the runtime value will be "---Auto---".
- Use the **Attribute Browser** to select an attribute and automatically insert the correct reference string for that attribute. For more information about using the Attribute Browser, see [Use the galaxy browser to reference objects](#).
- If you are configuring an I/O attribute for use with the Telemetry Server, see [ATS-configure I/O for use with AVEVA Telemetry Server Communication Drivers](#) for additional information.

If the output destination and the input source are not the same, click **Output destination differs from input source**. Enter a **Destination** attribute by using I/O auto-assignment, by typing in the reference string, or by using the **Attribute Browser** to search for the reference string in an object.

Important: If you clear the **Output destination differs from input source** check box, the **Write to** text box automatically shows "---". In the runtime environment, "---" is the same reference as the **Read from** value entered during configuration time. During runtime, you can change the **Read from** reference. During

configuration, do not lock the **Write to** parameter if you clear the **Output destination differs from input source** check box.

A Read/Write I/O Feature allows an attribute in a template or an instance to be configured so that its value is both read from and written to an external reference. The Read/Write I/O Feature monitors the value/quality of an input and sends outputs on state change.



The **Write to** (output) destination can be the same or different from the **Read from** (source). The references are always to another acceptable attribute type in the Galaxy.

You can add multiple Read/Write I/O Features to an object. However, you cannot add a Read/Write I/O Feature to an attribute that already has a Read or Write I/O Feature.

Note: You can add a Read/Write I/O Feature to lockable attributes, but they only function correctly during runtime if the configured attribute is unlocked.

When objects are on scan

When an object is On Scan, the value and quality of the attribute configured with a Read/Write I/O Feature mirrors the quality of the externally referenced attribute during a successful read. The data quality of the attribute is set to Bad when reads fail. Reads can fail because of communication errors or datatype conversion failures.

While the object is On Scan, the data can change quality. If an external set (for example, from a user) to an attribute changes either the value or quality, then a write of the attribute's value to the destination occurs during the next execute phase. The quality must be Good or Uncertain for a write to occur. For writes to occur because of a quality change, the quality change must be a transition from Bad or Initializing to Good or Uncertain.

The attribute called **WriteValue** is publicly exposed and plays an important role in driving outputs. When the object is Off Scan, quality is always Bad and user sets are accepted.

Use Read/Write I/O in Scripts

Two common types of scripts can be written on an attribute configured with a Read/Write I/O Feature: One can look at the input side and one can look at the output side.

The input side script uses the current value coming from the input source location and performs logic or calculations on it. This script refers directly to the attribute in its expressions. For example, if the attribute is "me.attribute1", the script refers directly to "me.attribute1" for data change conditions and for expressions within the script.

The output side script can modify output or validate a new requested output value. This script refers to the

"WriteValue" attribute configured on the attribute: "me.attribute1.writeValue".

To validate a new requested value to the attribute1, for example, a data change condition expression is written on "me.attribute1.writeValue". In addition, if the script wants to do clamping or validation, it can manipulate the "me.attribute1.writeValue" directly to clamp the output value. For example:

```
If (me.attribute1.writeValue > 100.0 ) then  
Me.attribute1.writeValue = 100.0;  
Endif;
```

The data change expression for this script is "me.attribute1.writeValue" because this value changes when a new value is about to be written to the field.

The script can intercept this value just before output and manipulate it. To prevent WriteValue from being written out, its data quality can be set to Bad with the SetBad() function.

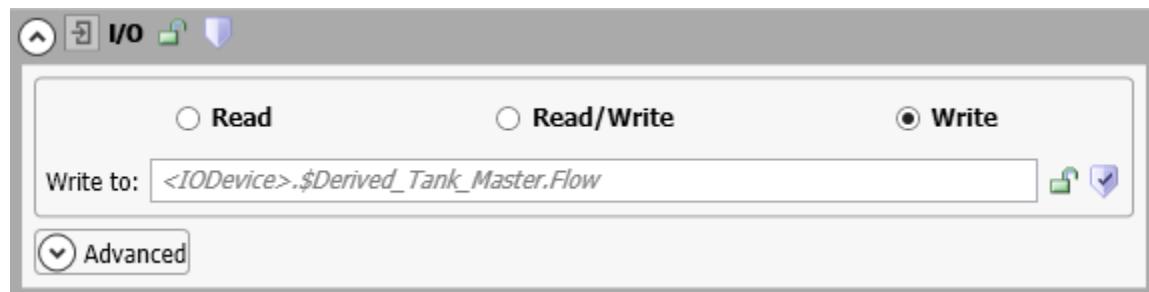
For more information, see [Work with Outputs](#).

Configure I/O as write-only

Select **Write** in the I/O parameters area. Define a **Write to (output) destination** by using I/O auto-assignment, by typing in the reference string, or by clicking the **Attribute Browser** button at the right.

- Use I/O auto-assignment to prepare the output destination for automatic assignment to a configured Device Integration (DI) object or other data source. For information about using I/O auto-assignment, see [I/O auto assignment](#).
- If the OutputDest attribute is locked in the parent template, the attribute cannot be updated with the resolved reference when the object is deployed, and the runtime value will be "---Auto---".
- Use the **Attribute Browser** to select an attribute and automatically insert the correct reference string for that attribute. For more information about using the Attribute Browser, see [Use the galaxy browser to reference objects](#).

Select the **Output Every Scan** check box if you want the attribute to write to the **Destination** attribute every scan period of the object. Otherwise, the write executes only when the value is modified or when quality changes from Bad or Initializing to Good or Uncertain.



Writeable and Calculated attributes can be configured with a Write I/O Feature. Arrays are not supported.

A Write I/O Feature can be added to a derived template or to an instance. If added to a template, the existence of the Write I/O Feature is automatically locked in derived objects. The output **Destination** attribute is separately lockable in templates.

If the data types of the configured and destination attributes are the same and only when the quality of the extended attribute is good, the two attributes are set to equal values according to the configured object's execution rate. If the two attributes are different data types, coercion rules are applied. If coercion fails, the extended attribute is placed into a configuration error and type mismatch state.

An attribute that is enhanced with a Write I/O Feature has the following characteristics:

- A value can be output only when quality is Good or Uncertain. The quality is not output, only the value is output, because quality is not output on sets.
- When the quality changes from Bad or Initializing to Good or Uncertain, the value is output, even if the value is not modified.
- When the quality changes from Good to Uncertain, with no value modification, the value is not output.
- When the object goes Off Scan, no output is done.
- When the extended object is Off Scan, quality is always Good and user sets are accepted.

Work with Outputs

The following information applies to the functionality of Read/Write and Write Features as well as to the output function of AnalogDevice objects.

If a single set request is made to a destination attribute during a single scan cycle, that value is sent to the destination. During a single scan cycle, though, more than one set request to the same destination is possible. In that case, folding occurs and the last value is sent to the destination.

During a single scan cycle, only the last value requested during a scan cycle is sent to its destination when the object executes. Its status is marked as Pending as it waits for write confirmation from the destination object. All other set requests during that scan cycle are marked as successfully completed.

If one or more new sets are requested during the next scan cycle, then the second scan cycle's value is determined as described in the preceding paragraphs. It is then sent to the destination when the object executes again and the value sent to the destination during the previous scan cycle is marked with successful completion status even if write confirmation is not received.

Within a single scan cycle, data is folded and only the last set requested is sent to the destination, unless buffering is enabled. For example, an {11,24,35,35,22,36,40} sequence of set requests results in a value of 40 being sent to the destination object. All other values result in successful completion status.

The exception to this behavior is when you enable buffering, either on a Galaxy-wide basis or for an individual attribute. For more information, see I/O Feature advanced properties information in [The I/O Feature](#).

Boolean data types are an exception to folding behavior when buffering is not enabled. This allows an unknown user input rate (for example, repeated button pushes) with a consistent object scan rate for outputs, and creates reproducible results.

In this case, a combination of folding as described plus maintenance of a queue of one element deep better meets the expectation of users. To begin with, the first value set after the object is deployed (the default True or False) is always written to its destination.

Subsequently, the following occurs during a single scan cycle: A two-tiered caching scheme of a Value to be Sent and a Next Value to be Sent is implemented. The Value to be Sent is based on data change as compared to the last value sent to the destination object. The Next Value to be Sent is based on data change as compared to the Value to be Sent value.

When the first data change occurs, the new value is cached in the Value to be Sent queue. Folding occurs if the same value is requested again. If another value change occurs, this second value is cached in the Next Value to be Sent queue. Again, folding occurs if the same value is requested again.

The Value to be Sent value is sent during the next scan cycle, and the Next Value to be Sent value is sent during the following scan cycle.

Note: In the case of Boolean data types used in Supervisory sets (sets between ApplicationObjects) or a mixture of Supervisory and User sets during a single scan cycle, the behavior is the same as the other data types.

For Boolean data types and User sets, the following examples apply:

Previous Scan Cycle Value Sent	Scan Cycle Set Requests	Value to be Sent	Next Value to be Sent
0	1,0,0,1,1	1	none
1	1,0,0,1,1	0	1
0	1,1,0,0	1	0
1	1,1,0,0	0	none

When the same attribute is extended with an Input extension and an Output extension, writes to the Output extension's **Destination** occur every scan regardless of whether the extended attribute has changed.

This behavior occurs even when the **Output Every Scan** check box is cleared, which may add more network traffic. The behavior does not apply to an Input extension.

I/O data quality

When the object is On Scan, the value and quality of an attribute configured with a Read I/O Feature mirrors the quality of the externally referenced attribute in the case of successful reads. The data quality of the configured attribute is set to Bad when reads fail because of communication errors or datatype conversion failures.

While the extended object is On Scan, it behaves as follows: If an external set (for example, from a user) to the configured attribute causes either the value or quality to change, then a write of the configured attribute's value to the destination occurs during the next execute phase.

The quality must be Good or Uncertain for a write to occur. For writes to occur because of a quality change, the quality change must be a transition from Bad or Initializing to Good or Uncertain. The attribute called WriteValue is publicly exposed.

When the configured object is Off Scan, quality is always Bad and user sets are accepted.

ATS-configure I/O for use with AVEVA Telemetry Server Communication Drivers

If you are using AVEVA Telemetry Server Communication Drivers to send and receive data from Application Server, you will need to configure I/O attributes, in accordance with the applicable communications protocol. The Telemetry Server communicates with outstations via the DNP3, IEC-60870, and/or Modbus protocols. Use the reference syntax that corresponds with the protocol you are using. Since some outstations may send data to the Telemetry Server only once per day, enable buffering for the Galaxy to ensure that data is not lost. See Auto-Configure Data Buffering for information about enabling buffering on a Galaxy-wide basis.

Note: If multiple attributes are configured for buffered data from the same data point in the Telemetry Server, or if auto-buffering is enabled, only one attribute will receive the buffered data from that data point at runtime.

Dynamic References

You can configure Telemetry Server data points using dynamic references. When you create and deploy an attribute with a dynamic reference in Application Server, the point is automatically generated on the Telemetry Server, and eliminates the need to manually create the same reference in two places.

Additional Information

For more information about enabling buffered data and creating Telemetry Server references, see the following sections:

- Auto-Configure Data Buffering
- Working with Buffered Data
- Telemetry Server Data References
- Telemetry Server Dynamic Tags

The History Feature

Any attribute that exists at run time and is not already historized can be configured with a history Feature.

Name: Attribute001

Description: Attribute description

Data type: Integer Array

Writeability: User writeable

Initial value: 0 Eng units:

Available features:

- I/O
- History**
- Limit alarms
- ROC alarms
- Deviation alarms
- Bad value alarm
- Statistics
- Log change

History

Description: me.Attribute001.Description

Force storage period: 0 ms

Value deadband: 0.0 EU

Trend high: 10.0 EU

Trend low: 0.0 EU

Enable swinging door

Rate deadband: 0.0 %

Interpolation type: SystemDefault

Rollover value: 0.0

A history Feature can be added to a template or an instance attribute. If added to a template attribute, the existence of the history Feature is automatically locked in derived objects.

You can configure Writeable and Calculated attributes of the following data types with a history Feature:

- Float, Double (stored as a Float)
- Integer
- Boolean
- String stored as Unicode, 512 character limit
- Custom Enumeration stored as an Integer
- ElapsedTime stored as seconds

You can configure the following attributes for a history Feature:

History Feature Attributes	Description
Description	<p>The attribute containing the description string, if any, associated with the attribute being configured.</p> <p>Enter the attribute manually, or use the browse button to open the Galaxy Browser to find a specific attribute.</p> <p>Example: For the description associated with Attribute001 that has been added to object Reactor31, you would enter or browse to "Reactor31.Attribute001.description".</p> <p>The description also can be a simple string without reference to an attribute.</p>
Force storage period	<p>The Historian normally stores values by exception. The force storage period is designed for situations where data is expected to change very infrequently, compared to the engine scan rate, for examples, setpoints and values which only toggle in abnormal cases.</p> <p>The force storage period sets a length of time, in milliseconds, after which the value must be historized, even if the value has not changed, or has not changed more than the Value deadband. Setting the value to 0 disables the parameter.</p> <p>Example: A setting of 3600000 (60 seconds/minute x 60 minutes/hour x 1000) will force the value to be stored once per hour (if the value remains unchanged), as measured from the time the object was last put onscan.</p> <p>The force storage period should be long relative to the AppEngine scan cycle. If the force storage period is shorter than the scan cycle, forced storage will occur at every scan period, as the value change is not recorded until the scan takes place.</p> <p>When the storage period is set to a value greater than the scan period and there is no change in value, or if the change in value is less than the Value deadband, the forced storage period artificially updates the timestamp on the previous (unchanged) value, and allows the Historian to store that value again with the artificial timestamp.</p> <p>In most cases, that is, in cases where a change in value occurs, values are associated with the timestamp from the source system (e.g., PLC), and that time is applied to the historized values. On the other hand, if a value</p>

History Feature Attributes	Description
	<p>is historized because of the force storage period, the value is instead stored with the system time of the engine scan when the force storage period was exceeded. Factors such as communications latency, clock differences, and detailed engine timing can result in force storage values being historized out of order or overlapping with those values historized due to value changes.</p> <p>It is also possible that the timeout for the forced storage period may coincide exactly with the timestamp of an actual exception update from the instrumentation. If this occurs, both values are stored with the associated timestamp. To avoid this, disable (set to 0) the force storage period in cases where the value normally updates frequently. In cases where the value rarely changes, set the force storage period to a much longer time period, for example, to a value between 1 and 48 hours (3600000 to 172800000 ms).</p>
Value deadband	<p>The threshold value, measured in engineering units, that the absolute value of the difference between the new and last-stored values must differ before storing the new value to history. The value deadband is intended to filter out low-level noise in the signal so that only significant changes are historized. When a value deadband is set, value changes that are within the value deadband will not be historized. Therefore, you should only specify a value deadband when the reduction of processing and storage overhead is more important than completeness of the data. If you choose to set a value deadband, the value should be set to as small a value as possible. The value should be based on how large (or small) the changes in value that are expected.</p> <p>A value of zero (0) is valid and means that any level of change results in the new value being stored.</p> <p>A change in Quality always causes a new record to be stored, regardless of whether the Value has changed.</p> <p>When combined with the Force storage period, new values are compared with the last historized value, regardless of whether the previous value was historized because it exceeded the value deadband or exceeded the force storage period.</p>

History Feature Attributes	Description
Trend high	<p>The default top of a trend scale.</p> <p>This value must be greater than or equal to the low value for the trend.</p> <p>If this value is changed at run time, the maximum engineering unit change is not reflected in the Historian until you redeploy the object.</p> <p>This attribute applies only to numeric data types.</p>
Trend low	<p>The default bottom of a trend scale.</p> <p>This value must be less than or equal to the high value for the trend.</p> <p>If this value is changed at run time, the minimum engineering unit change is not reflected in the Historian until you redeploy the object.</p> <p>This attribute only applies to numeric data types.</p>
Enable swinging door	<p>Enable and provide a valid swinging door rate deadband and the force storage period becomes the deadband override period.</p> <p>Boolean or string data types cannot be configured with a swinging door deadband.</p>
Rate deadband	<p>Available only if swinging door is enabled.</p> <p>The percentage rate of change deadband based on the change in the slope of incoming data values to the Historian.</p> <p>Example: A swinging door rate deadband of 10 percent means that data is saved to the Historian if the percentage change in slope of consecutive data values exceeds 10 percent.</p> <p>Default is 0.0, which indicates a swinging door rate deadband is not applied. Any percentage greater than 0.0 can be assigned to the rate deadband.</p>

History Feature Attributes	Description
Interpolation type	<p>The method used by the Historian to interpolate analog historical data. The interpolation type determines which analog value is selected during a Historian data retrieval cycle.</p> <p>Select an Interpolation type:</p> <p>System Default: The Historian system-wide interpolation setting is used. The system-wide setting must be either stairstep or linear interpolated.</p> <p>Stairstep: The last known value is returned with the given cycle time. If no valid value can be found, a NULL is returned to the Historian.</p> <p>Linear: Historian calculates a new value at the given cycle time by interpolating between the last known value prior to the cycle time and the first value after the cycle time.</p>
Rollover value	<p>A non-negative numeric value that represents a tag's reset limit when the Historian operates in counter retrieval mode.</p> <p>In counter retrieval mode the Historian uses a tag's rollover value to calculate and return the delta change between consecutive retrieval cycles.</p> <p>The default value is 0.0.</p> <p>Boolean and string data types cannot be configured with a rollover value.</p>

The Limit Alarms Feature

Select the **Limit alarms** Feature to add and configure a Limit Alarm on an attribute of Integer, Float, or Double data type. You can add a Limit alarm Feature to a template or instance. If added to a template attribute, the Limit alarm Feature is automatically locked in derived objects. Limit alarm Features cannot be added to attribute arrays.

Name: Attribute001

Description: Attribute description

Data type: Integer Array

Writeability: User writeable

Initial value: 0 Eng units:

Available features:

I/O History Limit alarms ✓ ROC alarms

Deviation alarms Bad value alarm Statistics Log change

Limit alarms

	Limit	Priority	Alarm message
<input checked="" type="checkbox"/> HiHi	90.0	500	me.Attribute001.Description <input style="border: none; width: 20px; height: 20px;" type="button" value="..."/>
<input checked="" type="checkbox"/> Hi	75.0	500	me.Attribute001.Description <input style="border: none; width: 20px; height: 20px;" type="button" value="..."/>
<input checked="" type="checkbox"/> Lo	25.0	500	me.Attribute001.Description <input style="border: none; width: 20px; height: 20px;" type="button" value="..."/>
<input checked="" type="checkbox"/> LoLo	10.0	500	me.Attribute001.Description <input style="border: none; width: 20px; height: 20px;" type="button" value="..."/>

Alarm deadband: 0.0

Time deadband: 00:00:00.0000000

You can enable up to four categories of Limit alarms. When enabled, alarms will be triggered when the value reaches the configured limit.

- HiHi
- Hi
- Lo
- LoLo

For each Limit alarm, specify the following parameters:

Limit Alarm Feature Parameters	Description
Limit	The limit that the attribute value must exceed to trigger an alarm. Enter the limit value for each enabled Limit alarm category.
Priority	A numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent. For more information about pre-configuring alarm priorities, see Configure priority ranges for alarm historization, mapping, and shelving .
Alarm message	Browse and select an existing attribute or type a text string as an alarm message. This text string appears in the InTouch alarm view.
Alarm deadband	The amount, in engineering units, that the attribute value must exceed the configured HiHi, Hi, Lo, or LoLo limit before an alarm is triggered.
Time deadband	The time, in seconds, that must elapse after the attribute value exceeds an alarm limit before the alarm is triggered.

For additional information, see [Configure Limit Alarms](#).

The Rate of Change (ROC) Alarms Feature

Select the **ROC alarms** Feature to add and configure a Rate of Change (ROC) alarm on an attribute of Integer, Float, or Double data type. You can add a ROC alarm Feature to a template or instance. If added to a template attribute, the ROC alarm Feature is automatically locked in derived objects. ROC alarm Features cannot be added to attribute arrays.

Name:	Attribute001																					
Description:	Attribute description																					
Data type:	<input checked="" type="checkbox"/> Integer	<input type="checkbox"/> Array																				
Writeability:	<input checked="" type="checkbox"/> User writeable																					
Initial value:	0	Eng units:																				
Available features:																						
 I/O	 History	 Limit alarms																				
 ROC alarms ✓	 Deviation alarms	 Bad value alarm																				
 Statistics	 Log change																					
ROC alarms																						
<table border="1"><thead><tr><th></th><th>Limit</th><th>Priority</th><th>Alarm message</th></tr></thead><tbody><tr><td><input checked="" type="checkbox"/> Up</td><td>0.0</td><td>500</td><td>me.Attribute001.Description <input data-bbox="1085 897 1117 939" type="button" value="..."/></td></tr><tr><td><input checked="" type="checkbox"/> Down</td><td>0.0</td><td>500</td><td>me.Attribute001.Description <input data-bbox="1085 960 1117 1003" type="button" value="..."/></td></tr><tr><td>Changes per:</td><td>Sec</td><td></td><td></td></tr><tr><td>Evaluate every:</td><td>5000</td><td>ms</td><td></td></tr></tbody></table>				Limit	Priority	Alarm message	<input checked="" type="checkbox"/> Up	0.0	500	me.Attribute001.Description <input data-bbox="1085 897 1117 939" type="button" value="..."/>	<input checked="" type="checkbox"/> Down	0.0	500	me.Attribute001.Description <input data-bbox="1085 960 1117 1003" type="button" value="..."/>	Changes per:	Sec			Evaluate every:	5000	ms	
	Limit	Priority	Alarm message																			
<input checked="" type="checkbox"/> Up	0.0	500	me.Attribute001.Description <input data-bbox="1085 897 1117 939" type="button" value="..."/>																			
<input checked="" type="checkbox"/> Down	0.0	500	me.Attribute001.Description <input data-bbox="1085 960 1117 1003" type="button" value="..."/>																			
Changes per:	Sec																					
Evaluate every:	5000	ms																				

You can enable up to two categories of ROC alarms. When enabled, alarms will be triggered when the value reaches the configured limit.

- Up: Rate of increase
- Down: Rate of decrease.

For each ROC alarm, specify the following parameters:

ROC Alarms Feature Parameters	Description
Limit	<p>The limit that the attribute rate of change value must exceed to trigger an alarm. Enter the limit value for each enabled alarm category.</p> <p>The down limit is a positive value, but indicates a negative rate of change, or a descending rate Hi limit.</p>
Priority	<p>A numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent.</p> <p>For more information about pre-configuring alarm priorities, see Configure priority ranges for alarm historization, mapping, and shelving.</p>
Alarm message	<p>Browse and select an existing attribute or type a text string as an alarm message. This text string appears in the InTouch alarm view.</p>
Changes per	<p>Select a unit of time for the rate of change calculation – seconds, minutes, hours, or days.</p>
Evaluate every	<p>The time interval, in milliseconds, at which the rate of change calculation and detection is performed.</p>

For additional information, see [Configure Rate of Change Alarms](#).

The Deviation Alarms Feature

Select the **Deviation alarms** Feature to add and configure a Deviation alarm on an attribute of Integer, Float, or Double data type. You can add a Deviation alarm Feature to a template or instance. If added to a template attribute, the Deviation alarm Feature is automatically locked in derived objects. Deviation alarm Features cannot be added to attribute arrays.

Name: Attribute001

Description: Attribute description

Data type: Integer Array

Writeability: User writeable

Initial value: 0 Eng units:

Available features:

I/O History Limit alarms ROC alarms

Deviation alarms ✓ Bad value alarm Statistics Log change

Deviation alarms

	Tolerance	Priority	Alarm message
<input checked="" type="checkbox"/> Minor	10.0	500	me.Attribute001.Description ...
<input checked="" type="checkbox"/> Major	15.0	500	me.Attribute001.Description ...

Target: 50.0

Deviation deadband: 0.0

Settling period: 00:00:30.0000000

You can enable up to two categories of Deviation alarms. When enabled, an alarm will be triggered when the attribute level deviates from a target value by a configured amount.

- Major: The major alarm deviation tolerance.
- Minor: The minor alarm deviation tolerance.

For each Deviation alarm, specify the following parameters:

Deviation Alarms Feature Parameters	Description
Tolerance	The major or minor deviation tolerance that the attribute value must exceed to trigger an alarm.
Priority	A numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent. For more information about pre-configuring alarm

Deviation Alarms Feature Parameters	Description
	priorities, see Configure priority ranges for alarm historization, mapping, and shelving .
Alarm message	Browse and select an existing attribute or type a text string as an alarm message. This text string appears in the InTouch alarm view.
Target	The setpoint value for deviation calculation.
Deviation deadband	The amount the attribute value must drop (measured in engineering units) below any of the deviation limits before the respective condition is set to FALSE.
Settling period	<p>The time, in seconds, allowed for the attribute value to return within allowable range after a Target change is made before the deviation alarm is triggered.</p> <p>A deviation alarm cannot occur again until the alarm clears.</p> <p>The value you specify will be converted to the following format: days hh:mm:ss.fffffffff.</p>

For additional information, see [Configure Target Deviation Alarms](#).

The State Alarm Feature

Select the **State alarm** Feature to add and configure a State alarm on an attribute of Boolean data type. You can add a State alarm Feature to a template or instance. If added to a template attribute, the State alarm Feature is automatically locked in derived objects. State alarm Features cannot be added to attribute arrays.

Name: Attribute001

Description: Enter attribute description

Data type: Boolean Array

Writeability: User writeable

Initial value: False

'False' label:

'True' label:

Available features:

- I/O
- History
- State alarm** (selected)
- Bad value alarm
- Statistics
- Log change

State alarm

Category: Discrete

Priority: 500

Alarm message: me.Attribute001.Description

Active alarm state: True

Alarm for Attribute: me.Attribute001

Time deadband: 00:00:00.0000000

You can set an alarm message and Priority for a state alarm. The time deadband is used to filter out rapid, transitory value spikes, and the active alarm state determines the trigger value (True or False) that will activate the alarm. You can also link the State alarm to an external attribute to capture alarm conditions detected in the PLC, rather than in the Application Server object.

For each State alarm, specify the following parameters:

State Alarms Feature Parameters	Description
Category	The alarm category. Select a category from the list box. See Configure State Alarms for additional information.
Priority	A numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent. For more information about pre-configuring alarm priorities, see Configure priority ranges for alarm historization, mapping, and shelving .
Alarm message	Browse and select an existing attribute or type a text string as an alarm message. This text string appears in the InTouch alarm view.

State Alarms Feature Parameters	Description
Active alarm state	The state of the active alarm, True or False, when the alarm is triggered.
Alarm for Attribute	<p>This lets you link to an external attribute, such as a PV attribute, to provide historical state alarm data from the PLC. This allows the PLC to pass alarm conditions it detects directly to Application Server. You can use the Attribute Browser to select which attribute to link.</p> <p>The Alarm for Attribute parameter is for use only with Historian Clients and is not compatible with InTouch HMI clients.</p> <p>For more information, see How to Configure an External Alarm Source.</p>
Time deadband	The time, in seconds, that must elapse after the attribute value exceeds an alarm limit before the alarm is triggered.

For additional information, see [Configure State Alarms](#).

The Bad Value Alarms Feature

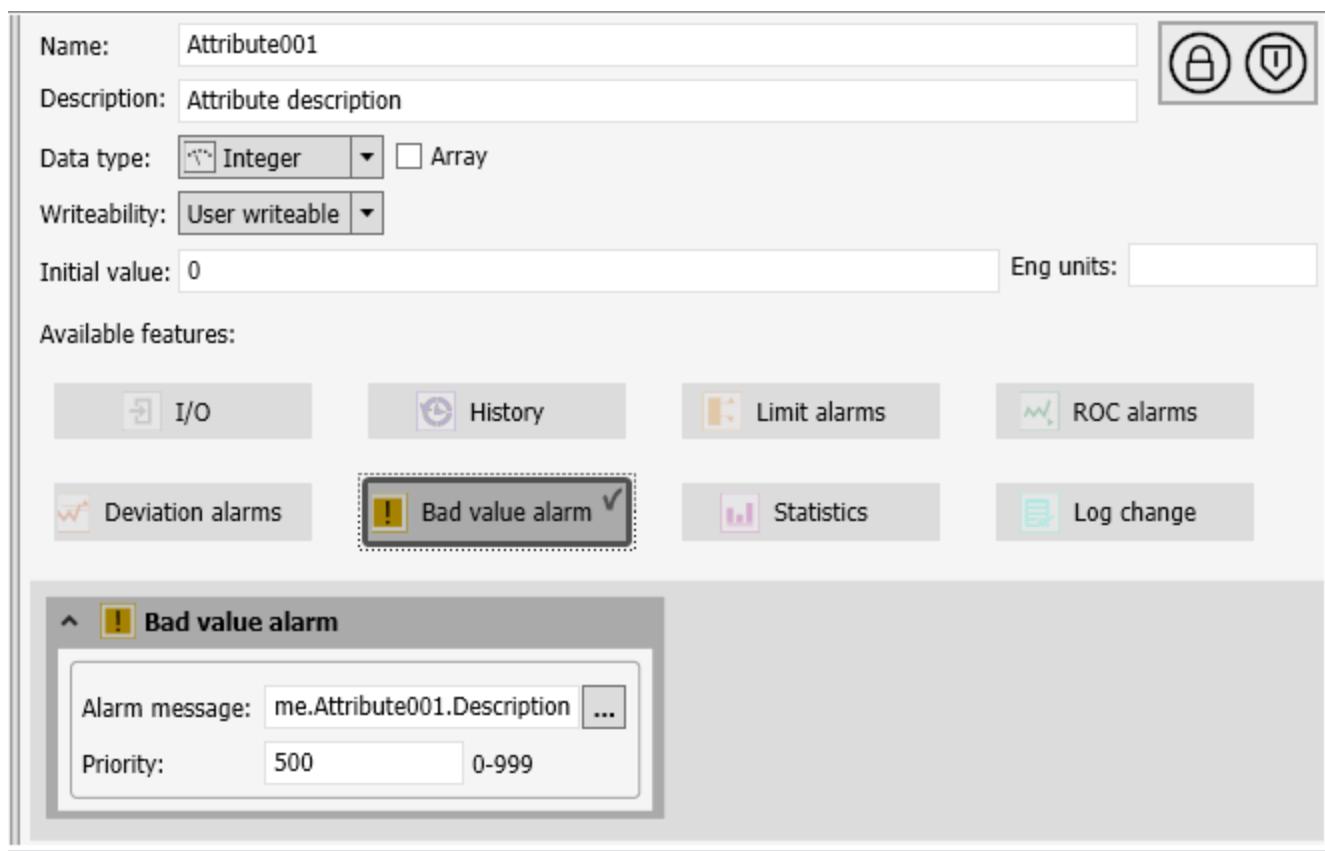
Select the **Bad value alarms** Feature to add and configure a Bad Value alarm on an attribute of Boolean, Integer, Float, or Double data type. If enabled, alarms will be triggered when the attribute has bad data value or data quality.

A bad value results from bad data quality. Bad quality can occur when:

- The PLC indicates that quality is bad (for example, if communications with a physical device is lost).
- There is a communications error (for example, an OPC or I/O server failure).
- There is a configuration error (for example, a broken I/O reference).
- An out of range value is recorded.
- A script sets data quality to bad.

You can configure graphic elements to highlight when a bad data quality state occurs, and its underlying cause. See [Use quality and status indicators to provide runtime feedback](#) for more information.

You can add a Bad Value alarm Feature to a template or instance. If added to a template attribute, the Bad Value alarm Feature is automatically locked in derived objects. Bad Value alarm Features cannot be added to attribute arrays.



In the **Alarm message** box, you can browse and select an existing attribute or you can type a text string as an alarm message. This text string appears in the InTouch alarm view.

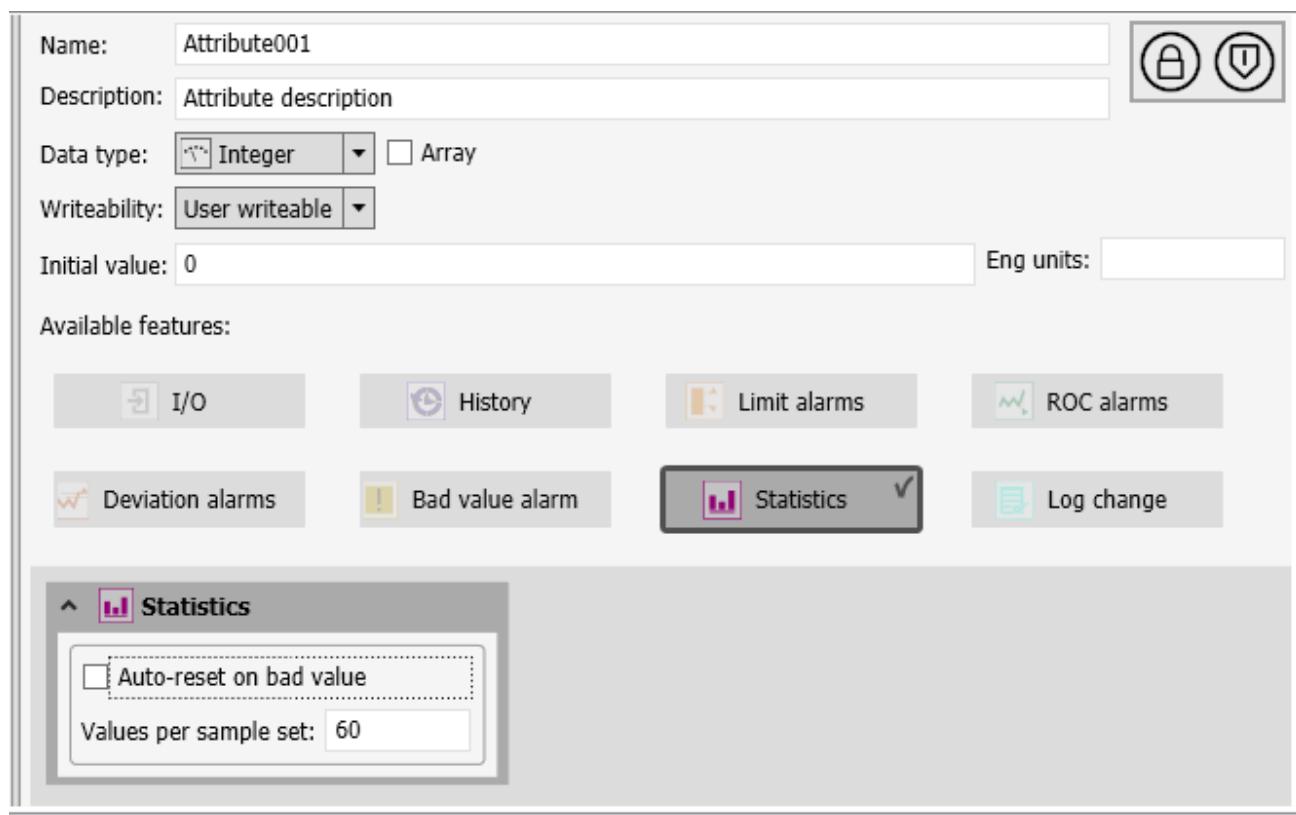
Specify a **Priority** for this alarm, a numeric value for the urgency of the alarm. Valid values are 1 through 999, with 1 being the most urgent.

For more information about pre-configuring alarm priorities, see [Configure priority ranges for alarm historization, mapping, and shelving](#).

The Statistics Feature

Select the **Statistics** Feature to add and configure calculation of statistics on an attribute of Boolean, Integer, Float, or Double data type. If enabled, the following statistics are calculated:

- Time since last transition: If no change in the input value, the elapsed time value increments to show the total elapsed time since the last transition. A change in input value resets the elapsed time since last transition.
- Average: Defined as the sum of the sample size divided by the number of sample size attributes in the data set.
- Standard deviation: Defined as the amount of variation or "scatter" from the average.



You can configure the following parameters for the Statistics Feature:

Statistics Feature Parameters	Description
Auto reset on bad value	If enabled, the input automatically triggers a reset command when the quality of an input goes from BAD or UNCERTAIN to GOOD.
Values per sample set	The maximum number of samples to collect in calculating the mean and standard deviation. The minimum number of samples is 2. A large sample set will impact run-time performance. This attribute is unavailable for a Boolean data type.

Using the Log Change Feature

Select the **Log Change** Feature to log system events for an attribute of Boolean, Integer, Float, or Double data type.



When enabled, the Log Change feature logs data change events as well as user events.

I/O auto assignment

Instead of configuring I/O references manually, or writing scripts to set them at runtime, you can use I/O auto assignment. Manual configuration of I/O references can be time consuming. Scripting these references eliminates the issues of manual configuration, but can significantly increase the time needed for deploying objects. With I/O auto assignment, you do not need to check out individual objects to configure I/O references, and you do not experience the runtime penalties associated with scripting.

Note: I/O auto assignment is the default setting for application and other system objects, such as area objects. Device Integration objects, such as the OPCClient object, must be manually configured.

When you add input or output attributes to an area or application object in the **Attributes** page of the Object Editor, the default setting prepares these attributes for I/O automatic assignment. The auto assignment reference appears in the I/O section of the **Attributes** page if you have enabled the I/O attribute feature. The default string for an input reference is:

`<IODevice>. [ObjectName]. [AttributeName]. InputSource`

where *[ObjectName]* is the hierarchical name of the application or system object, and *[AttributeName]* is the name of the attribute.

The default string for an output reference is:

`<IODevice>. [ObjectName]. [AttributeName]. OutputDest`

The string *<IODevice>* is a placeholder that indicates the I/O reference will be built automatically in a process called auto binding. The reference resolves automatically as an auto-bound reference when you link the object to a scan group and DI object, without having to manually enter or script the reference.

The following is an example of an I/O reference string before the object has been assigned to a scan group and DI

object:

```
<IODevice>.Mixer.Tank.Inlet.InputSource
```

Once you assign the object to a scan group, the reference resolves to include the assigned Device Integration object and scan group. For example, assigning the object to the scan group "Fast" under DI object "OPC001" will change the reference to:

```
OPC001.Fast.Mixer.Tank.Inlet.InputSource
```

Important: Do not lock InputSource or OutputDest attributes when using I/O auto assignment. If either InputSource or OutputDest attributes, or both, are locked in the parent template, the attributes cannot be updated with the resolved auto-bound reference when the object is deployed, and the runtime value will be "---Auto---".

I/O auto assignment is configured in the **IO Devices view**. Use this view to associate application and system objects with DI objects and scan groups.

Scan groups are contained by DI objects and help categorize devices that are associated with them on the basis of how often their I/O points are scanned. For basic information about DI objects, see [Device integration templates](#). For basic information about scan groups, see the OPCClient Object help file.

Note: A DI object will not appear in the **IO Devices view** unless at least one custom scan group has been defined for it.

Assign objects that need to be closely monitored or that have a faster rate of change to a scan group with a faster scan rate. Objects with slower rates of change can be assigned to a scan group with a slower scan rate. The scan rate is the rate at which the AVEVA runtime will perform scans. This may be different than the scan rate a PLC uses to monitor the hardware attached to it.

Note: Fast scan groups consume more network and computing resources than slow scan groups. In a large Galaxy, load balancing may require you to use multiple scan groups with similar scan rates.

The I/O references for the objects' attributes set to use I/O automatic assignment are dynamically generated when you assign an object to a scan group. The auto-bound reference replaces the placeholder string and is displayed in the IDE. It is generally in the following format:

```
<DIObject>.<ScanGroup>.<Object>.<Attribute>
```

<DIObject> and <ScanGroup> correspond to the name of the DI object and scan group to which you assign the object. For example, if the DI object name is DI01 and the scan group name is FastSG, then the auto-bound reference will start with DI01.FastSG.

<Object> may consist of multiple elements if there are contained objects. The following is an example of an object string:

```
Tank3.Valve1.Attr2
```

Here, Tank3.Valve1 corresponds to <object>. The complete auto-bound I/O reference would be:

```
DI01.FastSG.Tank3.Valve1.Attr2
```

Although you assign objects, references are generated for each I/O attribute associated with the object, and not for the object itself. Therefore, you may have to change the I/O references for a subset of an object's attributes. Once objects are associated with DI objects and scan groups, you can validate the auto-bound assignment for each of the object's attributes in the **Mapping view**, and if necessary, you can enter override values in this view.

Assign areas and objects to scan groups

The **IO Devices view** displays all application and system objects, as well as DI objects that have custom scan groups associated with them. Templates are not shown in this view, nor are DI objects that do not have custom

scan groups defined for them (DI objects with only the default scan group are not displayed in the IO Devices view).

Note: You cannot assign application and system objects to a default scan group; these assignments can only be made for custom scan groups.

In the **IO Devices view**, objects and areas are initially unassigned and placed in a flat view under the "Unassigned IO Devices" folder. You then select one or more of these unassigned objects or areas and drag and drop them onto a scan group. Deployed application and system objects cannot be assigned to a scan group. Undeploy them first.

Note: DI objects can remain deployed during the assignment process.

Once an object is attached to a scan group, the I/O references for the objects' attributes set to use I/O automatic assignment are automatically generated for you by concatenating the DI object, scan group, object (including contained objects), and attribute names.

You can assign multiple objects for auto-binding at once. Hold down the **shift** or **ctrl** key to select multiple objects and drag the selected items to a scan group. Object assignment is based on hierarchy and area inheritance, as follows:

- If all the objects you select are at the same hierarchical level, all contained (subordinate) objects that have not been assigned, except sub-areas, are also selected and will move to the same scan group.
 - To assign a sub-area, it must be specifically selected.
 - If some contained objects were previously assigned to a different scan group, these will retain their original assignments. To change an existing assignment, you must specifically select the assigned object and move it to a different scan group.
- If you select objects at different hierarchical levels for assignment to a scan group – for example, if you select an area, an object within the same area, and an object from another area – then only the selected items move to the scan group.
- If you select an area that contains another area, only the area selected will move. You must specifically select an area to assign it, regardless of its hierarchical level.

When an area or containing object is assigned to a scan group, the objects it contains are assigned as follows:

- Objects within the area that share the area's device linkage will be reassigned to the new device along with the area. This remains true even in the case where both the area and the contained objects are not currently linked to a device. Unassigned objects within the area will use the area's assignment.
- Objects within the area that are already assigned to a scan group that is different from that of the hosting area will retain their original assignments.
- If necessary, you can reassign contained objects to a different scan group after the initial assignment, or override assignments in the **Mapping view**.

Note: You cannot change scan group or DI object assignments for deployed objects in the **IO Devices view**. You must undeploy the object first.

Deleting a scan group will delete the I/O assignments of any objects that were previously assigned to it, including override values.

Caution: Creating or deleting scan groups within a derived DI object template may overwrite I/O assignments when these changes are propagated to instances of the DI object template. If this happens, all I/O auto assignment information for objects linked to the updated DI objects, including override values, will be lost. The objects will move to the "Unassigned I/O Devices" folder.

To assign an object or area to a scan group

1. In the **IO Devices view** or **Model view**, select application objects and areas for assignment.
2. Drag and drop the objects and areas to a scan group in the **IO Devices view**.
3. The object or area and all contained objects will be moved to the selected scan group.

Note: If the area or object has contained objects that were previously assigned to a different scan group, these will retain their original assignments. Contained areas will not move and must be overtly selected.

4. The I/O references for each attribute set for I/O auto assignment will be dynamically configured and displayed as auto-bound in the **Mapping view**.

Rename application, system, and device integration objects

You can rename application, system and DI objects in the **IO Devices**, **Model** and **Deployment** views. If an area or object has already been assigned to a scan group, its I/O references are updated automatically.

Rename scan groups

Changes to scan groups (renaming, adding, or deleting them) can only be done by editing the DI object that contains the scan group in the Object Editor. Use the DI object's **Scan Group** page to make any necessary changes.

If scan groups are renamed at the template level, the changes will propagate to the DI object instances and I/O assignments will be preserved.

Note: You cannot rename or delete the default scan group.

Edit and validate I/O assignments

The **Mapping** view is a table that displays I/O auto assignment references for application and system objects that are linked to DI object scan groups. Only auto-bound references are displayed in the **Mapping** view. Application and system objects not yet assigned to a scan group, and manually configured references and are not shown. To be visible in the **Mapping** view, the object must be selected in the **IO Devices** view.

When you initially open the **Mapping** view after starting the IDE, the table will scroll so the far right column is in view.

	Device	ScanGroup	Object	Attribute	I/O	Device.ScanGroup.Override	Object.Attribute.Override	Resulting Reference	Runtime Value
>	Simulator	Normal	4WA_100	ActualClosedL	I		B1000	Simulator.Normal.B1000	
	Simulator	Normal	4WA_100	ActualClosedL	I		B1001	Simulator.Normal.B1001	
	Simulator	Normal	4WA_100	ActualClosedL	I		B1002	Simulator.Normal.B1002	
	Simulator	Normal	4WA_100	ActualClosedL	I		B1003	Simulator.Normal.B1003	
	Simulator	Normal	4WA_100	ActualCmdSta	I		B1004	Simulator.Normal.B1004	

- Selecting a DI object in the **IO Devices** view lists I/O auto assignment attributes for all auto-bound objects linked to all scan groups under it.

- Selecting individual scan groups restricts the scope of the information displayed in the **Mapping** view to the auto-bound objects that have been linked to the selected scan groups.
- Selecting individual application or system object further restricts the scope of attributes displayed to only those associated with the selected object.

Note: You can select multiple items in the **IO Devices** view. Selected items can be at different hierarchical levels. Selecting a subordinate object will exclude non-selected objects within the device hierarchy, even though the parent object is selected.

In the **Mapping** view, you can view and validate I/O references for each automatically generated attribute, and you can override the automatically generated I/O references. As is the case in the **IO Devices** view, you do not have to check out objects to change their I/O assignments.

How to use the mapping view for I/O auto assignment

The **Mapping** view is divided into columns, each of which displays information for an I/O attribute that has been auto assigned.

Mapping									
	Device	ScanGroup	Object	Attribute	I/O	Device.ScanGroup Override	Object.Attribute Override	Resulting Reference	Runtime Value
Drag a column header and drop it here to group by that column									
>	Simulator	Normal	4WA_100	ActualClosedL	I		B1000	Simulator.Normal.B1000	
	Simulator	Normal	4WA_100	ActualClosedL	I		B1001	Simulator.Normal.B1001	
	Simulator	Normal	4WA_100	ActualClosedL	I		B1002	Simulator.Normal.B1002	
	Simulator	Normal	4WA_100	ActualClosedL	I		B1003	Simulator.Normal.B1003	
	Simulator	Normal	4WA_100	ActualCmdSta	I		B1004	Simulator.Normal.B1004	

Column Heading	Definition
Device	Device Integration Object (DIO) name
ScanGroup	Scan group name
Object	Application object or area object name
Attribute	Attribute name
I/O	Input or Output
Device.ScanGroup Override	Override setting: enter an override value to replace the DIO and scan group that was automatically assigned
Object.Attribute Override	Override setting: enter an override value to replace the object and attribute names
Resulting Reference	Concatenated I/O reference string (for example, "DIO1.SG2.Mixer3.Attr5" or "DIO1.SG2.Conveyor.40001")

Column Heading	Definition
Runtime Value	Data returned from runtime when the Validate References button has been clicked

You can change how I/O attributes are displayed by sorting, grouping, or filtering the attributes.

Attributes are initially sorted by Device name (DI object). To sort attributes using a different column, click on the column heading once to sort in ascending order. Click a second time on the same column heading to sort in descending order. A third click on the column heading turns off sorting for that column.

To group objects by I/O assignment criteria, drag a column heading to the top of the table in the **Mapping** view. For example, if you drag the heading "Attribute" to the top of the table, I/O assignments will be grouped by attribute name. In the figure below, I/O assignments are grouped by attribute name. Note each attribute grouping has been expanded.

	Device	ScanGroup	Object	Attribute	I/O	Device.ScanGroup.Override	Object.Attribute.Override	Resulting Reference	Run Value
▼ ActualOpenLS									
▼ ActualOpenLSPortA									
▲ ActualOpenLSPortB									
Simulator	Normal	4WA_100	ActualOpenLS	I		B1006		Simulator.Normal.B1006	
Simulator	Normal	4WM_100	ActualOpenLS	I		B1034		Simulator.Normal.B1034	
Simulator	Normal	4WS_100	ActualOpenLS	I		B1060		Simulator.Normal.B1060	
Simulator	Normal	V3A_100	ActualOpenLS	I		B1611		Simulator.Normal.B1611	

Note: Grouping is not available for the **Device.ScanGroup.Override**, **Object.Attribute.Override**, or **Resulting Reference** columns.

You can use more than one column heading to group I/O assignments. For example, you could group IO assignments first by attribute name and then by I/O type.

Columns can also be filtered to display only certain entries by clicking the filter symbol and then selecting the filtering criteria. When a filter is applied, the color of the filter symbol changes to orange.

Note: Filtering is not available for the **Device.ScanGroup.Override**, **Object.Attribute.Override**, or **Resulting Reference** columns.

The table contains additional information that you can reveal by clicking the **Show Details** button.

Details Column Heading	Definition
Attribute Data Type	Attribute type as defined by the attribute definition set in the Object Editor
Target Data Type	Attribute type of the target, as determined by the runtime
Runtime Quality	Data quality from runtime

Validate I/O references

Use the **IO Devices** pane and the **Mapping** pane to populate the **Resulting References** column in the **Mapping** grid. You can then validate auto-bound I/O attributes that have been assigned in **Mapping**. To begin I/O reference validation, press the **Validate References** button, available in the **Mapping** pane.

Validating I/O references is separate and distinct from the **Validate** function available in the **Template folder** and in the application views. That **Validate** function validates the object's references configuration against the Galaxy Repository. It is a configuration validation.

In contrast, validating I/O references is a runtime validation, and performs a one-time "get" from the runtime subsystem to retrieve the value and quality of the resulting references. A reference string in the **Resulting References** column is validated if a valid value and good quality are returned for that string. Objects should be deployed for an I/O reference validation to be meaningful.

Note: As a best practice, it is more efficient to validate each reference while performing I/O auto-assignment than it is to perform a single validation after configuring and deploying the entire application.

An estimate of the time needed for validation is displayed. The time is updated as validation progresses. When validation completes, the total number of unresolved I/O references is displayed along with the total number of auto-bound references checked, for example, "2 out of 5000 reference(s) failed to validate." To view references that did not validate during the validation period, click the "Show Advanced Columns" button. This will reveal the **Runtime Quality** column.

You can sort the **Runtime Quality** column to aggregate unresolved I/O references at the top of the list of references. See [How to use the mapping view for I/O auto assignment](#) for additional information on sorting. You can then move the associated objects to a different scan group or apply overrides. Recheck the references after completing any required reassessments and overrides.

Override I/O auto assignments

The **Mapping** view contains two override columns. To change an I/O assignment, enter the new parameter in one or both of the two override columns.

The information in other columns cannot be edited directly. You can also reassign an object to a different scan group in the **IO Devices** view, in which case the new assignment will be reflected in the **Mapping** view.

Note: If an object is already deployed, changing any of its I/O assignments in the **Mapping** view will change the object's status from "Deployed" to "Deployed with pending configuration changes."

To override the DI object and scan group assignment that was made in the **IO Devices** view, enter the value you want in the column **Device.ScanGroup Override**. Naming restrictions for the Device.ScanGroup Override value are as follows:

- The value must contain one DI object name and one scan group name contained by the DI object, separated by a period (.).
- The value cannot start or end with a period.
- The value cannot have any leading spaces.
- The I/O reference cannot be longer than 380 characters.

To override the object and attribute name, enter the value you want in the column **Object.Attribute Override**. Naming restrictions for the Object.Attribute Override value are as follows:

- The value cannot have any leading spaces.

- The I/O reference cannot be longer than 380 characters.

You can change attribute information using the Object Editor (after checking out the object), instead of overriding values in the **Mapping** view. However, if you enter the specific I/O reference for an attribute in the Object Editor (manual configuration), it will no longer appear in the **Mapping** view. Only attributes configured for I/O automatic assignment appear in this view.

Batch overrides of I/O assignments

If you have a large number of overrides to enter, information can be copied and pasted between the **Mapping** view and Microsoft Excel. However, you must be careful when moving data to and from the **Mapping** view. Observe the following conditions:

- Only data in the two override columns is editable. All other columns are read only.
- Even though you can only change the override columns, always copy all columns. This will help you verify that data is being copied correctly when you paste back into the **Mapping** view.
- When you copy data from the **Mapping** view, column headers are automatically included to help you while editing in Excel. When you copy your edited data back to the **Mapping** view, do not copy this header row, only copy the data.
- If you need to sort attributes, sort them in the **Mapping** view before you copy them into Excel.
- Do not change the sort order of the attributes in the **Mapping** view until you have pasted the edits made in Excel and applied them.
- Do not sort the attributes in Excel. Attributes can only be sorted in the **Mapping** view and the sort order cannot change until your edits have been applied.

WARNING! The **Mapping** view does not verify data that is copied to it. Therefore, if the order of attributes is different in Excel than it is in the **Mapping** view, the I/O references will be broken. Once changes are applied, there is no way to undo them.

To validate and edit I/O assignments

1. Click the **Validate References** icon to check that the I/O assignments in the **Mapping** table are valid. This action retrieves data and quality information from the runtime data acquisition subsystem.

Important: I/O assignments can only be validated if the Platform, AppEngine, and associated DI object have been successfully configured and deployed.

2. Check the data returned in the **Runtime Value** column. If the assignment is not valid or if the quality is bad, no data will be returned (the **Value** column will be empty), and "Not Connected" will be displayed in the **Runtime Quality** column (click the **Show Details** button to reveal this column).
3. To isolate invalid assignments, select the filter icon at the top of the **Runtime Quality** column filter for "Not Connected" messages.
4. Edit the I/O assignment of objects as needed.
 - To override a scan group and DI object assignment, enter the new value in the **Device.ScanGroup Override** column. The override value for this column cannot contain more than one period (.) to separate the DI object name from the scan group name.
 - To override an object and attribute name, enter the new value in the **Object.Attribute Override** column.
5. Click **Apply Pending Overrides** to save your changes.

- To discard the last unsaved override setting, click **Undo Last Pending Override**.
- To discard all unsaved override settings, click **Undo All Pending Overrides**.

Upload runtime configuration changes for auto-assigned objects

If changes are made during runtime to an auto-assigned I/O reference, you can upload these changes as overrides to the original assignments. Invalid changes are ignored, and an error message is sent to the OCMC Logger.

- Runtime changes to scan groups or DI objects are saved as **Device.ScanGroup** overrides.
- Runtime changes to objects or attributes are saved as **Object.Attribute** overrides.

When you look at the object in the **IO Devices view**, you will see that the original I/O auto assignment setting is retained and the object remains auto assigned to its original DI object and scan group.

The changes uploaded from runtime will appear in the **Resulting References** column of the **Mapping view**. The same information from runtime will appear when you examine the object's I/O attributes in the **Object Editor**. These changes are preserved across subsequent redeployments.

To upload runtime configuration changes

1. From the **Model** or **Derivation** view, select the object(s) with runtime changes you want to preserve.
2. Click **Upload Runtime Changes** from the **Object** menu. The **Upload Runtime Changes** dialog box will appear and provide status of the upload process.

I/O auto-assignment workflow example

This section describes the basic steps for implementing I/O auto-assignment, from start to finish. The basic steps are:

1. Use the **Attributes** page of the **Object Editor** to prepare system objects and application objects for I/O auto assignment.
2. Use the **Object Editor** to configure DI objects and scan groups.
3. Use the **IO Devices** view to assign system and application objects to scan groups (objects must be undeployed).
4. Use the **Mapping** view to validate I/O references. You can enter overrides in this view, if necessary.

To prepare objects for I/O auto assignment

1. Create a derived template from the \$UserDefined base template and open it in the Object Editor.
2. Add attributes as needed. For each attribute that requires input or output, click the I/O button.

The default I/O mode is **Read/Write**, and the default string indicating that the attribute is primed for I/O auto assignment appears in the **Read from / Write to** field.

3. Change the I/O mode as needed.

Important: Do not change the default string if you want to use auto assignment for the attribute. The default string indicates that auto assignment is not yet complete and the object is not linked to a PLC.

4. Save and check in the template, then close the Object Editor.

5. In the **Template folder**, create instances from the UDO derived template. The instances appear in the **Model view**, under Unassigned Area.

To configure DI objects and scan groups

1. In the **Template folder**, create a Device Integration (DI) object from the \$OPCClient, \$DDESuiteLinkClient, or \$Redundant DIOObject template. The DI object appears under "Unassigned Area" in the **Model view** and under "Unassigned IO Devices" in the **IO Devices** view.
2. Add one more scan groups to the DI object you created in the previous step. The naming convention for the scan groups must conform to the naming convention of the PLC with which you are connecting.
The scan groups will show as belonging to the DI object in the **IO Devices view**.

To assign system and application (user created) objects to scan groups

1. Create a new area and set it as the default. Do not deploy the area yet.
2. In the **IO Devices view**, drag and drop the area to a scan group.
3. Create new application objects. As you create new objects, they will appear in the **IO Devices view** under the default area.
The IDE will complete the references for each I/O attribute for you.
4. If you need to change the scan group assignment for an object, drag and drop it to the new scan group.

To assign system and application (user created) objects to scan groups - alternate method

1. Create a new area and assign it to a scan group.
2. Create new application objects. These will appear under the Unassigned folder in the **Model view**.
3. Assign the application objects to the area assigned to the scan group in step 1. All application objects assigned to the area will inherit the area's linkage to the DI object and scan group.
4. If you need to change the scan group assignment for an object in the area, drag and drop it to the new scan group.

To validate I/O references (and override auto assigned values, if necessary)

1. Open the **Mapping** view. This view opens automatically if you select an object, scan group, or DI object in the **IO Devices** view.
Each row in the **Mapping** table contains the I/O reference for one attribute.
2. Enter an override value for the attribute in one of the override columns.

Note: The **DI Object.Scangroup Override** column can only contain the DI object name and scan group name, separated by a period (.).

3. Deploy the platform and associated DI objects.
4. Click the **Validate References** button to check that communication is established with the PLC.
5. Deploy the objects.

I/O auto-assignment with OPC clients

Using I/O auto-assignment with the OPC communication protocol may require that you add overrides to allow

communication between the IDE and certain PLCs, such as Allen-Bradley Logix and Siemens S7 controllers.

If you are using auto assignment to communicate with a device connected to an OPC Server on a different node (for example, a Communication Driver on a remote Galaxy), you will use overrides to build valid I/O references with all required parameters to fulfill the OPC hierarchical path. If bulk edits are required, you can copy and paste between the **Mapping** view and Microsoft Excel.

I/O auto assignment uses the syntax: *<DI Object.ScanGroup>. <Object.Attribute>*. It is the first part of the syntax that limits how the OPCClient builds the I/O auto assignment reference, since additional parameters in the device hierarchy, such as Ethernet ports and backplanes are not included. While these parameters may be logically viewed as part of the *DI Object.ScanGroup* reference, you must add the parameters to the **Object.Attribute override** column (not the **DI Object.ScanGroup override** column) to create the fully qualified I/O reference.

Using DDESuiteLink instead of OPC will accommodate the automatically generated I/O reference to these PLCs, and in most cases, overrides will not be needed.

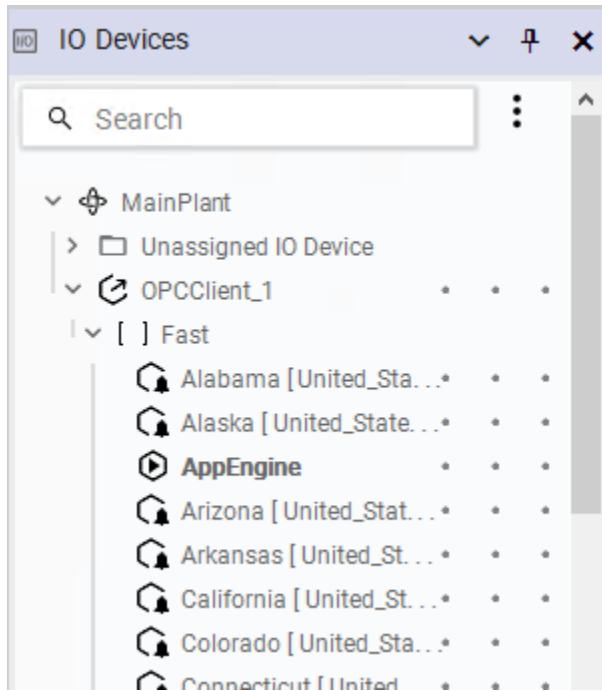
Caution: Do not change the order of references, either in Excel or in the **Mapping** view by sorting, grouping, or filtering the reference list while editing references.

Override scenario 1 — Siemens S7 controllers with an OPCClient object

The following scenario illustrates using I/O auto assignment with a Siemens S7 PLC. The Siemens S7 PLC in this example scenario includes a TCP/IP port that is in addition to the OPCClient object and scan group names captured by I/O auto assignment in the **IO Devices** view.

To configure the Siemens S7 PLC with the OPCClient object

1. Select the OPCClient object or scan group that contains the PLC in the **IO Devices** view.



I/O attributes for each object associated with the OPCClient or scan group (depending on which is selected in the **IO Devices** view) are shown in the **Mapping** view.

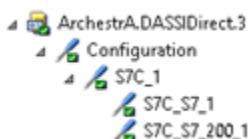
2. In the **Mapping** view, the resulting reference will be in the form:

OPCClientName.ScanGroupName.ObjectName.AttributeName

Example (**Resulting Reference** column):

OPCClient_01.SG1.UDO_01.Attribute_001

3. Do not edit the **Device.ScanGroup Override** column (leave it blank). This leaves the OPCClient Object and scan group assignment as defined in the **IO Devices** view.
4. Enter the fully qualified field reference in the **Object.Attribute Override** column. For example:
S7C_1.S7C_200_1.UDO_01.Attribute001



5. Apply the override. The resulting reference will be displayed. For example:

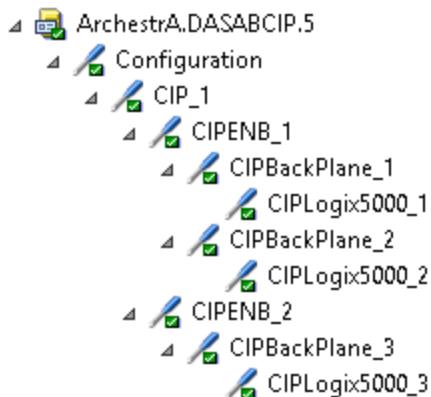
OPCClient_01.SG1.S7C_1.S7C_200_1.UDO_01.Attribute001

Override scenario 2 — Allen-Bradley CIP controllers with an OPCClient object

The following scenario illustrates using I/O auto-assignment with a Allen Bradley Logix PLC. The Allen Bradley Logix PLC in this example scenario includes a device hierarchy with a port object, ethernet port, backplane, and controller. These are in addition to the OPCClient object and scan group names captured by I/O auto assignment in the **IO Devices** view.

To configure the Allen-Bradley Logix PLC with the OPCClient object

1. Select the OPCClient object or scan group that contains the PLC in the **IO Devices** view.
I/O attributes for each object associated with the OPCClient or scan group (depending on which is selected in the **IO Devices** view) are shown in the **Mapping** view.
2. In the **Mapping** view, the resulting reference will be in the form:
OPCClientName.ScanGroupName.ObjectName.AttributeName
Example (**Resulting Reference** column):
OPCClient_01.SG1.UDO_01.Attribute_001
3. Do not edit the **Device.ScanGroup Override** column (leave it blank). This leaves the DI Object and scan group assignment as defined in the **IO Devices** view.
4. Enter the fully qualified field reference in the **Object.Attribute Override** column. For example:
CIP_1.CIPENB_1.CIPBackPlane_1.CIPLogix5000_1.UDO_01. Attribute001



5. Apply the override. The resulting reference will be displayed. For example:

```
OPCCClient_01.SG1.CIP_1.CIPENB_1.CIPBackPlane_1. CIPLogix5000_1.UDO_01.Attribute001
```

Write and edit scripts

You can write scripts that run commands and logical operations based on specified criteria being met to extend and customize your objects. For example, a script can be configured to start running when a key is pressed, a window is opened, or the value of an attribute changes.

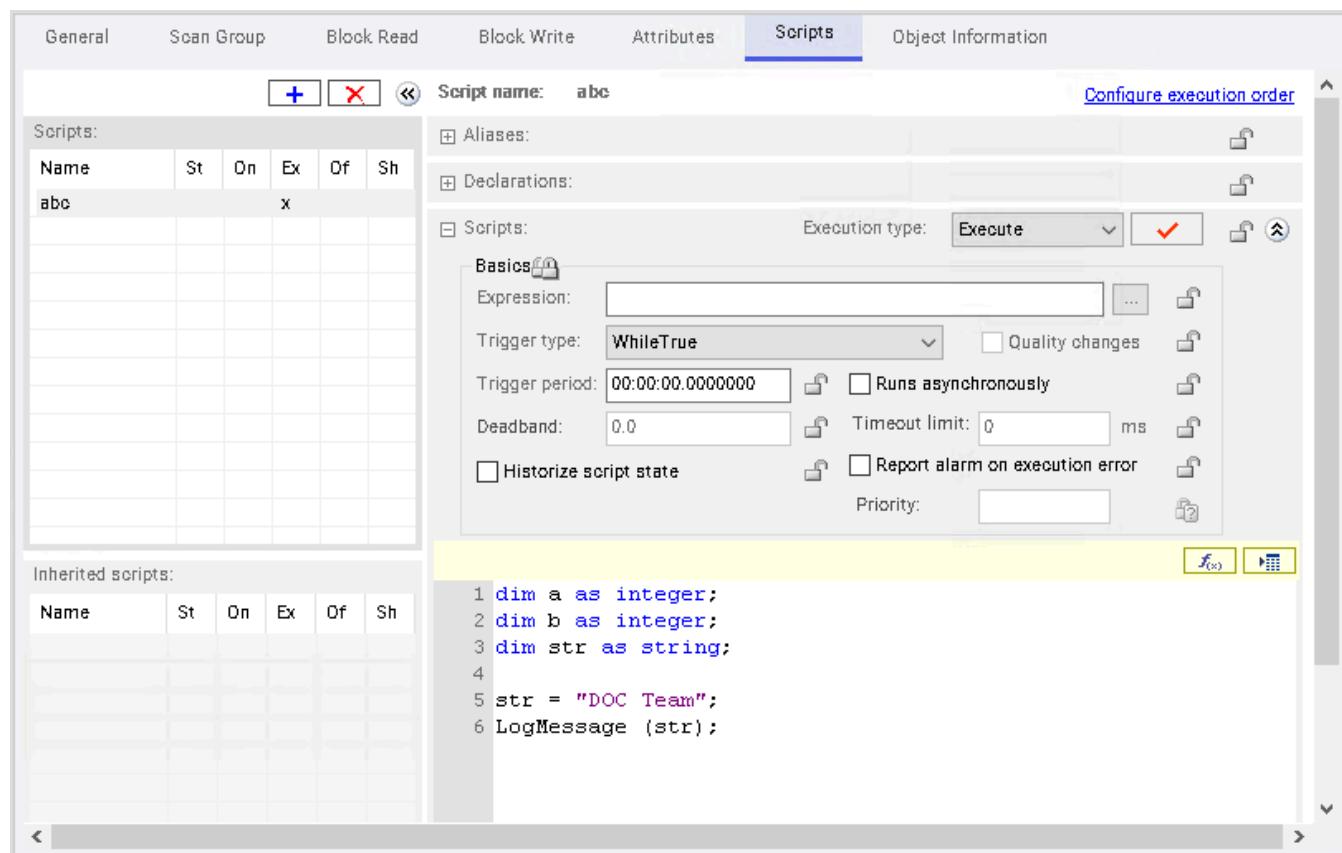
With scripts, you can create a variety of customized and automated system functions. A script adds behavior that runs when the object that contains the script is deployed and the object is either:

- On scan in the runtime environment or
- Changes scan or start/shutdown state

A script typically runs based on attributes of the object that contains it, but can be started by another script based on changing values of attributes of more than one object.

When a script condition is true, the script runs at least once immediately. The maximum length of a script trigger period is 49-days. A script never runs if the trigger period exceeds 49-days.

For specific information about writing scripts, including the scripting language, syntax, commands, and using .NET, see the *Application Server Scripting Guide*. For more information about the scripts page, see [Scripts page](#).



Important: You cannot pass attributes as parameters for system objects. Instead, use a local variable as an intermediary or explicitly convert the attribute to a string using an appropriate function call when calling the system object.

About object scripts

The following characteristics apply to the scripting environment:

- Script text has no length limitations.
- Selecting a script function from the **Script Function Library** dialog box adds it and its syntax to the script text where you can edit it.
- You can save a script with syntax errors, but the object cannot be deployed until you correct the script syntax errors.
- You can validate your scripts before using them. This helps you avoid syntactically correct but semantically incorrect combinations such as two statements declaring the same variable. Variables can be declared only one time in a single block.
- You can change the name of a script at any time by renaming it in the Object Editor.
- In the runtime environment, a script execution error stops the script's current execution. Script execution is retried on the next AppEngine scan.

Script execution

The existence and execution order of scripts associated with an object are inherently locked at each stage of development in the template, derived template, and instance. For example, a set of scripts associated with a template are treated as an ordered block in the **Configure Execution Order** dialog box when configuring execution order in a next-generation derived template.

New scripts in the derived template can be run in any order before and after the template's block of scripts. The derived template's execution order is treated as a block in any downstream derived templates or instances. Scripts cannot trigger any faster than the scan period of the AppEngine the script is associated with or faster than the scan period of the AppEngine that hosts the object that the script is associated with.

Scripts run in one of two modes:

- Synchronous scripting mode is the default for running scripts in the runtime environment. This mode runs scripts in order while an object is running on scan.
- Asynchronous scripting mode is a group of scripts running on the same, lower priority execution thread. These scripts only support Execute triggering and run independently from each other. Set the maximum number of independent threads in the AppEngine configuration editor. To use either scripting mode, you must select **Execute** as the **Execution Type** in the **Scripts** area on the **Scripts** page.

To create a script and associate it with an object

1. Add a script. On the **Scripts** page of the Object Editor, click the **Add (+)** button. A script is added to the **Scripts Name** list.
2. Type a name for the script and press Enter. Script names can be up to 32 alphanumeric characters, including periods. At least one character must be a letter.

Note: For detailed information about each item on the **Scripts** page, see [Scripts page](#).

3. Select a trigger that starts the script in the runtime environment.

Execution Type triggers include: Startup, On Scan, Execute, Off Scan and Shutdown.

- If you select **Startup**, **On Scan**, **Off Scan**, or **Shutdown**, the **Basics** group is unavailable. The script is triggered when the object starts, goes on scan, goes off scan, or shuts down.
If you select **Execute**, the **Basics** group is available.
If you selected **Execute** as the script trigger, select a **Trigger Type**. Depending on the type selected, you are required to enter an **Expression** and/or **Trigger Period** and **Deadband** values. When the combination of **Expression**, **Trigger Type**, **Trigger Period** and/or **Deadband** is satisfied in the runtime environment, the script starts running. See the following table for more information.

The **Trigger Period** format is as follows:

Hours:Minutes:Seconds:Milliseconds

For example, 3 hours, 5 minutes, and 10.5 seconds is:

03:05:10.5000000

Expressions are limited to one language statement in length and calling only synchronous mode script functions. Avoid using script functions with side effects in expressions because subtle behaviors can occur.

Trigger Type	Description
Periodic	When the object containing the script is going On Scan, a Periodic script evaluates its expression at the next scheduled scan period of the AppEngine. The script then runs periodically at the trigger interval specified in the Trigger Period box. A time interval of zero (0) starts the script during every scan. This trigger does not require an expression.
While True	When the object containing the script is going On Scan, a While True script evaluates its expression at the next scheduled scan period of the AppEngine. The script runs if true and then periodically thereafter at the trigger interval. The script continues running as long as the Expression value evaluates to true. A Trigger Period is required. Zero (0) evaluates the expression at the AppEngine scan period and non-zero means the expression is evaluated at the specified time interval.
On True	When the object containing the script is going On Scan, an On True script evaluates its expression at the next scheduled scan period. The script starts at the transition between the expression going from false to true.
On False	When the object containing the script is going On Scan, an On False script evaluates its expression at the next scheduled scan period. The script starts at the transition between the expression going from true to false.
Data Change	<p>Scripts run when the value or quality of the expression changes. The expression must evaluate to a single, non-arrayed value of the following types: integer, real, time, elapsedtime, string, double, Boolean, custom enumeration and quality. To allow execution based on quality, select the Quality changes check box.</p> <p>A deadband can be specified for all data types. Deadband units for time and elapsedtime types are milliseconds. Deadband is always ignored for strings because any change (even from "ABC" to "abc") is considered a change. Only major changes in quality (from Good/Uncertain to Bad/Initializing or vice versa) are considered changes.</p>

Trigger Type	Description
	After the object is put on scan, Data Change-triggered scripts start running at the AppEngine's next scan period and then at each subsequent scan period in which the value or quality changes.
While False	<p>When the object containing the script is going On Scan, a While False script evaluates its expression at the next scheduled scan period, runs if false, and then periodically thereafter at the trigger interval.</p> <p>The script runs as long as the Expression value evaluates to be false. A Trigger Period is required. Zero (0) evaluates the expression at the AppEngine scan period and non-zero means the expression is evaluated at the specified time interval.</p>

4. Select one or more of the following:

- Set the **Runs Asynchronously** and associated **Timeout Limit** parameters, as needed. The Timeout Limit is set in milliseconds. Note that if a script exceeds the Timeout Limit, it continues to execute all sequential statements, but generates a warning in the logger when it completes. If the script contains a loop, it will exit the loop and complete execution of the script, when the timeout limit is reached.
- Select **Report Alarm on Execution Error** and set a **Priority** for the alarm if you want the alarming function to alert you if a script execution failure occurs.
- Select **Historize Script State** to store the state of the script in your application's Historian.

5. In the **Declarations** area, type variable declarations about the script you are writing.

6. Set aliases for the reference strings in the **Aliases** area. This can simplify the script code and allows script code to be created and locked at a template level using alias names. When an individual instance of that template is created, you can link external attributes to the alias names.

In the **Aliases** area, click the **Add** button to add a new alias. An alias is added to the list. The name is shown in edit mode. Double-click the **Reference** entry, and enter a reference string for the alias. You can also click the **Browse** button at the end of the **Reference** block to open the Attribute Browser for easy selection of an object's attributes.

7. Write the script in the **Script Creation** box. Use the **Display Script Function Browser** and **Display Attribute Browser** buttons to help you insert script functions and object attribute references in your script. For help

with the specific commands and syntax, see "QuickScript .NET Functions" in the *Application Server Scripting Guide*.

Click the **Validate Script** button to validate if your script contains any syntax errors.

8. Order the scripts. If you have more than one script associated with a single object, click **Configure Execution Order**. Ordering does not apply to asynchronous scripts. If a script is added to an instance derived from a template that contains scripts, the new script automatically defaults to running after the derived scripts.
9. When you are done creating your script and setting its execution triggering parameters, save and close the Object Editor.

Script locks and change propagation

When you lock a script in a template, the following rules apply:

- The name of a script and its existence is implicitly always locked. This means:
 - You cannot delete the script in derived objects.
 - You cannot change the name of the script in derived objects.
 - If you rename the script in the template, the name changes in all derived objects.
 - If you delete a script from the template, the script is deleted from all derived objects.
 - If you add a script to the template, the script is added to all derived objects.
 - You can add scripts to derived objects. Adding scripts to derived objects does not affect parent object scripts.
- You can lock or unlock the script text in a template. There is script text for **Declarations**, **Execute**, **Startup**, **Shutdown**, **On Scan** and **Off Scan**.
- You cannot separately lock a script in the script editor. A single group lock is used to lock or unlock all scripts at once. Once a script is locked, derived templates and instances cannot modify any of the script text.
- You can lock alias names and alias references separately.
 - When you create aliases for a script, an AliasReference attribute is created for each script with aliases. To lock alias references, you must lock the AliasReference attribute.
 - Changes to alias names only propagate to derived objects when aliases are locked in the **Scripts** page.
 - Changes to alias references only propagate to derived objects when the AliasReference attribute is locked in the **Attributes** page.
 - If the AliasReference attribute is not locked, the alias references will be editable in derived objects, but changes made in the parent template will not propagate to derived objects.
- The script description, runs asynchronous flag, expression, trigger type, trigger period, deadband and execution error alarm are individually lockable and can be locked separately from the script text. A group lock is provided for this group of attributes.
- When you add a script to a template, all properties of the script are editable.
- When you add a script to an instance, all properties of the script are editable, except the lock properties (locks do not serve any purpose in instances since you cannot derive child objects from them).

Important: An expression typically uses attribute references. If you lock the expression and the associated script in a template, use aliases in both the expression and the script. This allows you to specify the attributes that the aliases point to on a per-instance basis while the script code is locked.

The following rules apply to the derivation behavior of locked script attributes:

- If an attribute is locked in a template, then all templates and instances derived from the template share the copy of the value of the locked attribute. A change to the value is only allowed in the template that locked it. The change propagates to all derived templates and instances.

For scripts, locking an attribute of the script, such as its script text or execution type, in a template means all derived templates and instances point to that locked attribute. Future changes to that locked attribute's value, such as modifying the script text, propagate and appear in all derived templates and instances.

If instances are deployed, they are marked pending update status. After they are redeployed, the change to the locked attribute in the template exists in the deployed instance.

- If an attribute is not locked in a template, then all templates and instances derived from that template receive their own copy of the value of that unlocked attribute. A change to that unlocked value is allowed in derived templates and instances because they own their own copy. Any change to the unlocked attribute value in the template does not propagate to any derived template or instance.

An unlocked attribute in a script (such as expression or script order) in a template means that all derived templates and instances have their own copy and the value of that unlocked attribute can change. Future changes to that locked attribute's value (for example, modified expression) in the template does not propagate to any derived template or instance. If instances are deployed, their status does not change to pending update. Redeploying them does not cause the value to change in the deployed instance.

Create and work with content

The **Content** pane of the **Attributes** page lets you:

- Add, modify, rename or delete object-owned graphics. Object-owned means that the graphic exists only for that particular object. Note that object-owned graphics are propagated to child objects.
- Link the object to content (graphics, layouts, External Content) in the Visualization folder. This is shared content, that can be linked to multiple objects. Links to content are propagated to child objects.
- View content inherited from parent templates. However, you cannot add, modify, or delete inherited content. To edit an inherited content item, you must check out and edit the derived template that owns the content.

You must have the derived template or object instance checked out in order to add, modify, or delete content associated with the object; otherwise, the content is read-only and you can only view it.

You can only add content to derived templates or instances, not to a base template.

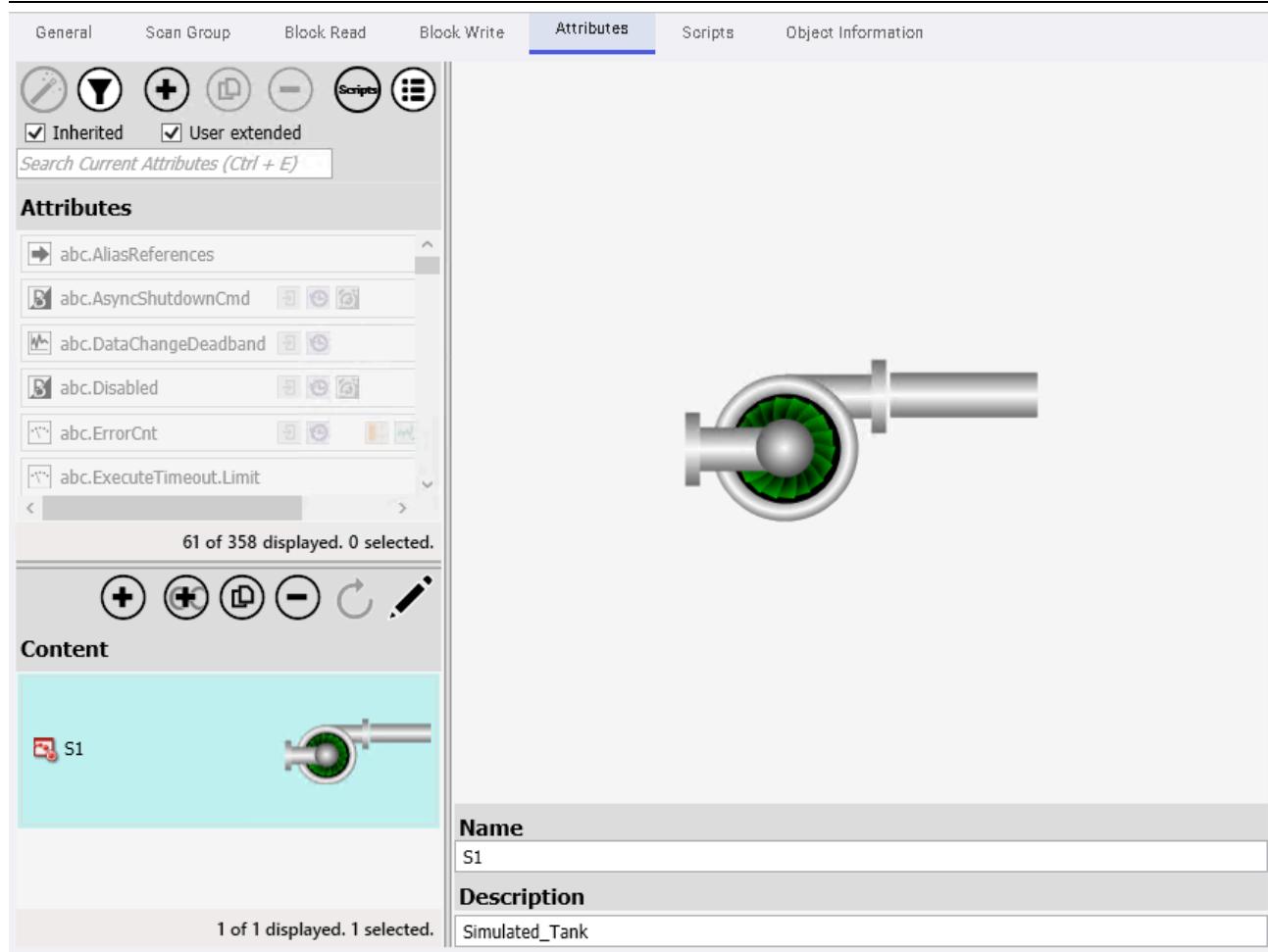
To add or link to content:

Linked content can be a graphic, an external content item, or a layout.

- To link to a content item in the Visualization folder to the object, select the **Link Content** button. This opens the **Galaxy Browser**. Then, select the graphic, layout, or External Content item to be linked to the object. Linked content can be shared by multiple objects. Relative references within the linked content will resolve correctly.

- To create an **object-owned graphic**, select the **Add Content**  button.
- To edit an owned-graphic, select the graphic and click the **Edit**  button.

Note: If you edit a linked graphic, layout or other shared content, the changes will appear in all objects that link to that content.



Add a graphic to an object

This procedure adds an owned graphic to the object. To link a graphic or external content item, see [Link to shared content in the visualization folder](#).

1. From the **Content** pane in the **Attributes** page of the Object Editor, click the **Add (+)** button. A new graphic name and graphic icon are added to the pane.
2. Select the new graphic in the **Content** pane.
3. In the **Name** field, edit the new local graphic name.
4. In the **Description** box, type a description for the graphic being added.
5. Click the **Edit** icon. The Graphic Editor opens. For instructions on using the Graphic Editor, see "About Creating and Managing Industrial Graphics" in the *Creating and Managing Industrial Graphics User Guide*.

Errors and warnings in an owned graphic

When you save an owned graphic that you have edited in the **Graphics Editor**, the owning object is also saved. If

there are any errors and warnings, they will be shown in the **Save Confirmation** dialog. However, the errors and warnings include not just the graphic you are saving, but because the owning object is also being saved, the dialog lists all errors and warning related to the owning object and other graphics owned by the object.

To determine if the warning or error is related to the graphic you are saving, or if it is related to the owning object (or other graphic owned by the object), check the prefix in the warning/error message. The first (left-most) entry in each error/warning message lists the graphic or object name.

Edit a graphic

All modifications made to graphics referenced by other objects are visible in the referenced objects.

To edit a graphic

1. On the **Attributes** page of the Object Editor, in the **Content** area, select the graphic to modify.
2. Click the **Edit Content** icon. The Graphic Editor opens, showing the selected graphic.
3. Make changes. See "About Creating and Managing Industrial Graphics" in the *Creating and Managing Industrial Graphics User Guide*.

Set graphics properties with a galaxy style library

To view and manage the settings for the Galaxy Style Library, select **Configure** on the the **Galaxy** menu. A Galaxy Style Library lets you set the appearance and behaviors of different graphics properties within a ViewApp at runtime.

- [Use element styles to modify graphics](#) — define graphics properties such as font size, font color, fill color, line weight, and line color.
- [Use quality and status indicators to provide runtime feedback](#) — provide visual cues about different quality and status states.
- [Use UI themes to set the appearance of WPF controls](#) — define graphics properties for WPF and navigation controls used in ViewApps.

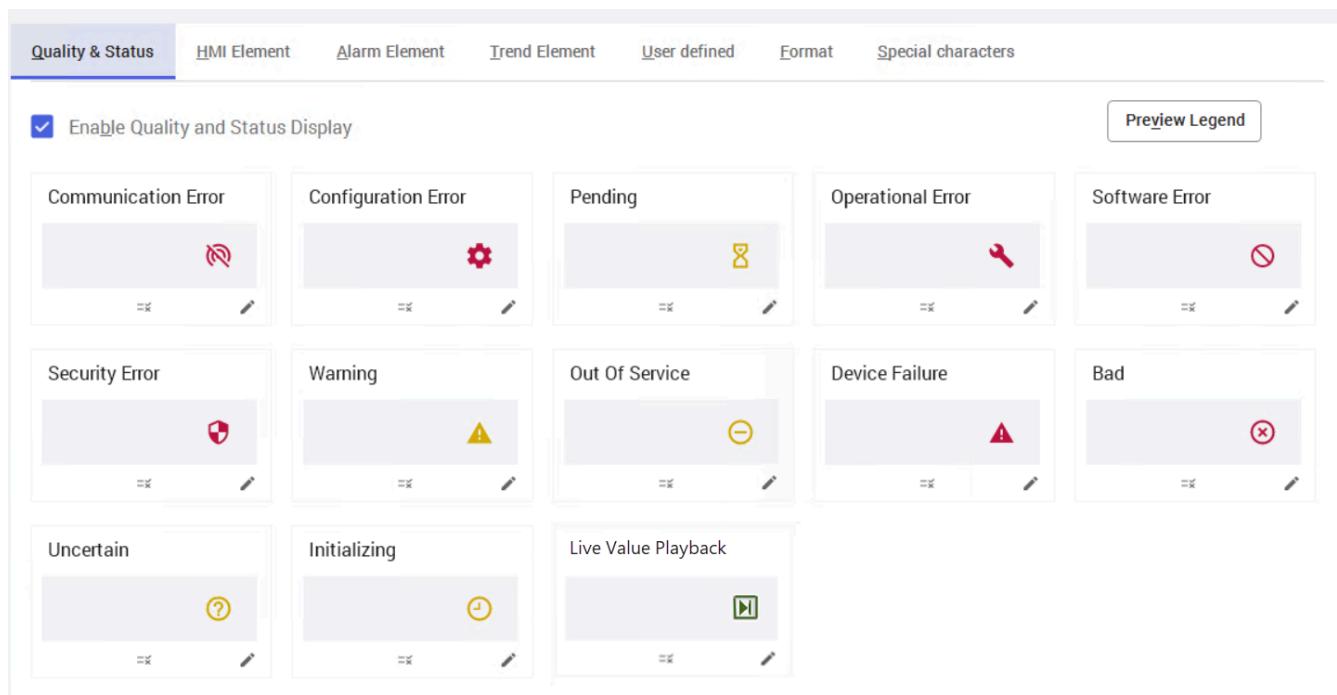
Use element styles to modify graphics

Element styles define one or more visual properties such as fill, line, text, blink, outline, and status properties of graphic elements. You can apply an element style to a graphic element to set preconfigured visual properties defined in that element style. Element styles establish consistent visual standards for graphics.

For more information about element styles, see [Apply element styles to elements](#).

Use quality and status indicators to provide runtime feedback

In the same way that element styles can be used to preconfigure consistent visual properties of graphic elements, Quality and Status (QS) indicators can be preconfigured to provide visual elements specific to quality and status states at runtime. Only the highest priority QS indicator is shown at runtime. Therefore, if two or more indicators are active, only the highest-priority indicator is shown. For example, if there are both operational and software errors, only the operational error is shown.



The indicators are listed in order of priority, with the highest priority indicator listed by row from left to right, and then top to bottom. You can configure text, fill, line, outline, blink, and associated display icons for the following QS indicators:

- Communication Error
- Configuration Error
- Pending
- Operational Error
- Software Error
- Security Error
- Warning
- Out of Service
- Device Failure
- Bad
- Uncertain
- Initiating
- Live Value Playback (used for the AVEVA OMI Historical Playback feature)

Use UI themes to set the appearance of WPF controls

UI Themes define the color settings that are applied to WPF controls used in ViewApps. In order to set the colors for a WPF control, the control must comply with the Material Design Spec. See the MSDN WPF Community Projects website for additional information.

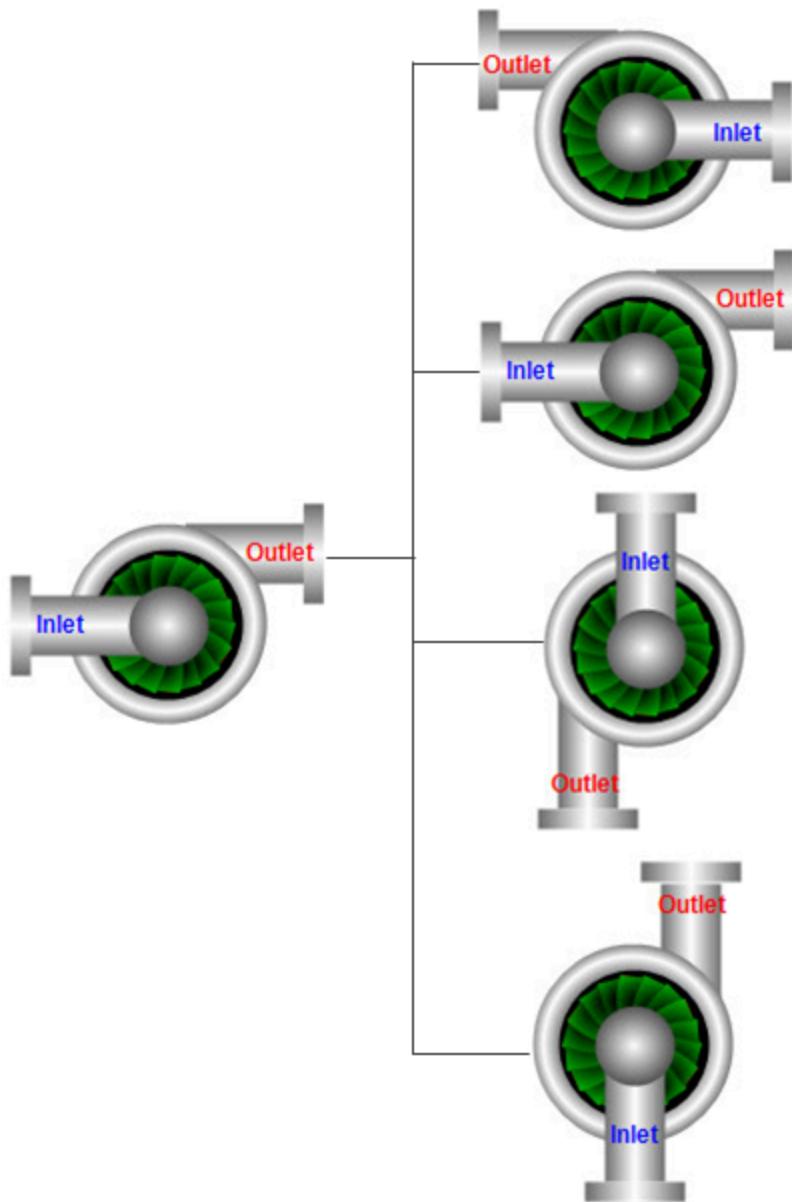
The UI Theme includes the following configurable parameters:

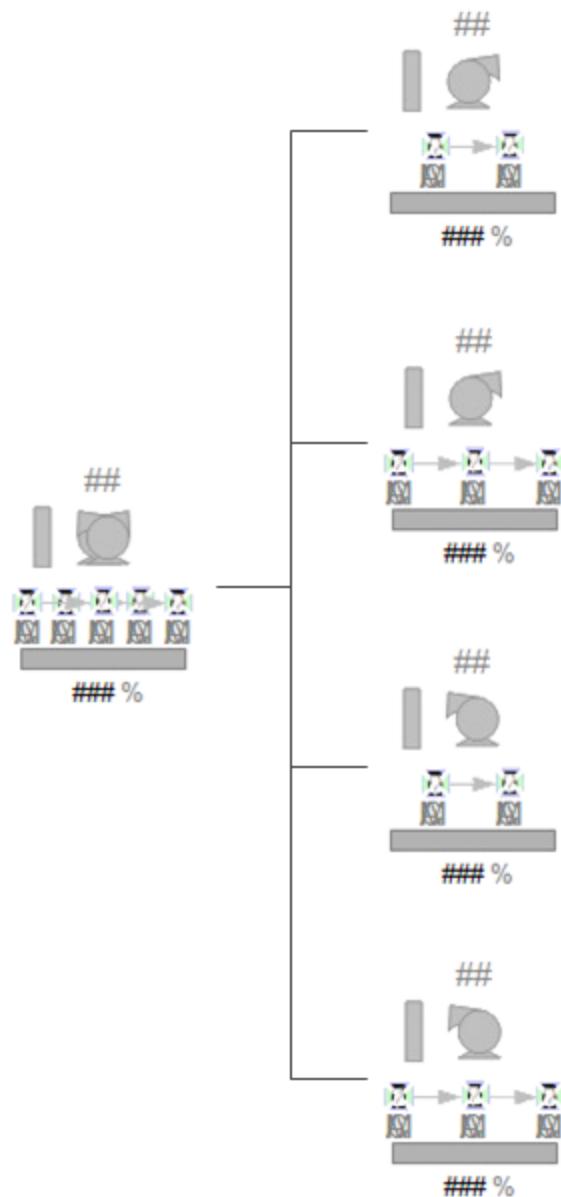
- **Theme** (light or dark): default – **light**
- **Background** color
 - Default **light** background color: grey, lightest setting
 - Default **dark** background color: blue-grey, second darkest setting
- **Primary** color – default **teal**
 - **Low.** This setting is only used by WPF controls that explicitly include the Primary Low property. The **low** setting is always darker than the **mid** setting. Default is the second darkest setting (second from left).
 - **Mid.** This setting is used by all WPF controls that use the Material Design Spec. Default is the middle setting.
 - **High.** This setting is only used by WPF controls that explicitly include the Primary High property. The **high** setting is always lighter than the **mid** setting. Default is the second lightest setting (second from right)
- **Accent** color – default **purple** (darkest setting)

Symbol Wizards

The Industrial Graphic Editor includes the Symbol Wizard Editor. Use this editor to create symbol wizards. A Symbol wizard contains multiple visual or functional configurations. A symbol wizard can be embedded into InTouch or OMI applications in the same way that you embed any Industrial Graphic.

Users select options from the symbol wizard to meet the requirements of an application. Incorporating multiple configurations in a single symbol wizard reduces the number of graphics needed to develop an AVEVA application.





The Symbol Wizard Editor can create symbol wizards from traditional Industrial Graphics and Situational Awareness Library graphics. Both types of graphics are located in the Visualization folder in separate folders. Symbol wizards are saved in the Visualization folder of the IDE and are not associated with any specific AVEVA object template or object instance. Except for the ability to select a specific graphic configuration, symbol wizards behave like standard Industrial Graphics.

Situational Awareness Library graphics provide an additional benefit of including defined properties and their associated attributes to more easily create configurations. Some Situational Awareness Library graphics include a Type property to assign a specific function for a configuration. For example, a meter graphic can be configured to represent a thermometer, a pressure meter, or flow meter by simply changing the attribute assigned to the Type property.

Typically, the process of creating and embedding a symbol wizard in an application requires the involvement of a Designer and a Consumer. A Designer creates symbol wizards using the Symbol Wizard Editor. A Consumer selects a configuration of a symbol wizard and embeds the instance of the graphic into an InTouch or OMI application.

Create Symbol Wizards

A Designer uses the Symbol Wizard Editor to define the various required graphic configurations based on a set of rules and graphic layers. A Designer defines a set of layers, which are used to group a set of graphic elements, custom properties, and named scripts. Graphic elements and other properties can be assigned to no layers or multiple layers. Graphic elements that are not assigned to any layer always appear in all configurations.

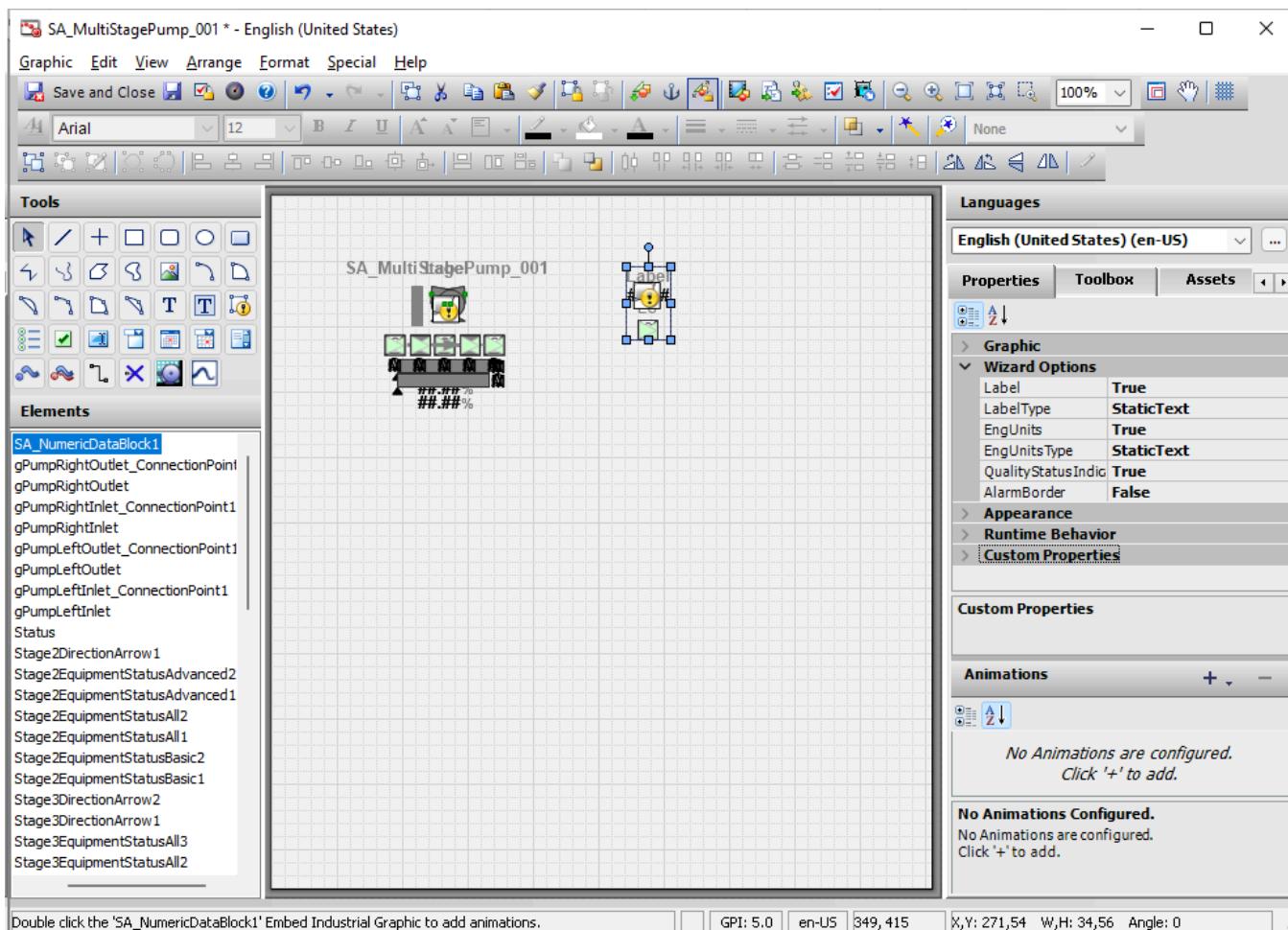
A Designer can create a rule for each layer that defines the conditions when the layer is included in a graphic configuration. Rules are assigned with Choice Groups, Choices, and Options. A Designer selects a configuration to be the graphic default that appears when the graphic is embedded in a managed InTouch application.

After creating all configurations, the Designer verifies each configuration of a graphic using the Symbol Wizard Preview. Designers set values in the **Wizard Options** view to verify that each configuration appears as designed based on the layer rules set for the graphic.

When a Symbol Wizard is finished, the Designer saves it to the Galaxy library so that it is available for use in InTouch and OMI applications.

Embed a Symbol Wizard into an application

Symbol wizards are stored in a Galaxy library just like standard Industrial Graphics. When a Consumer selects a symbol wizard and embeds it into a InTouch or OMI application, the default configuration of the Symbol Wizard is selected. The Consumer can change the graphic's configuration by changing the options from the **Wizard Options** section of the **Properties** view. Depending on the selected configuration, there can be additional configuration-related properties that can be selected by the consumer.



After selecting a graphic configuration and changing any properties, the Consumer saves the Symbol Wizard so that it can be embedded into a window from WindowMaker or into an OMI pane.

While the InTouch or OMI application is running, the Symbol Wizard appears as the configuration selected by the Consumer. A Symbol Wizard configuration cannot be changed during runtime.

For more information about how to embed Symbol Wizards into InTouch applications, see [Using Symbol Wizards in an Application](#) in the Industrial Graphic Editor help.

Rename a graphic

To rename a graphic:

1. On the **Attributes** page of the Object Editor, in the **Content** area, select the graphic to rename.
2. In the **Name** field, type the new name and press **Enter**. The new name is saved.

Delete a graphic

Caution: If you delete a graphic with embedded references, the links to the related objects will be broken. A "Symbol Not Found" message appears when you open objects whose embedded graphics have been deleted.

To delete a graphic

1. On the **Attributes** page of the Object Editor, in the **Content** area, select the graphic to delete.
2. Click the **Delete (-)** icon. The **Confirm Delete** dialog box appears. The left pane lists the graphic selected for deletion. The right pane shows all embedded references to the selected graphic.
3. Click **Delete**.

Rearrange objects

By default, a running ViewApp shows objects as navigation items that appear in alphabetic order by their assigned Galaxy names in the navigation model. An alphabetic order of navigation items may not be appropriate for operators. Objects can be rearranged to more closely resemble the functional organization of different components in a production process.

You can rearrange objects by dragging them to the desired location in the **Model** view of the System Platform IDE. After objects are moved, the **Assets** listings of objects in the ViewApp and Layout editors update to show the new arrangement.

Rearrange objects: drag and drop

You drag and drop an object with your mouse to move an object to a different location within the ViewApp navigation model. You can rearrange objects from the **Model** view of the System Platform IDE.

To rearrange objects from the System Platform IDE

1. Open the System Platform IDE and show the **Model** view.
2. Make sure that all objects shown in the **Model** view are undeployed.
3. Select the object to be moved within the **Model** view.
4. Drag the object to the desired location within the hierarchy of objects in the **Model** view.

A horizontal line shows the hierarchical level within the navigation model where the object will be placed.

The horizontal line does not appear when you attempt to make an invalid move.

About rearranged objects

Rearranged objects maintain their location within a list of objects for the following Galaxy operations:

- Export and import of a Galaxy
- Back up and restore a Galaxy
- GRLoad and GRDump operations

Also, objects can be programmatically moved to another location within a list of objects using the GRAccess ToolKit.

Rearranged Objects in the Export and Import of a Galaxy

When rearranged objects are exported to an aaPKG file, rearranged objects are retained when the aaPKG file is imported into another Galaxy. If the imported Galaxy includes rearranged objects, but the existing Galaxy does not, the object order of the imported Galaxy takes precedence.

Rearranged Objects in Backup and Restore of a Galaxy

When rearranged objects are backed up to a CAB file, rearranged objects are retained when the CAB file is restored to another Galaxy.

Rearranged Objects in a GRLoad and GRDump

When one or more objects are selected and exported through GRDump, then GRLoad can be used to import the contents of GRDump into another Galaxy. Objects retain their rearranged locations in the GRLoad import operation in the new Galaxy.

Rearrange an Object Programmatically using GAccess Toolkit

You can use the GAccess object model to write programs that automate the configuration of local and remote Galaxies. You can move an object in the Model view hierarchy using the SortOrder property that programmatically moves an object after a specified predecessor object in the list of objects shown in the **Model** view.

Class

IgObject

Syntax

[C#]

```
string SortOrder { set; get; }  
[Visual C++]  
[propget]  
HRESULT SortOrder(  
[out, retval] BSTR* thePredecessorName  
);  
[propput]  
HRESULT SortOrder(  
[in] BSTR newPredecessorName  
);
```

Parameters

thePredecessorName

The current predecessor of this object.

newPredecessorName

The new predecessor of this object. The object does not need to be checked out.

Remarks

SortOrder sets or returns the preceding object name in the navigation hierarchy. When the sort order is set, the tagname of the asset is displayed in the ViewApp navigation hierarchy just below the predecessor.

Assign alias name to an object

A running ViewApp can show objects as navigation items that appear by their assigned object names in the navigation model. Object names may not be intuitive and easily recognizable to operators. Objects can be assigned more understandable alias names.

Alias names are assigned from the **Model** view of the System Platform IDE. You can toggle the **Model** view to

show the objects by their tagname or alias names. In the example below, the Plant_Area area object has been assigned a more specific Building_A alias name.

The screenshot displays two side-by-side navigation panels. The left panel is titled 'Objects by Tagname' and the right panel is titled 'Objects by Alias Names'. Both panels have a search bar at the top and a tree view of plant assets under 'MainPlant' and 'Enterprise'. The 'Alias Names' panel shows the 'Building_A [Plant_Area]' alias for the 'Plant_Area' object, while the 'Tagname' panel shows the original 'Plant_Area' tagname. The 'Alias Names' panel also includes a status bar with tabs for 'Model - Alias', 'Deployment', and 'Derivation'.

Alias names can be shown from the ViewApp and Layout editors but cannot be modified. Also, if an alias was assigned to an object and the ViewApp or Layout editors were open, the editors must be closed and reopened to show the new alias name.

An alias name appears in the navigation model of the ViewApp during runtime or in preview mode.

The screenshot shows a 'Navigation' tree on the left and a detailed view of the 'Startup' item on the right. The 'Startup' item is highlighted in blue. The detailed view shows sub-items: HOME, AREA_001, AREA_002, ReactorComplex34, R35, and Historical. The 'Navigation' tree also includes items like AREA_001, AREA_002, ReactorComplex34, R35, Historical, ALARMS_CURRENT, and ALARMS_EVENTS_RECENT.

Alias names are associated with navigation attributes that have the word Title in their names. For example, the CurrentTitle attribute returns the alias name of the current navigation item in a running ViewApp. You can write scripts using alias names to change the current focus of a ViewApp. For more information about navigation

attributes that reference alias names, see [About ViewApp navigation](#).

Also, alias names can be hidden from the navigation model of a running ViewApp or in preview mode by placing an underscore (_) as the first character of the alias name. Hiding an alias name is useful for navigation items like a system node in the navigation model that does not provide any useful ViewApp information to an operator. For more information about configuring the visibility of an alias name, see [Hide assets in ViewApp navigation during runtime](#).

Steps to Assign or Remove an Alias Name

You can assign an alias name to a derived instance of an object that appears in the **Model** view of the System Platform IDE. The System Platform IDE includes a **Show Alias** option that enables you to toggle the **Model** view to show the original tagname and alias names of objects.

An alias name must adhere to the following rules:

- An alias name can be up to a maximum of 128 characters in length.
- An alias name can consist of alphanumeric characters and special keyboard characters (!,@,#, \$,%,&,*,(,),-,_,=,+).
- An alias name can start with any alphanumeric or special character.
- An alias name can contain blank spaces.
- Multiple objects can be assigned the same alias name in a Galaxy.

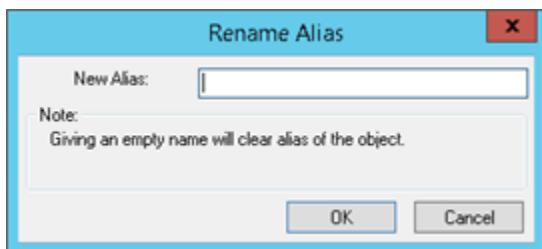
To assign an alias name to an object

1. Open the **Model** view of the System Platform IDE to show the list of derived instances.
2. Ensure the derived instance you want to assign an alias name is undeployed.
3. Select the object instance from the **Model** view.
4. Right-click on the object to show the shortcut menu and select the **Rename Alias** option.

You can also assign an alias name by other methods:

- **Keyboard shortcut:** Ctrl + Shift + F2
- **Menu bar option:** Select **Edit** from the menu bar and select the **Rename Alias** option.

The **Rename Alias** dialog box appears with a field to assign a new Alias name.



5. Enter an alias name that meets the format rules and select **OK**.

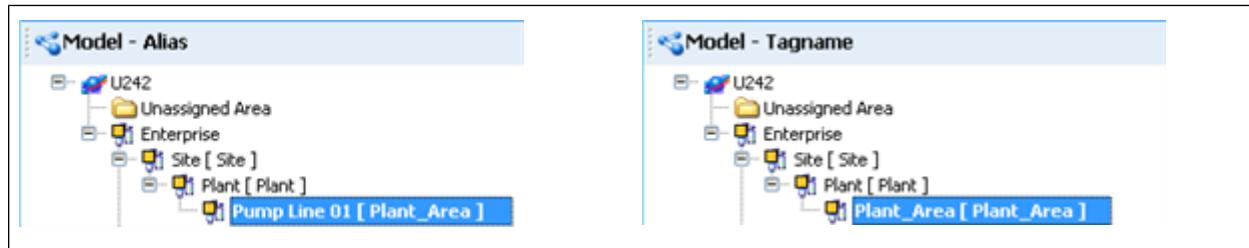
The object should appear in the **Model** view with its assigned alias name.

6. Verify the alias name has been assigned to the object.

- a. Right-click on the object with an assigned alias name and select the **Show Alias** option from the shortcut menu.

You can also verify an alias name by other methods:

- a. **Keyboard shortcut:** Ctrl + Shift + F3
- b. **Menu bar option:** Select the **Alias\Tag name** button  on the menu bar.
- b. Toggle the **Show Alias** option to show the alias name and the tagname of the object. The title bar of the **Model** view indicates whether you are looking at the alias or tagnames of the objects.



About alias names during operations

Objects retain their aliases when objects are exported, backed up, and loaded as part of a Galaxy operation. Also, object aliases can be programmatically set and returned using the GAccess ToolKit.

Alias retention in galaxy operations

Alias names assigned to Galaxy objects are retained for the following Galaxy operations:

- Export and import of a Galaxy
 - Back up and restore a Galaxy
 - Migrate a Galaxy
 - Exporting instances to and importing them from a CSV file
- Alias names in the export and import of a galaxy

When objects assigned alias names are exported to an aaPKG file, object Alias names are retained when the aaPKG file is imported into another Galaxy with the following exception:

AliasName is a reserved keyword in System Platform 2020 R2 and newer versions. If an import package contains previous versions of objects with name "AliasName", the import process skips the attribute and logs a warning message in the OCMC log. Attributes with name AliasName are not imported.

Alias names in backup and restore of a galaxy

When objects assigned alias names are backed up to a CAB file, object alias names are retained when the CAB file is restored to another Galaxy.

Migrate a galaxy

When objects assigned alias names are migrated, object Alias names are retained when migrated to another Galaxy with the following exception:

AliasName is a reserved keyword in System Platform 2020 R2 and newer versions. If a migration package contains previous versions of objects with name "AliasName", the migration process skips the attribute and logs a warning message in the OCMC log. Attributes with name AliasName are not migrated.

Alias names when exporting to and importing from a csv file

When one or more objects are selected and exported to a CSV (Comma-Separated Value) file, you can import the contents of the CSV file into another Galaxy. Objects retain their alias names when imported to the new Galaxy.

Assign an alias name programmatically using the GRAccess Toolkit

You can use the GAccess object model to write programs that automate the configuration of local and remote Galaxies. You can rename object aliases using the Alias property that programmatically gets or sets the alias name of an object.

Class

IgObject

Syntax

[C#]

```
string AliasName { set; get; };
```

[Visual C++]

[propget]

```
HRESULT Alias(
```

```
 [out, retval] BSTR* theAlias
```

```
);
```

[propput]

```
HRESULT Alias(
```

```
 [in] BSTR newAlias
```

```
);
```

Parameters

theAlias

The current alias of this object.

newAlias

The new alias of this object, or blank if the object does not have an alias. The object does not need to be checked out. The alias rename operation is immediate.

Object Wizards

Adding an Object Wizard to a template provides a streamlined method that lets users quickly and easily derive and configure assets and their associated graphics from the template.

Create an Object Wizard 	About Object Wizards	Object wizard components	Conditional visibility expressions for choice groups, choices, and options
Add Content to an Object Wizard 	Add content for use with an object wizard	Attribute and graphic overrides	Object wizard trimming

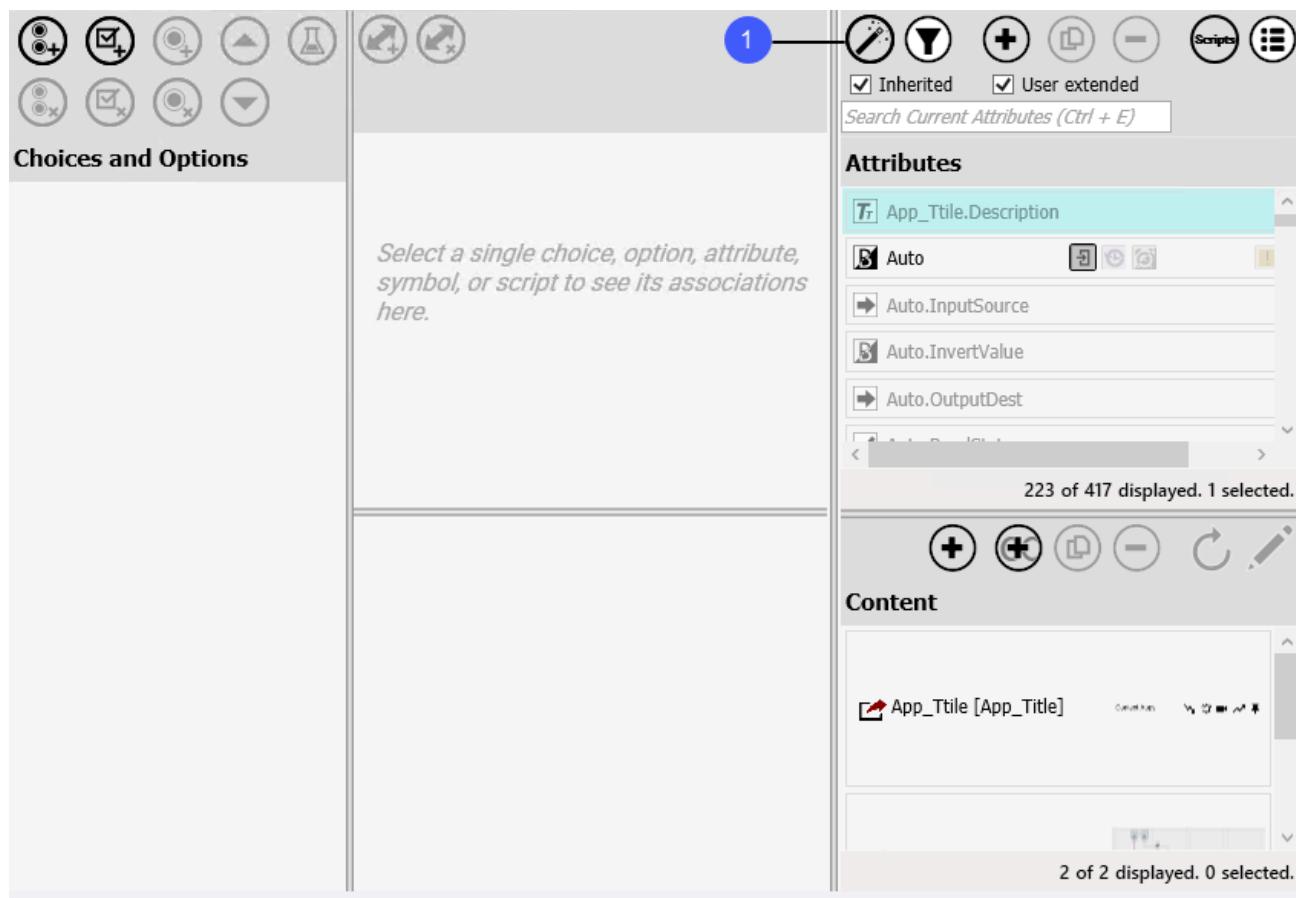
Derive a Template	Object wizard derivation	Object wizard test mode	Propagation of object wizard changes to derived objects
			
Configure an Instance	Configure instances	Create a new instance graphically	
			

About Object Wizards

Object Wizards consist of a series of user-selectable choices and options that are used to customize a deployable instance. Each choice and option may have one or more attributes, graphics, other content types and/or scripts associated with it.

An Object Wizard can be added to any derived template and provides a simplified user interface for configuring instances (assets) from the template. To access the Object Wizard editor:

1. Open a template in the **Object Editor**.
2. If the template does not open to the **Attributes** page by default, select it.
3. If necessary, select the **Configure Wizard Options** icon (1) to open the Object Wizard editor. The **Attributes** tab divides into multiple panes that are used for building the Object Wizard.



Object Wizards can be used either within the IDE, or from the **Configure New Asset** editor that also lets you configure an associated graphic as you configure the object.

Note: The **Configure New Asset** editor can only be used if 1) the template contains a graphic, and 2) your security setting allows you to edit graphics. It cannot be used with other content types, such as layouts and external content.

A single template with an Object Wizard can replace a number of derived templates for configuring a variety of similar instances. You can add an Object Wizard to any derived template.

Depending on their level of permissions, users can:

- Create and configure instances from within the IDE, and add them to the Galaxy, or modify existing instances. A user simply opens an instance and answers a series of prompts or questions. See [Configure instances](#) for additional information.
- Use the **Configure New Asset** editor to create and configure instances and a representative graphic, simply by dragging an associated (linked or embedded) graphic into another graphic, and then answering a series of prompts or questions. See [Create a new instance graphically](#) for additional information.

Note: The drag and drop method is only valid for graphics, and not for other linked content types such as layouts and external content.

- For more information about linking to a graphic, see [Link to shared content in the visualization folder](#).
- For more information about embedding a graphic, see [Add a new object-owned symbol](#).

Important! InTouch HMI does not support directly adding assets to an InTouch HMI window if the asset is derived from an object wizard with a linked Symbol Wizard, such as the graphics in the Situational

Awareness Library. **With InTouch HMI, the Symbol Wizard must be contained in an overview graphic** when deriving assets from an object wizard. In contrast, AVEVA OMI does support the the use of linked Symbol Wizards when adding assets to AVEVA OMI layouts from an object wizard.

Object Wizard features

An Object Wizard contains choices and options that guide users through the process of configuring instances. You can replace many derived templates by using a single template with an Object Wizard, since the Object Wizard can contain a wide variety of configuration possibilities. Attributes and features contained in the Object Wizard that are not needed in a particular instance are removed and are not a part of the runtime object. Finally, and perhaps most importantly, an Object Wizard is easy to use and can reduce user training and system maintenance costs.

Object Wizards provide the following benefits:

- Simplified workflow for configuring instances
- Automatic propagation of changes, without having to lock templates
- Unnecessary attributes can be trimmed from instances to reduce the footprint of deployed objects
- Unused graphics can be hidden from users to reduce clutter
- Reduced system maintenance
- Reduced training requirements for plant personnel

Create a basic Object Wizard

An Object Wizard can include Choice Groups, each with two or more Choices, and/or Options. A Choice Group is a radio button list that lets the user select one choice. An Option contains a checkbox that a user can select or clear.

To add an Object Wizard to a template

1. In the IDE, open a derived template in the Object Editor.
2. If the **Attributes** page is not already selected, select it now. See [Create derived templates](#).

Note: Only templates derived from the \$UserDefined base template open in the Object Editor with the **Attributes** page selected. The Object Editor always opens the first page of the object. For example, the Object Editor opens the **General** page of the \$AnalogDevice template.

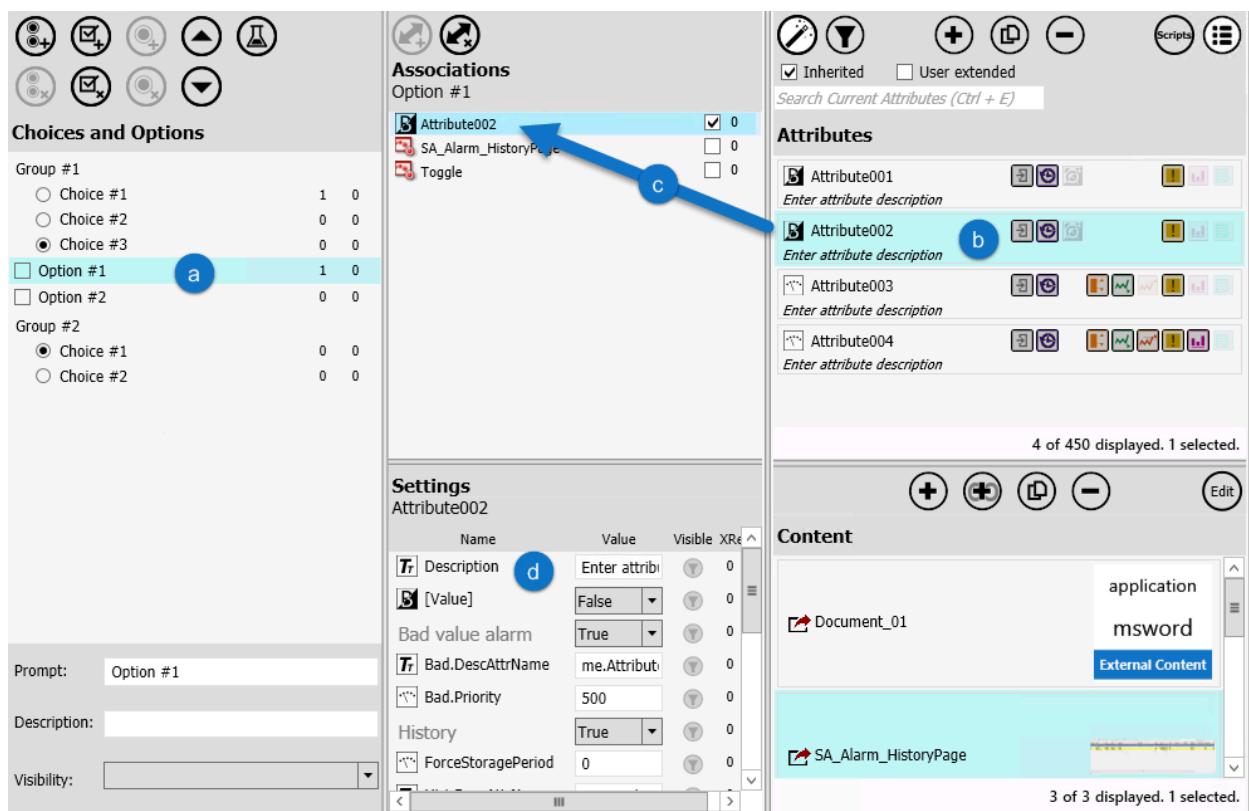
3. Configure the template by adding:
 - Attributes
 - Graphics or other content such as links to documents and videos
 - Scripts

See [Add content for use with an object wizard](#) for additional information.
4. If the Object Wizard editor is not already open, select the **Configure Wizard Options** button.
See [Object wizard GUI panes](#) for additional information.
5. Add a Choice Group by selecting the **Add Choice Group (1)** icon. With the Group name selected, change the name of the Group shown in the **Prompt (5)** text field. See [Add a Choice Group](#) for more information.



6. To add more Choices to a Choice Group, select the Choice Group and then select the **Add Choice (2)** icon. To change the name of a Choice, select the Choice and enter the new name in the **Prompt** field. See Add a Choice to a Choice Group for more information.
7. Add Options by selecting the **Add Option (3)** icon. Change Option names the same way that you change Choice Group and Choice names. See Add an Option for more information.
8. Use the **Up and Down arrow (4)** icons to change the order of Choice Groups, Choices (within a Choice Group), or Options.
9. To associate an attribute or content with a Choice or Option:
 - a. Select the Choice or Option.
 - b. Select the attribute or content (graphic or other type of content) you want to associate with the Choice or Option.
 - c. Select the **Add Association** icon (or drag the attribute or content to the **Associations** pane).
 - d. Settings for the selected attribute or content are shown in the **Settings** pane, once the association is made. Configure visibility and value overrides, as needed.

Note: No settings are shown for graphics that do not contain custom properties or a Symbol Wizard.



Object wizard configuration best practices

An Object Wizard can be added to any derived template. However, to maximize the benefits of using Object Wizards, add your Object Wizard to the template as close to the top level of the derivation hierarchy as possible. To build an Object Wizard you must:

- Add and configure attributes, graphics, links to external content, and scripts before you build the Object Wizard.
- An Object Wizard requires at least one Choice Group or Option. See [Object wizard components](#) for additional information.
- Configure Choices and Options by associating attributes, graphics/content, and scripts with Choices and Options. We recommend that you limit the number of items you associate with each Choice or Option to avoid configuration mismatches.
- When configuring Choices and Options, keep the workflow as simple and direct as possible. Remember that users will be working sequentially through the Object Wizard. If you enable a feature at one level of the Wizard hierarchy, you cannot disable the same feature at a subsequent level.

For example, if a graphic has Custom Property "X" enabled and is associated with a Choice, do not associate the same graphic with "X" disabled to an Option further down in the hierarchy. Instead, leave "X" disabled and override the setting when deriving an instance.

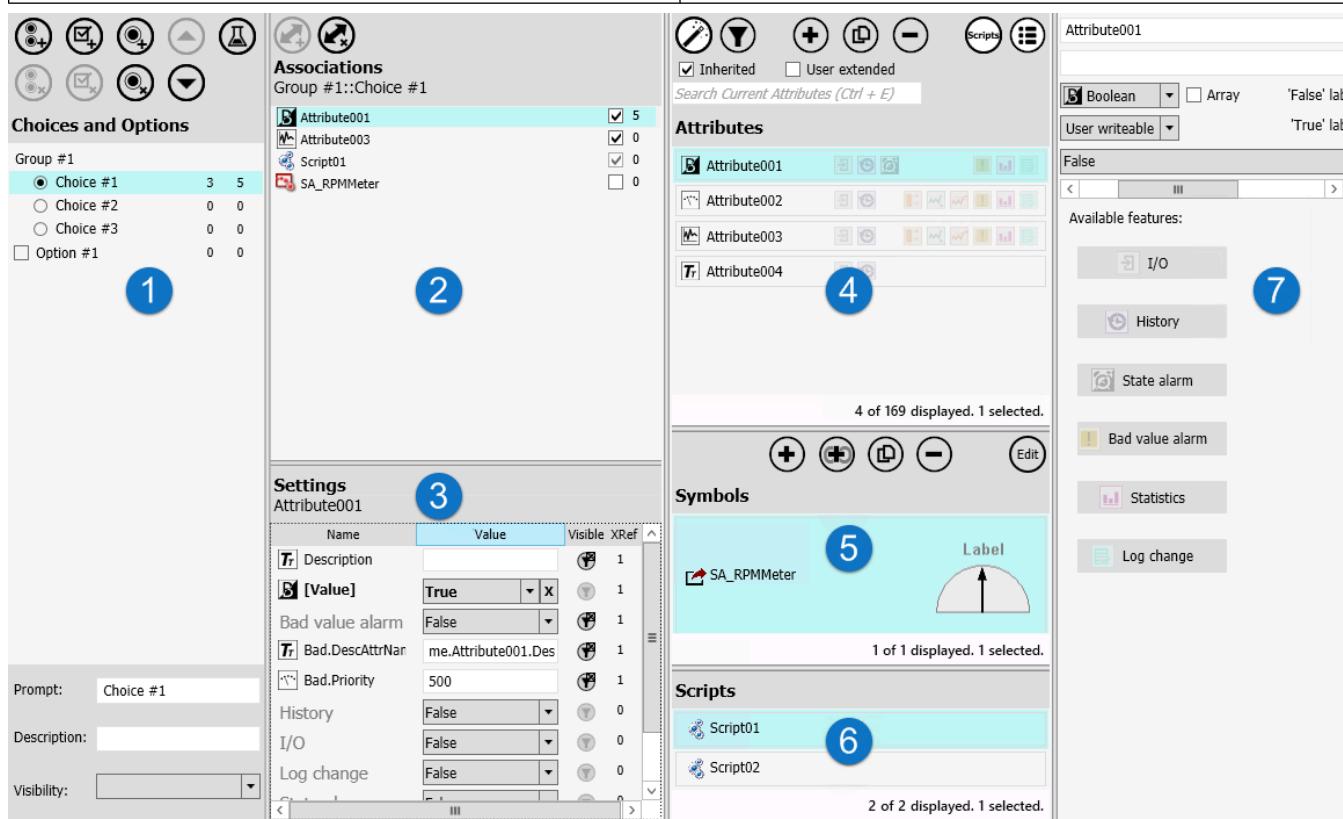
- Configure the [Conditional visibility expressions for choice groups, choices, and options](#) of Choices and Options in derived instances.
- Configure settings for the associated attributes and graphics. Scripts and external content do not have settings that can be configured.

- Select whether each attribute, graphic, and other content can be [Object wizard trimming](#) from derived instances. Scripts are always configured as trimmable.

Object wizard GUI panes

The user interface for configuring Object Wizards is accessed through the **Attributes** page within the Object Editor of the Integrated Development Environment (IDE).

	Select the Configure Wizard Options icon to open the Object Wizard user interface. The Attributes page divides into multiple panes for use in creating and editing an Object Wizard. See Object wizard user interface details for additional information.
	Select the Scripts icon (located at the top right of the Attributes pane) to open the Scripts pane.



Ref	Name	Description
1	Choices and Options pane	Add, rename, reorder, and delete Choice Groups, Choices, and Options that comprise the Object Wizard in the Choices and Options pane (1) . You can also set the logic for displaying Choices and Options.

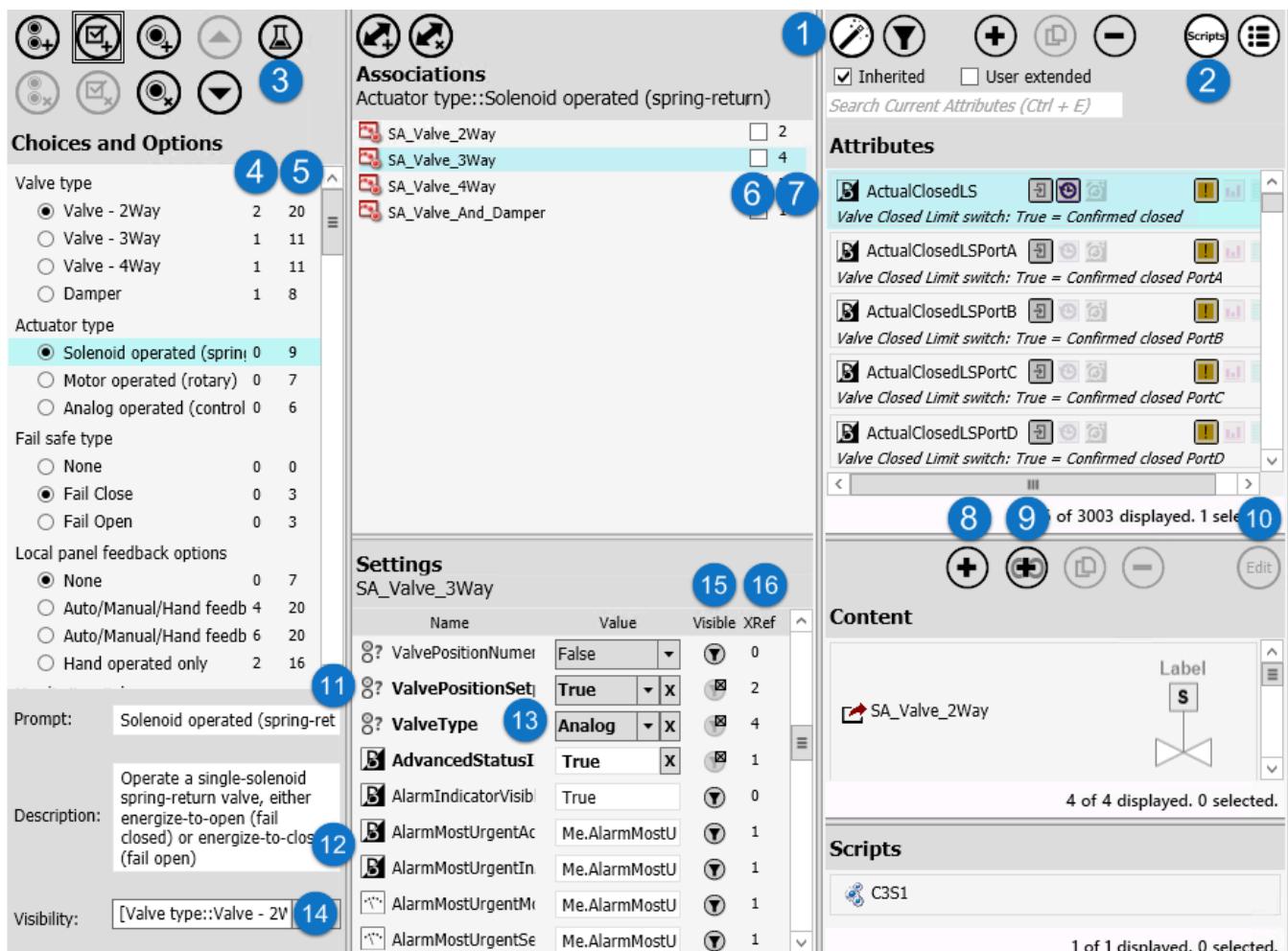
2	Associations pane	<p>Drag and drop attributes, content, and scripts to the Associations pane (2) to associate them with a choice or option; you can use the Add button at the top of the pane instead of dragging and dropping. You can select multiple attributes, graphics, or scripts at once. You will also enable and disable trimming of associated attributes and graphics in this pane. See Object wizard trimming for additional information.</p> <p>You can view associations for individual attributes, graphics, and scripts in this pane by selecting an attribute, graphic, or script. Every association of the item is displayed. Select an association to view the item's configuration.</p>
3	Settings pane	<p>Configure attribute and graphic overrides in the Settings pane (3). Overrides are set individually for each choice or option with which the attribute or graphic is associated. Default settings for attributes are set in the Details pane (7).</p> <p>You can resize columns by selecting a column header (in this case, Value) and moving its edges to widen or narrow it.</p>
4	Attributes pane	<p>The Attributes pane (4) lists the attributes contained in the template. You can create new attributes in this area. To associate attributes with the Object Wizard, highlight a Choice or Option, and then select and drag one or more user-created attributes from the Attributes pane to the Associations pane (2). See Associate Attributes with Choices and Options for additional information.</p> <p>Default configuration settings for a selected attribute are displayed to</p>

		<p>the right of this pane, in the Details pane (7). See About the attributes page for additional information.</p> <p>Configured settings for the selected attribute that apply to a selected Object Wizard Choice or Option are shown in the Settings pane (3). You can also enable attribute features (historization, I/O, alarms, etc. in the Attributes pane. Multi-select can be used to enable features for attributes, as well as to associate attributes. To use multi-select for features, select the attributes in the Attributes pane and enable the feature in the Details pane.</p>
5	Content pane	<p>The Content (5) pane lists graphics and external content items contained in, or linked to, the object. To associate graphics with the Object Wizard, highlight a Choice or Option, and then select and drag one or more items from the Content pane to the Associations pane (2). You can add a new graphic to the object from this pane, or link to a graphic in the Visualization folder. See Associate content with a choice or option for additional information.</p>
6	Scripts pane	<p>To display the Scripts pane (6), select the Scripts button. The Scripts pane lists scripts contained in the object. To associate a script with the Object Wizard, highlight a Choice or Option, and then select and drag one or more scripts from the Scripts pane to the Associations pane (2). See Associate content with a choice or option for additional information.</p> <p>To add a script to the object, select the Scripts tab at the top left of the Object Editor.</p>
7	Details pane (Attribute Editor)	<p>The Details pane (7) displays details of a selected attribute,</p>

		graphic, or other content. This pane lets you configure individual attributes, but is not usually used while configuring an Object Wizard. See Configure Objects for additional information.
--	--	--

Object wizard user interface details

The Object Wizard user interface provides a number of visual elements to help you as you build an Object Wizard.



Ref	Icon	Name	Description
1		Configure Wizard Options button	Opens/closes the Object Wizard editor (Choices and Options , Associations , Settings , Attributes , and Content)

			panes.
2		Scripts button	Opens/closes the Scripts pane so you can associate scripts with Object Wizard Choices and Options.
3		Test mode button	Changes the Object Wizard editor to test mode. This lets you verify the behavior of the Object Wizard. See Object wizard test mode for additional information.
4	<#>	Number of associations	Number of trimming-enabled attributes, graphics/content, and scripts associated with a Choice or Option. This can help in the maintenance of complex Object Wizards by letting you quickly identify where trimming has been enabled.
5	<#>	Total configured settings	Total number of configured settings for all attributes and graphics that are associated with each Choice or Option . This can help in the maintenance of complex Object Wizards by letting you quickly identify where overrides have been configured.
6	<input type="checkbox"/> <input checked="" type="checkbox"/>	Trimming checkbox	Place a check in the checkbox to enable trimming for an attribute or graphic. Trimming is always enable for scripts, and is enabled by default for attributes. See Object wizard trimming for additional information.

7	<#>	Number of configured settings	Number of configured settings for each attribute or graphic that is associated with the selected Object Wizard Choice or Option. Scripts and external content items do not have configurable settings.
8		Add graphic button	Add (create new) graphic button. The new graphic is owned by the object. See Add content for use with an object wizard for additional information.
9		Link content button	Link to an existing graphic or external content item in the Visualization folder. The graphic/content can be shared with other objects. See Add content for use with an object wizard for additional information.
10		Edit graphic button	Opens the selected graphic in the Graphic Editor .
11		Symbol Wizard icon	Symbol Wizard settings are indicated by a unique icon. See Attribute and Graphic Icons , below.
12		Attribute / Custom Property icons	Attribute and graphic custom property icons are displayed for Boolean, integer, float, double, string, time, elapsed time, and internationalized string. An icon is also displayed for Symbol Wizard settings. See Attribute and Graphic Icons , below.

13		Setting Override indicator	Attribute, custom property, and Symbol Wizard settings that contain an override are indicated by bold text and an X next to the value. To remove the override, select and clear the X. See Override Icons , below.
14	<text field>	Visibility Expression	Sets the conditional visibility of Object Wizard elements. See Conditional visibility expressions for choice groups, choices, and options for additional information.
15	Visibility indicator icons    	Visible Visible - override Hidden Hidden - override	When the icon is unmuted, the setting is visible to the user when configuring instances. When muted, the setting is hidden. See Attribute and graphic overrides and Configurable settings for attribute and graphic values for additional information. A small x in the top right corner of the icon indicates that the visibility setting is an override, and may differ from the setting in the parent template. Note: In derived templates, visibility indicators replace visibility expressions for Object Wizard choices and options. See Object wizard derivation for additional information.
16		XRef	Total number of configured overrides for the selected graphic or attribute. If the attribute or graphic is associated

			with multiple Object Wizard Choices or Options, hover over the XRef number to see which Choices and Options have overrides configured.
--	--	--	--

Attribute and Graphic Icons

Icon	Description / Data Type	Attribute	Graphic
	Symbol Wizard Choice or Option		X
	Boolean data type	X	X
	Integer data type	X	X
	Float data type	X	X
	Double data type	X	X
	String data type	X	X
	Time data type	X	X
	Elapsed time data type	X	X
	Internationalized string data type	X	X
[No icon]	Large type with no icon indicates that the item is an attribute feature. Attribute features are selected in the Details pane (see Object wizard GUI panes) and include I/O, History, State Alarms, Bad Value Alarms, Hi/Lo Alarms, etc.	X	

Note: For custom properties, string, time, and elapsed time data types can be toggled between reference/expression and string literal. This does not apply to attributes.

	Reference/expression
	String literal

Override Indicators

Indicator	Description
	This is the default setting (no override). If the setting is changed in a parent template, this value will also change.
	Setting has been overridden. Changes made in a parent template will not cause the this value to change. The value is shown in bold and an X appears next to the value.
	No override: Setting is visible and is not overridden. If the visibility of this item is changed in a parent template, the visibility of this setting will also change.
	No override: Setting is hidden and is not overridden. If the visibility of this item is changed in a parent template, the visibility of this setting will also change.
	Override has been applied: The visibility setting ("visible") is locked in this template. If this item is hidden in a parent template, this setting will remain visible (setting will not change).
	Override has been applied: The visibility setting ("hidden") is locked in this template. If the this item is made visible in a parent template, this setting will remain hidden (setting will not change).

Object wizard components

An Object Wizard consists of Choice Groups and Options. Choice Groups contain at least two Choices, and Object Wizard users must select one Choice from every Choice Group.

Once you create Choice Groups, Choices and Options, you can associate attributes, graphics, layouts, external content, and scripts with the wizard to create an Object Wizard workflow. Typically, you create and configure the required attributes, scripts and other content prior to adding an Object Wizard to the template. See [Configure Objects](#) for additional information.

Choices are chosen by selecting the radio button next to the Choice. Only one Choice can be selected from a Choice Group. When users select a Choice, they are also selecting any elements (attributes, etc.) associated with

the Choice, even though those elements may not be visible to them.

Options are selected by placing a check in the Option's checkbox. Options can be either true (checked) or false (unchecked). Attributes, graphics, other content, and scripts can only be associated with the true state. No action is implemented when an Option is set to false.

You can rename Choice Groups, Choices, and Options as needed and add a description (the description is visible to users as a tooltip). Choice Groups and Options must be uniquely named; a Choice Group and an Option cannot have the same name. Choices within a single Choice Group must also be uniquely named.

Add content for use with an object wizard

You add attributes, graphics, layouts, scripts, and other content to a template object with an Object Wizard the same way that you add content to any other object. After adding the content, you can associate individual attributes and other content to individual Choices and Options. This association allows content that is not needed to be trimmed when you derive instances from the template. This helps to reduce the footprint of runtime objects. See [Object wizard trimming](#) for additional information.

Note: Only user-created attributes can be associated with Object Wizard Choices and Options.

When you associate content with Choices and Options, the propagation of these items to derived instances is determined by:

- User selection of Choices and Options during configuration.
- Trimming configuration for attributes and graphics, as they apply to the selected Choices and Options.

Content is trimmed from a derived instance when the instance is saved. Note that all items are propagated to derived templates, regardless of configuration and trimming settings, since unlike instances, templates are not runtime objects.

Both attributes and graphics can have unique settings for each association. The visibility of these settings, which determines if the setting can be overridden by a user deriving instances from the template, can also be configured.

For detailed information about custom properties and Symbol Wizards, see [Using custom properties with Application Server](#) and [Symbol Wizard Editor](#).

Content can either be added or linked to an object.



When you select the **Add Content** button, a new graphic is created that is owned by the object. Any changes made to the graphic are only applicable to the owning object. Changes to the graphic in the parent object will cascade to child objects. The **Add Content** button does not let you create an object-owned external content item.



When you select the **Link Content** button, the Galaxy Browser opens the Visualization folder. Selecting a graphic, layout or external content object from the Visualization folder creates a link between the object and the content, but the object does not own the content. Note that relative references will resolve

correctly for the linked graphic or layout. Changes to the content will cascade to every object with which the item is linked, regardless of the hierarchical relationship of the objects.

For information on adding scripts to a template, see [Write and edit scripts](#).

The settings configured in the **Attribute Editor**, **Graphic Editor**, and **Layout Editor** provide the default settings for each attribute and content item. You can override the default values for attributes and graphic custom properties in the **Settings pane**. Note that you can set different values for each Choice or Option with which the content is associated.

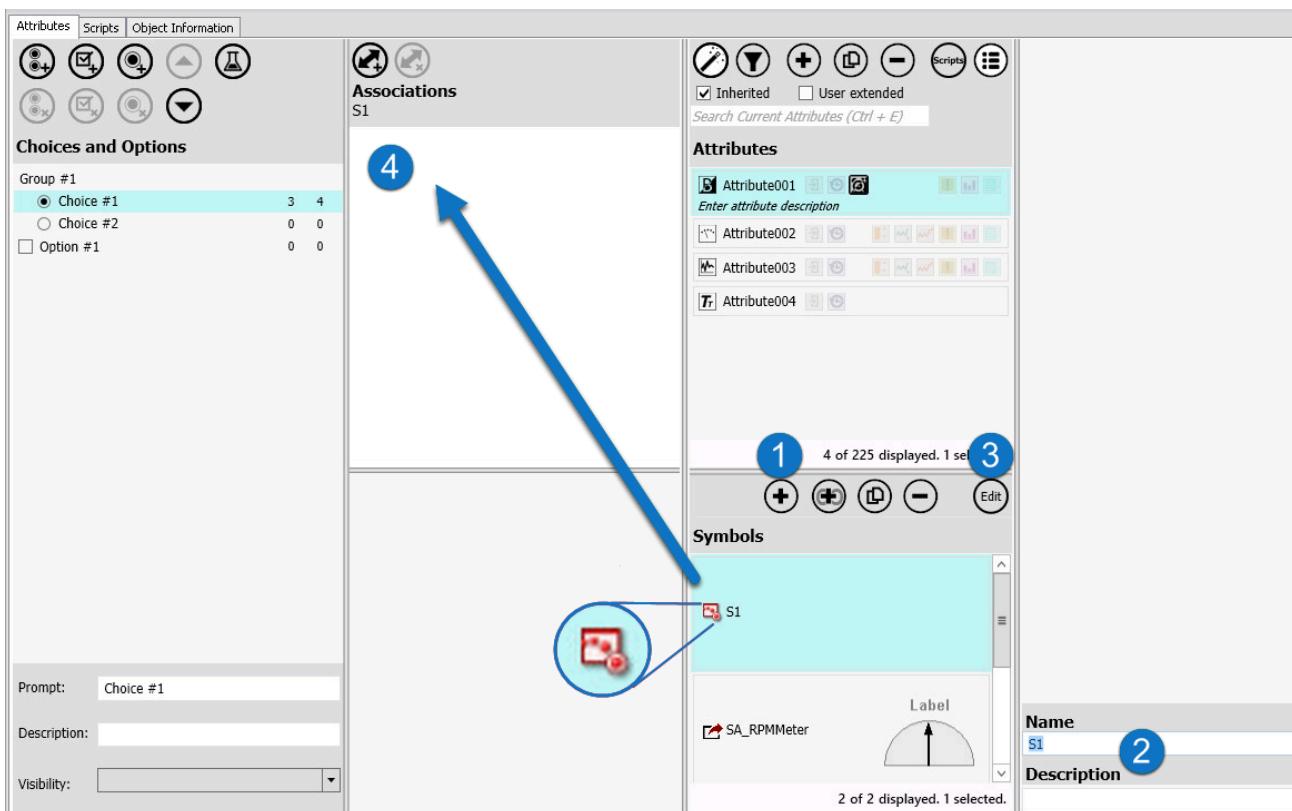
In addition, you can allow the user to override the default values in derived instances. See [About the attributes page](#) and [Add attributes to an object](#) for information about adding and configuring attributes.

Add a new object-owned symbol

To add a new object-owned graphic

1. In the **Content** pane, select the **Add Content (1)** button. The graphic is added with a default name (based on the previously added graphic, if any).
2. In the **Details (2)** pane, edit the name of the graphic and add a description.
3. In the **Content** pane, select the **Edit (3)** button to open the **Graphic Editor**. See *Creating and Managing Industrial Graphics User Guide* for information about using the editor to create graphics. When you are done editing the graphic, save and close the Graphic Editor.
4. Associate the graphic with one or more Choices or Options, as described in [Associate content with a choice or option](#).

The icon next to the graphic name in the **Content** pane indicates that the graphic is owned by the object.



Link to shared content in the visualization folder

You can link an object to content in the Visualization folder. Content can be a graphic, a layout or an external content object.

To link to an existing content item in the Visualization folder:

1. In the **Content** pane, select the **Link Content** button

The **Galaxy Browser** opens. Select an existing graphic, layout, or external content object from the **Visualization folder**.

2. In the **Details** pane, edit the default name of the content as needed. The edited name applies only to the object to which it is linked. The name of the content item remains unchanged in the Visualization folder. You can also add a description .

Note: If you choose to link to the same content item more than once, it is automatically renamed within the object with an incremental number appended to the original name. Its name in the Visualization folder is not changed, and is displayed in brackets in the following format: OriginalName1 [OriginalName]. In most cases, linking to the same item more than once is not an efficient use of an Object Wizard.

3. **Optional:** To edit the linked content, select the **Edit** button to open the **Graphic Editor** (for graphics), the **Layout Editor** (for layouts), or the **External Content Editor** (for external content items). Any change you make to the linked content item changes the configuration of the content, and the change affects any object that links to the item.

Note: Editing protected graphics, such as graphics contained in the **Situational Awareness Library**, is not permitted. If you edit an item in the **Visualization folder**, the changes will affect every object that links to the

item.

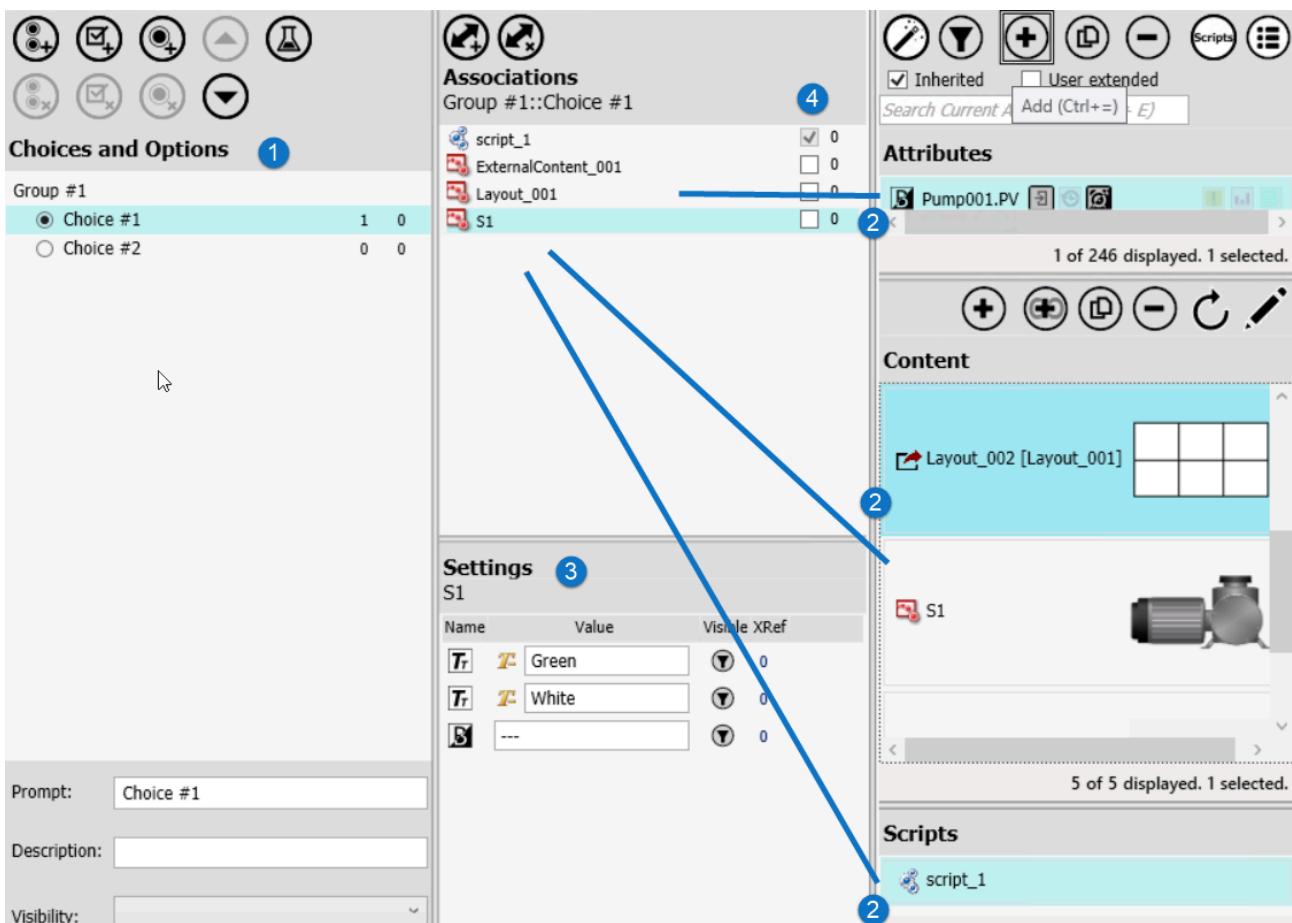
Associate content with a choice or option

In most cases, you will have added content to the object prior to adding the Object Wizard. Once you have begun configuring the Object Wizard, you can associate the following types of content with Object Wizard Choices and Options:

- Attributes: shown in the **Attributes** pane
- Graphics: shown in the **Content** pane
- External Content Objects: shown in the **Content** pane
- .NET Controls: shown in the **Content** pane
- Scripts: shown in the **Scripts** pane

To associate content with a Choice or Option:

1. Highlight a Choice or Option in the **Choices and Options** pane.
2. Select and drag one or more item from the **Attributes**, **Content**, or **Scripts** pane to the **Associations** pane. Use the **Shift** or **Ctrl** key to select multiple items in a single pane.



Note: Individual items can be associated with more than one Choice or Option. However, a Choice or Option does not have to have any items (attributes, etc.) associated with it. For example, a Choice labeled "No" or

"None" typically would not have any content associated with it.

Note: A content item (attribute, graphic, layout, etc.) does not have to be associated with an Object Wizard Choice or Option. Unassociated items are simply a part of the object and are propagated to all derived objects.

3. If applicable, highlight an attribute or graphic and enter overrides for the default values. The overrides affect only the Choice or Option for which they are entered. For example, a graphic can have one label for one Choice, and a different label for another. The override applies only to the associated Choice or Option. See [Attribute and graphic overrides](#) for additional information.
-

Note: You can resize columns in the **Settings** pane by selecting the column header and moving its edges to widen or narrow the column.

4. If applicable, override the default trimming setting for each associated item.
 - Trimming is disabled by default for graphics, layouts, and external content (checkbox is unchecked).
 - Trimming is enabled by default for attributes (checkbox is checked).
 - Trimming is enabled for scripts and cannot be disabled.
 - Note that the trimming defaults for graphics and attributes are different. See [Object wizard trimming](#) for additional information.

Disassociate content from a choice or option

To remove the association of a content item (attribute, graphic, etc.) with a Choice or Option:

1. Locate the Option, or the Choice within its Choice Group and select it.
 2. In the **Associations** pane, select the content item you want to remove from the Choice or Option.
 3. Select the **Remove Association** button. The item is no longer associated with the Choice or Option.
-

Note: Removing an association does not remove the content item. The content remains a part of the template, and if the content is associated with other Choices or Options, those associations remain intact. However, removing an association may affect the trimming of content within instances (see [Object wizard trimming](#) for additional information). If the content is completely unassociated with any Object Wizard Choice or Option, it is always included in derived instances.

To completely remove an attribute from a template, see [Delete content from an object](#).

Multiple content item associations

Attributes, graphics, and other content can be associated with more than a single Choice and/or Option. Each association can use different override values (settings) for a particular attribute or graphic. To change a setting for a particular association, enter overrides of the defaults in the settings pane.

You can use different overrides for each Choice or Option with which an attribute or graphic is associated. For example, an integer attribute could have a default value of 10, but could have an override of 5 for one Choice and an override of 20 for a different Choice.

Map attribute and graphic configurations to a choice or option

You can associate a particular attribute, layout, or graphic configuration with an Object Wizard selection (Choice

or Option). In this way, pre-configured attributes and content can be added to the derived instance simply by selecting an Object Wizard Choice or Option. Different configurations can be applied to different Choices and Options.

Mapping these configurations to Object Wizard Choices and Options may require additional planning and testing. This is especially important if multiple configurations of the attribute or content are being mapped. Mismatches between configurations can result in different behaviors in the derived instance than intended. Use **Test mode** to catch potential mismatches.

Attribute and graphic overrides

Override values for an attribute or graphic (with a Symbol Wizard or custom properties) can be configured for each associated Choice or Option. When you associate an attribute or graphic, the default values are listed in the **Settings** pane. To configure values that apply to the selected Choice or Option, enter the values in the **Settings** pane. If the attribute or graphic is associated with multiple Choices or Options, different values can be used for each one. Values for the Choices and Options are saved when you save the object. If the setting is marked as visible, users can override your setting (or override the default, if you have not configured a setting) as they derive and configure instances from the Object Wizard.

Graphics with overrides that affect their shape, for example, certain Situational Awareness Library graphics, are rendered with the applicable shape-changes as defined by the overrides.

Note: If you override a Symbol Wizard setting, the only way to make the setting visible again is to remove the override. Attributes and custom properties with overrides can be made visible again by selecting the visibility checkbox.

A configured setting is defined as any change from the default configuration of the attribute or graphic. This includes changing a value or visibility. For example, you might change the default Boolean value from true to false, or you might change an integer value from 5 to 10, or change engineering units from liters to cc. Changing the visibility of a setting also counts as a configured setting.

- The default visibility setting for attribute values is hidden (Visibility icon is muted (grayed-out) in the **Settings** pane).
- The default visibility setting for Symbol Wizard and custom property values is visible (Visibility icon is unmuted in the **Settings** pane).
- Visibility overrides are indicated by a small X in the upper right quadrant of the visibility icon.

The total number of configured settings for each individual attribute and graphic is displayed in the **Associations** pane. The total number of configured settings for ALL attributes and graphics associated with an individual Choice or Option is displayed in the **Choices and Options** pane.

Delete content from an object

You can delete (completely remove) a content item from an object. When you delete content, all settings and associations to the items are also removed.

If there are any instances or templates derived from the template that use the deleted item, the deletion of the item will propagate to all the derived objects. That is, all objects derived from the template will no longer contain the deleted content.

Delete an attribute or content from a template

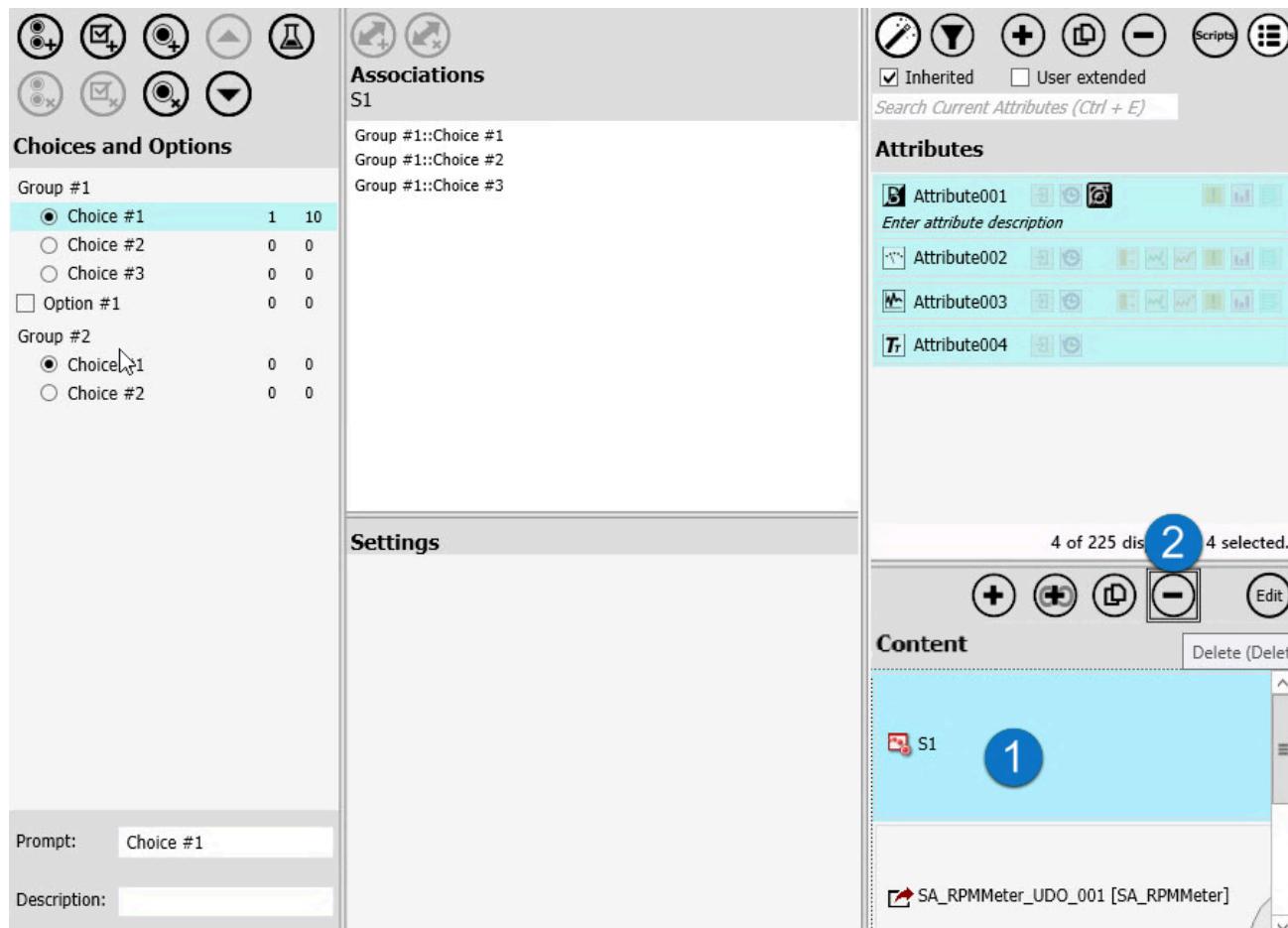
You can delete an attribute or content (object-owned or linked) from a template that has an Object Wizard. If there are objects derived from the Object Wizard that use the deleted attribute or content, the deletion propagates to all the derived objects.

Linked graphics, layouts, and external content items will still exist in the Visualization folder. Deleting an object-owned graphic completely removes it. Once you save the template that contains the deleted object-owned graphic, the graphic cannot be retrieved.

To delete an attribute or content from a template:

1. Select the content item from the **Content** pane. To delete an attribute, select it from the **Attributes** pane. You can select more than one attribute or graphic at a time.
2. Select the **Delete** button. This removes the selected graphic from the object. Use the **Delete** button in the **Attributes** pane to delete attributes.

Note: Added (object-owned) graphics are permanently deleted once the object is saved. Linked content is removed from the object but remains in the Visualization folder.

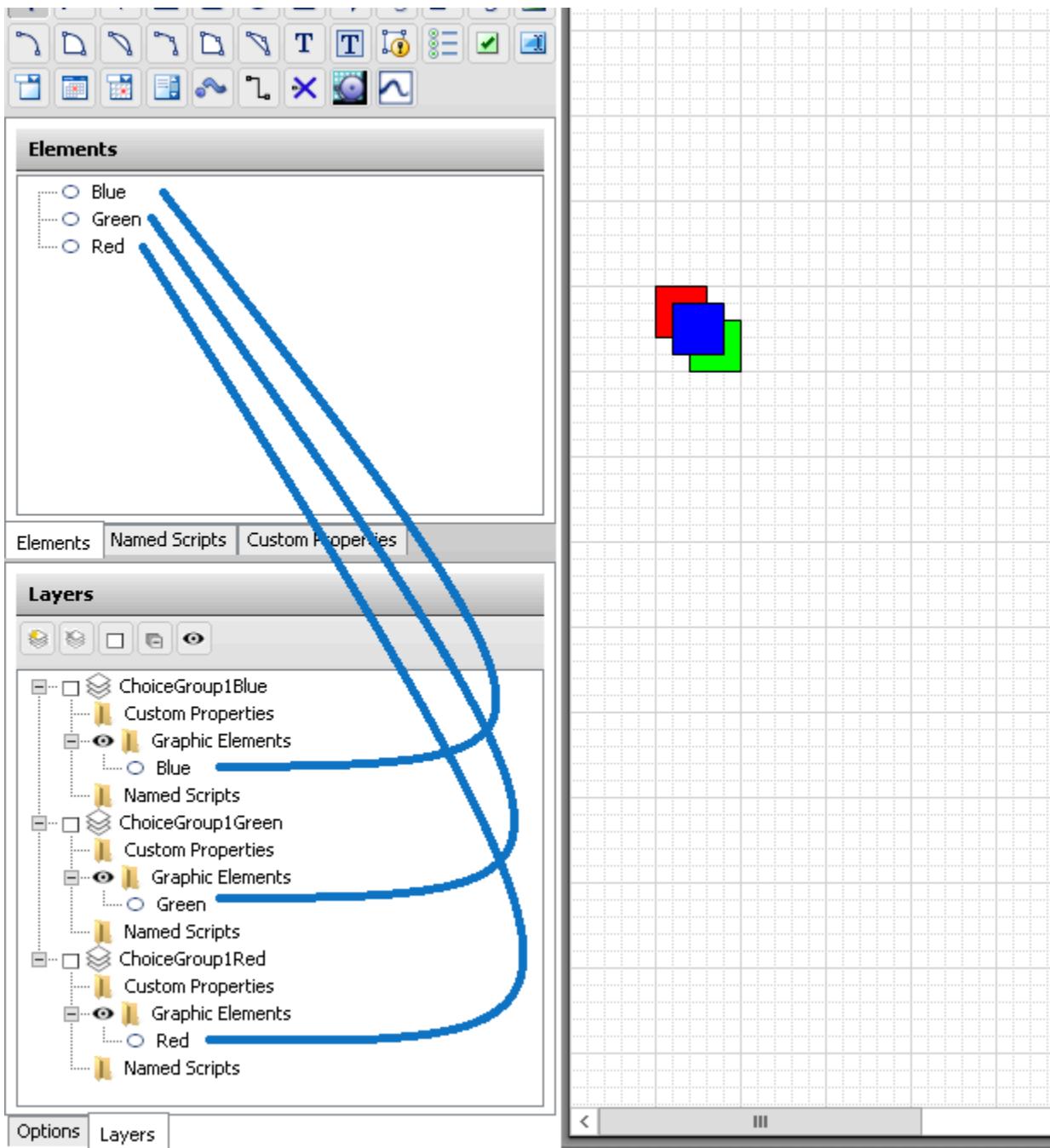


Example1: Map a symbol wizard to an object wizard

This example shows how to create a simple Symbol Wizard, set configurations, and map the configurations to

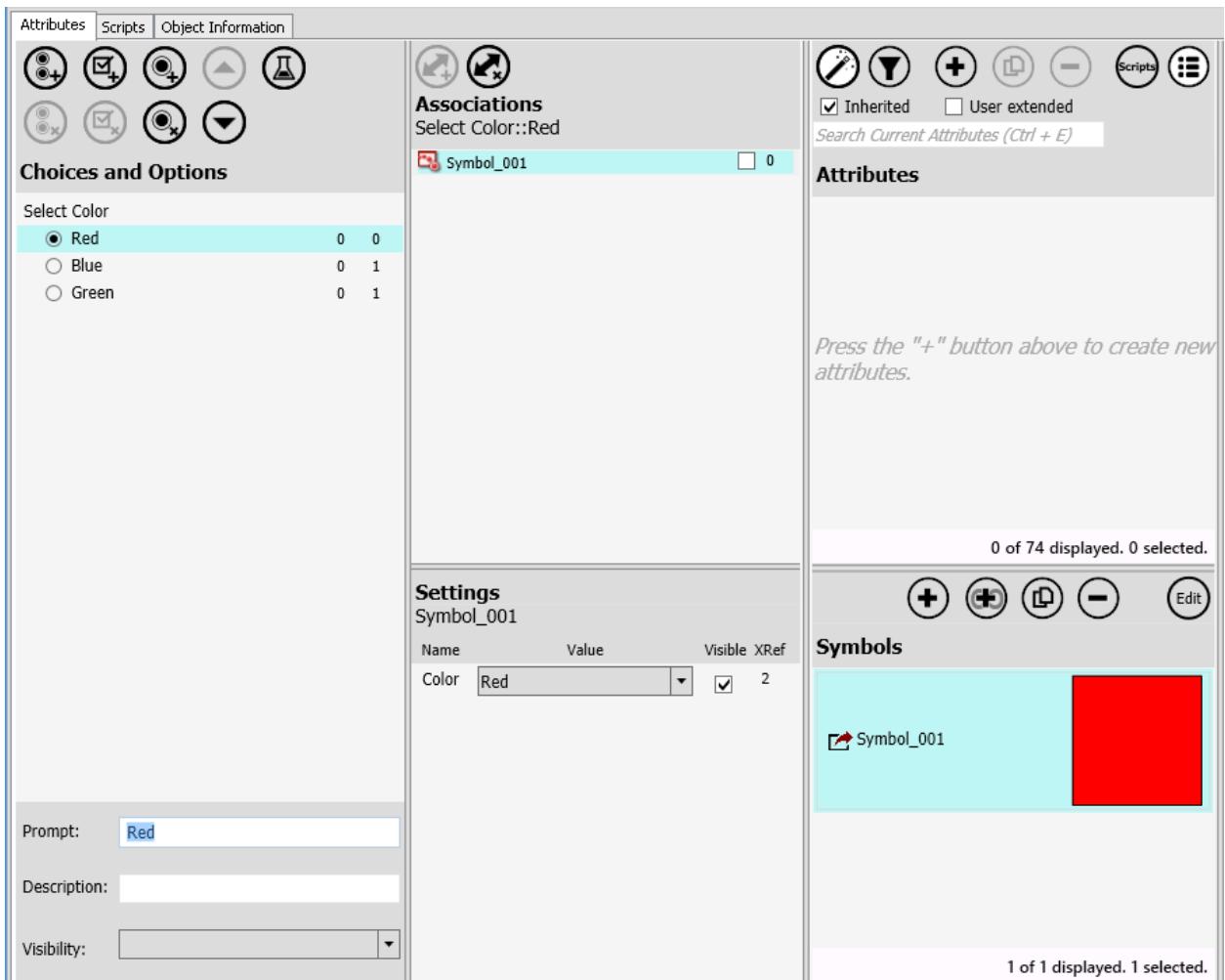
Object Wizard Choices.

1. Create a new graphic in the Visualization folder.
2. Build a simple Symbol Wizard in the **Graphic Editor**. by adding a Choice Group with three Choices named Red, Green, and Blue.
3. Add three shapes, filled with colors that correspond to the Choices. Rename the shapes so you can identify them.
4. Assign each shape to the corresponding Choice by dragging the shape name from the **Elements tab** to the corresponding Choice in the **Layers tab**.



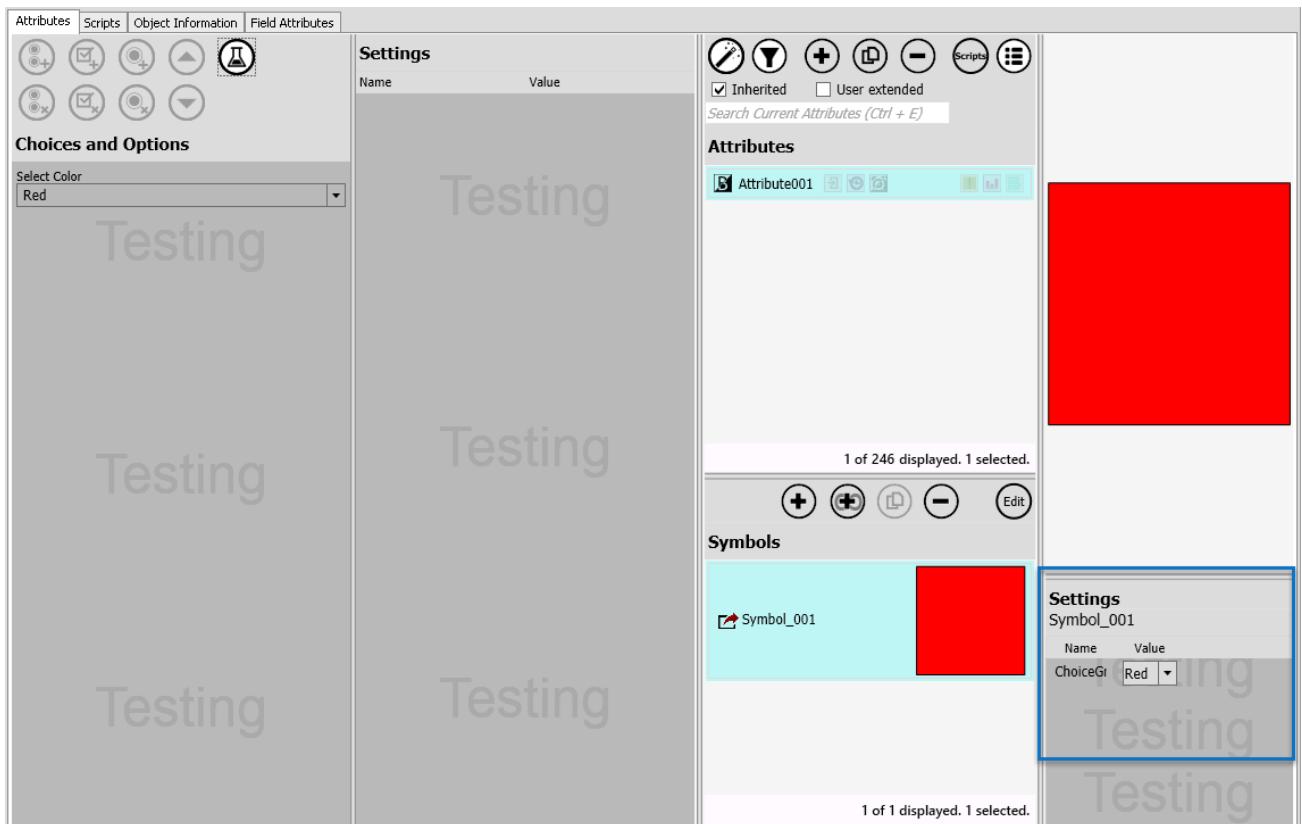
5. Save and close the **Graphic Editor**.

6. Derive a new template from the Template Folder, and add an Object Wizard with:
 - One Choice Group. Change the name of the Choice Group to "Select Color."
 - Three Choices: Red, Green and Blue.
7. Link the Symbol Wizard you created in step 1 to the derived template.
8. Configure each Symbol Wizard association to match the corresponding Object Wizard Choice.
 - Red Symbol Wizard Choice selected for Red Object Wizard Choice.
 - Blue Symbol Wizard Choice selected for Blue Object Wizard Choice.
 - Green Symbol Wizard Choice selected for Green Object Wizard Choice.



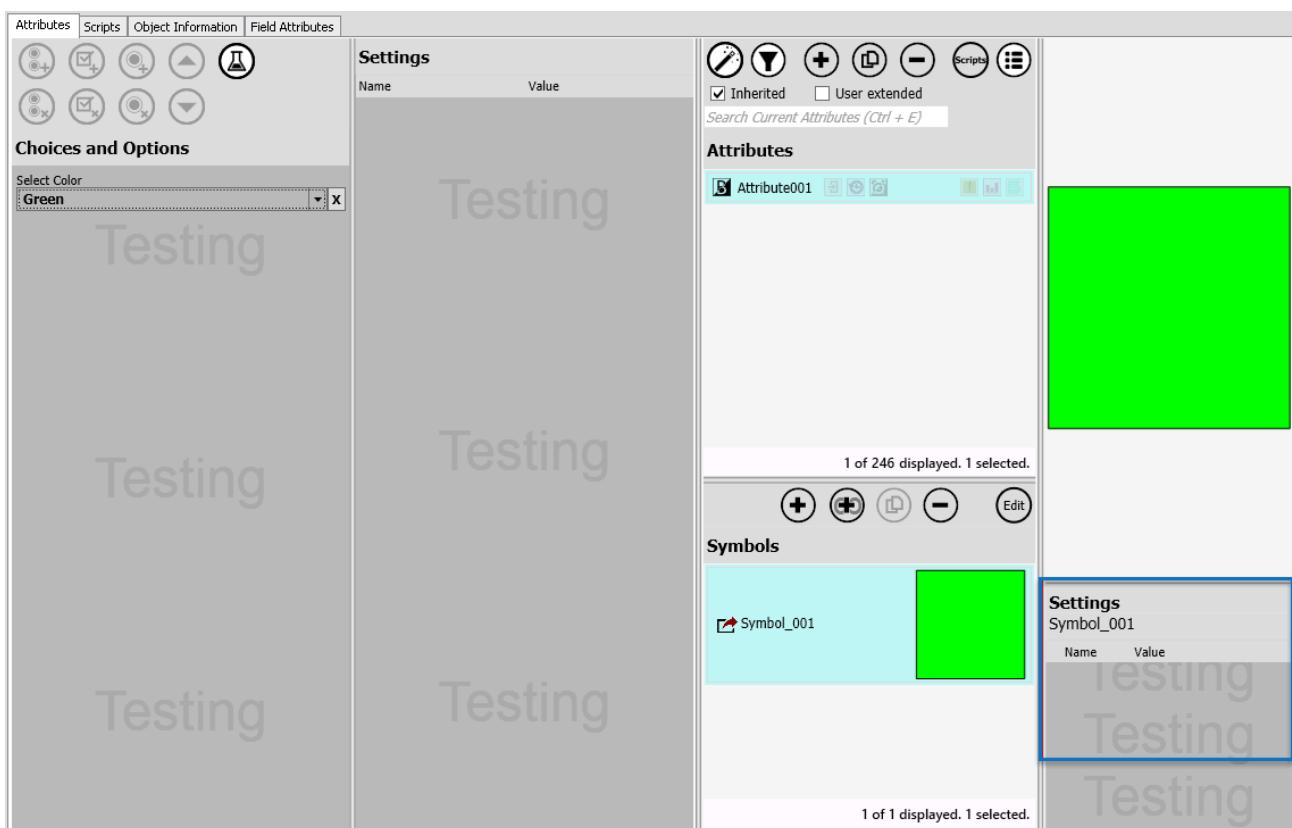
Note: Only the default selection can be marked as visible (you can remove the checkmark to hide the default selection). Selections with override are not visible. Users cannot modify selections that are not visible.

9. Switch to Test Mode.
10. With the default Option Wizard Choice selected (Red, in this example), you should see something like this:



The Symbol Wizard Choice Group can be changed in the **Symbol Settings** pane, since we did not turn off visibility for the default selection.

- With any of the non-default Option Wizard Choices selected, you should see something like this:



In this case, Symbol Wizard settings cannot be changed because a non-default configuration of the Symbol Wizard is associated with the Green Object Wizard Choice. When a non-default selection is made in the Symbol Wizard, its visibility setting is turned off and disabled.

Example2: Configure a linked graphic associated with multiple choices

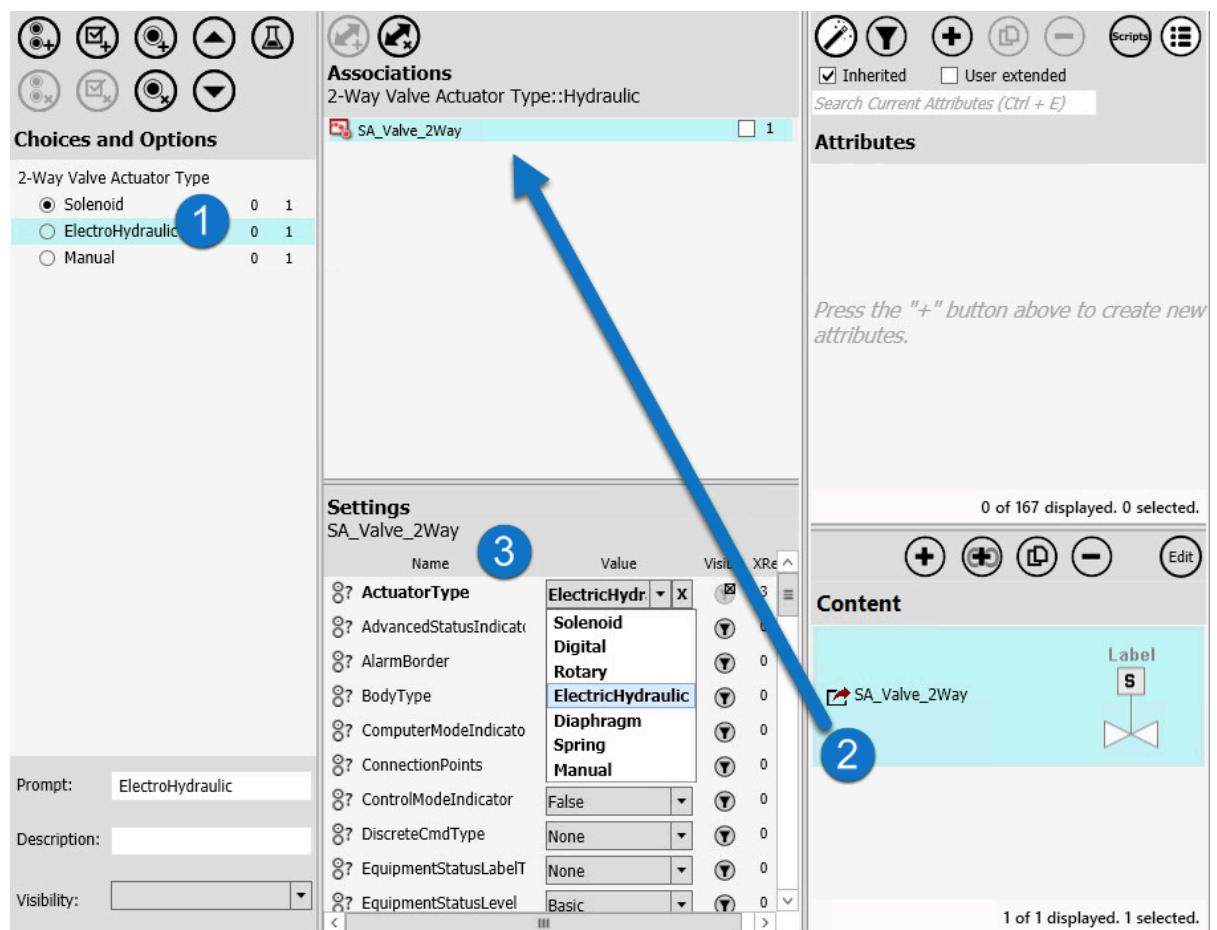
This example shows the steps required to link and configure a graphic from the **Visualization folder**, set configurations, and associate different configurations with Choices and Options. This example uses a graphic from the **Situational Awareness Library**. You could also link other types of content, such as layouts and external content items. However, only graphics allow you to configure and associate different configuration settings within the Object Wizard.

Note: Most graphics in the **Situational Awareness Library** are Symbol Wizards.

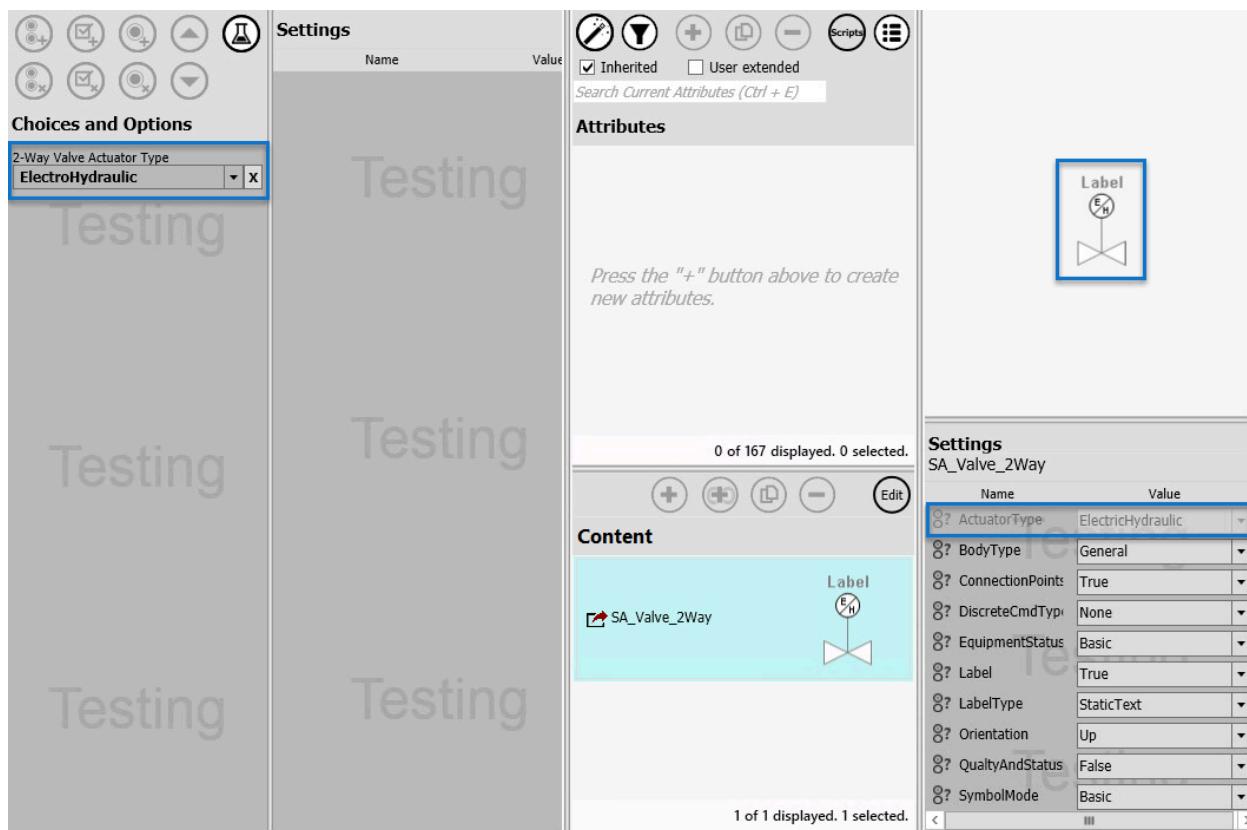
To configure a linked graphic associated with multiple choices

1. Derive a template from the \$UserDefined template, then:
 - a. Add an Object Wizard Choice Group and name it "2-Way Valve Configuration."
 - b. Add three Choices, named: "Solenoid," "ElectroHydraulic," and "Manual."
2. Navigate to the Equipment folder of the Situational Awareness Library (in the **Visualization** folder), and link to the SA_Valve_2Way graphic. The graphic is added to the **Content** pane.
Associate the linked graphic with each of the Choices created in step 2, and set the ActuatorType in the **Settings** pane to match the associated Object Wizard Choice.
3. Set visibility and overrides for each association:

- For Solenoid:
 - Select the Solenoid Choice from the **Choices and Options** pane.
 - Select SA_Valve_2Way from the Associations pane.
 - In the Settings pane, make sure that ActuatorType is set to the default (Solenoid), then turn off visibility.
- For ElectroHydraulic and Manual:
 - Select the ElectroHydraulic or Manual Choice from the **Choices and Options** pane.
 - Select SA_Valve_2Way from the Associations pane.
 - In the **Settings** pane, set the ActuatorType to match the Choice in the **Choices and Options** pane. Since both ElectroHydraulic and Manual are overrides, visibility of these settings automatically changes to hidden.



4. Switch to **Test mode** and check each of the Choices in the **Choices and Options** pane.
 - ActuatorType under graphic **Settings** should be grayed-out to indicate it is hidden
 - The graphic setting should match what you have selected in the **Choices and Options** pane.
 - The graphic thumbnail should match the selected ActuatorType.



Conditional visibility expressions for choice groups, choices, and options

You can set whether or not a user will be able to see individual Choice Groups, Choices and Options as they use the Object Wizard. Visibility expressions are scripted and use simple logical expressions to determine if users will be able to see a particular Option or Choice when configuring instances. The visibility expression determines if the Object Wizard component will be visible or hidden, based on the user's previous Object Wizard selections. For example, selecting Choice #1 may hide (or show) Choice Groups, Choice or Options that follow that Choice in the Object Wizard.

Note: You cannot add a visibility expression to the first listed item (Choice Group or Option) in an Object Wizard; the first item is always visible.

When a user uses the Object Wizard to configure an instance, the default behavior of any elements hidden by visibility expressions remains in effect. For example, if Choice #1 is the default of a Choice Group, and the Choice Group is hidden, Choice #1 is still selected for the configured instance, even though it is not visible to the user. If the default state of an Option is True but the Option is hidden, the Option remains True for the configured instance.

Visibility is set through logic expressions assigned to Choices and Options. These logic expressions use the Boolean operators AND, OR, and NOT.

In derived templates that contain an Object Wizard from the parent object, icons are used to implement visibility settings that are in addition to the settings made in the parent object. Note that the visibility settings made in the parent Object cannot be changed in the derived template. See [Object wizard visibility settings in a derived template](#) for additional information.

Add a basic visibility expression

To add a basic visibility expression to a Choice Group, Choice or Option:

1. Highlight the item for which you are setting visibility (Choice Group, Choice or Option). You can select any item except the initial one. That is, you cannot make the visibility conditional of the first item (Choice Group or Option) in the Object Wizard. If the first item is a Choice Group, you cannot make it or any of its Choices conditional.
2. Place the cursor in the Visibility text box of the selected item, and type "[" (open bracket character). A drop-down list of all preceding items (Choices and Options) is displayed.

Note: Choices are listed in format "*Choice Group Name::Choice Name*", for example, Choice Group #1::Choice #1. Options are listed in the format "*Option::State*", for example Option #1::True.

Note: While you can simply select an item for a simple visibility expression without typing an open bracket, we recommend that you always type the open bracket as a matter of habit. As you begin building more complex visibility expressions, you risk deleting the expression if you select an item from the drop-down without first typing an open bracket.

3. Select the item from the drop-down list that will be used to control the visibility of the selected item. For example, you can make Option #2 visible only if Choice #1 of Choice Group #1 has been selected (Group #1::Choice #1), and/or if Option #1 is set to True (Option #1::True).

For additional information, see [Operators, rules, and behavior for visibility expressions](#).

Note: When a user configures an instance, the default behavior of elements remains in effect, even if they are hidden from the user. The default Choice of a hidden Choice Group is still selected, and hidden Options remain in their default states.

Add a visibility expression with logical operators

To add a visibility expression that uses logical operators to a group, choice or option:

1. Highlight the item for which you are setting visibility (Choice Group, Choice or Option). You can select any item except the initial one. That is, you cannot make the visibility conditional of the first item (Choice Group or Option) in the Object Wizard. If the first item is a Choice Group, you cannot make it or any of its Choices conditional.
2. Place the cursor in the Visibility text box of the selected item, and type "[" (open bracket character). A drop-down list of all preceding items (Choices and Options) is displayed.

Note: Always type an open bracket before selecting an item from the drop-down list. Simply selecting an item without first typing an open bracket will overwrite the existing expression.

Note: Choices are listed in format "*Choice Group Name::Choice Name*", for example, Choice Group #1::Choice #1. Options are listed in the format "*Option::State*", for example Option #1::True.

3. Select the item upon which you are basing the visibility of the selected item. It will be entered in the visibility expression. To expand the expression:
 - Place the cursor at the front of the expression (before the open bracket) and enter the NOT operator, or
 - Place the cursor at the end of the expression (after the close bracket) and enter the AND or OR operator.
4. Type "[" (open bracket character) and select another item from the drop-down. All Object Wizard items must be enclosed in brackets. Enclose items and operators in parentheses as needed to ensure that the

expression resolves as intended.

Choices and Options

Group #1

<input checked="" type="radio"/> Choice #1	3	2
<input type="radio"/> Choice #2	1	2
<input type="radio"/> Choice #3	2	0
<input type="checkbox"/> Option #1	1	2
<input type="checkbox"/> Option #2	1	2

Prompt: Option #2

Description:

Visibility: [Group #1::Choice #1] or

Associations

Option #2

B OptionA2 2

a single option/choice association above its settings here.

The visibility expression can only refer only to existing choices.

- Add additional operators, parentheses, and Object Wizard items as needed to complete the visibility expression. For example:

[ChoiceGroup1::Choice1] AND ([Option1::True] OR [Option2::True] OR [Option3::False])

Note: When a user configures an instance, the default behavior of elements remains in effect, even if they are hidden from the user. The default Choice of a hidden Choice Group is still selected, and hidden Options remain in their default states.

For additional information, see [Operators, rules, and behavior for visibility expressions](#).

Operators, rules, and behavior for visibility expressions

As you build an Object Wizard, you can create visibility expressions that selectively hide Choice Groups, Choices, and Options. This can simplify the workflow by only showing Choices and Options that are applicable, based on Choices and Options that the user has already made. When an Option or Choice Group is hidden, the system applies the configured default values for attributes, graphics, and scripts associated with the hidden Option or Choice Group. You can leverage this capability to hide advanced functions from users tasked with configuring instances from templates that contain Object Wizards.

The **Visibility** prompt lets you assemble expressions to hide or show Choice Groups, Choices, or Options based on what the user has previously selected. Note that Visibility cannot be conditional for the first Object Wizard item (Choice Group or Option). The first Object Wizard item is always visible. Available logical operators are AND, OR, and NOT.

Observe these basic rules and behaviors when building visibility expressions:

- Always start by typing the "[" (open bracket character) in the visibility text box. This will bring up a list of all

Choices (from other Choice Groups) and Options that precede the selected item. Do not simply highlight an element without first typing the open bracket character.

- Choices are listed with their Choice Group. Each Option is listed twice, for both states (True and False). Use fully qualified Option/Choice selections surrounded by brackets. The separator between Choice Groups and Choices, and between Options and their states is ":" For example:

[Valve Type::3-Way] OR [Valve Type::4-Way] AND [Option1::True]

- Valid logical operators:

- Valid binary operators are "AND" and "OR".
- Valid unary operator is "NOT".
- "AND" has a higher precedence than "OR".
- "NOT" has a higher precedence than "AND" or "OR".
- You can use "&" or "&&" instead of "AND".
- You can use "|" or "||" instead of "OR".
- You can use "!" instead of "NOT".

- Parentheses can be used to build complex expressions. For example:

[ChoiceGroup1::Choice1] AND ([Option1::True] OR [Option2::True] OR [Option3::False])

Note that without parentheses, this expression would resolve very differently, since "AND" has a higher precedence than "OR". If parentheses were not used, the expression would resolve this way:

[ChoiceGroup1::Choice1] AND [Option1::True] OR [Option2::True] OR [Option3::False]

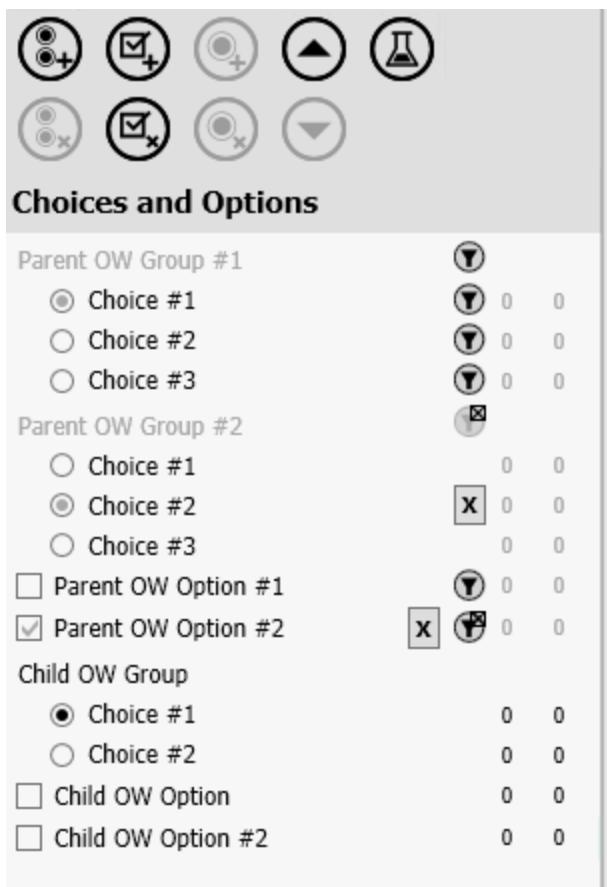
- Literals and operators are not case sensitive.
- Spacing between literals and operators is optional.
- The visibility expression editor validates the logical expression as you enter it, but does not check if the expression is possible. For example, the expression "[Valve Type::3-Way] AND [Valve Type::4-Way]" would not be flagged since it is logically correct, even though the condition is impossible (you cannot select more than one Choice from a single Choice Group).

Object wizard visibility settings in a derived template

When you derive a template from a template with an Object Wizard, derived Object Wizard elements can no longer be edited, and Object Wizard group names are grayed out. Other noticeable changes are:

- A visibility icon  appears next to each derived wizard item.
- You can override visibility settings by clicking on the visibility icon.
- The item counts are reset to zero; changes made in the derived object will increment the counters.
- If you change a default selection, the setting override indicator  appears next to the changed setting.

See [Object wizard user interface details](#) for more information about visibility icons and the setting override indicator. For additional information about derivation and visibility, see [Object wizard derivation](#).



The visibility expressions that were configured in the parent Object Wizard remain in effect. You can, however, hide Choice Groups, Choices, and Options. Instead of using Boolean expressions to set the visibility of Groups, Choices, and Options, use the visibility icon . When changing the visibility settings of a derived Object Wizard:

- The behavior of a hidden Choice or Option does not change when configuring an instance from the derived Object Wizard. The default state of a hidden Option remains in effect when the instance is configured. If the Option's default state is True, any attributes, graphics, or scripts associated with it are automatically configured for the instance. Similarly, the default Choice in a hidden Choice Group is automatically selected, along with its associated attributes, graphics, and scripts.
- You cannot hide (turn off the visibility of) a default Choice. You can, however, hide the entire Choice Group.
- If every Choice except the default is hidden, the entire Choice Group is hidden.
- If you change a default choice, the entire Choice Group is automatically hidden. You can turn visibility on again for the Choice Group. Once the Choice Group has been made visible, you can hide individual Choices.

Object wizard trimming

Trimming refers to the ability of the Object Wizard to remove unneeded attributes, graphics, and scripts from derived instances. By removing unneeded items, the size of instances can be minimized, thus allowing more efficient utilization of runtime system resources. This can also result in faster deployments.

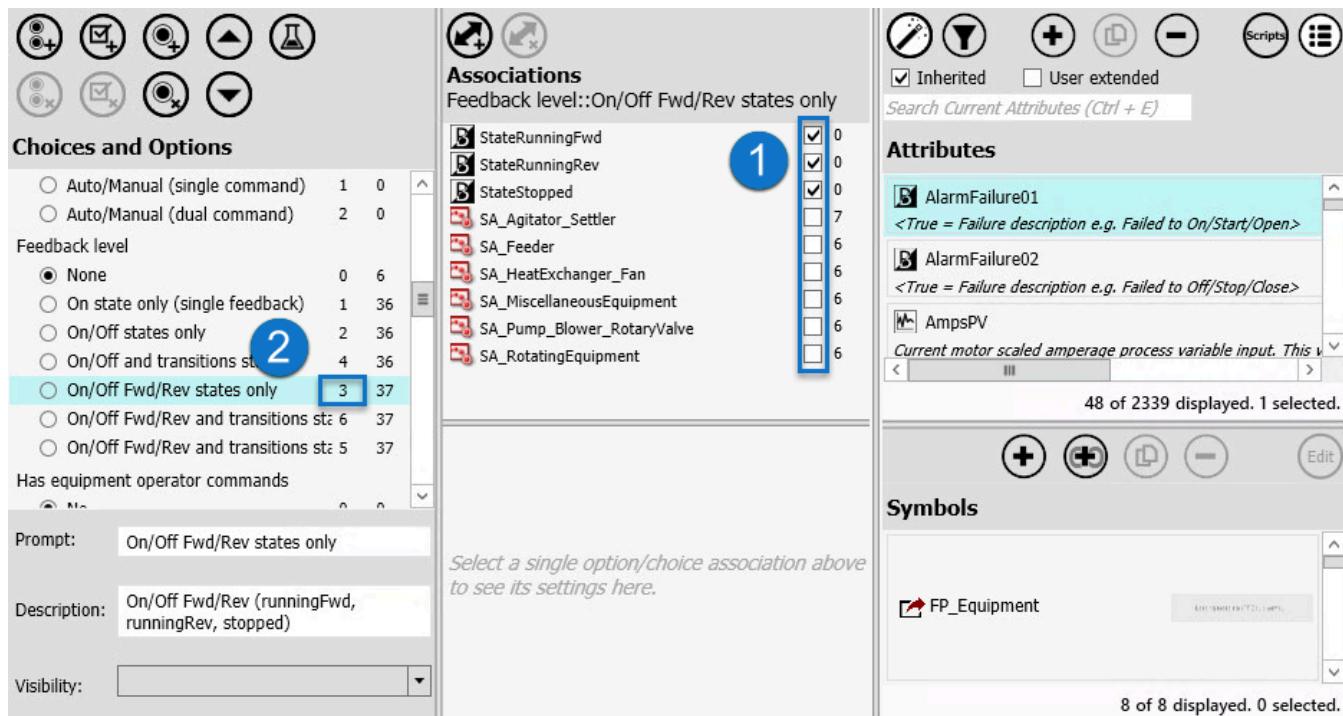
- When you associate an attribute with a Choice or Option, the **Trim** checkbox (1) is checked by default. This means that trimming is enabled, and the attribute will be evaluated for trimming.

- When you associate a graphic with a Choice or Option, the **Trim** checkbox is cleared by default. This means that trimming is DISabled, and the graphic will NOT be evaluated for trimming.
- When you associate a script with a Choice or Option, the **Trim** checkbox is checked. Trimming cannot be disabled for scripts.

Note: Graphics do not exist as runtime objects, but do exist as part of a ViewApp. Trimming unnecessary graphics can help to reduce system resource requirements.

The total number of trimmable items (attributes, graphics, and scripts) associated with each Object Wizard Choice and Option is displayed in the **Choices and Options** pane (2). Items that are not enabled for trimming are not included in this count. The default trim state of associated graphics is disabled.

Note: The count of trimmable items resets to 0 in derived templates.



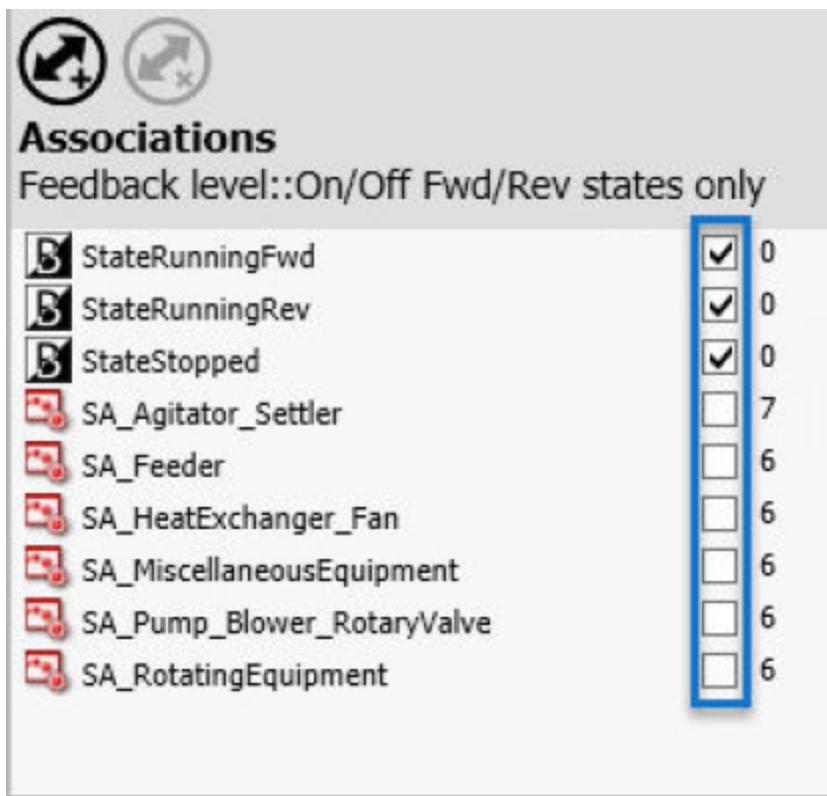
Enable trimming

Whether or not an attribute or graphic associated with a Choice or Option will be evaluated for trimming is determined by the state of its trimming checkbox in the **Associations** pane. See [Object wizard user interface details](#) for additional information.

Note: Scripts are always evaluated for trimming.

To enable trimming:

- Select the **Trim** checkbox in the **Associations** pane.
 - When checked, trimming is enabled.
 - When unchecked, trimming is disabled.



Graphics and attributes are evaluated for trimming when an instance is derived, in accordance with the following behaviors (scripts are always evaluated for trimming):

- If none of its associations have trimming enabled, the item is not evaluated for trimming and is always propagated to derived instances.
- The item is propagated to the instance if any Choice or Option where trimming is enabled is selected.
- The item is trimmed if at least one of its associations has trimming enabled, and none of the associations with trimming enabled are selected (even if associations with trimming disabled have been selected).
- Trimming occurs only when the instance is saved.

Note: Trimming is not applicable to derived templates, since these are not runtime objects.

Attribute and graphic overrides

You can associate different configurations of graphics and attributes with different Object Wizard Choices and Options. You can also allow users to override values that you previously set as they configure derived objects. See [Configurable settings for attribute and graphic values](#) for additional information about setting overrides and visibility for Symbol Wizard and custom property values. For detailed information about Symbol Wizards and custom properties, see [Using custom properties with Application Server](#) and "Working with Symbol Wizards" in the *Creating and Managing Industrial Graphics User Guide*.

The **Settings** pane lets you configure setting overrides and visibility for associated attributes and graphics.

Attribute Overrides

For attributes, you can override default settings and set visibility for:

- Attribute description
- Data type
- Initial value
- Engineering units (if applicable)
- Activate/deactivate features, such as I/O and alarms
- Feature values

Attribute Setting Visibility

When you associate an attribute with an Object Wizard Choice or Option, its settings are **hidden by default**. You can make settings visible by selecting the checkbox next to the setting. You can also make overrides visible.

Graphic Overrides

For graphics, you can override default settings and set visibility for:

- Symbol Wizard Choices and Options
- Custom properties

Graphic Setting Visibility

When you associate a graphic with an Object Wizard Choice or Option, Symbol Wizard and custom property settings (if any) are **visible by default**. If you override a graphic custom property or Symbol Wizard setting, its visibility automatically turns off (the setting is hidden). You can make an overridden custom property setting visible again by selecting the checkbox next to the setting.

Note: If you override a Symbol Wizard setting, the only way to make the setting visible again is to remove the override.

Custom properties can be Boolean, integer, float, string, etc. See [Object wizard user interface details](#) for additional information. For detailed information about custom property data types, see [Using custom properties with Application Server](#). When configuring string, time, and elapsed time custom properties, a toggle is provided in the **Settings** pane to configure the custom property as either:

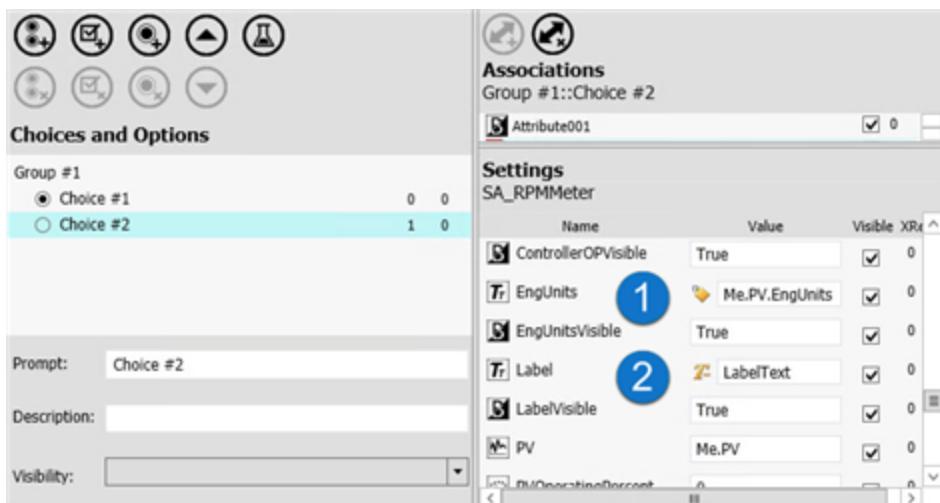


Reference string or an expression (1). Reference strings will be evaluated to determine their meanings. For example, Me.Tank1.InletValve.PV.



String literal (2). String literals are passed as static text.

To change the string type, select the icon next to the text field.



Enter overrides and set visibility for attribute and graphic values

For both attributes and graphics that contain a wizard or custom properties, you can enter overrides that apply only to the associated Object Wizard Choice or Option. While the following procedure shows how to configure a graphic, the procedure for configuring attributes is virtually identical. See also [Attribute and graphic overrides](#).

To configure an Override for a Choice or Option

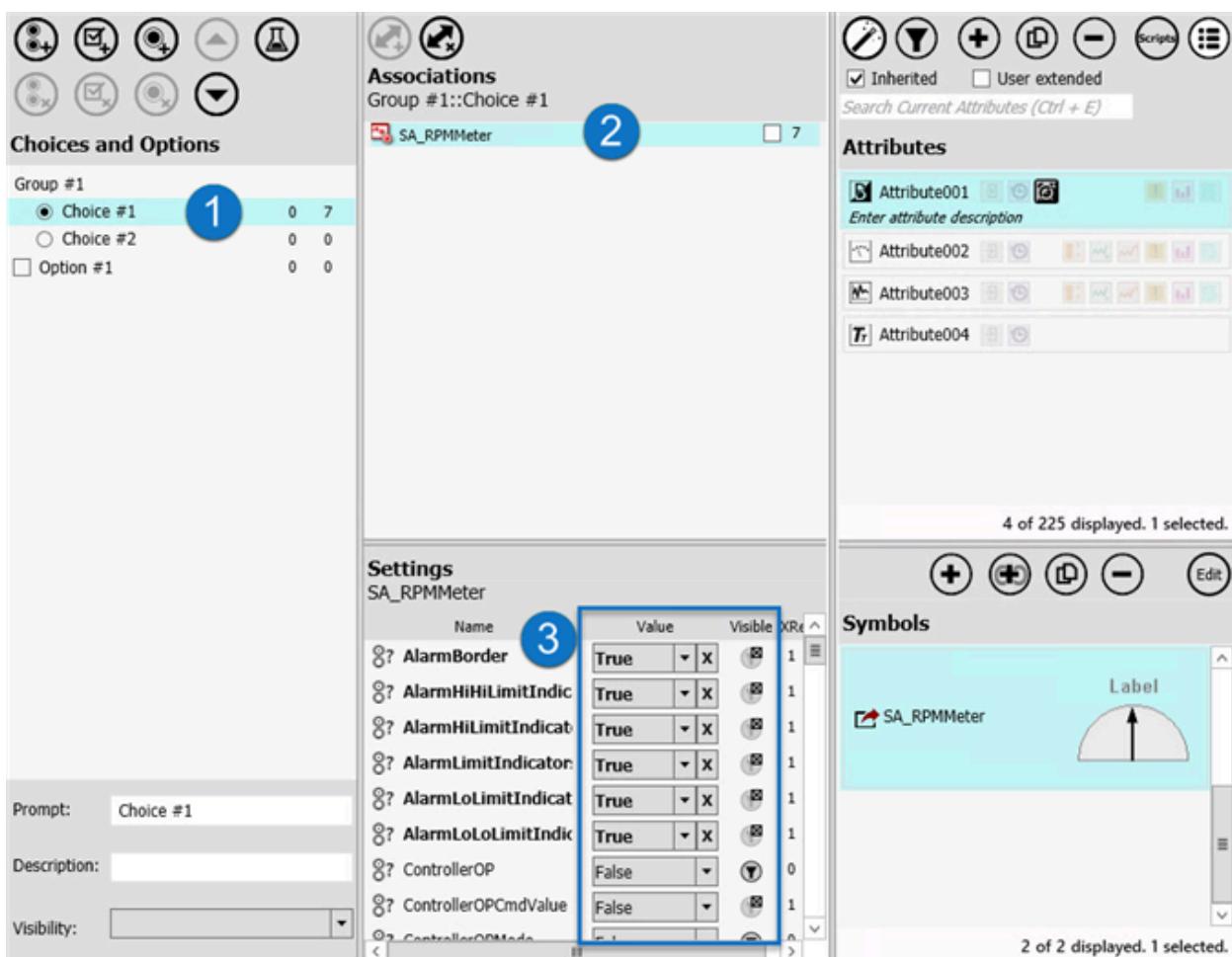
1. Select the Choice or Option you want to configure (**Choices and Options** pane).
2. Select a graphic or attribute (**Associations** pane) associated with the Choice or Option. See [Associate Content with a Choice or Option](#) for information about associating graphics with Object Wizards.

This does not apply to scripts since they can only be associated with a Choice or Option. There are no configuration or visibility settings that can be configured.

3. In the **Settings** pane, override the default values as needed. Overrides are shown in **bold** and by an **X** next to the new value. See
 - You can resize columns in the **Settings** pane as needed by selecting and moving the boundary between column headers.
 - Symbol Wizard and Custom Property settings are visible by default. Overrides are automatically hidden from users.
 - You can only restore the visibility of a Symbol Wizard setting with an override by removing the override.

Note: This behavior is unique to Symbol Wizard overrides. You can change visibility settings for both Custom Property overrides and attribute overrides.

- You can restore the visibility of a Custom Property with an override by selecting the Visibility icon until it is unmuted. The icon will have a small X in its upper right quadrant to indicate that Visibility has been overridden.
- Switching a Custom Property string type between expression/reference and static text is an override (value is displayed with **bold** type with an **X** next to the value and in the Visibility setting).
- Attribute settings are hidden by default. You can override the Visibility settings for attributes that use their default setting, as well as for attributes that have an override.



Note: Attribute, Symbol Wizard, and Custom Property values that are marked as visible can be overridden by users as they configure instances.

Note: The total number of configured settings (visibility settings and value settings) is shown in the **Associations** pane, and is reflected in the **Choices and Options** pane. See [Attribute and graphic overrides](#) for additional information.

Attribute and graphic overrides

Override values for an attribute or graphic (with a Symbol Wizard or custom properties) can be configured for each associated Choice or Option. When you associate an attribute or graphic, the default values are listed in the **Settings** pane. To configure values that apply to the selected Choice or Option, enter the values in the **Settings** pane. If the attribute or graphic is associated with multiple Choices or Options, different values can be used for each one. Values for the Choices and Options are saved when you save the object. If the setting is marked as visible, users can override your setting (or override the default, if you have not configured a setting) as they derive and configure instances from the Object Wizard.

Graphics with overrides that affect their shape, for example, certain Situational Awareness Library graphics, are rendered with the applicable shape-changes as defined by the overrides.

Note: If you override a Symbol Wizard setting, the only way to make the setting visible again is to remove the override. Attributes and custom properties with overrides can be made visible again by selecting the visibility

checkbox.

A configured setting is defined as any change from the default configuration of the attribute or graphic. This includes changing a value or visibility. For example, you might change the default Boolean value from true to false, or you might change an integer value from 5 to 10, or change engineering units from liters to cc. Changing the visibility of a setting also counts as a configured setting.

- The default visibility setting for attribute values is hidden (Visibility icon is muted (grayed-out) in the **Settings** pane).
- The default visibility setting for Symbol Wizard and custom property values is visible (Visibility icon is unmuted in the **Settings** pane).
- Visibility overrides are indicated by a small X in the upper right quadrant of the visibility icon.

The total number of configured settings for each individual attribute and graphic is displayed in the **Associations** pane. The total number of configured settings for ALL attributes and graphics associated with an individual Choice or Option is displayed in the **Choices and Options** pane.

Configurable settings for attribute and graphic values

Graphics and attributes can be configured for each Choice or Option with which they are associated. When you initially associate an attribute or graphic with a Choice or Option, the default settings of the attribute or graphic are listed in the **Settings** pane.

Note: Icons for attribute and graphic settings are defined in [Object wizard user interface details](#).

To override default values, enter the new values in the **Settings** pane (1). If the attribute or graphic is associated with multiple Choices or Options, different overrides can be used for each one. These configured (override) values are saved when you save the template.

An override is defined as any change from the default graphic configuration. For example, you might change the default Choice of a Symbol Wizard, or change a custom property or attribute value. Changing the visibility of a setting is also considered an override. By default, graphic settings are visible in derived objects, while attribute settings are hidden by default.

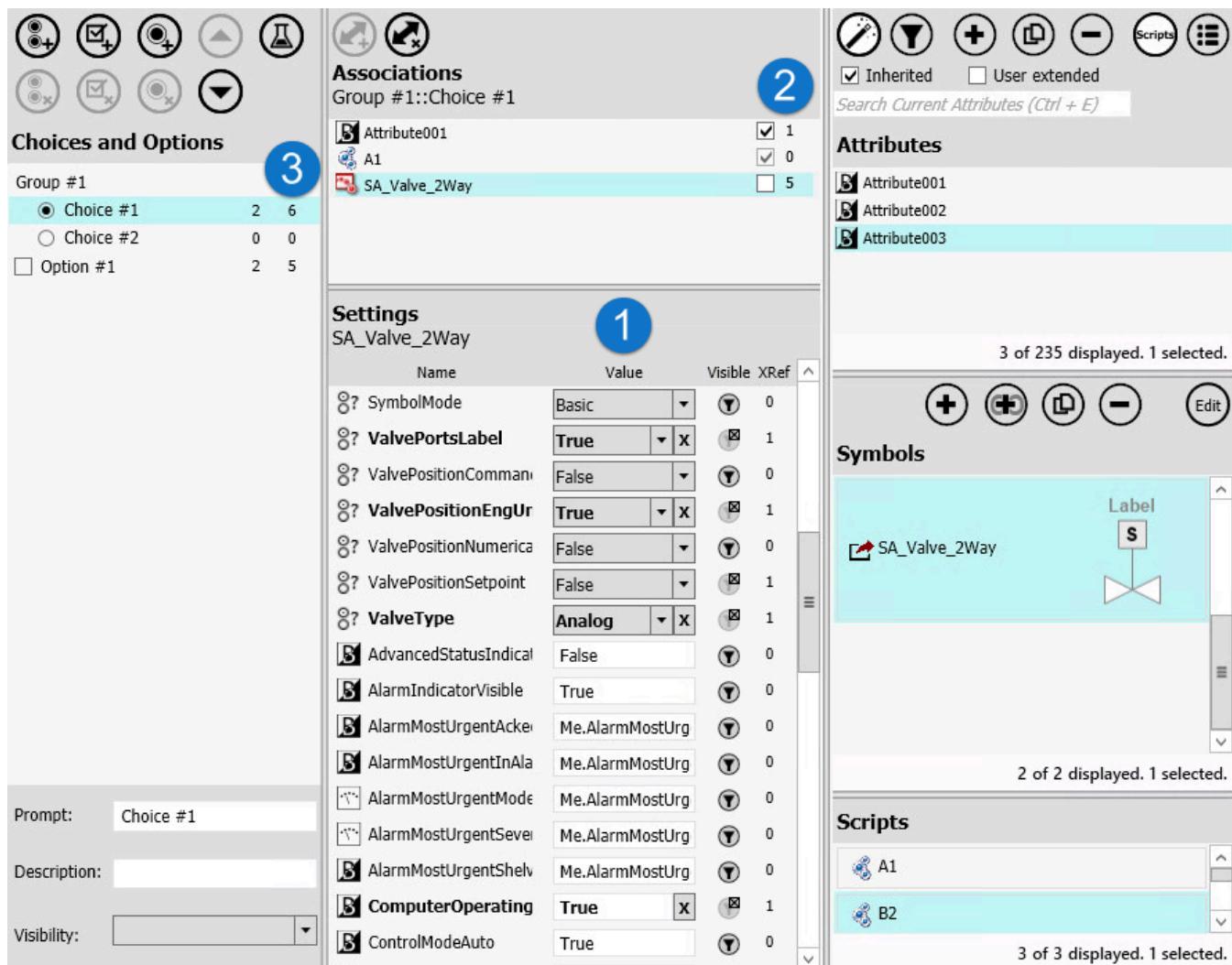
If the setting is marked as visible, users can enter overrides when they use the **Create New Asset** editor to configure a new instance. If the setting is hidden, it is not exposed to users and therefore the default value cannot be overridden.

Symbol Wizard overrides are automatically hidden and cannot be changed back to visible, except by restoring the original setting. The names of configured settings are displayed in bold type, and have an "X" next to the setting. To restore the original setting, clear the X.

Configured visibility settings have a small x in the upper right quadrant of the visibility icon. To restore the original, click on the icon twice to clear the override. If the original visibility state was visible, the icon will cycle through override - hidden, override - visible, and then return to its original state of visible (no override).

Custom property and attribute overrides are also hidden automatically. However, you can override the visibility setting for a setting that has an override value and make it visible again.

The total number of configured settings for each individual attribute and graphic is displayed in the **Associations** pane (2). The total number of overrides for ALL attributes and graphics associated with a Choice or Option is displayed in the **Choices and Options** pane (3).



Object wizard derivation

You can derive child templates from templates with Object Wizards. Creating a second (or deeper) level of derivation from a template with an Object Wizard lets you extend the functionality of the original Object Wizard, and make specialized versions for specific applications. See [Reasons to derive a template from an object wizard](#) for additional information.

Set visibility of choices and options in a derived template

The original Object Wizard cannot be changed in the child derived template. However, you can set the visibility of Choice Groups, Choices, and Options. You can also set different defaults for Choice Groups, and you can turn Options on or off.

To set Object Wizard visibility in a derived template, do any of the following:

1. Select the visibility icon next to a Choice Group, Choice, or Option.
 - When you hide a Choice Group, all its Choices are hidden as well.

- When you hide a Choice, the remaining Choices in the Choice Group are still visible.
- When only one Choice is visible in a Choice Group, the entire Choice Group is hidden.

Note: The system applies the default values and actions of hidden Options and Choice Groups when a user derives an instance from the template. Visibility does not change configuration, it only affects aspects of the wizard that are visible to the user.

2. Change the default Choice in a Choice Group. This hides the Choice Group.

You can select the visibility icon to make the Choice Group visible again. You can then set visibility of the Choices individually.

3. Change the default state of an Option. This hides the Option.

You can select the visibility icon to make the Option visible again.

Reasons to derive a template from an object wizard

There are two main reasons for deriving a child template from a template that contains an Object Wizard:

1. To extend the existing Object Wizard and thus, make it more capable. Available actions to accommodate the specific needs of a target solution include:
 - Add Choice Groups or Options.
 - Add attributes, graphics, or associations.
2. To create specialized versions of templates that preconfigure choices, and thus simplify the task of configuring instances. For example, you can define common configuration aspects that are applicable to a set of similar instances, or modify some of the characteristics of the existing Object Wizard. Available actions to simplify configuration include:
 - Change default Choices or settings.
 - Hide Choice Groups, Choices, or Options.
 - Enter overrides for attribute or graphic settings, or change their visibility settings.

Using derived templates lets you propagate changes made to a parent template, as long as the child template does not have an override for what has changed. See [Propagation of attribute and graphic settings](#) for additional information.

If you are familiar with using Application Server, you will notice one other advantage of using derived Object Wizards: there is no need to lock objects. Changes automatically propagate to derived objects (provided the child template does not have an override for what changed in the parent), without explicitly locking attributes in the parent object.

Changes to object wizards in derived templates

After you create a derived template from a template with an Object Wizard, you can only make certain changes to the existing Object Wizard in the derived object.

- You can set overrides for attributes and graphics.
- You can hide or show settings for attributes and graphics.
- You can change visibility and defaults for Choices and Options.

- You can add new Choice Groups and Options (you cannot add new Choices to an existing Choice Group).
- You can add new attributes, graphics, and scripts, and associate them with the existing Object Wizard.
- You cannot remove existing associations from the Object Wizard.
- You cannot remove any existing Choice Groups, Choices, or Options.
- You cannot change the order of any existing Groups, Choices, or Options.
- You cannot change attribute, graphic, or script associations with any existing Choices or Options.

When a Choice or Option is changed from the default, the change is indicated by an X next to the modified Choice or Option. To revert to the default, select the X.

Similarly, when a visible setting for an associated attribute or graphic has been changed from the default, the change is indicated by an X next to the modified setting. To revert to the default, select the X.

Behavior of object wizard components in a derived template

The following list summarizes actions that you can and cannot perform in derived templates that contain an Object Wizard.

- Inherited Groups, Choices and Options cannot be edited.
 - Groups, Choices and Options cannot be renamed or deleted.
 - Their order cannot be changed (cannot be moved up or down).
 - Their prompts, descriptions, and visibility rules cannot be edited.
- You can change the default Choice, or the default state of an Option.
- Associations of inherited items (attributes, graphics, and scripts) with Choices and Options cannot be changed, and inherited associations cannot be removed.
- You can associate an inherited item with additional inherited Choices and Options
- Inherited attribute and graphic values and visibility settings can be overridden.
- Inherited values and visibility settings with overrides can be overridden.

Note: Even if a value is overridden in the parent object, the value will appear as a default value in the child object. This is because the override from the parent becomes the default for the child.

Object wizard test mode

Test mode lets you simulate how an Object Wizard will behave when a user configures a derived instance. Use Test mode as you develop your Object Wizard to ensure that it behaves as expected. For complex Object Wizards, associated attributes and graphics may not always appear as you had planned. Test mode lets you verify that derived instances will behave as intended.

- Select the **Test** button to enter Test mode.

- Deselect the button to exit Test mode.

Visibility expressions determine which Choice Groups, Choices, and Options are visible or hidden (along with

associated graphics, attributes, settings, and configured overrides), based on previous selections.

Verify behavior of visibility expressions

1. Open the template, containing the Object Wizard to be tested, in the **Object Editor**.
2. Select the **Test** mode button. The background color of the **Choices and Options** and **Settings** panes darkens, and the **Associations** pane is no longer shown. Choice Groups and Options are displayed, based on their visibility settings.



3. Check that the Choice Groups and Options that you expect to be visible are shown.
 - Regardless of the construction of the Object Wizard, the first item (Choice Group or Option) in the Object

Wizard is always visible.

- Choice Groups are displayed as drop down lists, with only the default Choice visible.
4. Check that the expected default Choice for each visible Choice Group is shown.
 5. Check that the default state (checked or unchecked) of each visible Option is correct.
 6. Cycle through different Choice and Option selections, and verify that subsequent dependent items are hidden or shown as designed.
 7. Verify that the settings for associated attributes and graphics are shown in the **Settings** pane, and that settings that should be visible are marked as visible. Verify that settings with visibility turned off are marked as hidden.

About object wizard behavior in test mode

At the most basic level, you can check that Object Wizard elements (Choice Groups, Choices, and Options) operate as intended. Choices and Options should appear in their default states. That is, Options that should be enabled by default should appear with their checkbox checked, and Choice Groups should show the default Choice. You can change these selections while in Test mode, to ensure that Choices and Options are correctly configured, including visibility settings and associated attributes, graphics, and scripts. Attribute and graphic settings that should not be changeable by consumers of the Wizard should not be visible.

- Check conditional visibility expressions. See [Conditional visibility expressions for choice groups, choices, and options](#) for additional information.
- Check that attributes and graphics are shown in their correct configuration when the corresponding Choice or Option is selected. See [Enter Overrides and Set Visibility for Attribute and Graphic Values](#) for additional information.
- Check that attributes and graphics are correctly trimmed. See [Object wizard trimming](#) for additional information.

Propagation of object wizard changes to derived objects

Most changes made to an Object Wizard propagate through the object hierarchy to all objects derived from the Wizard. Examples of changes that are propagated include:

- Adding, editing, renaming, changing the order of, or deleting Choice Groups, Choices, or Options.
- Adding, removing, or modifying an association (attribute, graphic, or script) with a Choice or Option.
- Changing the default state (setting) of an Option.
- Changing the visibility setting of a Choice or Option.
- Changing the default value or visibility setting of an attribute or graphic associated with a Choice or Option.
- Changing the visibility setting of a value in the **Settings** pane.

Important: With the exception of the first two items in the above list, changes DO NOT propagate to derived templates if the setting is marked as overridden in the derived template.

Object wizard behavior when a default choice is deleted

If you remove a default Choice from a Choice Group, the topmost Choice automatically becomes the new default, unless you explicitly select a different one. The new default propagates to child objects (templates and instances).

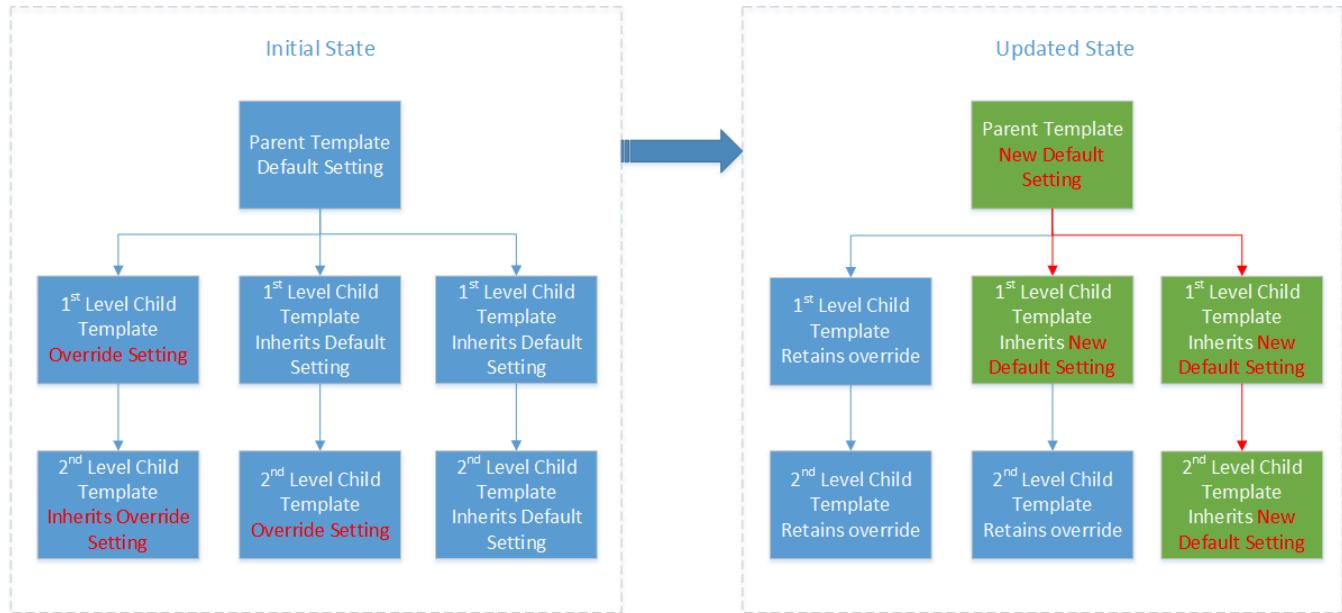
You cannot delete a Choice if there are only two Choices in the Choice Group; you must delete the entire Choice Group (a Choice Group, by definition, must contain at least two Choices). In this case, any associated attributes or graphics that were associated with the deleted Choice Group become a part of the object; they no longer have an association with the Object Wizard. Attributes and graphics that are not associated with the Object Wizard are not trimmed from derived instances.

Propagation of attribute and graphic settings

Changes to the default value and visibility setting of an attribute, Symbol Wizard, or custom property value are propagated to child objects, unless the value or visibility setting has been overridden in a derived (child) object. We assume that you do not want to lose an override you entered for a child object due to a change made to the parent.

If a default value is changed in the parent object, but the value has not been overridden in a derived object, the new default value propagates from the parent to child objects where the default value has not been modified.

Likewise, if a visibility setting is changed in a parent template, the change propagates to unmodified child templates. If the visibility setting was modified in the child, the change in the parent does not propagate.



Configure instances

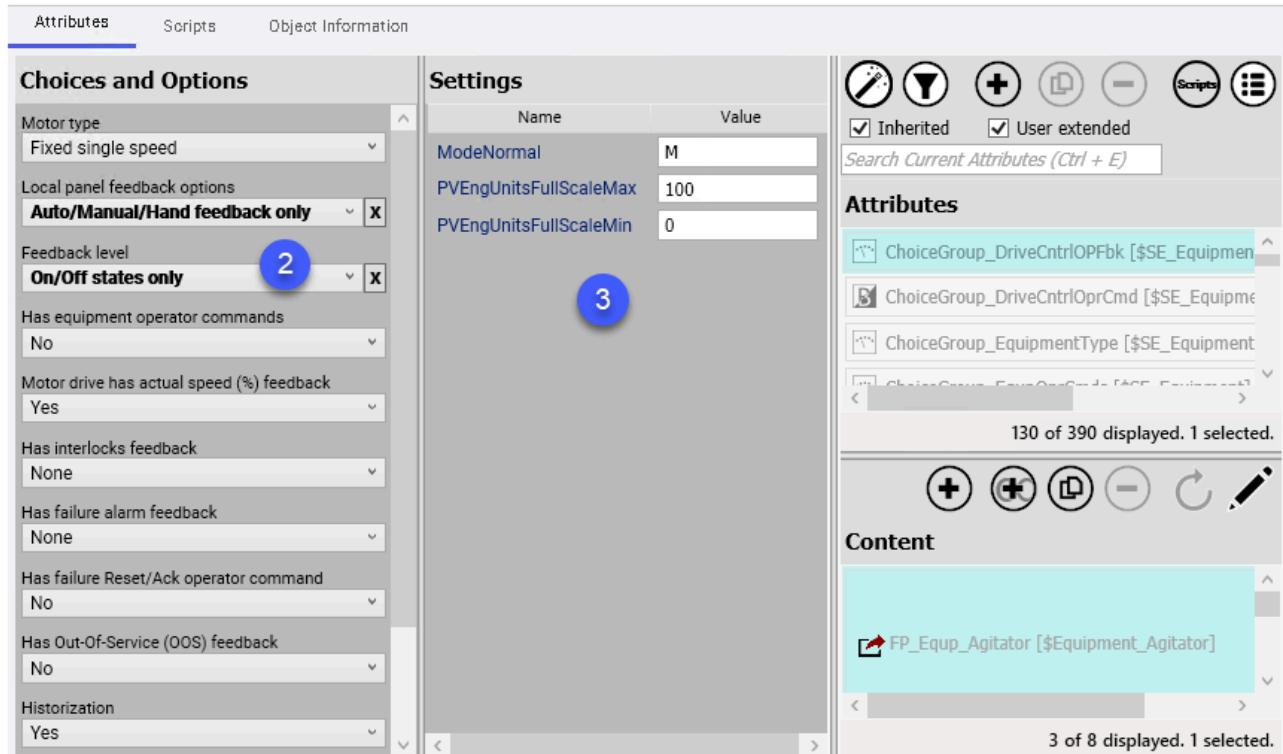
You can use the IDE to configure instances from a template with an Object Wizard, or you can graphically create instances using an editor that is started from the Graphic Editor. If you are using AVEVA OMI for your runtime environment, you can also access the editor through the ViewApp Editor or Layout Editor. See the AVEVA OMI online help for more information about these editors.

- From the IDE, the Object Wizard presents a series of prompts or questions, from which you select an answer to build a representation of a piece of field equipment, such as a pump, valve, PLC, etc.
- If you are using the Graphic Editor or one of the AVEVA OMI editors, select a graphic from the **Toolbox** tab and drag it onto an overview graphic. See [Create a new instance graphically](#) for more information.

Configure an instance with the IDE

To configure an instance using an Object Wizard in the IDE:

- In the IDE, open the instance you need to configure.
 - To create and configure a new instance:
Select the template you want to use, and derive an instance from it. Then, open the instance. See [Create an instance](#) for additional information.
 - To change the configuration of an existing instance:
Select the instance you want to use and open it. See [Create an instance](#) for additional information.
- Start at the top of the **Choices and Options pane**. Either select a Choice from the first Choice Group, or select/deselect the first Option as needed to match the physical asset you are modeling. Depending on the selections you make, attributes associated with the Choice or Option may appear in the **Attributes pane**.
 - If you change from the default Choice, the override is indicated by an **X** and the override is shown in **bold** type.
 - If you change the default state of an Option (selected or not selected), the override is indicated by an **X**.
 - You can restore a default setting by clearing the **X**.
- If an associated attribute contains configurable settings, these are listed in the **Settings pane**. Overrides are also indicated by an **X** next to the changed setting and are indicated by **bold** type.



4. Continue sequentially down the list of Choices and Options.
5. Save and close the instance, and then deploy it.

Guidelines for configuring instances with an object wizard

Keep in mind the following behaviors and guidelines when you use an Object Wizard to configure instances:

- Start at the top, with the first item of the Object Wizard.
- The answers you provide to each prompt may cause subsequent prompts to change. Therefore, if you skip selecting a choice or option, the subsequent selections you make could be changed or removed when you return to select a skipped step.
- Some prompts could be added or removed, based on the answers provided to previous prompts.
- If an attribute or graphic is associated with a selected Choice or Option, you may have the option to override certain settings.

Override default settings

When you override a default attribute or graphic setting in the Object Wizard, the override is marked by an X, and the setting and its description are displayed in bold.

The screenshot shows the AVEVA Operations Management Interface. On the left, the 'Choices and Options' tab is visible, showing settings for 'Wizard Mode' (Basic), 'Valve Type' (2-Way), and 'Limit Switches (2-Way)' (None). The main area is the 'Settings' tab, which displays a table of attributes and their values. One attribute, '2WayAttribute', has its value set to 'False' and is highlighted with a blue border. To the right of the table are buttons for 'Inherited' (checked) and 'User extended' (unchecked). Below the table is a search bar: 'Search Current Attributes (Ctrl + E)'. To the right of the search bar is the 'Attributes' section, which lists the selected attribute '2WayAttribute' with its value '2Way'. At the bottom of the interface are sections for 'Symbols' and 'Scripting'.

Name	Value
2WayAttribute.@Description	2Way
2WayAttribute	False X
2WayAttribute.#Bad value alarm	False
2WayAttribute.#History	False
2WayAttribute.#I/O	True
2WayAttribute.@Buffered	False

Settings can be different types, including Boolean, integer, string, etc. Settings can be used to turn on or off a feature for the object, such as I/O, Historization, or Alarms. Changing one setting can cause other settings to be shown or hidden.

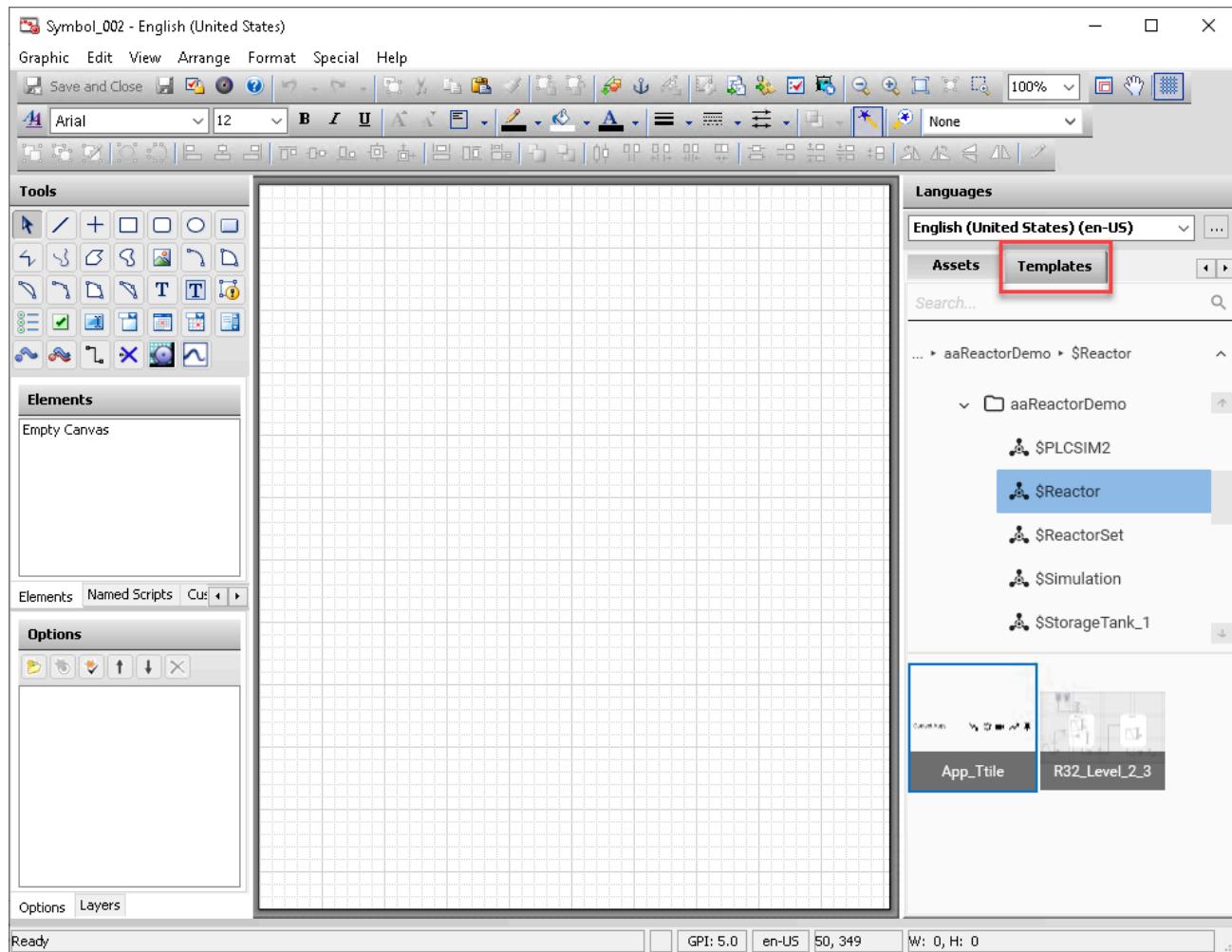
Create a new instance graphically

You can quickly and easily add and configure new instances from templates that contains an **Object Wizard** and *at least one graphic*, simply dragging a thumbnail image from the template onto the Graphic Editor canvas. Dropping the graphic thumbnail onto the canvas opens the **Configure New Asset** editor, where you can configure the Object Wizard settings. You can also configure Symbol Wizard and custom property settings, if applicable to the graphic you select. With this editor, you do not need to use the **System Platform IDE** to configure object properties, or spend a lot of time configuring graphic properties in the **Graphic Editor**.

Note: You can add an asset configured from an Object Wizard with a linked Symbol Wizard directly to an AVEVA OMI pane. However, InTouch HMI does not support adding and configuring an asset in an InTouch HMI window that has been derived from an Object Wizard with a linked Symbol Wizard. If you are using InTouch HMI, you must instead embed the symbol in an overview symbol. InTouch HMI does not support directly linking an Object Wizard to graphics in the Visualization folder that use a Symbol Wizard, such as those in the Situational Awareness Library.

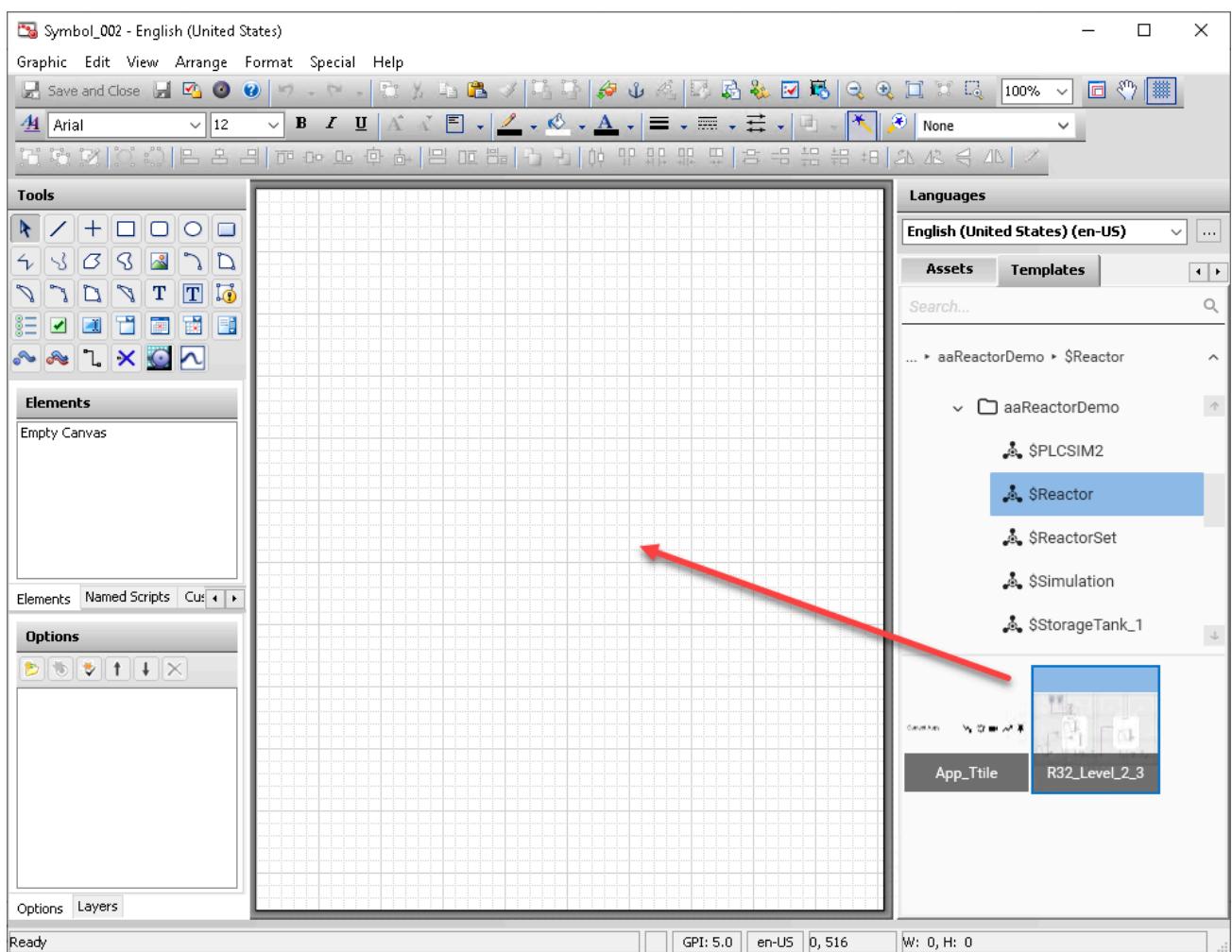
To start the **Configure New Asset** editor:

1. Select a new or existing symbol in the **Visualization** folder. The **Graphic Editor** opens.
2. If necessary, use the arrow keys until the **Templates** tab is shown, then select it.

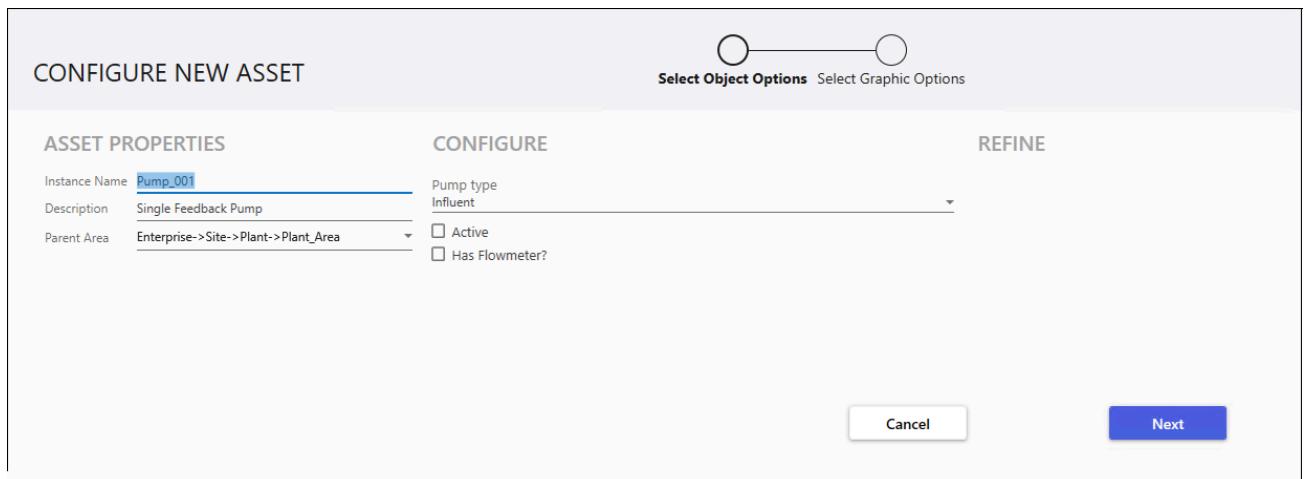


3. Navigate to the template you want to use. The template you select must contain at least one graphic.

4. Select the graphic to use and drag it onto the canvas.



5. As soon as you drop the image onto the canvas, the **Configure New Asset** editor opens to the **Select Object Options** page.



Under **Asset Properties** (left column) edit the following entries:

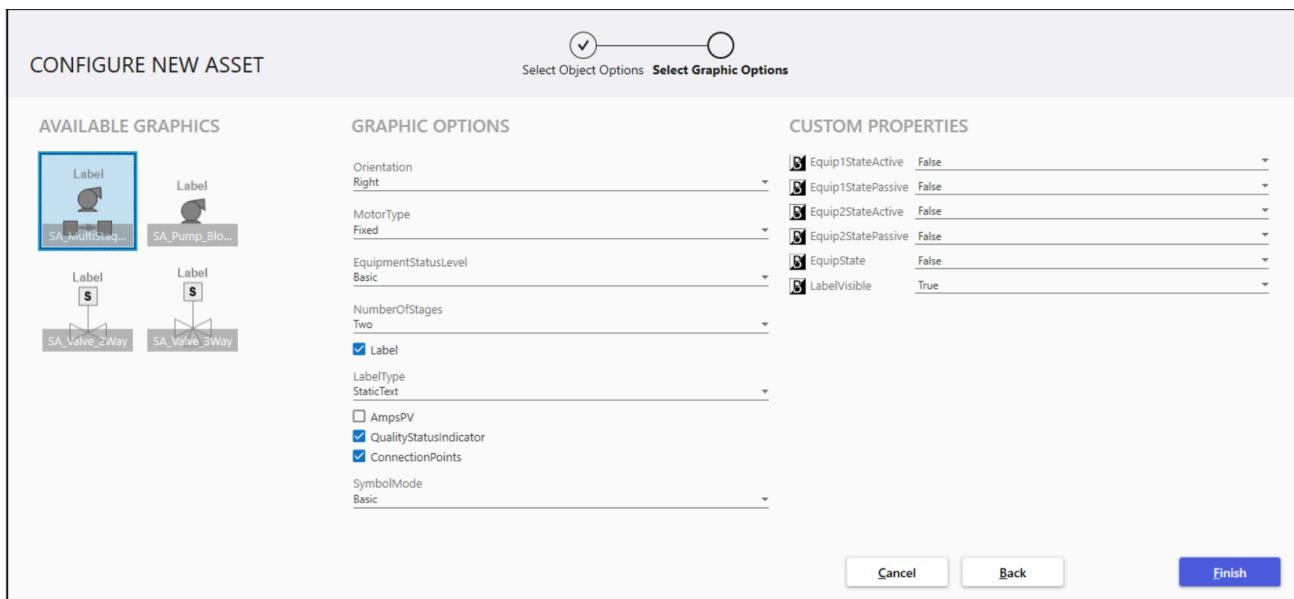
- **Instance Name** (name of the new asset). The Instance Name is the same as the object tag name. Each Instance Name must be unique within the selected **Parent Area**.

- **Description** (a description of the new instance).
- **Parent Area** (select the appropriate parent area for the new instance).
- Configure the **Object Options** (center column).
- Configure any properties listed under **Refine** (right column—there are none shown in the example above), then click **Next**.

Object options determine which graphic options can be used. Only the applicable graphic (or graphics) will be available. For example, if an object includes options that allow it to be configured as a pump or as a conveyor belt, you will not be able to add a pump graphic if you have selected a conveyor belt asset on the **Object Options** page.

See [VisualBuild Object options](#) for additional information.

6. The **Select Graphic Options** page opens. If more than one graphic is displayed, select the one to use, then configure its graphic options. Click **Finish** when you are done.



Graphics options are Symbol Wizard options. Select the Symbol Wizard options applicable to the asset you are adding to the overview symbol. Once you select the graphic options, you can configure any applicable custom properties.

7. After you click **Finish**, the **Configure New Asset** editor closes. Adjust the size and position of the graphic on the **Graphic Editor** canvas as needed. Add additional symbols and graphics as needed.
8. When you have finished adding symbols and other edits, select **Save and Close** to exit from the **Graphic Editor**.

VisualBuild Object options

The **Select Object Options** page consists of three columns for **Asset Properties**, configuration (**Configure**), and settings (**Refine**).

CONFIGURE NEW ASSET

ASSET PROPERTIES

Instance Name: Pump_001

Description: Single Feedback Pump

Parent Area: Enterprise->Site->Plant->Plant_Area

CONFIGURE

Pump type: Influent

Active

Has Flowmeter?

REFINE

Cancel

Next

The **Asset Properties** column contains settings for:

- **Instance name** (what the instance will be named within the Galaxy). The instance name is the object's tagname, and can be up to 32 alphanumeric characters, with at least one letter and no spaces. \$ cannot be used as the first character.
- **Description** (of the instance). There are no restrictions on the length or characters in the description.
- **Parent area**: This is a drop-down selection that lets you move the instance to the appropriate area of the Galaxy.

The **Configure** column contains object property selections, arranged as a series of prompts with drop-down answers. Start at the top and work down to the end. As you make selections, additional prompts may appear in the Configure area (always below the current prompt).

The **Refine** column contains additional settings that you can set. Typically, the settings will include things like upper and lower limits for scaling and alarms, on/off settings, or other items that further define settings that were selected in the **Configure** column. Some configurations do not contain any settings under the **Refine** column.

VisualBuild graphic options

The **Select Graphic Options** page consists of three columns for **Available Graphics**, **Graphic Options**, and **Custom Properties**.

CONFIGURE NEW ASSET

Select Object Options **Select Graphic Options**

AVAILABLE GRAPHICS	GRAPHIC OPTIONS	CUSTOM PROPERTIES
SA_MultiStage...	Orientation: Right MotorType: Fixed EquipmentStatusLevel: Basic NumberofStages: Two <input checked="" type="checkbox"/> Label LabelType: StaticText <input type="checkbox"/> AmpsPV <input checked="" type="checkbox"/> QualityStatusIndicator <input checked="" type="checkbox"/> ConnectionPoints SymbolMode: Basic	Equip1StateActive: False Equip1StatePassive: False Equip2StateActive: False Equip2StatePassive: False EquipState: False LabelVisible: True
SA_Pump_Blo...		
SA_Valve_2Way		
SA_Valve_3Way		

Cancel Back **Finish**

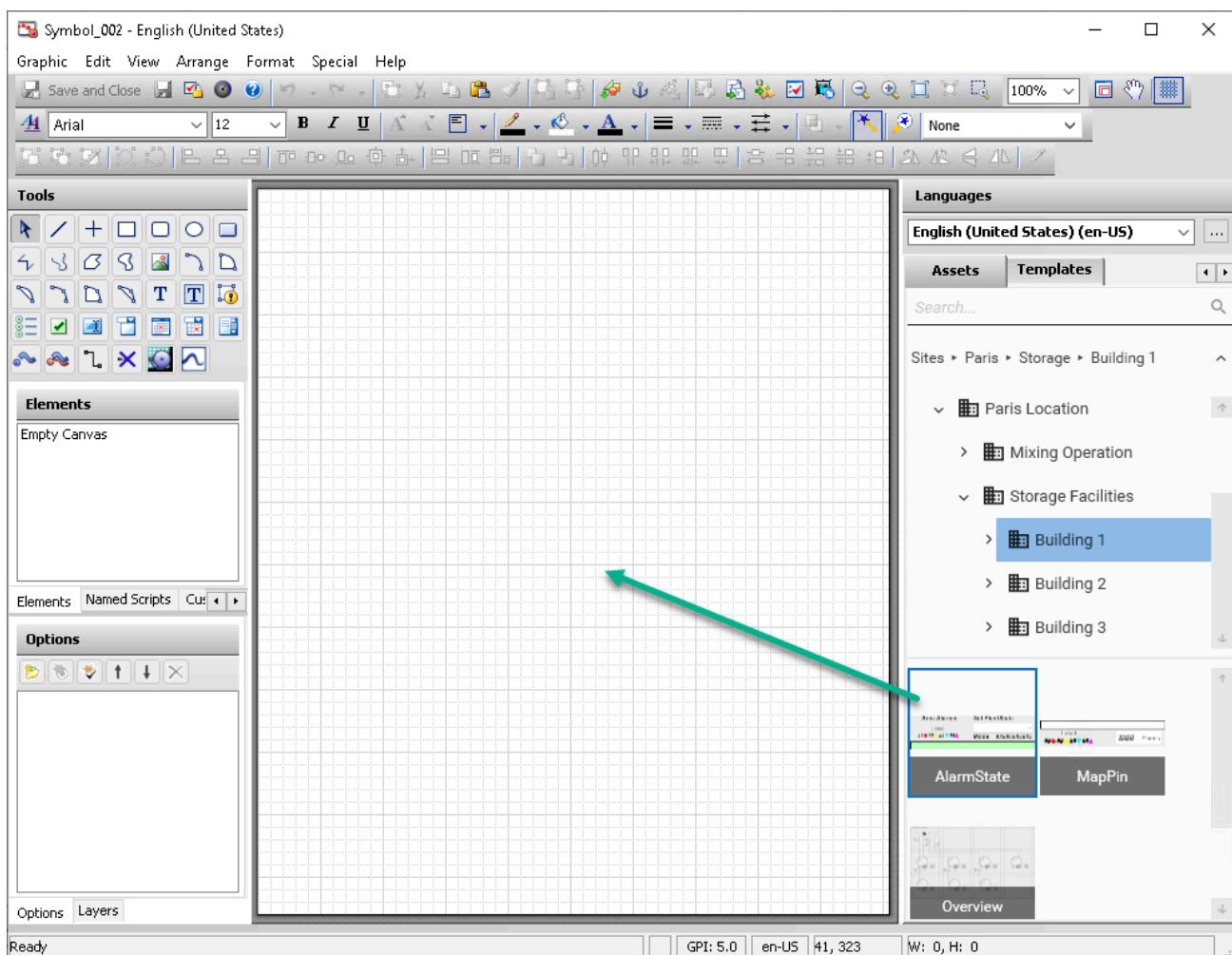
The **Available Graphics** column displays the graphic or graphics that are associated with the object as configured on the **Select Object Options** page. If more than one graphic is displayed, select the one you want to use. Only one graphic can be selected for each asset you configure.

The **Graphic Options** column contains Symbol Wizard configuration selections that apply to the graphic. Typically, these are used to control things like orientation of the graphic, whether or not it has a visible label, to add alarm borders, etc.

The **Custom Properties** column contains additional refinements to set parameters for the graphic. This includes items like alarm limits and priority levels.

Add a configured asset to an overview graphic

1. To add a graphic from a configured instance, select the **Assets** tab in the **Graphic Editor**.



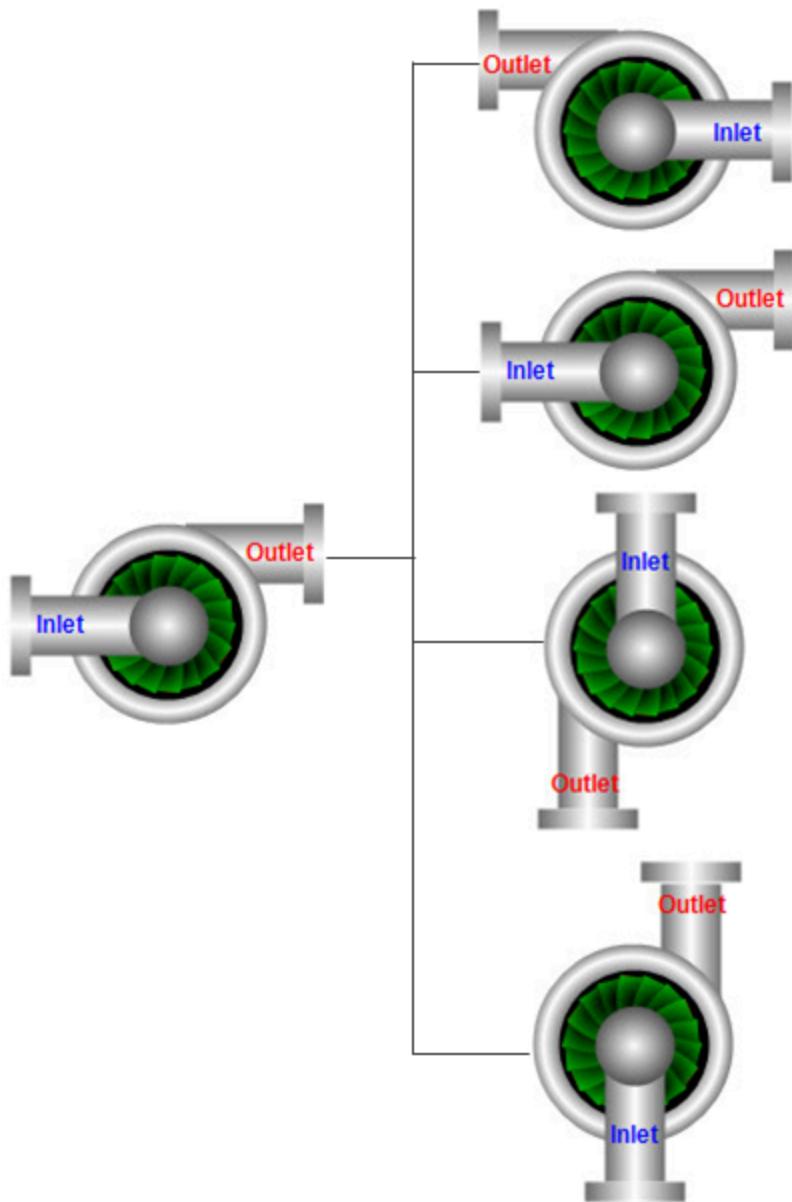
2. Navigate to the asset you want to add to the open graphic and drag it onto the canvas.
3. Adjust the size and position of the graphic on the **Graphic Editor** canvas as needed. Add additional graphics as needed.
4. When you have finished adding graphics and other edits, select **Save and Close** to exit from the **Graphic Editor**.

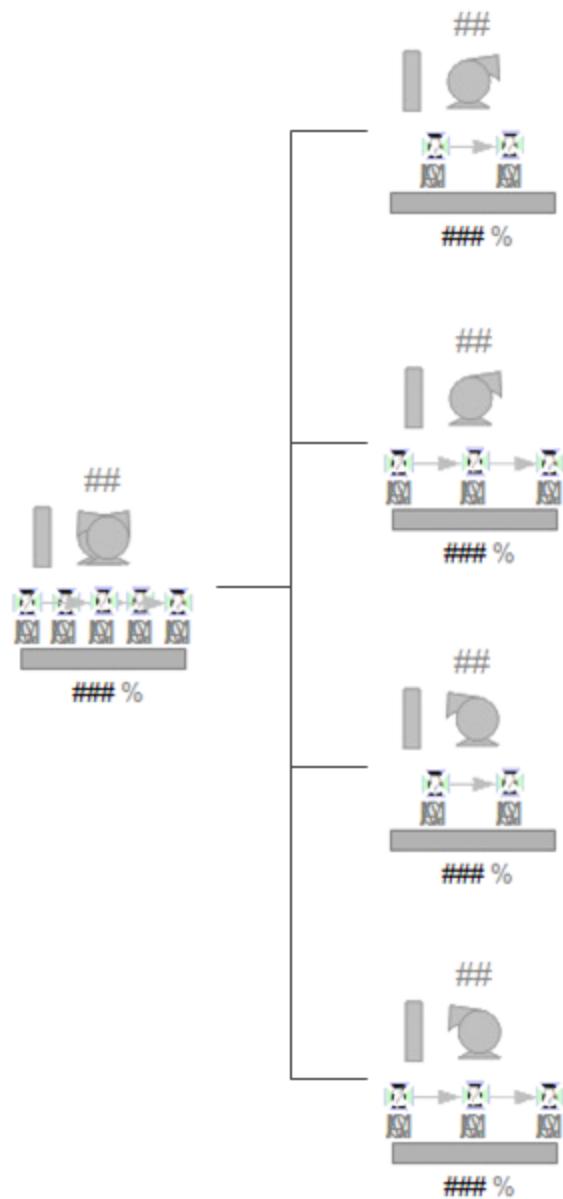
Note: InTouch HMI does not support adding a graphic derived from a Symbol Wizard linked to an object wizard directly onto an InTouch window. The graphic must be contained in an overview graphic.

Symbol Wizard Editor

The Industrial Graphic Editor includes the Symbol Wizard Editor, which can be used to create graphics containing multiple visual or functional configurations called Symbol Wizards. A Symbol Wizard can be embedded into managed InTouch applications like a standard Industrial graphic.

A Symbol Wizard's configuration is selected to meet the requirements of an application. Incorporating multiple configurations in a single Symbol Wizard reduces the number of graphics needed to develop an AVEVA application.





The Symbol Wizard Editor can create Symbol Wizards from traditional Industrial Graphics and Situational Awareness Library graphics. Both types of graphics are located in the Visualization folder in separate toolkit folders. Symbol Wizards are saved in the IDE's Visualization folder and are not associated with any specific AVEVA object template or object instance. Except for the ability to select a specific graphic configuration, Symbol Wizards behave like standard Industrial Graphics.

Situational Awareness Library graphics provide an additional benefit of including defined properties and their associated attributes to more easily create configurations. Some Situational Awareness Library graphics include a Type property to assign a specific function for a graphic configuration. For example, a meter graphic can be configured to represent a thermometer, a pressure meter, or flow meter by simply changing the attribute assigned to the Type property.

Typically, the process of creating and embedding a Symbol Wizard in an application requires the involvement of a Designer and a Consumer. A Designer creates Symbol Wizards using the Symbol Wizard Editor. A Consumer selects a configuration of a Symbol Wizard and embeds the instance of the graphic into a managed InTouch application.

The topics in this section include some general information about creating Symbol Wizards and using them with applications and Object Wizards. For more details and instructions on using Symbol Wizards, see "Working with Symbol Wizards" in the Industrial Graphic Editor help.

Create Symbol Wizards

A Designer uses the Symbol Wizard Editor to define the various required graphic configurations based on a set of rules and graphic layers. A Designer defines a set of layers, which are used to group a set of graphic elements, custom properties, and named scripts. Graphic elements and other properties can be assigned to no layers or multiple layers. Graphic elements that are not assigned to any layer always appear in all configurations.

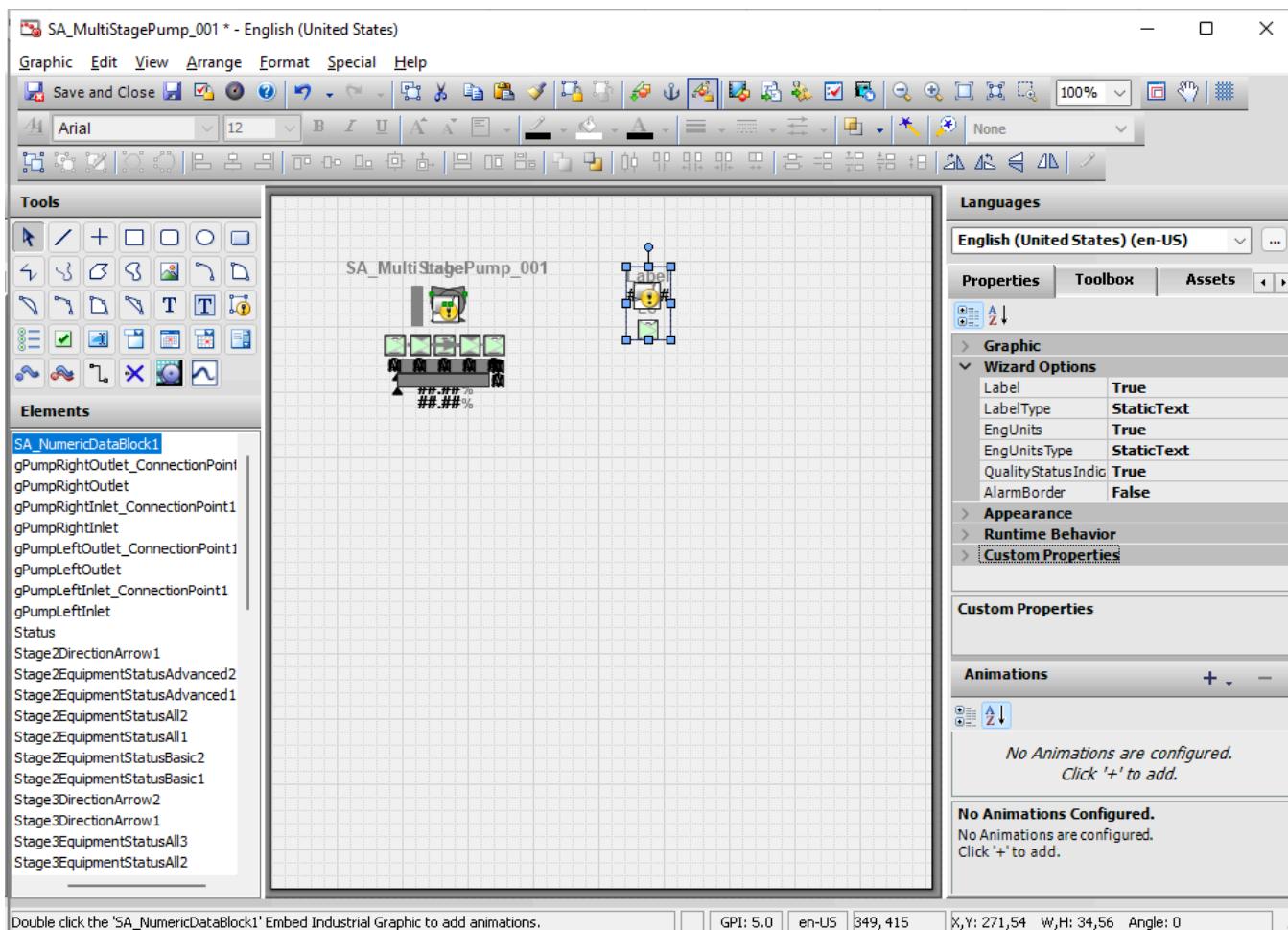
A Designer can create a rule for each layer that defines the conditions when the layer is included in a graphic configuration. Rules are assigned with Choice Groups, Choices, and Options. A Designer selects a configuration to be the graphic default that appears when the graphic is embedded in a managed InTouch application.

After creating all configurations, the Designer verifies each configuration of a graphic using the Symbol Wizard Preview. Designers set values in the **Wizard Options** view to verify that each configuration appears as designed based on the layer rules set for the graphic.

When a Symbol Wizard is finished, the Designer saves it to the Galaxy library so that it is available for use in InTouch and OMI applications.

Embed a Symbol Wizard into an application

Symbol wizards are stored in a Galaxy library just like standard Industrial Graphics. When a Consumer selects a symbol wizard and embeds it into a InTouch or OMI application, the default configuration of the Symbol Wizard is selected. The Consumer can change the graphic's configuration by changing the options from the **Wizard Options** section of the **Properties** view. Depending on the selected configuration, there can be additional configuration-related properties that can be selected by the consumer.



After selecting a graphic configuration and changing any properties, the Consumer saves the Symbol Wizard so that it can be embedded into a window from WindowMaker or into an OMI pane.

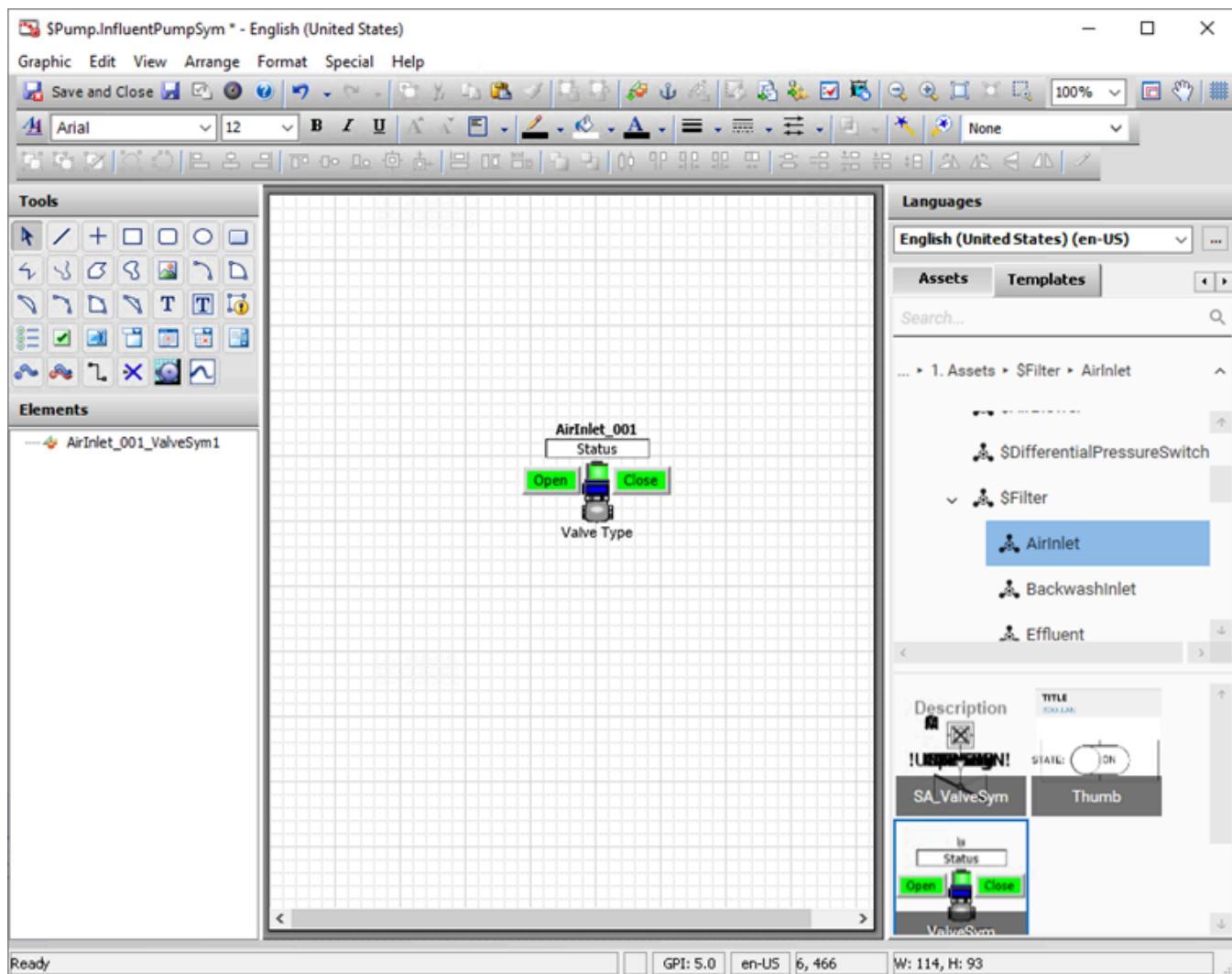
While the InTouch or OMI application is running, the Symbol Wizard appears as the configuration selected by the Consumer. A Symbol Wizard configuration cannot be changed during runtime.

For more information about how to embed Symbol Wizards into InTouch applications, see [Using Symbol Wizards in an Application](#) in the Industrial Graphic Editor help.

Integrate Symbol Wizards with Object Wizards

Symbol Wizards can be combined with object wizards to give users a simple configuration workflow when adding new assets to an overview graphic. The *Application Server User Guide* describes how to build an object wizard that can be added to any object template. Then, you can associate object wizard choices with different configuration values to be applied to a linked Symbol Wizard.

Users access Object Wizards through the **Templates** tab of the **Graphic Editor** to derive instances. The configured instance is placed on the overview graphic. For detailed instructions about using Symbol Wizards with object wizards, see the topic "Create a New Instance with the Configure New Asset Editor" in the *Application Server User Guide*.



Note: AVEVA OMI lets you add a configured asset directly to a pane through the Layout Editor. However, InTouch HMI does not support adding a graphic derived from a Symbol Wizard linked to an object wizard directly onto an InTouch window. The graphic must be embedded in another graphic (for example, contained in an overview graphic) before you add it to an InTouch window.

About the galaxy style library

A Galaxy Style Library applies a set of visual and numeric styles to ViewApps in the current Galaxy. Styles can be set for text, lines, fill colors, outlines, etc. The Galaxy Style Library contains the following style elements:

- The **Quality and Status** tab sets the physical appearance of quality and status icons. Also, the **Quality and Status** tab includes a Status group to assign the visual characteristics to error condition status icons.
- The **HMI Element** tab sets the appearance of various HMI aspects, such as background color colors, bad values, default fonts, etc.
- The **Alarm Element** tab sets the appearance of alarms.
- The **Trend Element** tab sets the appearance of the pens and background grid used to show historical trends.
- The **User defined** tab can be used to create custom style elements.

- The **Format** tab specifies the format of numbers that appear in a ViewApp.
- The **Special characters** tab sets characters to be shown at runtime for bad quality/no value and values too large for a fixed field.

The screenshot shows the 'Format' tab selected in the top navigation bar. On the left, a sidebar lists various data types: Real, FixedDecimal, Integer, Exponential, Hex, Binary, UserDefined1 through UserDefined10. 'Real' is currently selected. The main panel displays settings for 'Real' data. It includes a dropdown menu set to 'Real', a checkbox for 'Enable fixed width' which is unchecked, and a table for defining 'Value range' and 'Precision'. The table rows are as follows:

Value range	Precision
Less than 1 (from 0 to 0.999)	4
In the ones (from 1 to 9)	2
In the tens (from 10 to 99)	2
In the hundreds (from 100 to 999)	2
In the thousands (from 1,000 to 9,999)	2
In the ten thousands (from 10,000 to 99,999)	0
In the hundred thousands (from 100,000 to 999,999)	0
In the millions (from 1,000,000 to 9,999,999)	0
In the ten millions (from 10,000,000 to 99,999,999)	0
One hundred million or greater (from 100,000,000 on)	0

Below the table are two buttons: 'Reset settings' and 'Reset to default'.

Open the galaxy style library

- Select the **Galaxy** tab.
- Select the **Configure** option, then select **Galaxy**.
- Select the **Styles** option.

Manage galaxy style libraries

To manage your collection of Galaxy Style Libraries, select the vertical ellipse above the **Galaxy Style Libraries** field to show a list of management commands.

- Make default (enabled only if more than one Galaxy Style Library is available)
- Rename
- Duplicate

- Export

The screenshot shows the 'Configure' screen for a Galaxy. On the left, there's a sidebar with icons for Security, Galaxy, Styles (which is selected), Alarms, and Languages. The main area shows 'Standard_Style (Default)' and 'Standard_Style1'. A context menu is open for 'Standard_Style1', listing options like 'Make default', 'Rename', 'Duplicate', 'Export', 'Trend Element', and 'User'. At the bottom, there's a checkbox for 'Enable Quality and Status Display'.

Duplicate a galaxy style library

You can duplicate an existing Galaxy Style Library to add it to the list of other Galaxy Style Libraries defined for the Galaxy.

To duplicate a galaxy style library

1. Select the Galaxy Style Library to be duplicated.
A blue background indicates the Galaxy Style Library is selected.
2. Select the vertical ellipse below the Galaxy Style name.
3. Select the **Duplicate** command from the context menu.
4. Click **Save** to save the duplicated library.

The duplicated Galaxy Style Library appears with a default name of StandardStyle n , where n is a single digit. A blue dot next to the Galaxy Style is shown until the new style is saved.

Set the default galaxy style library

The default Galaxy Style Library shown from the list is active when a ViewApp starts in a Galaxy.

To set a default galaxy style library

1. Select the Galaxy Style Library to be set as the default.
A blue background indicates the Galaxy Style Library is selected.
2. Select the vertical ellipse below the Galaxy Style name.
3. Select the **Make default** command from the context menu.
4. Default is shown in parentheses next to the name of the Galaxy Style Library, and no longer appears next to the previous default library name.

Rename a galaxy style library

Galaxy Style Libraries should be named using a convention that suggests the theme of the element styles or their purpose. Also, a new Galaxy Style Library should be renamed from its default name after it has been added.

To rename a galaxy style library

1. Select the Galaxy Style Library to rename.

A blue background indicates the Galaxy Style Library is selected.

2. Select the vertical ellipse below the Galaxy Style name.

3. Select the **Rename** command from the context menu.

4. Enter the new name for the library.

A warning message is shown if you type an invalid name.

5. Click in the Galaxy Style Libraries area to deselect the Galaxy Style Library and verify if you typed a valid name. You can also press **Enter** to validate the new name.

Export a galaxy style library

A single Galaxy Style Library can be exported as an xml file to a designated folder location.

To export a galaxy style library

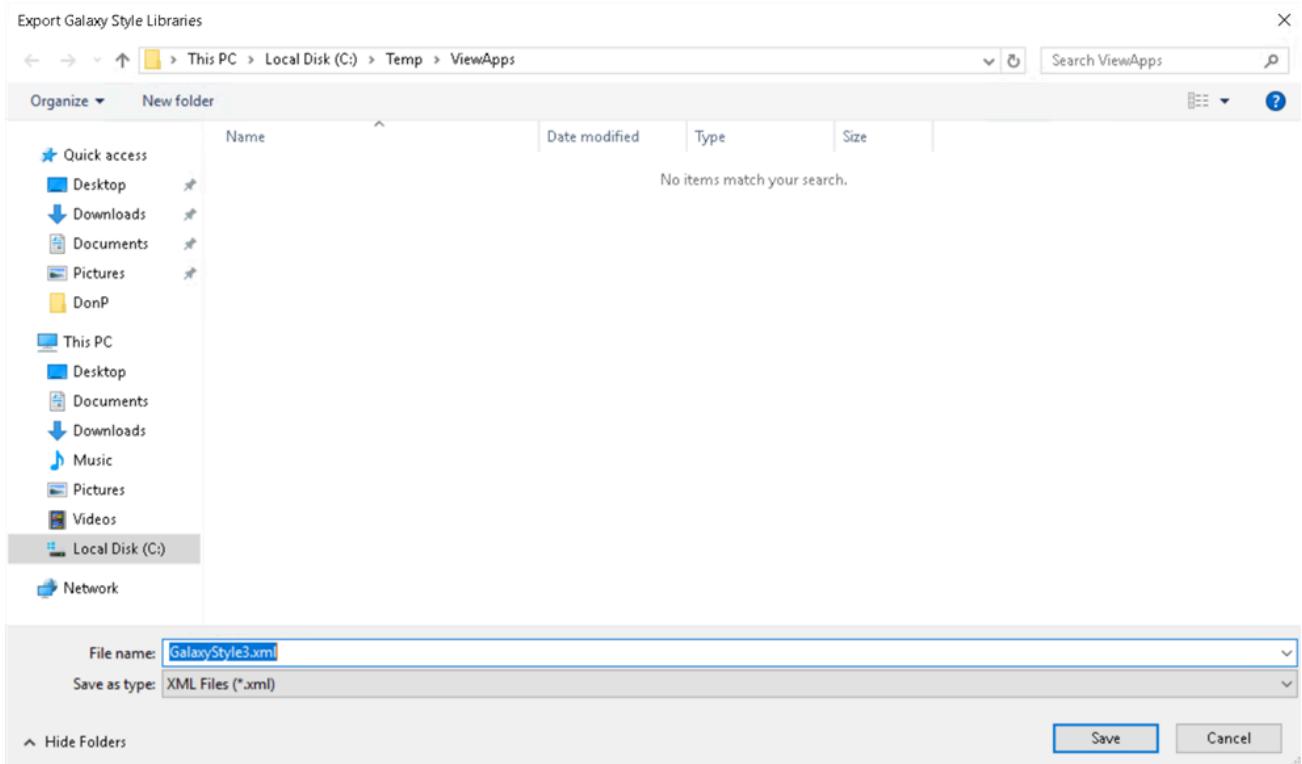
1. Select the Galaxy Style Library you want to export.

A blue background indicates the Galaxy Style Library is selected.

2. Select the vertical ellipse below the Galaxy Style name.

3. Select the **Export** command from the context menu.

4. In the **Export Galaxy Style Libraries** dialog, select the folder where you want to save the xml file.



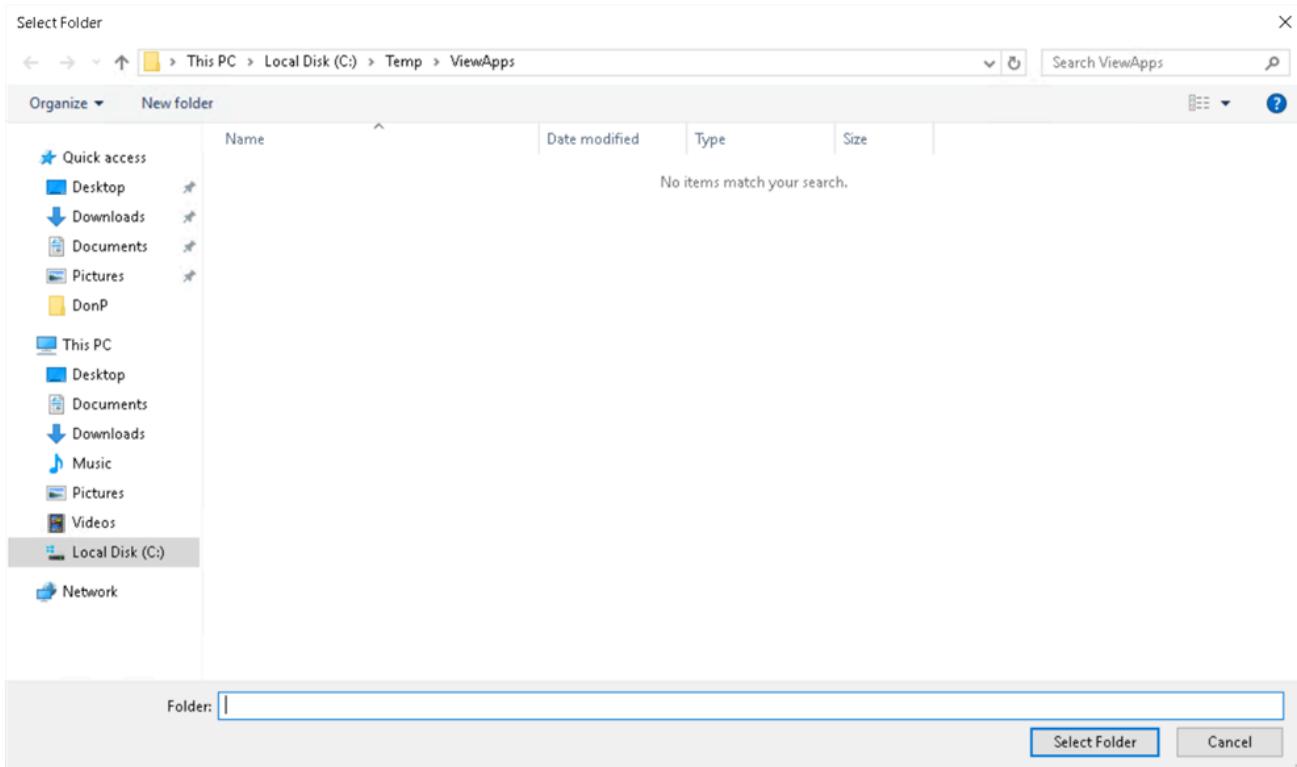
5. The default name of the exported xml file is the name of the exported Galaxy Style Library. You can edit this in the dialog.
6. Click **Save**.

Export all galaxy style libraries

All Galaxies shown in the **Galaxy Style Libraries** list can be exported together to a common folder location.

To export all galaxy style libraries

1. Select the **Galaxy** tab in the System Platform IDE.
2. Select the **Export** option, then select **All Style libraries**.
The **Export all Styles** dialog opens.
3. Select the folder where you want to save the exported xml files and click **Select Folder**.



An xml file for each Galaxy Style Library in the Galaxy is created in the folder you selected. The name of each exported xml file is the name of the exported Galaxy Style Library.

Import a galaxy style library

You can import a Galaxy Style Library from an XML file.

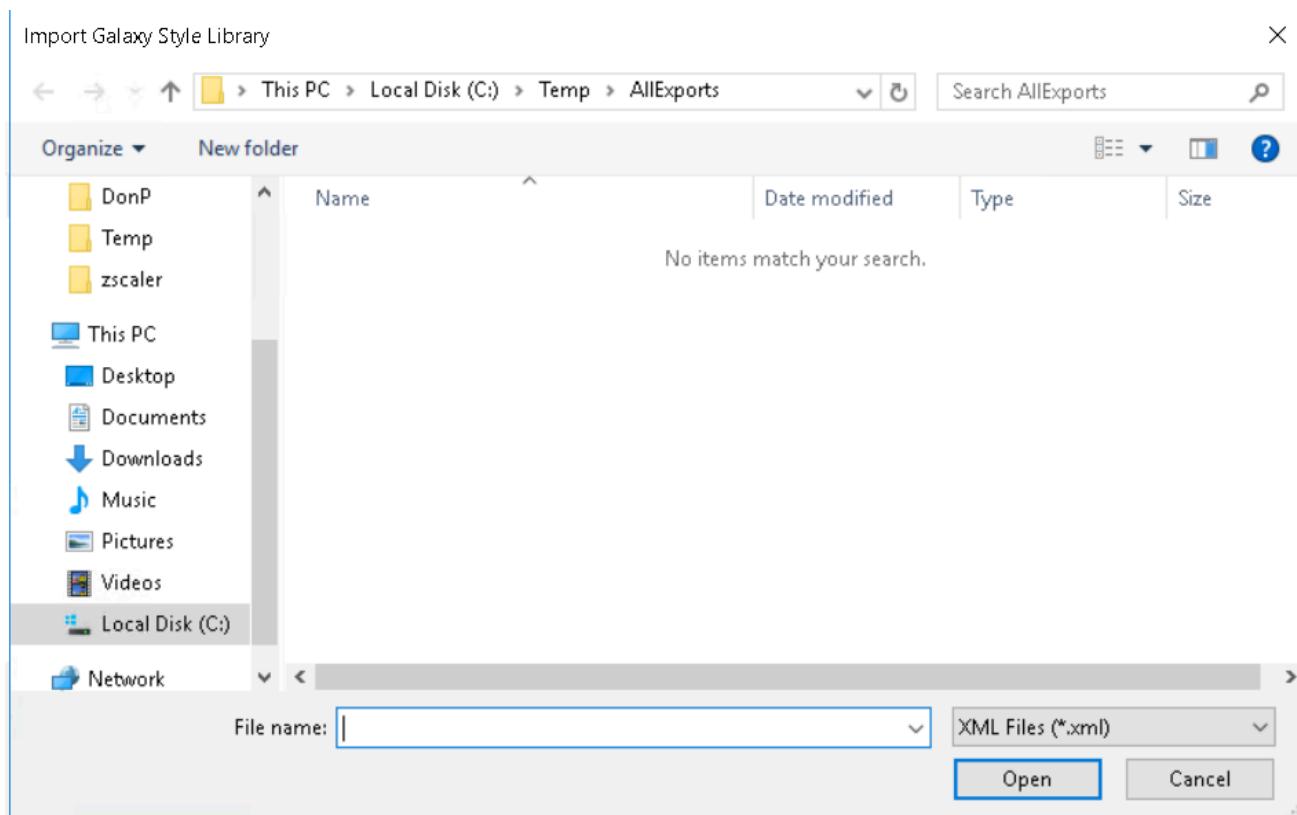
Optional Galaxy Style Library XML files are installed with System Platform. These optional Galaxy Style Libraries are located in the ...\\Program Files (x86)\\ArchestrA\\FrameWork\\Bin\\AdditionalElementStyles folder.

The names of the optional Galaxy Style Library XML files suggest the primary color schemes of the element styles within each optional Galaxy Style Library.

To import a galaxy style library

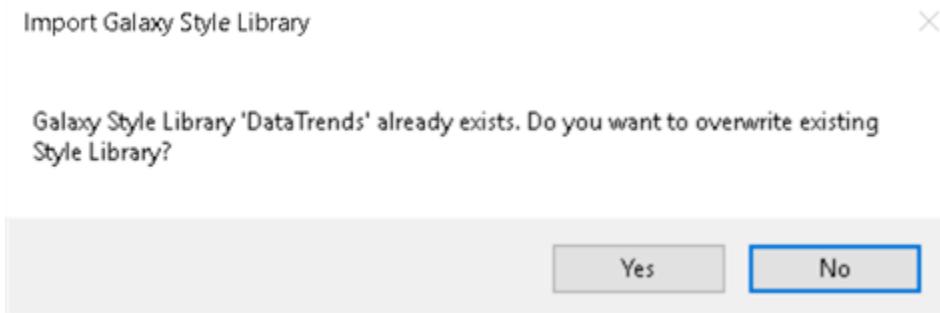
1. Select the **Galaxy** tab in the System Platform IDE.
2. Select the **Import** option, then select **Style Libraries**.

The **Import Style Libraries** dialog opens.



3. Select the folder containing the Galaxy Style Library to be imported.
4. Select the Galaxy Style Library xml file you want to import, then click **Open**. You can select multiple files for import.

If the Galaxy Style Library to be imported is a duplicate of a Galaxy Style Library that is already listed, a warning message appears with options to overwrite the existing Galaxy Style Library or not.



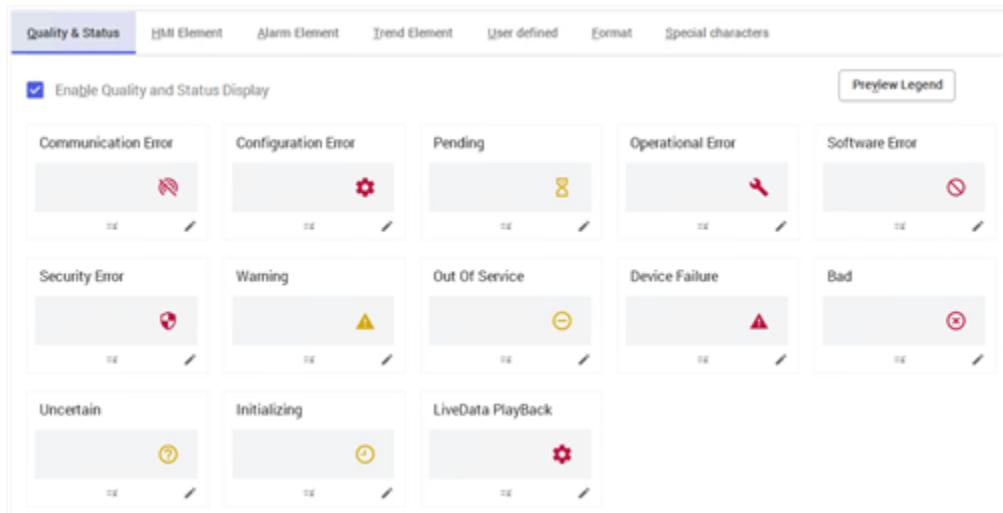
After successfully importing a Galaxy Style Library, its name appears in the **Galaxy Style Libraries** list.

About the quality and status tab

Data quality and status can be represented by a set of icons that appear in a running ViewApp. The physical appearance of icons indicates current quality and status of the data associated with an icon. The **Quality and Status** tab shows a list of status style overrides that can be modified to set the visual characteristics of quality and status states at runtime.

Quality and Status (QS) indicators are listed by order of priority (left to right, with the first row listing the highest

priority indicators). At runtime, if more than one QS indicator is active, only the highest priority indicator is shown.



The appearance of text, fill, line, outline, blink can be set as overrides of the following **Quality and Status** icons:

- Communication Error (highest priority)
- Configuration Error
- Pending
- Operational Error
- Software Error
- Security Error
- Warning
- Out of Service
- Device Failure
- Bad
- Uncertain
- Initializing
- LiveData Playback (lowest priority)

Show data status

To show a specified status or quality at runtime, you can:

- Use a Status element that shows an icon. It indicates the status or quality of specified attributes directly or those used indirectly in elements.
- Change the appearance of animated elements based on the status and quality of AVEVA attributes they use.

Show quality and status with the status element

The Status element cannot monitor attributes of:

- Elements that are not in the same hierarchy level in the Elements List.
- Elements that use the attributes in scripts.
- Elements that are invisible at run time.

For more information on how to configure status on an element, see [Configuring Animation for a Status Element](#).

For more information on how to configure the appearance of a status element, see [Setting the Appearance of a Status Element](#).

Show quality and status by overriding

You can override the appearance of an animation depending on its configured attributes by:

- Overriding the animation or changing the appearance of the element.
- Drawing an outline around the element.

Overriding the appearance of animations also applies to:

- Elements contained in groups.
- Elements in graphics embedded in other graphics.

Overriding the appearance of animations does not apply to:

- Elements that use the monitored attribute in scripts.
- Elements that are invisible at run time.

For more information, see [Overriding Element Appearance Depending on Quality and Status of its Attributes](#).

Quality and status styles

You can configure your graphic to show non-good status and quality of attributes in different ways:

- A status element shows a specific icon depending on the quality and status of configured attributes or elements. For more information see [Use status elements](#).
- The text, fill, or line appearance of elements can be overridden depending on the quality and status of the attributes they reference.
- Elements are drawn with an outline depending on the quality and status of the attributes they reference.

Use status elements

Status elements show a specified graphic depending on the quality and status of:

- Attributes and tags configured for specific animated elements.
- One or more specified attributes or tags.

You can assign status elements to an animation in three steps:

1. Draw the status element on the canvas.
2. Associate the status element with animated elements on the canvas and/or attributes that provide the quality and status data to be monitored.
3. If needed, configure the appearance of the status element.

Draw the Status Element on the Canvas

You can easily place a status element on the canvas to show an icon that indicates quality and status of attributes or tags contained in selected animated elements and/or specified attributes and tags.

You do this as you would with any other element. For more information, see [Drawing and Dragging Elements](#).

Configure the Status Element

You can associate the status element with:

- Animated elements that use attributes and tags that provide the quality and status that is to be monitored.
- Attributes and tags that provide the quality and status to be monitored.

In both cases, the appearance is set by the settings in your application. For example, in AVEVA ViewApps it is set in the **Quality and Status** tab of the **Configure Galaxy Style Library** dialog box.

For more information on how to configure this animation, see [Configuring Animation for a Status Element](#).

Set the Appearance of a Status Element

You can set the appearance of a status element depending on the quality and status of its referenced attributes and tags and/or attributes and tags used in its referenced elements.

You can also preview the appearance of a status element. For more information, see [Previewing all Status Appearances](#).

You can reset the appearance of a status element to its default. For more information, see [Reset an Override Appearance to its Default](#).

Override Element Appearance in Accordance with the Quality and Status of its Attributes

You can configure any animated element to appear differently depending on the quality and status of its associated attributes and tags if your HMI application supports it.

For animated elements, you can:

- Override the appearance of the text font, style, and blinking.
- Override the appearance of the fill style and blinking.
- Override the appearance of the line style, weight, pattern, and blinking.
- Preview all status appearances in one dialog box.
- Reset the status appearances to their defaults.
- Use an outline to indicate a specified status or quality.

Note: Instead of overriding the appearance of elements on the canvas, you can use a status element. The

status element shows an icon representing quality and status of monitored attributes and tags.

Override the Text Appearance of Elements to Indicate Non-Good Status or Quality

If your application supports it, you can configure the text appearance of animated elements with attributes that have non-good status or quality are overridden with a specific text appearance.

Override the Fill Appearance of Elements to Indicate Non-Good Status or Quality

If your HMI application supports it, you can configure the fill appearance of animated elements with attributes that have non-good status or quality are overridden with a specific fill appearance.

Override the Line Appearance of Elements to Indicate Non-Good Status or Quality

You can configure a Galaxy so that the line appearance of animated elements with attributes that have non-good status or quality are overridden with a specific line appearance.

To override the line appearance of elements specified by a Status element

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select **Enable Quality and Status Display**.
5. Select the edit icon  for a status or quality.
6. Click the **Line** tab.
7. To override the line pattern, select the **Line Pattern** checkbox and select a line pattern from the adjacent list.
8. To override the line weight, select the **Line Weight** checkbox and type a new line weight in the adjacent box. Alternatively, you can increment or decrement the line weight by clicking the adjacent boxes.
9. To override the line color:
 - a. Select **Line Color** checkbox.
 - b. Click the color box.
 - c. Select a line color from the **Select Line Color** dialog box. For more information, see [Setting Style](#).
10. To override the line blink behavior:
 - a. Select **Blink**.
 - b. Select a blinking speed from the **Speed** list (Slow, Medium, or Fast).
 - c. Click the color box.
 - d. Select a line blink color from the **Select Blink Color** dialog box. For more information, see [Setting Style](#).
11. Click **Apply**.

Add Outlines to Elements to Indicate Non-Good Status or Quality

You can configure the Galaxy so that animated elements with attributes that have non-good status or quality are shown with an outline.

To add outlines to elements to indicate non-good status or quality

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select **Enable Quality and Status Display**.
5. Click the **Outline (OI)** tab.
6. Select the edit icon  for a status or quality.
7. Select **Show Outline**.
8. To set the line pattern, select **Line Pattern** and select a line pattern from the dropdown list.
9. To set the line weight, select **Line Weight** and enter a line weight. Alternatively, you can increment or decrement the line weight by clicking the adjacent boxes.
10. To set the line style:
 - a. Click the color box next to **Line Color**.
 - b. Select a line color from the **Select Line Color** dialog box. For more information, see [Setting Style](#).
11. To set the line blink behavior:
 - a. Select **Blink**.
 - b. Select a blinking speed from the **Speed** list (Slow, Medium, or Fast).
 - c. Click the color box.
 - d. Select a line blink color from the **Select Blink Color** dialog box. For more information, see [Setting Style](#).
12. Click **Apply**.

Preview all Status Appearances

You can preview the appearance of all status overrides by showing the **Status Legend** dialog box.

To preview all override appearances

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Click **Preview Legend**. The **Status Legend** dialog box appears.
5. Click **Close**.

Reset an Override Appearance to its Default

For any status, you can reset the default appearance:

- Text, fill, and line overrides.
- Outline settings.
- Status element settings.

To reset a status or quality to its default appearance

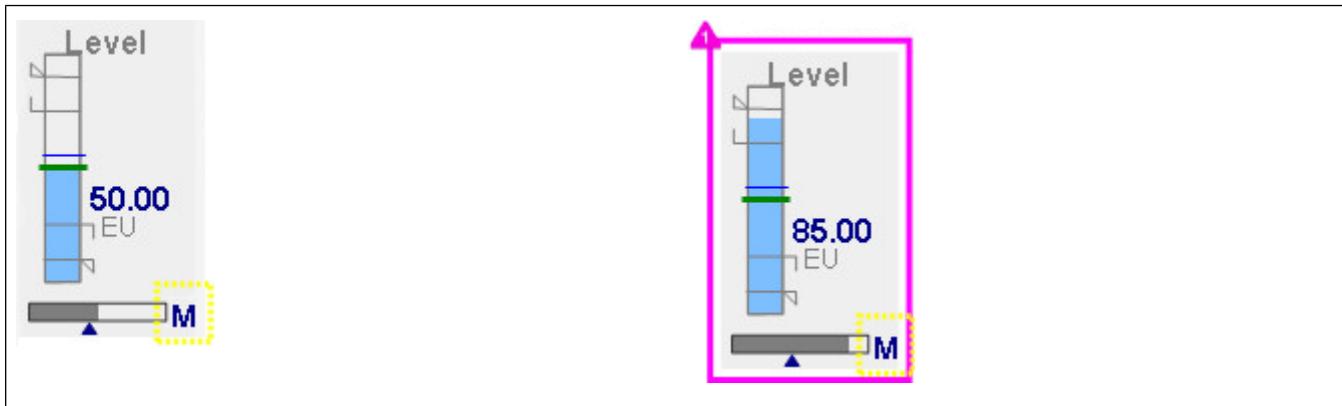
1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select the edit icon  for a status or quality.
5. Click **Reset to default**. All text, fill, and line overrides, status element icons, and outline settings are reset to their defaults.

About element styles

An element style defines a set of visual properties that determine the appearance of text, lines, graphic outlines, and interior fill shown in Industrial Graphics or graphic elements. An element style can be applied to a graphic that sets pre-configured visual property values that take precedence over a graphic's native visual properties.

Element styles provide the means for developers to establish consistent visual standards in their AVEVA OMI ViewApps. An element style can define the same visual properties of text, lines, fill, and outlines for all graphics included in a ViewApp.

Likewise, element styles can be applied to a graphic during runtime to show the changing status of a piece of equipment in response to events or alarms. The following example shows an alarm border animation element style around a meter when the process value shown by the meter transitions into an alarm state.



Element styles details

A Galaxy Style Library includes pre-defined and user-defined element styles, plus numerical format styles and special characters used to show exceptions at runtime.

- Pre-defined element styles

Pre-defined element styles have been configured with default values. You can modify each element style's font, fill, line, and outline properties.

Pre-defined element styles are organized in groups belonging to the **Element Style Overrides** list by their functional purpose:

- **Quality and Status**
- **HMI Elements**

- **Alarm Elements**
- **Trend Elements**

Many of the element styles apply to the graphic elements that comprise Situational Awareness Library graphics. An element style can define the visual properties of a graphic like its background color. An element style can also indicate a graphic's current operating state. For example, there is a set of pre-defined element styles that set the visual characteristics of machinery alarm states represented by a graphic.

- User-defined element styles

User-defined element styles do not contain preset font, fill, line, and outline properties. You must specify each pre-defined element style's properties to meet the needs of your applications.

- **Format** styles are applied to numerical values, such as set points and process variables. There are pre-defined styles for real, Fixed Decimal, Integer, Exponential, Hex and Binary numbers. You can also configure user-defined formats. Format styles define range and precision for values.
- **Special characters** define two different special characters that are shown in alarm mode at runtime for the following conditions:
 - Bad quality with no value: default character is "!" (exclamation point)
 - Value too large for fixed field: default character is "*" (asterisk)

You can create your own Galaxy Style Library by overriding the default style configurations and exporting the library. For information about setting element style overrides, see Change the visual properties of an element style (,).

If you want to restore the original default visual properties for your graphic elements, each of the styles pages includes a **Reset to Default** button. The UI Theme page does not have a button to restore the default setting.

Format Styles tab

The **Style Library** page of the backstage includes the **Format Styles** tab. **Format Styles** provides options to individually configure Galaxy-wide styles for common types of numbers used in industrial applications.

Each global number style is assigned a unique name, which cannot be changed. A number style can be applied by name in design time and runtime for an analog data type in User Input and Value Display animations. Also, grouped elements support global number styles.

The screenshot shows the 'Format styles' configuration dialog. At the top, there's a ribbon with tabs: Quality & Status, HMI Element, Alarm Element, Trend Element, User defined, Format (which is selected and highlighted in blue), and Special characters. Below the tabs, the title 'Format styles' is repeated. On the left, a vertical list of numeric format types is shown: Real, FixedDecimal, Integer, Exponential, Hex, Binary, UserDefined1 through UserDefined10. 'Real' is currently selected and highlighted in light blue. To the right of this list is a main panel titled 'Format styles'. It includes a dropdown menu labeled 'Formatting as: Real'. Below it is a checkbox labeled 'Enable fixed width'. The main area is a table with two columns: 'Value range' and 'Precision'. The rows define ranges for different numerical values, each with a precision of 4 digits. The table has 10 rows corresponding to the user-defined formats from UserDefined1 to UserDefined10. At the bottom of the main panel are 'Reset settings' and 'Reset to default' buttons. At the very bottom of the dialog are 'Cancel' and 'Save' buttons.

Configure a global numeric format

You can configure a global number style by changing the values assigned to the properties for each type of number.

To configure a global numeric format style

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Format** tab to show numeric format options.

Format styles

Value range	Precision	Up	Down
Less than 1 (from 0 to 0.999)	4	▲	▼
In the ones (from 1 to 9)	2	▲	▼
In the tens (from 10 to 99)	2	▲	▼
In the hundreds (from 100 to 999)	2	▲	▼
In the thousands (from 1,000 to 9,999)	2	▲	▼
In the ten thousands (from 10,000 to 99,999)	0	▲	▼
In the hundred thousands (from 100,000 to 999,999)	0	▲	▼
In the millions (from 1,000,000 to 9,999,999)	0	▲	▼
In the ten millions (from 10,000,000 to 99,999,999)	0	▲	▼
One hundred million or greater (from 100,000,000 on)	0	▲	▼

Format styles

Formatting as:
Real

Enable fixed width

Reset settings

Reset to default

Cancel Save

- Select a numeric format style from the **Format styles** list.

The **Format** tab updates to show options for **Fixed Width**, **Precision**, and/or **Bits From**, or **To** based on the selected format style.

- Update the fields shown for the selected format style.
- Click **Save** to save your changes.

Format list

The **Format** list includes a set of styles that can be individually configured to create customized number styles. The Real number format is the default for the user defined styles.

Based on the selected number style, the following fields appear on the **Format** dialog box to change the default properties.

- **Fixed Width**

Appears for every number format style. When selected, the length of a number cannot exceed the text length of the text element (Text, TextBox, or Button) in design time. Numbers that exceed design time text length will show the special character specified in the **Value too large for Fixed Field**. The character is defined in the **Special character** tab of the **Styles** dialog.

For the Real number style, the length of the fractional part of the number is truncated to fit the design time length. If the length of the number is still too large after removing the entire fractional part, then the number will show the special character specified in the **Value too large for Fixed Field**.

- **Precision**

Appears for the **Real**, **FixedDecimal**, and **Exponential** number styles. **Precision** sets the possible number of digits in the fractional part of a number to the right of the decimal point and can be set from 0 to 8.

- **Bits From and To**

Appear for the **Hex** and **Binary** number formats. **Bits From** sets the starting bit position (0-31) of a hex or binary number shown during runtime. **To** sets the ending bit position of a hex or binary number shown during runtime.

User-Defined Global Number Styles

The Galaxy Style Library includes a set of 25 user-defined number styles. User-defined styles appear towards the bottom of the list of the **Format Styles** list and are named UserDefined1 to UserDefined25.

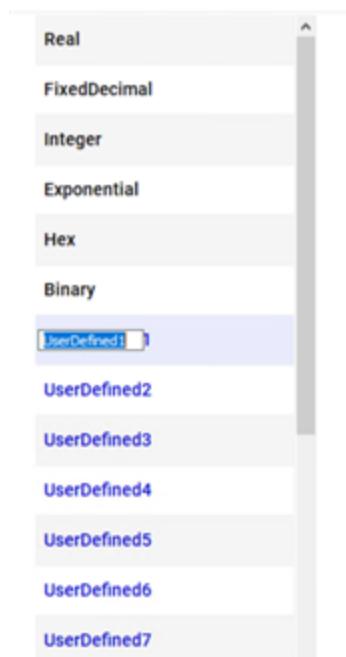
Rename User-defined Global Number Styles

Rename a style to provide a descriptive name, to provide a group of related styles names, or to suit any other specific application needs. The renamed style will appear during configuration. For example, the renamed style will appear in the Industrial Graphics Editor when configuring an animation using Number Styles.

The renamed style will function the same during run time as it did prior to the name change.

To rename a user-defined element style

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. On the **Format** tab, navigate to the user-defined style you want to rename.
4. Right-click the UserDefined format style to select the style name and enable the text box for editing.



5. Enter a new style name.

Import and Export User-Defined Global Number Styles

You can import and export your renamed user-defined Number Style as a normal part of the Galaxy Style Library. For more information, see [Importing and Exporting Galaxy Style Libraries](#).

Switch to another galaxy style library during runtime

AVEVA OMI includes the Style namespace with two attributes.

- CurrentStyle is a string that is the name of a Galaxy Style Library. CurrentStyle can be included in a script to change to another Galaxy Style Library during runtime.

CurrentStyle is retentive. A ViewApp starts with the Galaxy Style Library that was active when the ViewApp was previously stopped.

The CurrentStyle attribute operates in the MyViewApp.Style namespace. CurrentStyle is specified in a script with the MyViewApp.Style prefix in the form MyViewApp.Style.CurrentStyle.

- Styles.CSV is a comma-separated string that is a list of the defined galaxy styles. The value is read-only and is non-retentive. It is defined by the list of galaxy styles defined in the Galaxy Style Library. This list is configured in the IDE.

Note: The CurrentStyle and Styles.CSV attributes are available for use only in AVEVA OMI ViewApps. The CurrentStyle attribute cannot be used in an InTouch HMI application because the MyViewApp. prefix is regarded as a configuration error.

Attribute	Data Type	Read/Write	Initial Value	Description
CurrentStyle	String	Yes	Retentive	Loads the name of a Galaxy Style. During runtime, the value of CurrentStyle can be changed to the name of another Galaxy Style. The appearance of graphics in the running ViewApp are updated with the element styles of the new, active Galaxy Style.
Styles.CSV	String	Read Only	No initial value	Gets the list of available Galaxy Styles in CSV format. During runtime, shows the styles as a comma separated string. The value is automatically updated when new styles are added to the Galaxy Style Library in the IDE.

Build applications

You can build applications with the objects and graphic standards that you built during the design phase of your project. These standards represent reusable components that can be inserted into a ViewApp in an almost assembly line process.

Note: Some of these links go to the relevant sections in other AVEVA help systems.

Create a ViewApp Framework	About screen profiles	AppFramework layouts	About ViewApp security	About alarms
				
Work with Graphic Elements	About graphic elements	About the graphic element property commonalities	Specific Graphic Element Properties	Use animation
				
Work with Graphics and Other Media	Create and manage graphics	Manage industrial graphics	About custom properties for graphics	About using an External Content item
				
Write Scripts	Script Editor		Add a script to a graphic	
				
Assemble a View Application	ViewApp editor	Switch ViewApp language at runtime	ViewApp Navigation	About client controls embedded into a graphic
				
Show ViewApps with InTouch Access	About InTouch Access Anywhere	About InTouch Access Anywhere	Build an InTouch Access Anywhere	About using InTouch Access Anywhere

Anywhere

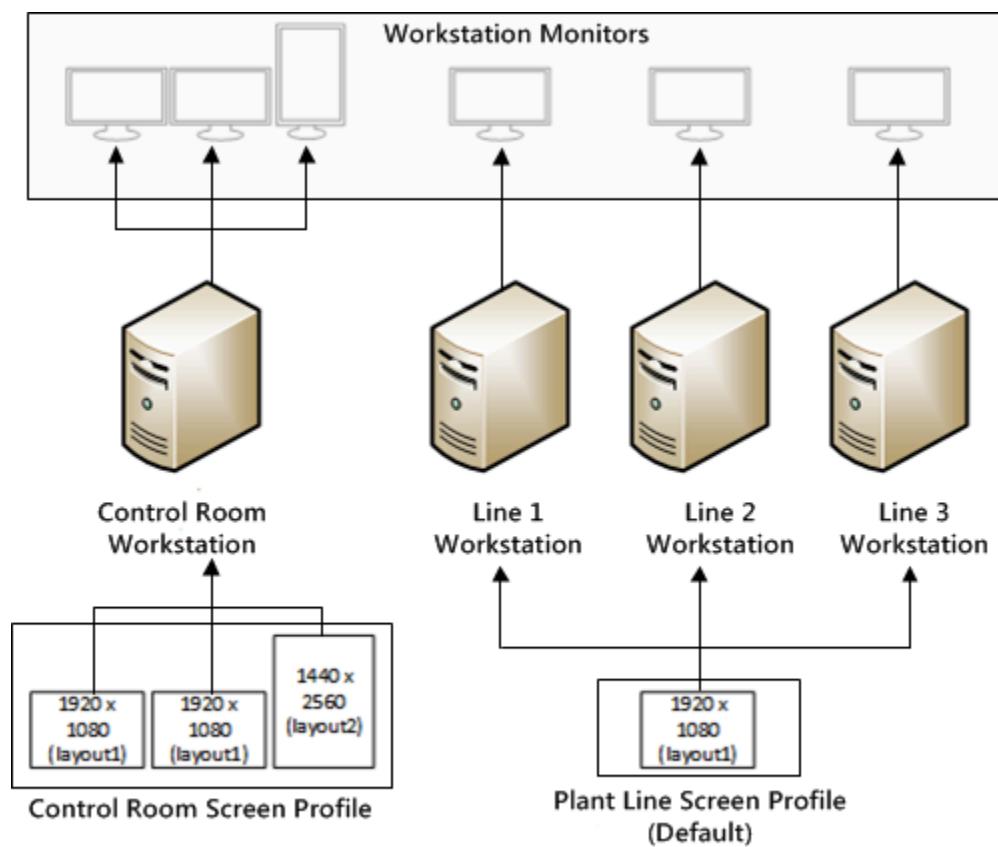
architecture

Secure
Gateway

About screen profiles

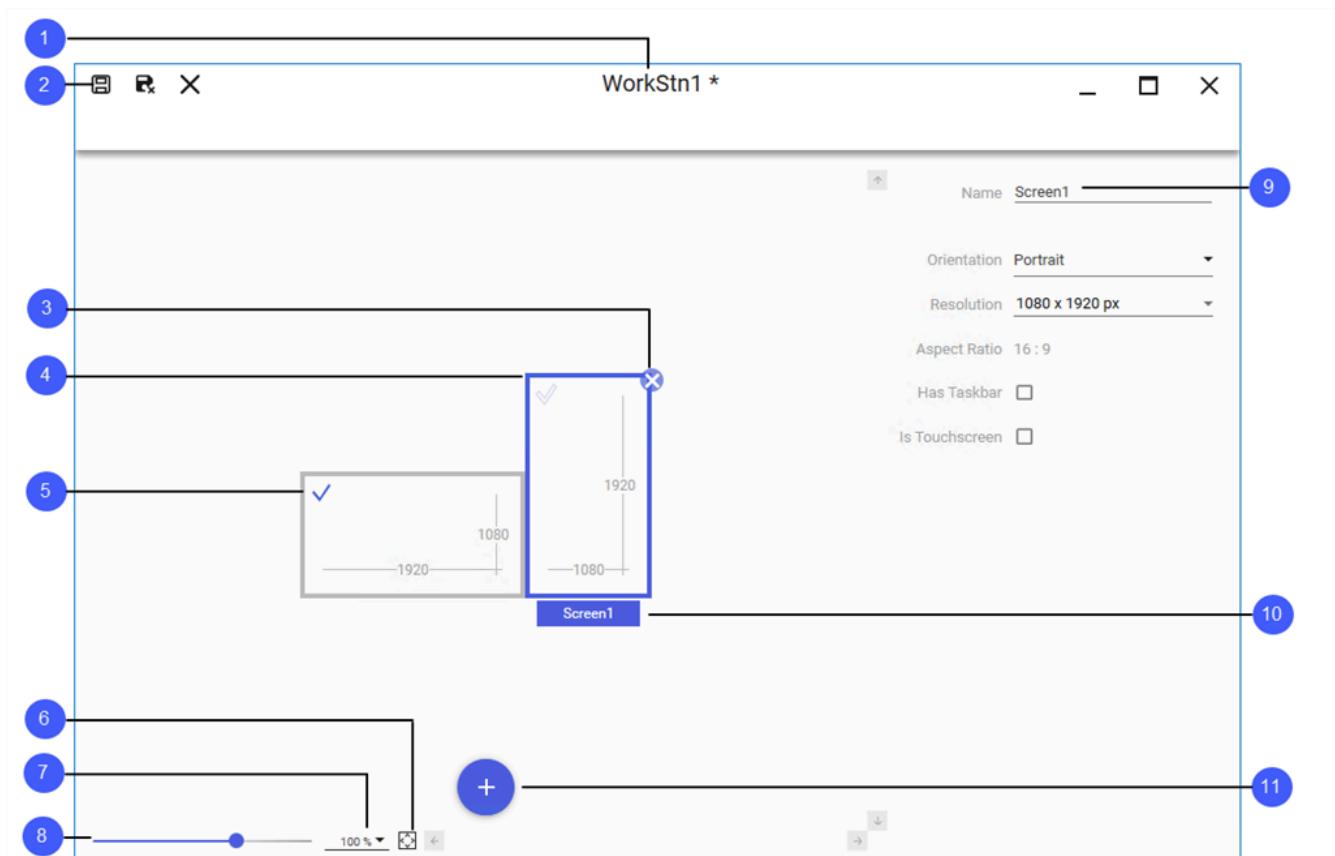
A screen profile defines the physical characteristics of one or more client workstation screens that will show a running ViewApp and how these screens are arranged with respect to each other. Each screen icon that appears in a screen profile represents a physical screen.

You create, manage, and configure screen profiles with the Screen Profile Editor. You must assign values to the physical properties of each screen in a profile. Also, you must arrange the screen icons to represent the actual configuration of screens attached to a client workstation with the Screen Profile Editor.



About the Screen Profile Editor

The figure below shows the various components of the Screen Profile Editor to create and configure a profile by adding screens, arranging the screens to each other, and assigning property values to each screen.

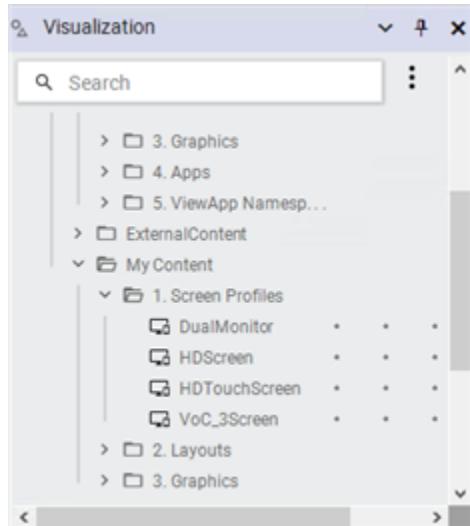


1	Screen Profile Name	Name assigned to the screen profile.
2	Command Bar	Commands to save, save and close, and close the screen profile in the Screen Profile Editor.
3	Delete Screen	Selecting the Delete Screen icon removes the screen from the profile. The primary screen cannot be deleted from a profile .
4	Selected Screen	A blue border indicates the current selected screen in the profile. Tapping or clicking on a screen icon within a profile sets the focus of the Screen Profile Editor to the selected screen.
5	Primary Screen Indicator	A solid blue check mark icon identifies the primary screen in the profile.
6	Zoom to Fit	Selecting the Zoom to Fit icon changes the size of all screens in a

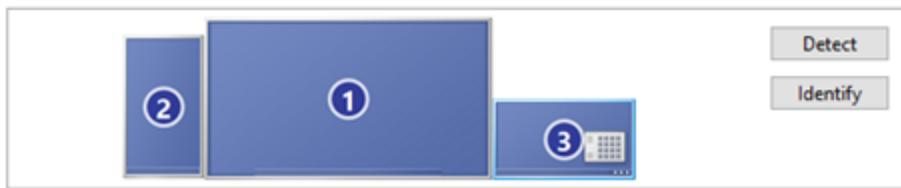
		profile to fit entirely within the edit area of the Screen Profile Editor.
7	Zoom Percent Picker	Drop-down list of zoom percentages to resize all screens shown in a profile.
8	Zoom Percent Slider	Moving the Zoom Percent Slider left or right changes the size of screens shown in a profile
9	Selected Screen Properties	Properties of the selected screen in a profile. See About screen properties for a description of each screen property.
10	Screen Name	Name assigned to a screen as the value of the screen's Name property.
11	Add Screen	Selecting the Add Screen icon adds a screen to a profile immediately to the right of the selected screen. The added screen is a duplicate of the selected screen.

Create a screen profile

You create a screen profile from the **Visualization folder**. After creating a screen profile, you continue editing and configuring the screen profile by selecting it from the list of objects shown in the **Visualization folder**.



A screen profile should replicate the display properties configured in Windows for the computer .



To match the computer's display properties, a screen profile must include values assigned to the following properties:

- Resolution of each screen in the profile: a drop down menu provides preset display resolutions, ranging from 1024 x 768, to 7680 x 4320. Or, you can enter a custom resolution. The default is 1920 x 1080.
- Orientation of each screen: landscape or portrait.
- Physical arrangement of screens: side by side, stacked vertically, or a combination.
- Primary screen: the primary screen should correspond to display 1 configured in Windows.
- Touch screens: enable this property if the screen is a touch-sensitive.

Create a screen profile from the visualization folder

You create a screen profile from the **Visualization folder** of the System Platform IDE. After a screen profile is created, it appears in the **Visualization folder** with an identifying icon.

To create a screen profile

1. Open the **Visualization folder**.
2. Select **Galaxy** from the menu bar, select the **New** option, and then select **Screen Profile**.

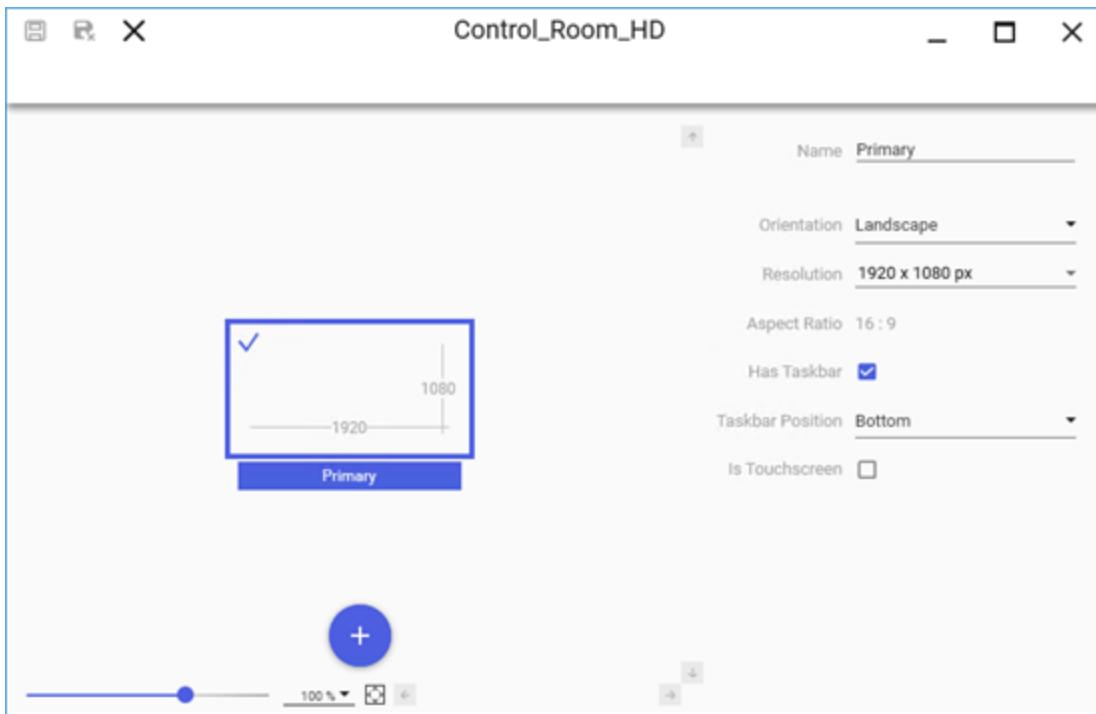
You can also create a screen profile by other methods:

- **Keyboard shortcut:** Ctrl + Shift + E
- **Shortcut menu:** Right-click within the **Visualization folder** and select **New**, and then **Screen Profile**.

A screen profile appears in the **Visualization folder** with a default name of ScreenProfile_001 for the first screen profile that has been created. A blue background around the screen profile name indicates the name can be changed.

3. Select the screen profile from the list and rename it.
4. Select the screen profile and press Enter to open the screen profile in the Screen Profile Editor.

After opening a screen profile, the Screen Profile Editor shows a single screen in landscape orientation with a 1920 x 1080 resolution.



You can begin configuring the screen profile for your View Applications. For more information about configuring a screen profile, see [Configure a screen profile](#).

Configure a screen profile

Video Tutorial: Configure a screen profile in AVEVA OMI

https://player.vimeo.com/video/992308648?badge=0&autoplay=0&player_id=0&app_id=58479

A screen profile specifies the physical characteristics of one or more workstation screens. For each screen defined in a profile, you must specify a screen name, the screen's orientation, resolution, and whether the screen includes a task bar and supports touch gestures. Also, you must arrange the screen icons in a screen profile to represent the configuration of actual client workstation screens, as configured within the Windows display settings (you can configure the Windows settings either before or after you configure the screen profile). For the details of each screen configuration property, see [About screen properties](#).

Name	Primary
Orientation	Landscape
Resolution	1920 x 1080 px
Aspect Ratio	16 : 9
Has Taskbar	<input checked="" type="checkbox"/>
Taskbar Position	Bottom
Is Touchscreen	<input type="checkbox"/>

Configuration Note:

A computer's Windows display properties do not necessarily need to be defined before you create a screen profile. You can create a screen profile first, and then configure the hardware screens for the computer afterwards. You can start testing with a single 1080P screen and add screens to the profile later in the testing cycle.

Add a screen to a screen profile

You can add up to a maximum of 20 screens to a screen profile. When a new screen is added, a screen icon appears to the right of the currently selected screen. The physical properties of the selected screen are assigned to the added screen.

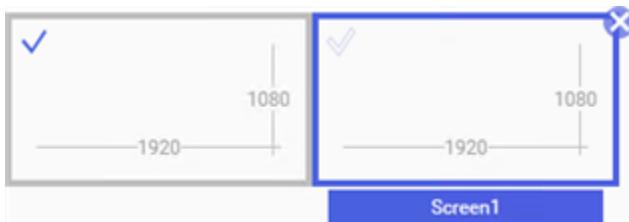
If another screen is located immediately to the right of the selected screen, the added screen is inserted and the existing screen is moved to the right.

To add a screen to a screen profile

1. Open the **Visualization folder**.
2. Open the screen profile for editing.
3. Select the screen in which you want to add a new screen to the right of it.
4. Click the **Add Screen** icon that appears beneath the screen icons.



A screen icon appears immediately to the right of the selected screen and is named Screenx, where x is the number of the screen added to the profile.



5. Rename the added screen.
6. Modify the physical properties of the new screen in the right column of the editor window.
7. Move the screen (in relation to the primary screen) so it mimics the computer display configured in Windows.
8. Save your changes to the screen profile.

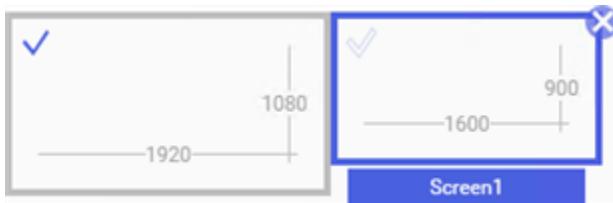
Delete a screen from a screen profile

You can delete all screens within a profile except the primary screen designated with a checkmark in the top left corner of the screen.

To delete a screen from a screen profile

1. Open the **Visualization folder**.
2. Open the screen profile that contains a screen you want to delete.

3. Select the icon of the screen you want to delete.



The primary screen shows a check mark in the screen's upper left corner. The primary screen cannot be deleted from a screen profile. If the screen you selected can be deleted, a circle with an X appears at the screen's upper right corner.

4. Click the circle on the screen to be deleted or press the Delete key.

The screen is deleted from the screen profile and its icon no longer appears in the edit area.

5. Click **Save**.

Rearrange screens in a screen profile

The screen configuration area shows the placement of screens defined in a screen profile in relation to each other. The placement of the screen icons should represent how actual screens are arranged on the workstation that will run a View Application, as configured within the Windows display settings. The primary screen should correspond to display 1 in the Windows settings.

You use a drag and drop operation with a mouse or touch gesture to arrange the screen icons to match the position of the screens of a workstation. Typically, you arrange screens by aligning the corner of one screen to the corner of another screen.

A valid screen arrangement must adhere to the following rules:

- At least one corner of a screen must be aligned with the corner of another screen.
- All screens in a profile must be arranged to touch at least one other screen.
- Screens must be arranged without overlapping any other screens.
- A screen arrangement supports up to a maximum of 20 screens.

To rearrange screens in a screen profile

1. Open the **Visualization folder**.

2. Open the screen profile.

3. Select the screen icon that you want to move.

The border of the icon turns blue to show that it is selected.

4. Drag the icon to its new location.

While dragging the icon, the border turns red. It turns blue again when the icon is aligned with the corner of another screen icon.

If you drop the icon at an invalid position, it returns to its original position.

5. Click **Save** to save the changes.

Configure a screen

The Screen Profile Editor shows the properties of the selected screen in the right column. You assign a name to a screen and configure its properties by updating the values shown in the property fields.

To configure a screen

1. Open the **Visualization folder**.
2. Open the screen profile for editing.
3. Select the screen that you want to configure properties for.

The default Primary screen includes the following properties.

The screenshot shows the 'Primary' screen profile in the AVEVA Operations Management Interface. The properties listed are:

- Name: Primary
- Orientation: Portrait
- Resolution: 1080 x 1920 px
- Aspect Ratio: 16 : 9
- Has Taskbar:
- Taskbar Position: Bottom
- Is Touchscreen:

Additional touch property fields appear when the **Is Touchscreen** property is selected.

The screenshot shows the 'Primary' screen profile with the 'Is Touchscreen' property selected. The additional touch property fields are:

- Is Touchscreen:
- Touch Lock Level: Two Handed Operation
- Touch Lock Position: Bottom Left

For a complete description of all properties, see [About screen properties](#).

4. Type a name for the screen and assign values to the screen's properties.
5. Click **Save**.

About screen properties

A set of screen properties appears in the Screen Profile Editor. These properties define the physical and functional characteristics of each screen in a profile. Assigning a value to each property is required to configure a screen.

Name Property

Each screen defined in a screen profile must be uniquely named. The border of the **Name** field turns red if you enter an invalid screen name.

Orientation Property

The orientation of a screen can be either landscape or portrait. A screen's width is greater than its height when oriented in landscape mode. In portrait mode, screen height is greater than its width.

Changing a screen's orientation changes the width and height values shown in the **Resolution** property field. For

example, when the resolution is 1920 w x 1080 h, changing the orientation from Landscape to Portrait changes the resolution to 1080 w x 1920 h.

Resolution Property

The **Resolution** property field shows common screen resolutions in a drop-down list. The ratio of the width of a screen by its height in pixels is the aspect ratio. Changing orientation reverses width and height values shown in a screen resolution.

Aspect Ratio

The **Aspect Ratio** property is an automatically calculated ratio of a screen's width to height by pixel counts. The Screen Profile Editor shows the aspect ratio of a screen in the form of a well known value. If an aspect ratio does not fall in a range of a well known value, then the aspect ratio is represented by a double data type rounded to two decimal points

Has Task Bar

The **Has Taskbar** property indicates whether the selected screen shows a task bar or not. Task bar information is considered when adjusting the size of a layout to fit a maximized screen. As task bar can differ in size based on themes or physical size and is only an approximation.

If you select the **Taskbar** property then the **Taskbar Position** field appears with options to position the taskbar at the Top, Bottom, Left or Right of a screen border.

Is Touchscreen

A screen touch lock determines how screen touch gestures are supported while a ViewApp is running.

A touch lock can be configured to:

- Support all touch interactions of a ViewApp.
- Prevent users from interacting with machinery while interacting with related controls on touch screens. The risk of accidentally entering a machinery command with a touch gesture is reduced by forcing the user to keep the touch lock active by pressing the lock icon continuously while operating screen controls.
- Restrict access to a subset of supported touch gestures to prevent accidental interactions with a ViewApp.

Touch Lock Configuration

You can set three Touch Lock properties when configuring a screen profile.

- **Is Touchscreen**

- **Disabled**

Disabled is the default. No touch lock appears on the screen while a view application is running. You can interact with a ViewApp or its contents using all supported touch gestures.

- **Enabled**

A touch lock appears on the screen and restricts the usage of touch gestures to protect a ViewApp.

- **Touch Lock Level**

The **Touch Lock Level** property becomes accessible only when the **Is Touchscreen** property is enabled.

- **None**

The touch lock is disabled and users can interact with a view application or its contents to the full extent to which touch is supported. **None** is the default and no touch lock icon appears on the screen of a running ViewApp.

- **Commands Locked**

In **Commands Locked** mode, all touch gestures are blocked when the touch lock is active. A touch lock icon is green to indicate the touch lock is active or in a locked state. Mouse or keyboard interactions are still possible.

Pressing the touch lock icon during runtime deactivates or *unlocks* the touch lock to enable you to interact with a view application using all supported touch gestures. A touch lock icon turns red to indicate the touch lock is in an unlocked state.

Commands Locked mode does not block writes to I/O in any way, shape, or form.

- **Two Handed Operation**

In **Two Handed Operation** mode, single finger content selection gestures are blocked by the touch lock, but two and three finger pane level gestures are permissible. Pressing on the touch lock icon deactivates the touch lock enabling you to interact with a view application using all touch gestures including single finger content gestures.

- **Touch Lock Position:**

Touch Lock Position specifies the placement of the touch lock icon on the screen. This property is enabled only if the **Touch Lock Level** property is set to **Commands Locked** or **Two Handed Operation**.

Touch Lock Position provides options to place the Touch Lock icon at the bottom left or bottom right of a screen.

Scope of a Touch Lock During Runtime

- One Touch Lock can be configured for each screen assigned to a ViewApp.
- When Touch Lock is present during runtime, the possible touch gestures are specified by the configuration of that screen.
- When Touch Lock is absent or deactivated, a user can perform all supported touch actions on the corresponding touch screen.
- If there are multiple layouts per screen, then touch interactions on all layouts depend on the state of Touch Lock.
- If there are layouts that span multiple touch screens, the configuration of the Touch Lock on the primary screen of the layout determines touch functionality for other screens.

Resize a screen

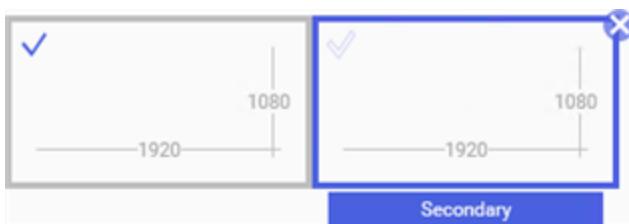
You can change the size of a screen by selecting new horizontal and vertical dimensions in the **Resolution** field. You can also change screen dimensions by changing the screen's orientation between Landscape and Portrait.

Changing the size of a screen affects the positioning of surrounding screens. A resize may result in screens that are not aligned to the edge or ones that are completely detached. These screens will then snap to the closest point that does not overlap any other screens.

The scrollable extents of the viewable screen area are updated if a screen is resized, which may cause other screens to extend beyond the boundaries of the Screen Profile Editor's viewable area.

To change the size of a screen in a screen profile

1. Open the screen profile for editing.
2. Select the screen to resize.



3. Select new horizontal and vertical dimensions from the **Resolution** field or change the screen's orientation.

1600 x 900 px
1024 x 768 px
1280 x 1024 px
1366 x 768 px
1440 x 900 px
1600 x 1200 px
1680 x 1050 px
1920 x 1080 px
1920 x 1200 px
2560 x 1440 px
2560 x 1600 px

Note: You can also type a custom width and height in the **Resolution** field.

4. Click **Save**.

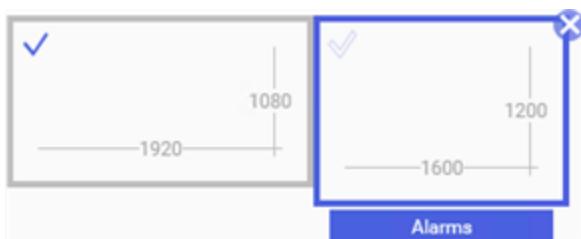
Set a primary screen

Every screen profile includes a primary screen. A primary screen represents the main screen of a workstation that will show a running View Application. The primary screen is used as the reference point when applying a screen profile to the workstation, and corresponds to display 1 in the Windows settings. The primary screen cannot be deleted from a screen profile.

The primary screen of a screen profile is identified by a solid check mark in the screen icon's upper left corner. When a non-primary screen is selected, an empty check mark is shown instead. Clicking on an empty check mark selects the screen as the primary screen of the profile.

To set the primary screen of a screen profile

1. Open the screen profile for editing.
2. Select the screen icon to designate as the primary screen of the profile.

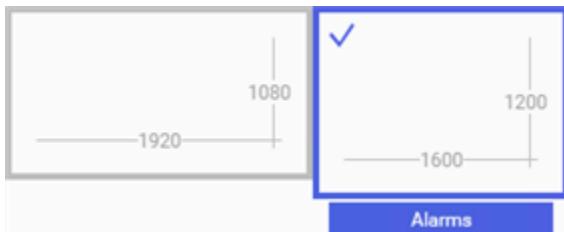


3. Move your mouse over the empty check mark at the top left of the screen icon.

A tooltip message indicates the screen can be set as the primary screen.

4. Click the check mark.

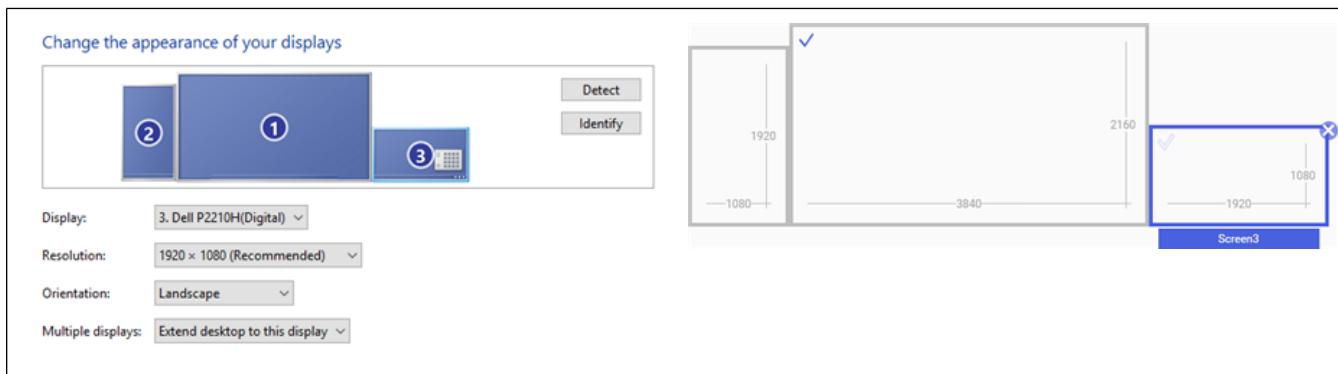
The check mark turns blue to indicate that this screen is now the primary screen.



5. Save your changes.

Set additional screens

The following figures show examples of a computer's Window display properties and a corresponding screen profile that replicates the display properties.



In this example, the Windows display properties of the three screens attached to a workstation are:

Screen 1	3840 by 2160 resolution, Landscape orientation
Screen 2	1080 by 1920 resolution, Portrait orientation
Screen 3	1920 by 1080 resolution, Landscape orientation, Primary screen

Typically, there are three major tasks to configure a screen profile:

- Add the screens
- Position the screens

- Configure the properties of the screens

The following steps demonstrate these general tasks by building an example screen profile that replicates a computer's Windows display properties shown above.

- Open a screen profile that you created, which shows the default **Primary** screen.
- Rename the **Primary** screen to **Screen1**
- Add a screen, which appears to the right of the **Primary** screen and is named **Screen2**.
Screen2 has same properties as **Screen1** except its name and it does not contain a task bar. Also, **Screen2** is the selected screen.
- Add another screen, which appears to the right of **Screen2** and is named **Screen3**.



- Set **Screen3** as the primary screen in the screen profile.
- Drag **Screen2** to the left of **Screen1**.

The three screens of the profile have been created and are in the approximate configuration that matches the Windows display properties.

- Set the properties of **Screen1** to match Screen1 of the computer's Window display properties.

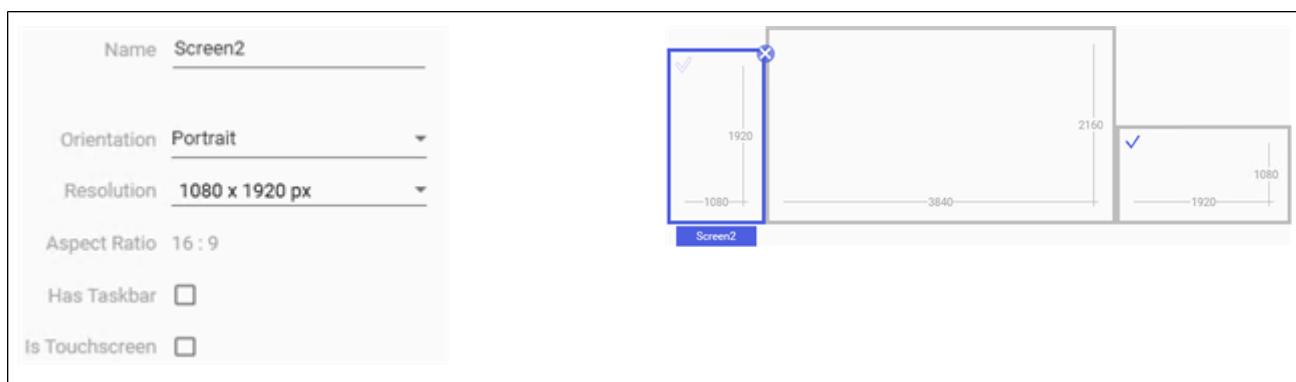
A screenshot of a software interface for configuring a screen profile. On the left, there is a list of properties for 'Screen1':

- Name: Screen1
- Orientation: Landscape
- Resolution: 3840 x 2160 px
- Aspect Ratio: 16 : 9
- Has Taskbar:
- Taskbar Position: Bottom
- Is Touchscreen:

On the right, there is a preview window showing three screens. The first screen is labeled 'Screen1' at the bottom and has a blue checkmark in its top-left corner. The second screen is labeled 'Screen2' at the bottom and has a blue checkmark in its top-left corner. The third screen is labeled 'Screen3' at the bottom and has a blue checkmark in its top-left corner. Each screen has a coordinate system with '1920' on the horizontal axis and '1080' on the vertical axis.

If necessary, drag **Screen2** and **Screen3** to align both screens to the bottom of **Screen1**.

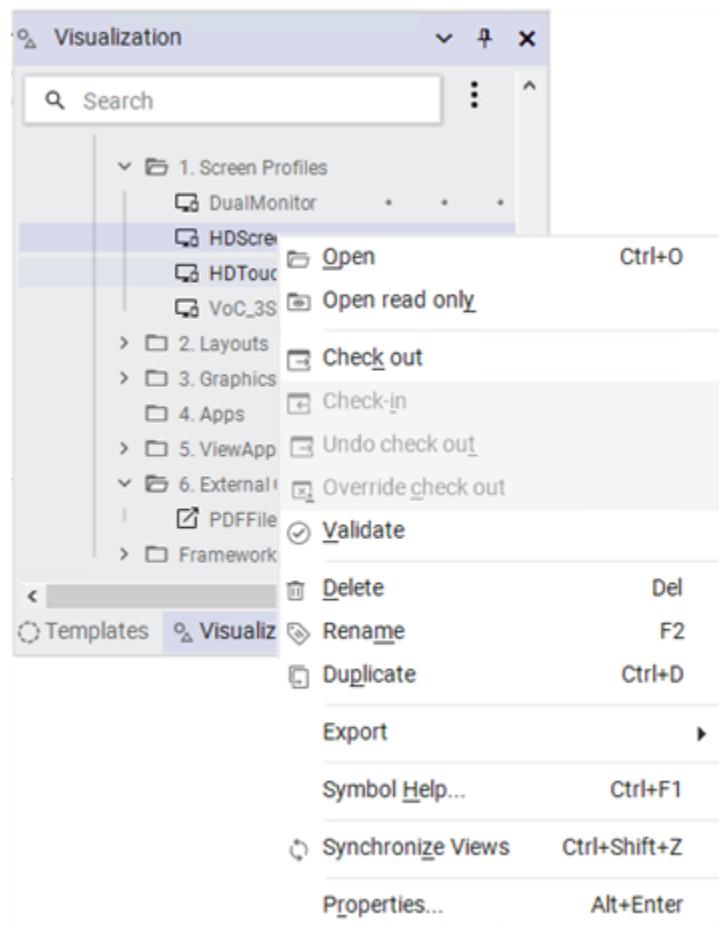
- Set the properties of **Screen2** to match the properties of screen 2 of the Windows display properties.



Your screen profile should match the Windows display properties.

Manage a screen profile

You can perform routine tasks to manage a screen profile from the **Visualization folder** by selecting shortcut menu options.



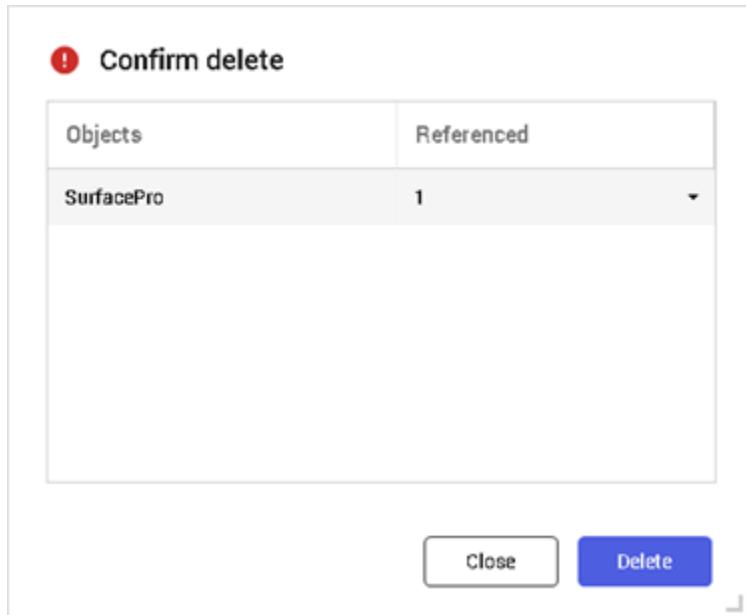
Delete a screen profile

You can delete a screen profile from the Visualization folder.

To delete a screen profile

1. Open the **Visualization folder**.
2. If the screen profile you want to delete is open for editing, close it.
3. Select the screen profile(s) that you want to delete from the **Visualization folder** list.
4. Press the **Delete** key or right-click and select **Delete** from the shortcut menu.

The **Confirm Delete** dialog box appears. The **Referenced** column shows the number of objects that the screen profile is referenced by. To see what those objects are, click the down-arrow at the right side of the **Referenced** list. This shows any layouts, ViewApp templates, or ViewApp instances that refer to the profile.



Note: If you delete the screen profile associated with a layout, it sets the layout's **Target** field to a red background to indicate an error condition. The resolution of the target screen is cached with the layout.

5. Click **Delete** to delete the screen profile(s).

Rename a screen profile

You can rename a screen profile by selecting a profile from the Visual Graphics list and pressing F2 or by clicking on the name to enter edit mode.

A new screen profile name is validated to ensure it meets naming requirements. If a new name is a duplicate of an existing name or breaks any name rules, a message appears indicating there is a problem with the name.

Pressing the Esc key cancels a rename request and the screen profile reverts to the original name. Committing an empty name cancels the rename request.

To rename a screen profile

1. Select the screen profile that you want to rename from the **Visualization folder** list.
2. Click the screen profile name or press **F2** to enter edit mode. A blue background appears around the screen profile name.
3. Type the new name of the screen profile.

4. Commit the new name by clicking outside of the screen profile name.
If the new name is valid, it appears in the **Visualization folder** list. If the name is invalid, you see a dialog box that describes the problem.
5. If you typed an invalid screen profile name, click **OK** and re-enter another name.

Duplicate a screen profile

You can create a duplicate of a screen profile and then rename it. Duplicating a screen profile reduces the effort to create an entirely new screen profile by using an existing screen profile as a template.

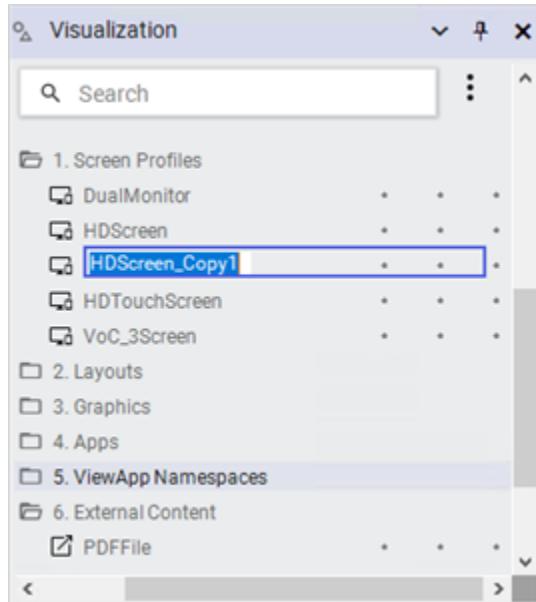
To duplicate a screen profile

1. Open the **Visualization folder** and select the screen profile that you want to duplicate.
2. In the **Home** ribbon **Edit** area, select **Duplicate**.

You can also duplicate a screen profile by other methods:

- **Keyboard shortcut:** **Ctrl + D**
- **Shortcut menu:** Right-click the name of the screen profile to duplicate and click **Duplicate**.

A new screen profile appears in the **Visualization folder** list with **Copy1** appended to its name.



3. Type the name to give the duplicate screen profile.

Export a screen profile

You can export one or more screen profiles from the Visualization folder to an aaPKG file. You can then import the screen profiles from the export file to another project.

To export a screen profile

1. Open the **Visualization folder**.
2. Select the screen profiles to export.

3. In the **Home** ribbon **Export** area, select **Selected**, then select **As package**.

You can also right-click the screen profile name and select **Export**, then **Selected object(s)**.

The **Export Selected objects** dialog box appears with a folder tree to specify where the exported file should be saved and a field to name the export file.

4. Select the folder to save the export file.

5. Assign a name to the export file.

The default export file name is the name of the first selected screen profile from the **Visualization folder**.

6. Click **Save**.

A horizontal bar shows the progress of the screen profiles being loaded into the export file.

7. Verify the export aaPKG file has been placed in the folder you specified.

Import a screen profile

You can import screen profile created in another project to the Visualization folder of the System Platform IDE. Exported screen profiles are saved in an aaPKG file.

During the import process, you have the choice to keep or overwrite any screen profiles currently in the Visualization folder with the imported screen profiles.

To import one or more screen profiles

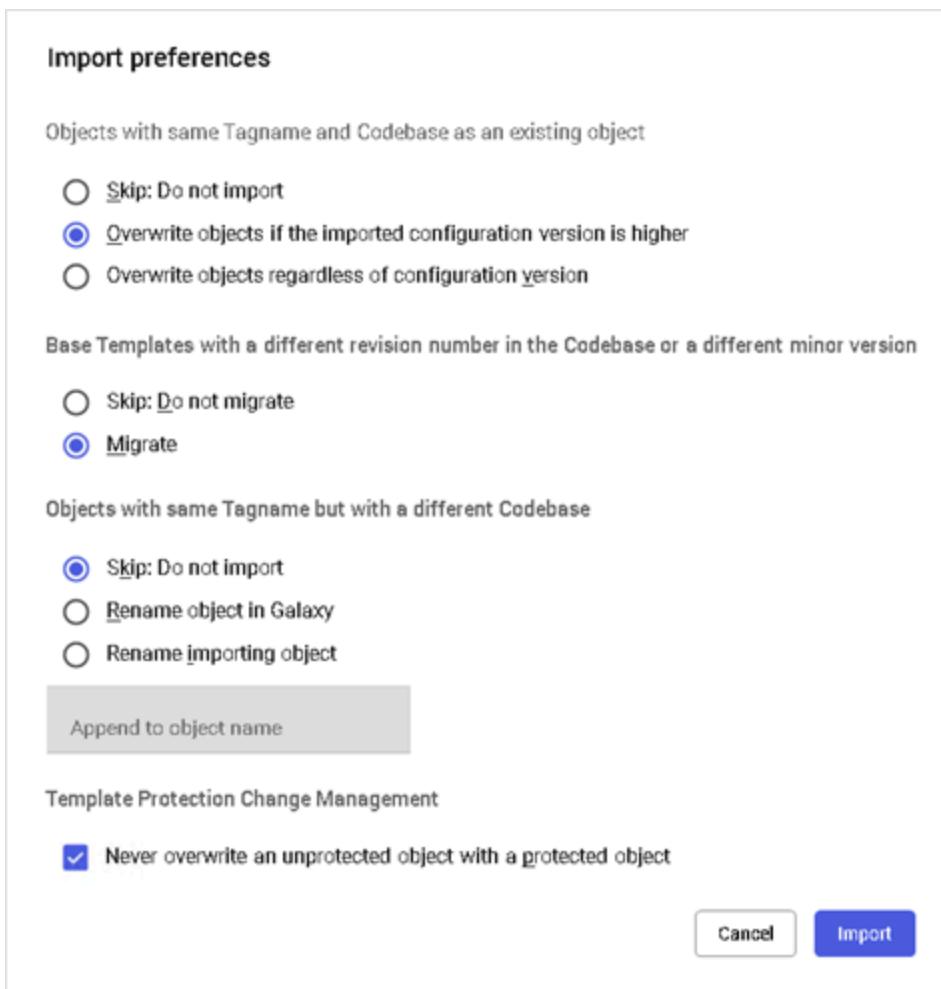
1. Open the **Visualization folder**.
2. Select the **Galaxy** ribbon, then select **Import**.
3. Select **Objects**, then **From package**.

The **Import Objects from package** dialog box opens.

4. Go to the folder where the aaPKG file containing screen profiles is located.
5. Select the aaPKG file and click **Open**.

The **Import Preferences** dialog box opens with options to import screen profiles. You cannot have two screen profiles with the same name or more than one copy of the same version of a profile in the same Galaxy.

When you import a screen profile, you can choose options from the **Import Preferences** dialog box how you want naming and version conflicts handled.



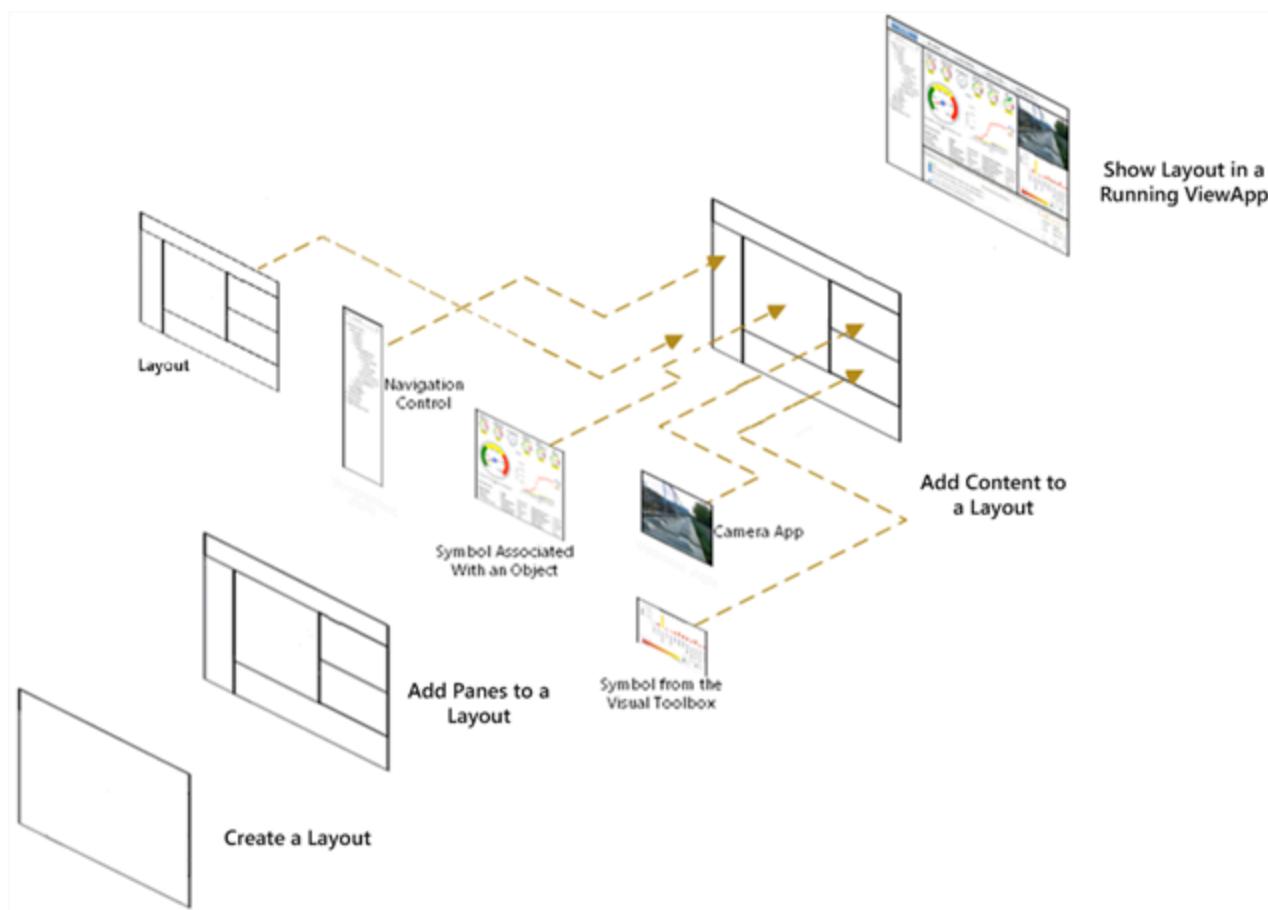
6. Click **OK** to import the screen profiles.

A progress bar shows the screen profiles being imported into the Visualization folder. After finishing the import, you should see the screen profiles in the list of Visualization folder objects.

AppFramework layouts

A layout consists of one or more rectangular areas called panes that contain content shown in a ViewApp. Horizontal and vertical lines called dividers set the boundaries between adjacent panes in a layout. As part of building a ViewApp, you associate a layout to a screen in a selected screen profile. The Layout system encompasses an editor and an engine that displays a layout and its content while a ViewApp is running.

During design time, you use the Layout Editor to configure one or more layouts for a ViewApp. The Layout Editor includes controls to add panes and adjust their size and arrangement to other panes within the layout's window. The Layout Editor provides properties to configure the layout itself and each individual pane. You can also drag and drop content into panes as part of configuring a layout.



In addition to containing content such as the apps and graphics that you add to layout panes, a layout can itself function as content that you add to another layout. See [Add content to panes](#) for more information.

You can also associate layouts with an Application Server object, as you would any other content item. For more information about using layouts as content, see [Use a layout as content](#).

About the Layout Editor

The Layout Editor consists of **Build** and **Script** pages, and includes controls to create and configure panes that appear in a layout.

The Layout Editor provides Layout, Content, and Script editing modes. The Layout editing mode provides a set of components and properties to define the visual and behavioral aspects of each pane.

Content editing mode includes two different modes:

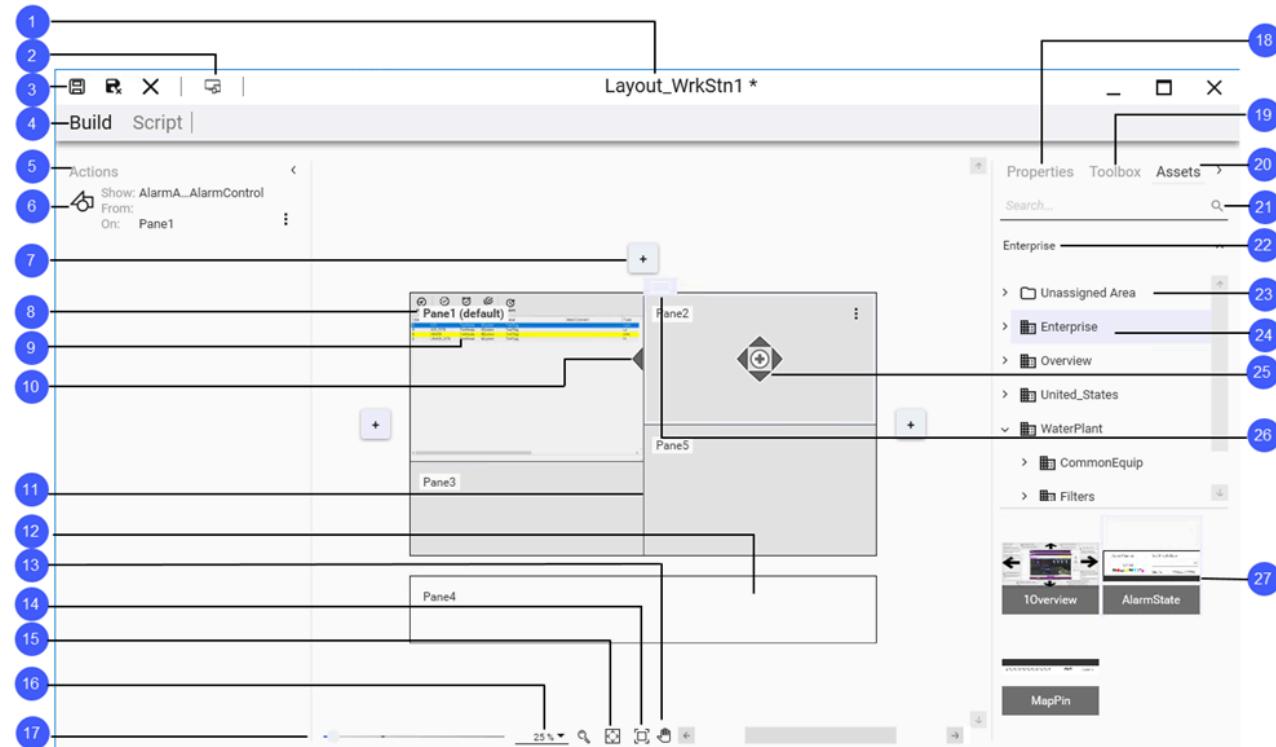
- Fixed Layout (default): This is the best choice for adding content if the ViewApp that uses the layout will be viewed exclusively on desktop-type monitors or similar displays.
- Responsive Layout: This is best choice if the layout will be used in one or more ViewApps that will be viewed by a mix of devices, such as smart phones, tablets, and monitors.

For more information about whether to use the fixed and responsive mode, see [About fixed vs. responsive layouts](#).

Once you have begun adding and configuring content, switching from Fixed to Responsive or visa versa will clear

all panes and any configuration changes that you have made will be lost. Unsaved layout scripts are not affected. In either content editing mode, you can select the content to appear in a pane by selecting the **Toolbox** and **Assets** tabs and dragging and dropping objects, controls, apps, or graphics into panes.

The figure below shows the various components parts of the Layout Editor when Fixed mode is selected. For more information about using the Layout Editor to configure Responsive Layouts, see [About the responsive layout mode](#).



1	Layout Name	Name assigned to the displayed layout.
2	Fixed Layout/Responsive Layout Button	Sets/resets the <code>IsResponsive</code> layout property and changes the build view from the default fixed layout view to the responsive layout view (and back again) in the Layout Editor. Note that any configuration changes made in one view are not preserved when you switch between fixed and responsive views, since the two views cannot be reconciled. If you switch views after configuring the layout, the configuration is lost.
3	Menu Bar	Commands to save, save and close, and close the layout.

4	Build/Script Selection	<p>Switch between the Build and Script pages of the Layout Editor.</p> <ul style="list-style-type: none"> In the Build page, you can select between fixed and responsive layout modes, and configure the layout by adding, removing, and resizing panes, including slide-in panes, and add content such as graphics and apps to individual panes. You can also set properties for the layout, panes, and content. In the Script page, you can add layout scripts that control how panes and their content execute in runtime.
5	Action List	<p>List of navigation actions associated with the panes of a layout. Actions are added to pane when content is dragged from Toolbox or Assets and dropped on a pane or by using the pane-drop-down menu to move content between panes.</p>
6	Pane Action	<p>Selected pane of a layout listed in the Pane Action list, which contains options to edit or remove the current content of the pane.</p>
7	Add Slide-in Pane Button	<p>Adds a slide-in pane on the side of the layout where the Add Slide-in Pane button is located. A slide-in pane can contain content. During runtime, users can select a tab on the interior border of a layout window to show or hide a slide-in pane.</p>
8	Pane Name	<p>Name assigned to a layout pane.</p>
9	Pane Content	<p>Thumbnail graphic representing the content added to a layout pane.</p>
10	Extender Button	<p>Selecting the triangular extender button extends the source pane's size in the direction pointed by the</p>

		triangle to fill the area previously occupied by the target pane.
11	Pane Divider	Line representing the boundary of a pane. During configuration, a layout's pane size can be changed by moving its divider. Also, a divider's visual properties can be configured and whether users can move a divider within a pane during runtime.
12	Slide-in Pane	Optional panes that can be configured on each side of a layout window. During runtime, users select a tab to show or hide a slide-in pane, which appears within the boundaries of the layout window.
13	Pan Lock	Places a pane into pan mode to enable the user to move the view box and show content outside the boundaries of the pane.
14	Zoom to 100%	Sets the size of content shown in a layout to its actual size in the Layout Editor.
15	Zoom to Fit	Adjusts the size of content to fit entirely within a pane in the Layout Editor.
16	Zoom Box	The Zoom Box enables a selected area to be zoomed to the full width or height of the current editing area of the Layout Editor. The cursor shows a magnifying glass when the Zoom Box is selected. Drawing a rectangle by mouse or by touch within the editing area shows the content within the rectangle at the full width or height of the current editing area.
17	Zoom Percent Slider	Moving the Zoom Percent Slider left or right zooms the size of the layout shown in the Layout Editor.
18	Properties grid	Selecting the Properties grid shows a list of properties of the selected

		<p>item within the Layout Editor.</p> <ul style="list-style-type: none">• Layout selected: Tap or click outside the layout window to show layout properties (fixed layout only)• Pane selected: Tap or click within a layout pane to show pane properties• Divider selected: Tap or click on a divider within the layout to show divider properties (fixed layout only)• Content in a pane selected: Tap or click content within a pane to show the content's properties
19	Toolbox Tab	Selecting the Toolbox tab shows the hierarchical folder organization of the Visualization folder. Selecting a folder shows the content within it.
20	Assets Tab	Selecting the Assets tab shows the hierarchical organization of objects shown in the Model view of the Template folder.. Selecting an object shows any content associated with it.
21	Search Field	<p>Data entry field to enter a search string to locate content with a matching name.</p> <ul style="list-style-type: none">• Search from Properties grid: Locates selected property type by name.• Search from Toolbox tab: Locates content within the Visual Toolbox by name.• Search from Assets tab: Locates objects in the Model view by name.

22	Breadcrumb	When the Toolbox tab is selected, a breadcrumb shows the serial list of graphic folder names to the content selected from the list. If the Assets tab is selected, a breadcrumb shows the serial list of Areas or other objects to the selected item from the list.
23	Asset List	The Asset List shows a hierarchy of Area and other types of assets shown by the Template folder Model view. An arrowhead to the left of an asset within the list indicates an asset that can be expanded to show the contained objects.
24	Selected Asset	Touching or clicking an asset shown in the list selects it. A shaded background identifies the selected asset from the list.
25	Add Pane Control	Selecting an arrowhead of the Add Pane Control splits the currently selected pane in half and the new pane is inserted in the direction of the arrowhead. Panes can be added up, down, left, or right of the existing pane with the Add Pane Control .
26	Pane Presentation Style Selector	Selecting the Pane Presentation Style Selector shows a drop-down list of options to select how content is hosted and presented in a pane. In Single mode, a pane can host one piece of content. In Multiple or Tabbed mode, a pane can host multiple items.
27	Asset Content	The content associated with the asset selected from the list.

About fixed vs. responsive layouts

The Layout Editor lets you choose between fixed layout mode and responsive layout mode.

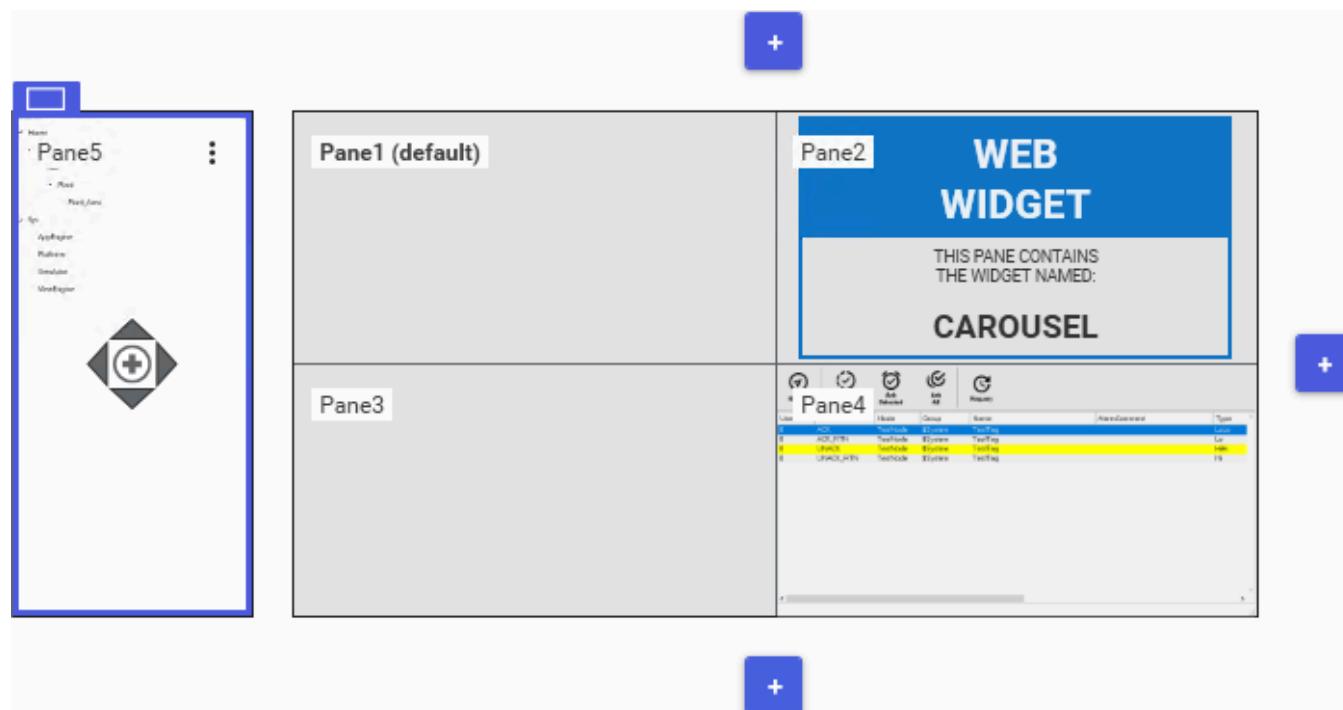
- Use the fixed layout mode if the layout you are designing is intended for use with only a single screen layout or screen form factor. For example, if the layout is only for use in a single ViewApp that will be viewed on a monitor, then using the fixed layout mode is appropriate.
- Use the responsive layout mode if the layout is intended for use with multiple screen form factors. For example, if the layout will be used in multiple ViewApps designed for viewing on different screen form factors, or if the ViewApp will be viewed on hand-held devices through the OMI web client.

The two modes are mutually exclusive, and any unsaved changes made in one mode will be undone if you switch to the other mode. You can toggle between the responsive and fixed layout modes, by clicking or touching the **Fixed Layout/Responsive Layout** button.

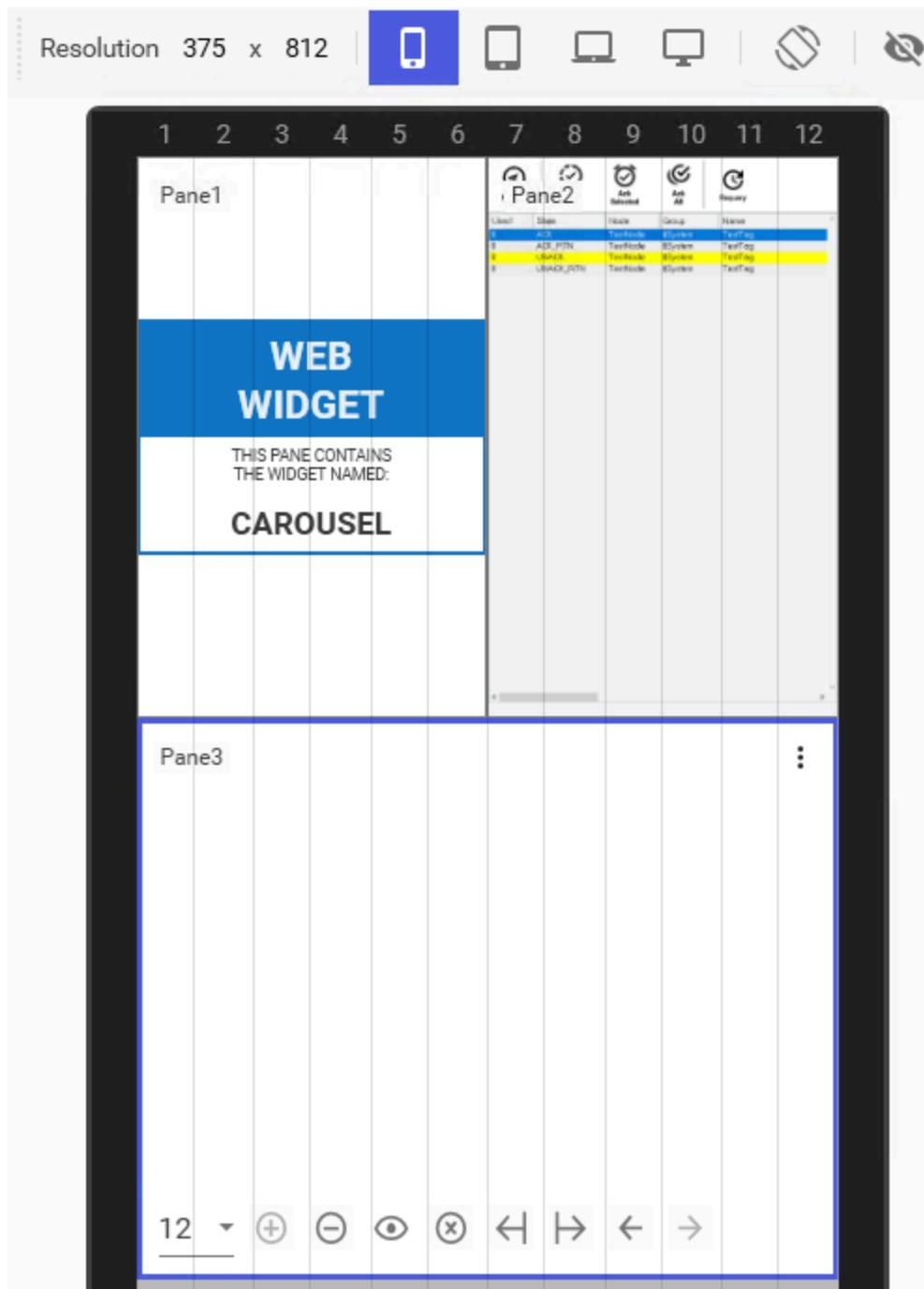
You can also switch between modes by enabling or disabling the **IsResponsive** behavior layout property. To view responsive layout properties, tap or click above the layout window, in the area that shows screen resolutions. To view layout properties, tap or click anywhere outside the layout window.

Important: All changes to the layout are discarded when you switch between modes. Note that if you switch from responsive to fixed or vice versa and then save the layout, you will lose any changes from the previous layout type, even changes that were saved. There is no undo.

You will see something similar to following image when you add content to panes in Fixed Layout mode.



You add content the same way to panes in the responsive layout mode as in fixed layout mode. In the responsive layout mode, panes are now divided into 12 columns of equal width (there is a single pane initially, as is the case with fixed layouts). Panes can be added, and width of each pane can adjusted to use anywhere from one to 12 columns. Additionally, panes in different sized screens can be configured to use a different number of columns.



For more information about the responsive layout mode, see [About the responsive layout mode](#).

About responsive layouts

When configuring a responsive layout, some of the important differences between it and a fixed layout are:

- In the responsive layout mode, as in the fixed layout mode, you begin with a single pane. However, panes are added to a responsive layout by selecting an Add Pane button. In the fixed layout mode, you add panes by dividing the original pane

- Different screen sizes are defined in a responsive layout. Panes can be configured to use different numbers of columns for each screen size. For example, a user on a mobile device may be shown content in a pane that takes up the entire width of the screen, while a user on a laptop is shown the same content in a pane that takes only a fraction of the screen width.
- In the responsive layout mode, you define pane widths by number of columns, not by percentage (or number of pixels) of the total layout width. A pane can be as wide as 12 columns, or as narrow as one column. A pane that is 12 columns wide takes the full width of the layout. You can set a different number of columns for each of five different screen sizes, from extra small (mobile) to large (desktop).
- You can configure multiple panes in a side by side arrangement if the total number of columns the panes use is 12 or less. For example, you can have three panes side by side if each pane four columns wide.

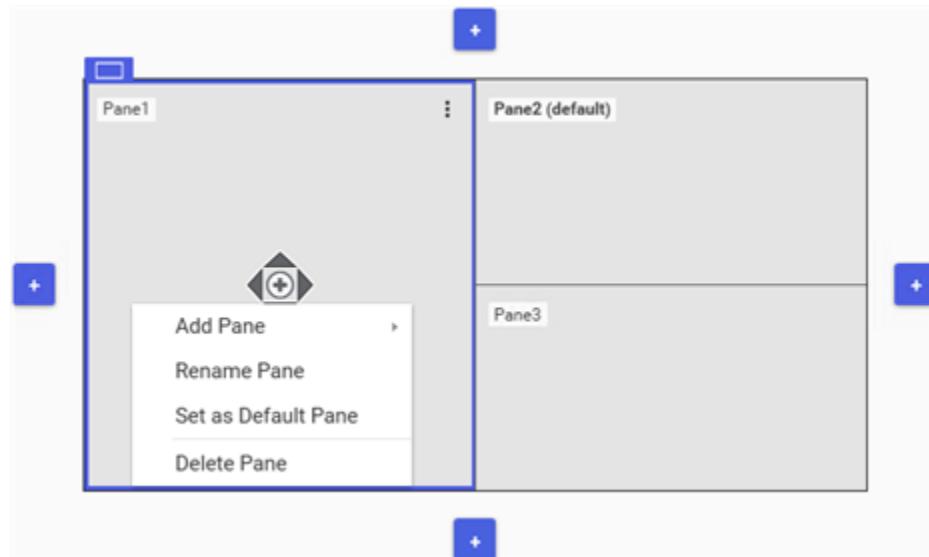
See [About the responsive layout mode](#) for more information.

Design-time editor actions: fixed layout

During design time when you are editing a layout, the Layout Editor provides shortcut menus, a Pane Options button, and shortcut keys to modify the structure of a layout and the properties of panes.

Shortcut Menu

When a layout is opened in the Layout Editor, you can show a shortcut menu by right clicking on the selected pane of a layout.



In Edit Layout mode the shortcut menu includes the following actions:

- **Add Pane**

A sub-menu provides options to add the new pane to the Left, Right, Above, Below the selected pane.

- **Rename Pane**

A blue background appears behind the pane name to indicate the pane can be renamed. The new name you enter must follow System Platform naming conventions.

- **Set As Default Pane**

This action appears only if the pane you select is not the default pane of the layout. The word default placed within parentheses appears next to the pane name after it has been selected as the default pane.

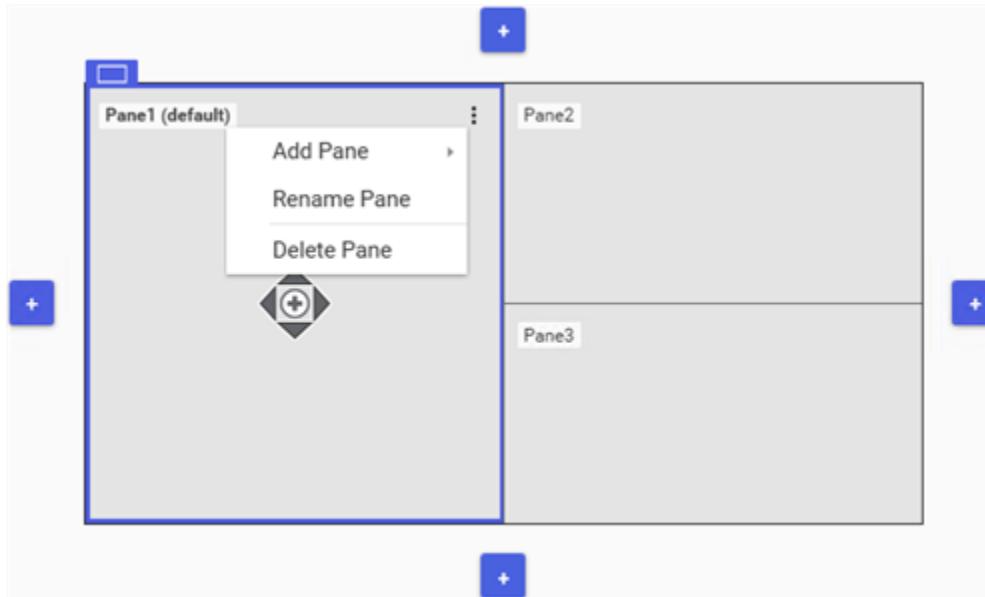
- **Delete Pane**

Selecting **Delete Pane** immediately removes the pane from the layout and the space within the layout window is reassigned to the remaining panes.

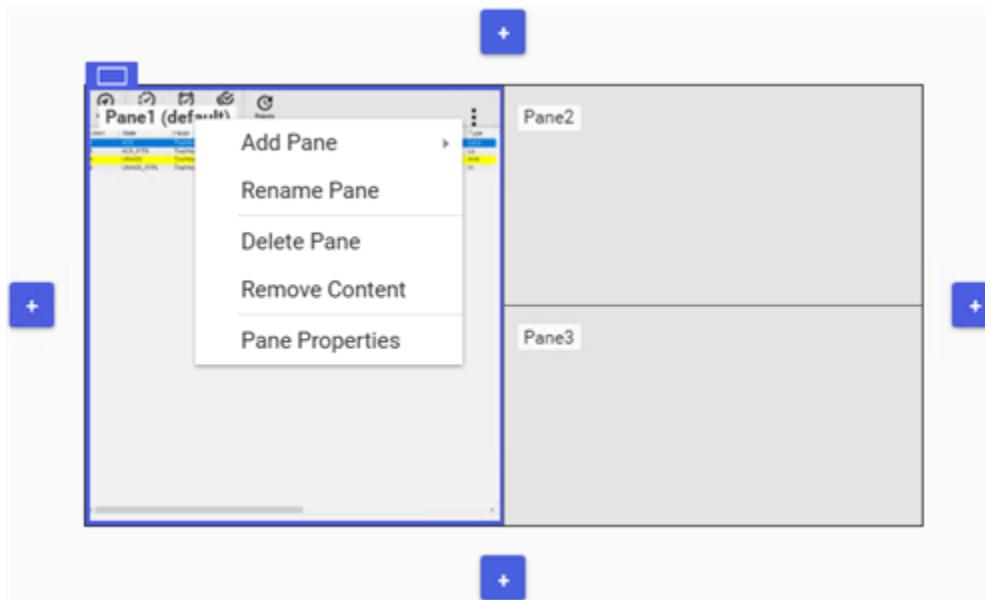
In Edit Content mode the shortcut menu provides a single action to clear the content from the selected pane.

Pane Menu Button

The Pane Options button appears at the top right corner of a pane when a pane is selected within a layout,. When a pane is not selected, placing the mouse over it shows its **Pane Options** button. Clicking or tapping the button shows a list of the same actions as a shortcut menu.



In Edit Content mode, the Pane Options button includes additional commands to remove content from the pane and edit the pane's properties.



Keyboard Shortcuts

The following keyboard shortcuts are available when a pane is selected in Layout Edit mode:

- Delete key: Remove the selected pane and all of its content.
- F2 key: Rename the selected pane
- Alt + Up, Down, Left, or Right: Split the pane in that direction if applicable.

Configure the Layout Editor

You can use a set of controls to adjust the editing areas of the Layout Editor while you are working on a layout.

Adjust the Size of Layout Editor Areas

The Layout Editor consists of three columns that show an Action List, a graphic of the layout being edited, and a tabbed area at the right to show properties.

Dividers are vertical or horizontal lines that separate the different areas of the Layout Editor. You can resize a Layout Editor area by moving its divider with a drag and drop mouse action or by touch.

Place the cursor directly over a divider to select it. The shape of the cursor changes to a double arrowhead when a divider is selected. Using a mouse, left click and hold to drag the divider to another position. Using touch, select the divider and move the divider with your finger to another position.

Show or Hide Layout Editor Fields

A triangle icon to the left of the name of a Layout Editor field indicates the field can be expanded or hidden by selecting the triangle.

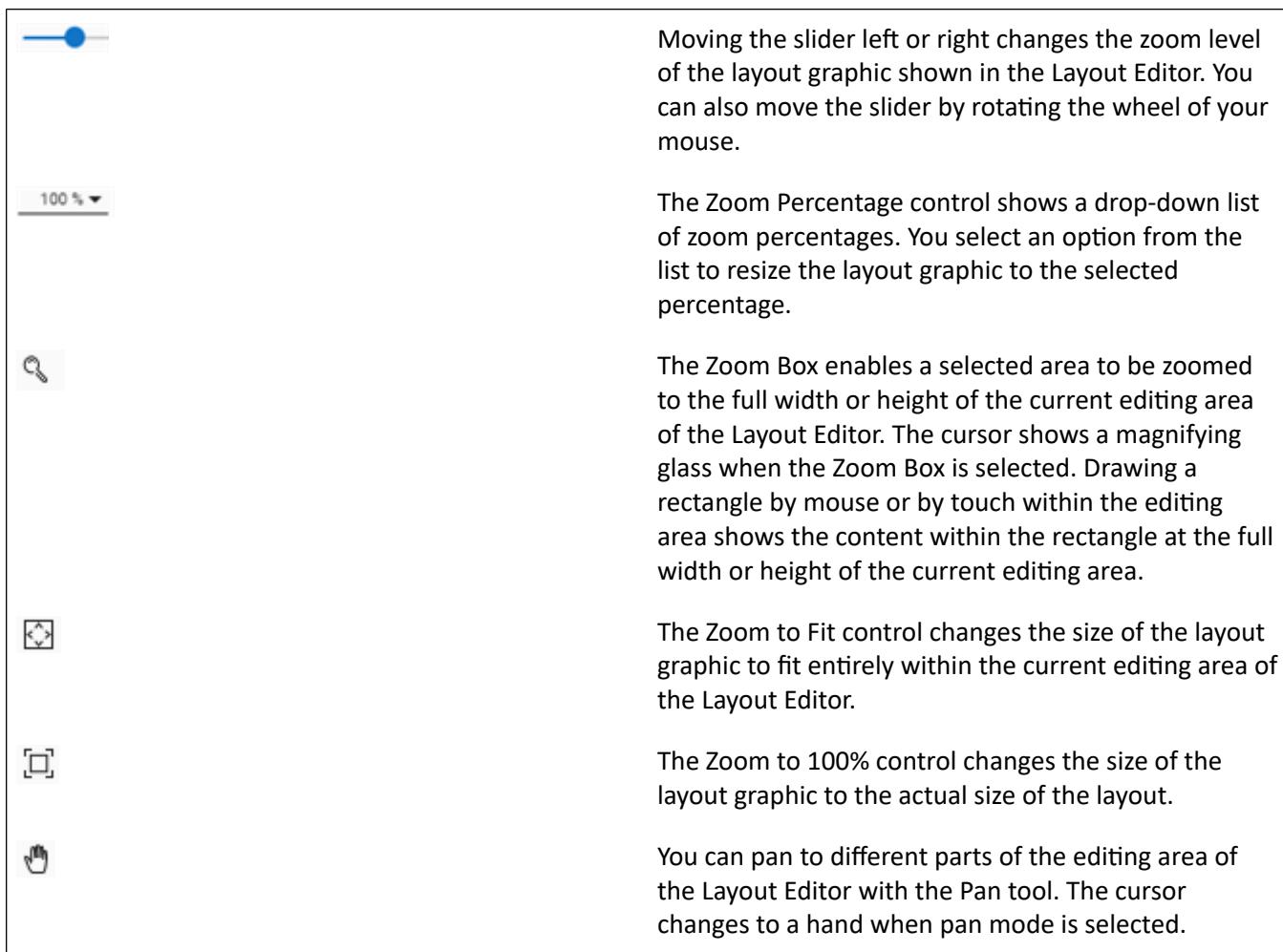


The triangle points down when the Layout Editor field is expanded to show its content.

Change the Size of a Layout Shown in the Layout Editor

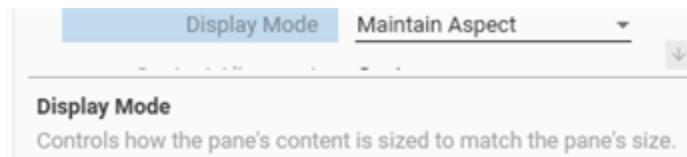
The middle area of the Layout Editor shows a graphic representation of the layout currently being edited. A set of pan and zoom controls appear at the bottom of the field to change the size of the Layout graphic. Panning or zooming a layout is useful when the layout is designed for a much larger or smaller screen than the editing area of the Layout Editor.





View Descriptions of Layout Properties

You can see a brief description of a selected property by selecting the name of a property shown in the **Properties** grid view. The description appears beneath the list of properties.



Create a layout 1

You access the Layout Editor from the **Visualization folder** to create a screen layout. After creating a layout, you continue editing and configuring the layout by selecting it from the content list shown in the **Visualization folder**. For more information, see [Create a layout](#).

Create a layout 2

Video Tutorial: Create a layout in AVEVA OMI

https://player.vimeo.com/video/992308927?badge=0&autoplay=0&player_id=0&app_id=58479

You create a layout from the **Visualization folder** of the IDE. After a layout is created, it appears in the list of visual objects in the **Visualization folder** with an identifying icon.



To create a layout

1. Open the **Visualization folder**.
2. On the **Home** ribbon, in the **Create** area, select **Layout**.

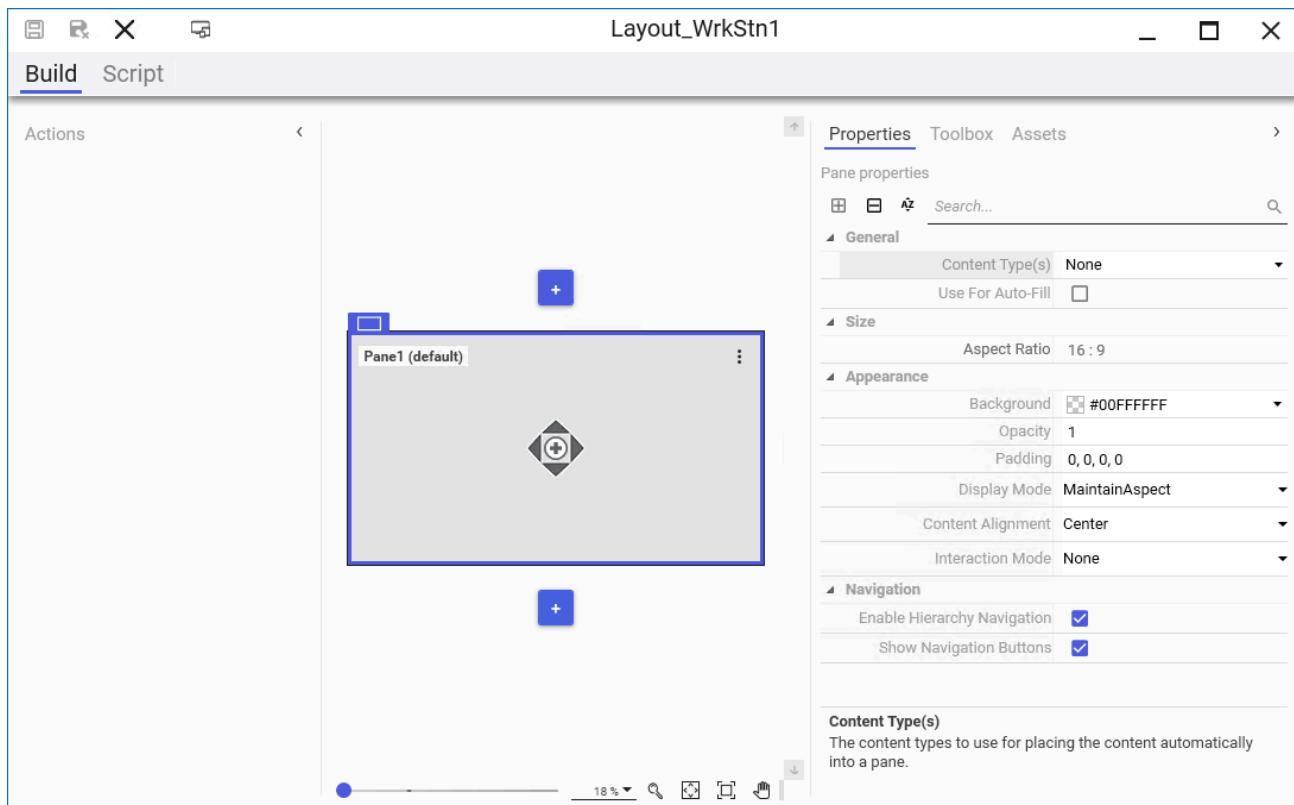
You can also create a layout by other methods:

- **Keyboard shortcut:** Ctrl + Shift + L
- **Shortcut menu:** Right-click in the **Visualization folder** and select **New**, then **Layout**.

A layout appears in the **Visualization folder** list with a default name of Layout_001 for the first layout that has been created. A blue background around the layout name indicates the name can be changed.

3. Select the layout from the list and rename it.
4. Double-click the layout or right-click and select **Open** to show the layout in the Layout Editor. You can also press Enter to open a layout in the Layout Editor.

After opening a layout for the first time, the Layout Editor shows a single pane with its properties listed in the right column of the editor.



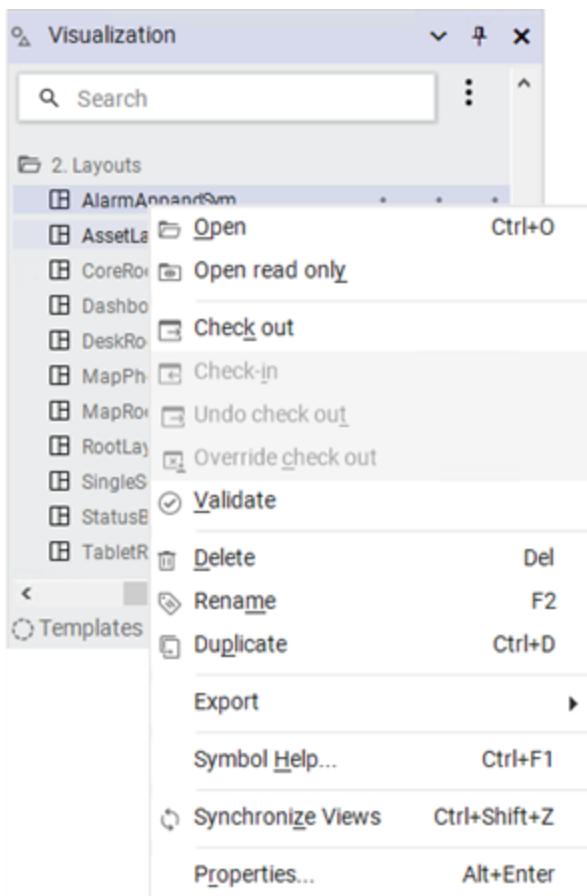
You can begin configuring the layout for your View Applications. For more information about configuring a layout profile, see [Configure a layout](#).

5. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

Manage a layout

You can perform routine tasks to manage a layout from the **Visualization folder**. Management tasks include:

- [Delete a layout](#)
- [Rename a layout](#)
- [Duplicate a layout](#)
- [Export a layout](#)
- [Import a layout](#)



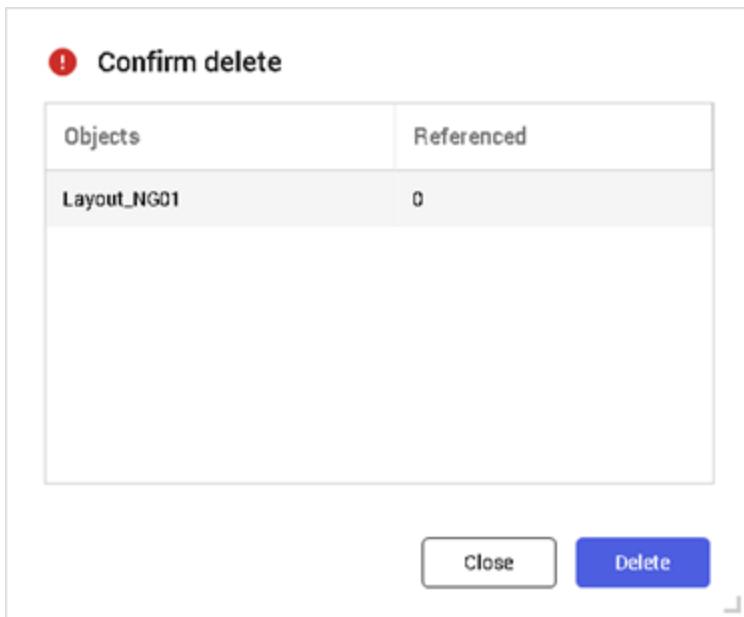
Delete a layout

You can delete a layout from the Visualization folder.

To delete a layout

1. Open the **Visualization folder**.
2. If the layout you want to delete is open for editing, close it.
3. Select the layout you want to delete.
4. Press the Delete key or right-click the layout and select **Delete**.

The **Confirm Delete** dialog box appears. The **Referenced** column shows the number of objects that the layout is referenced by. To see what those objects are, click the down-arrow at the right side of the **Referenced** list. This shows any screen profiles, ViewApp templates, or ViewApp instances that refer to the layout.



5. Click **Delete** to delete the layout.

Duplicate a layout

You can duplicate to reduce the effort to create an entirely new layout by using an existing layout as a template.

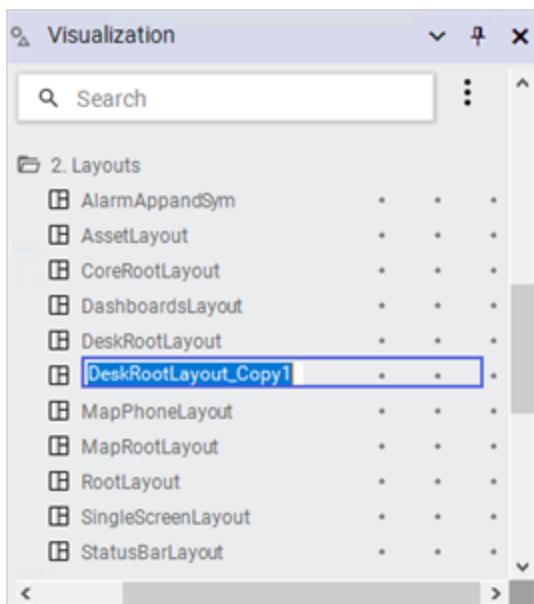
To duplicate a layout

1. Open the Visualization folder and select the layout that you want to duplicate.
2. On the **Home** ribbon, in the **Edit** area, select **Duplicate**.

You can also duplicate a layout by other methods:

- **Keyboard shortcut:** **Ctrl + D**
- **Shortcut menu:** Right-click the name of the layout and click **Duplicate**.

A new layout appears in the Visualization folder list with Copy1 appended to its name.



3. Select the duplicate layout and rename it.

Rename a layout

You can select a layout from the Visualization folder list and rename it by pressing F2 or by clicking on the name to enter edit mode.

A new layout name is validated to ensure it meets naming requirements. If a new name is a duplicate of an existing name or breaks any name rules, a message appears indicating there is a problem with the name.

Pressing the Esc key cancels a rename request and the layout reverts to the original name. Committing an empty name cancels the rename request.

To rename a layout

1. Select the layout that you want to rename from the **Visualization folder** list.
2. Click the layout name or press **F2** to enter edit mode. A blue background appears around the layout name.
3. Type the new name of the layout.
4. Commit the new name by clicking outside of the layout name.
If the new name is valid, it appears in the **Visualization folder** list. If the name is invalid, you see a dialog box that describes the problem.
5. If you typed an invalid layout name, click **OK** and re-enter another name.

Export a layout

You can export one or more layouts from the Visualization folder to an aaPKG file. You can then import the layouts from the export file to another Galaxy.

To export a layout

1. Open the **Visualization folder**.

2. Select the layouts to be exported.
3. In the **Home** ribbon **Export** area, select **Selected**, then **As package**.

You can also export a layout from the shortcut menu by right-clicking the layout name, then selecting **Export**, and then **Selected objects**.

The **Export Selected objects** dialog box appears.

4. Select the folder where you want to save the export file.
5. Assign a name to the export file.
The default export file name is the name of the first selected layout from the **Visualization folder**.
6. Click **Save**.

A horizontal bar shows the progress of the layouts being loaded into the export file.

Import a layout

You can import a layout created in another Galaxy to the Visualization folder of the System Platform IDE. Exported layouts are saved in an aaPKG file.

During the import process, you have the choice to keep or overwrite any layouts currently in the Visualization folder with the imported layouts.

To import one or more layouts

1. Open the **Visualization folder**.
2. Select the **Galaxy** ribbon, then select **Import**.
3. Select **Objects**, then, **From package**.
The **Import Objects from package** dialog box appears.
4. Go to the folder where the aaPKG file containing layouts is located.
5. Select the aaPKG file and click **Open**.

The **Import preferences** dialog box opens with options to import layouts. You cannot have two layouts with the same name or more than one copy of the same version of a profile in the same Galaxy. When you import a layout, you can choose options from the **Import preferences** dialog box how you want naming and version conflicts handled.

Import preferences

Objects with same Tagname and Codebase as an existing object

Skip: Do not import
 Overwrite objects if the imported configuration version is higher
 Overwrite objects regardless of configuration version

Base Templates with a different revision number in the Codebase or a different minor version

Skip: Do not migrate
 Migrate

Objects with same Tagname but with a different Codebase

Skip: Do not import
 Rename object in Galaxy
 Rename importing object

Append to object name

Template Protection Change Management

Never overwrite an unprotected object with a protected object

6. Click **Import** to import the layouts.

A progress bar shows the layouts being imported into the Visualization folder. After finishing the import, you should see the layouts in the list of Visualization folder objects.

Configure a layout

The Layout Editor includes several editing modes to change the characteristics of the entire layout, a specific pane in a layout, a pane divider, and pane content. The following table lists the different methods of selecting an edit mode within the Layout Editor.

Edit Modes	Edit Mode Selection Methods
Layout	<ul style="list-style-type: none">Click or tap outside the boundary of a layout shown in the Layout Editor.
Pane	<ul style="list-style-type: none">Click or tap an empty pane to show the selected pane's properties.

	<ul style="list-style-type: none"> Select Pane Properties from the Pane Options menu of a pane containing content
Divider	<ul style="list-style-type: none"> Click or tap on a pane divider within a layout.
Content	<ul style="list-style-type: none"> Click or tap within a pane containing content to edit content properties. Select Edit content_name from the Pane Options menu to edit graphic content within the Industrial Graphic Editor.

When switching edit modes, the **Property** field automatically adjusts to show the properties appropriate for the selected object in the Layout Editor.

This topic describes the different tasks to configure the physical appearance of a layout. For more information about editing the properties of a layout or a pane, see [Set layout properties](#) or [Set pane properties 1](#). See [Add content to panes](#) for the steps to add content to a pane.

Add panes to layout

Fixed Layout Mode

When in **fixed layout mode**, you can add panes by a variety of methods:

Method	Add Pane Above	Add Pane Below	Add Pane Right	Add Pane Left
Add Pane Control				
Keyboard Shortcuts	Alt + Up	Alt + Down	Alt + Right	Alt + Left
Shortcut Menu	Right-click Add Pane , then Above	Right-click Add Pane , then Below	Right-click Add Pane , then Right	Right-click Add Pane , then Left
Pane Options Button	Click Pane Options button, select Add Pane , then select Above	Click Pane Options button, select Add Pane , then select Below	Click Pane Options button, select Add Pane , then select Right	Click Pane Options button, select Add Pane , then select Left

Initially the Layout Editor shows a single pane. As panes are added to a layout, dividers are automatically created between adjacent panes. Dividers are one pixel thick, but can be changed by setting the divider's **Thickness** property. By default, the newly added pane will be one pixel less in size.

When a pane is added, the Layout Editor divides the selected pane in half regardless of whether it is variable or fixed. If the selected pane has a minimum size that is more than half its current size, then the selected pane will

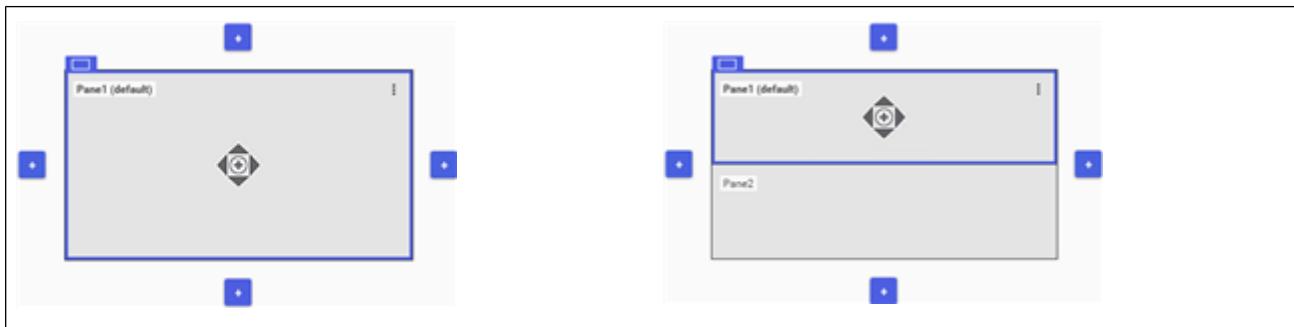
be set to its minimum size and the new pane will take the remaining available size.

To add a pane to a fixed layout

1. Open a layout from the **Visualization folder**.

The **Add Pane** control appears in the center of the default pane with arrows to create a new pane above, below, right, or left of the current pane.

2. Select an arrow of the **Add Pane** control to add a pane to the layout.



The new pane that is added is named **Pane<n>**, where *n* is a number from 1 to 50.

Note: A fixed layout supports up to a maximum of 50 panes. You will not be able to add panes beyond the 50 pane limit.

3. Select **Save** to save your changes to the layout.

Responsive Layout Mode

When in **Responsive layout mode**, just click the **Add Pane** button to add a new pane. Each new pane is added below the last pane added.



To add a pane to a responsive layout

1. Open a layout from the **Visualization folder**.

2. Switch to responsive mode by enabling the **IsResponsive** layout property, or by selecting the **Fixed/Responsive Layout** button. The **IsResponsive** property is visible when layout properties are selected (click above the layout to switch to the layout properties view).

3. Press the **Add Pane** button.

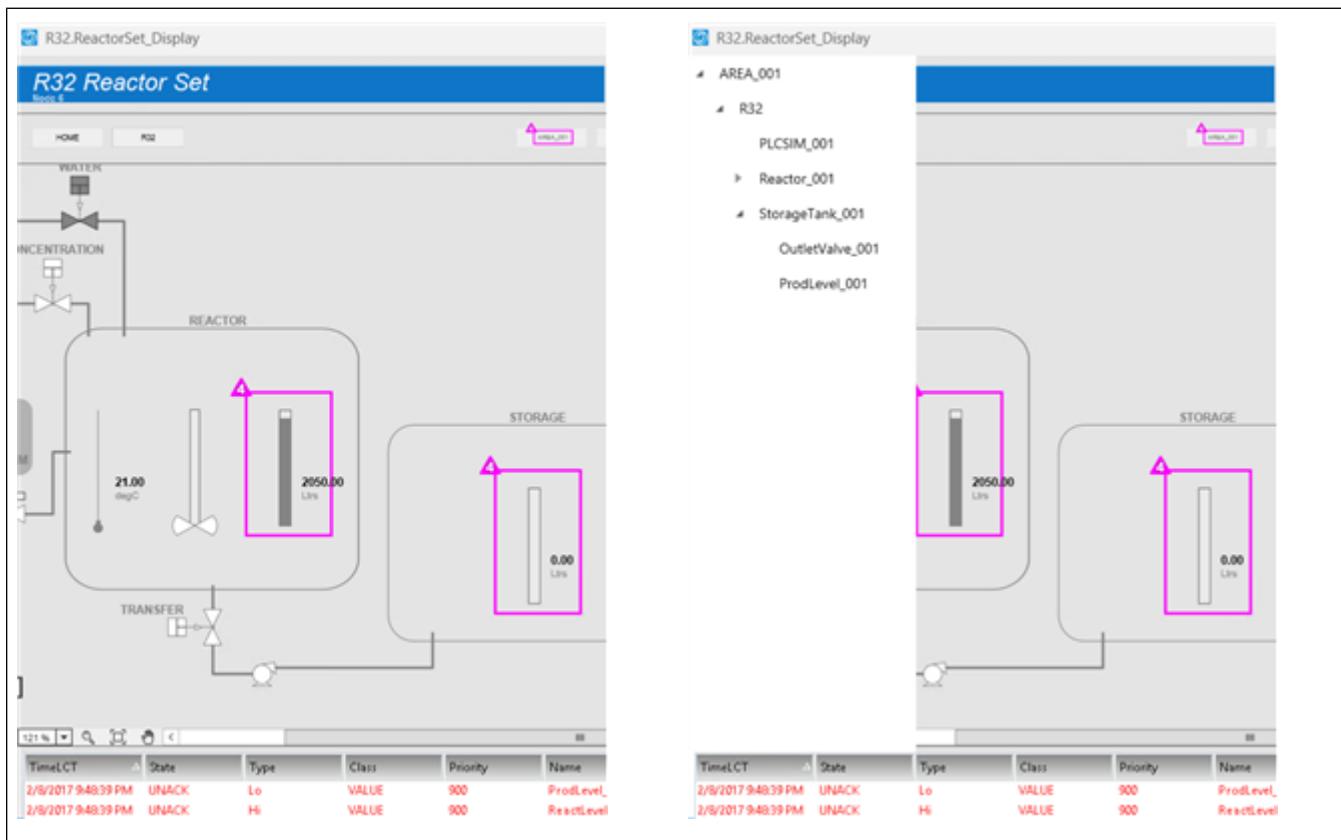
The new pane that is added is named **Pane<n>**, where *n* is an integer.

Note: A responsive layout does not automatically limit the number of panes you can add. However, as you add panes and content, more computer resources are consumed which can impact performance, both during configuration and at runtime.

Configure a layout: slide-in panes

A layout can have a slide-in pane configured at each side of a layout window. During runtime, slide-in panes are hidden until the user selects the tab at the side of a layout to extend the slide-in pane and show its contents. In

In the example below, selecting the tab at the left border of the layout shows a NavTree control in a slide-in pane. Clicking or tapping outside of the slide-in pane hides it.



Configure a layout: add slide-in panes

You can add up to a maximum of four slide-in panes to a layout.

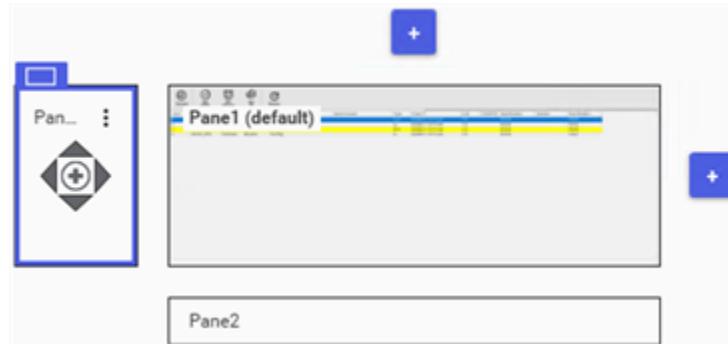
To add slide-in panes to a layout

1. Open a layout in the Layout Editor.

The Layout Editor shows four Add buttons with a plus sign at each side of a layout.

2. Select the square on the side of the layout in which you want to add a slide-in pane.

The added slide-in pane appears next to the layout.



All standard layout configuration operations can be done to a slide-in pane. The one exception is that a slide-in

pane cannot be set as the layout's default pane.

Slide-in panes are included in a layout's maximum pane count. If the maximum pane count has been reached, the pane Add buttons are disabled within the Layout Editor.

A slide-in pane always spans the full height or width of a layout window. A slide-in pane can be divided into smaller panes, but the frame of the slide-in pane is always the full width or height of the layout window.

The initial width of a slide-in pane on the left or right is 25% of the layout's width, and the initial height of a slide-in pane on the top or bottom is 25% of the layout's height.

When the dimensions of a layout are resized, slide-in panes are resized accordingly. When the width is changed, the top or bottom slide-in panes are resized to match the new width. When a layout's height is changed, the left or right slide-in panes are resized to match the new height.

Right clicking a pane within the slide-in pane or selecting the **Pane Options** menu button and selecting **Delete Slide-In Pane** removes the slide-in pane from the layout. Any navigation actions associated with the slide-in pane are removed from the Action List. The default pane in the layout is selected and the layout is zoomed to fit.

Resize slide-In panes

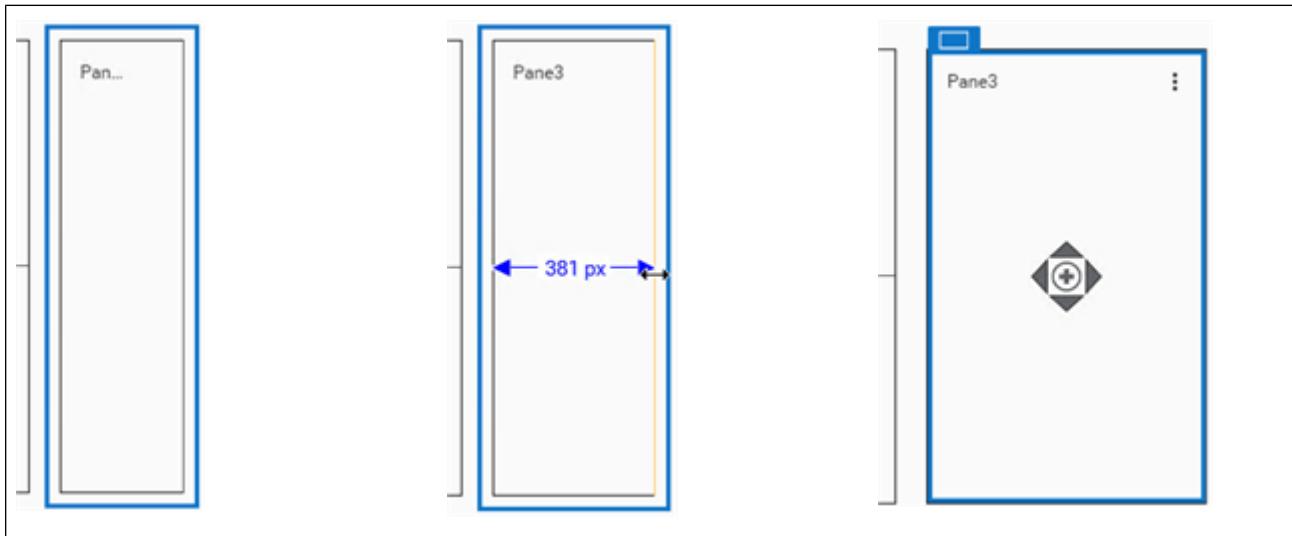
By default, a slide-in pane extends to 25 percent of a layout's width or height during runtime. You can resize slide-in panes with the Layout Editor. The upper and lower slide-in panes can be resized vertically to change their height. The left and right slide-in panes can be resized horizontally to change their width.

The size of a slide-in pane can be increased up to the maximum layout width or height minus 16 pixels. This small 16 pixel band provides sufficient space within the layout window for the user to click outside of the slide-in pane to hide it during runtime.

To resize a slide-in pane

1. With the Layout Editor, open the layout containing the slide-in pane to resize.
2. Tap or click the outside border of the slide-in pane.

The cursor changes to a double arrowhead and the outside border of the selected slide-in pane changes color to indicate the pane can be resized. The pixel width of the slide-in pane also appears while it is being resized.



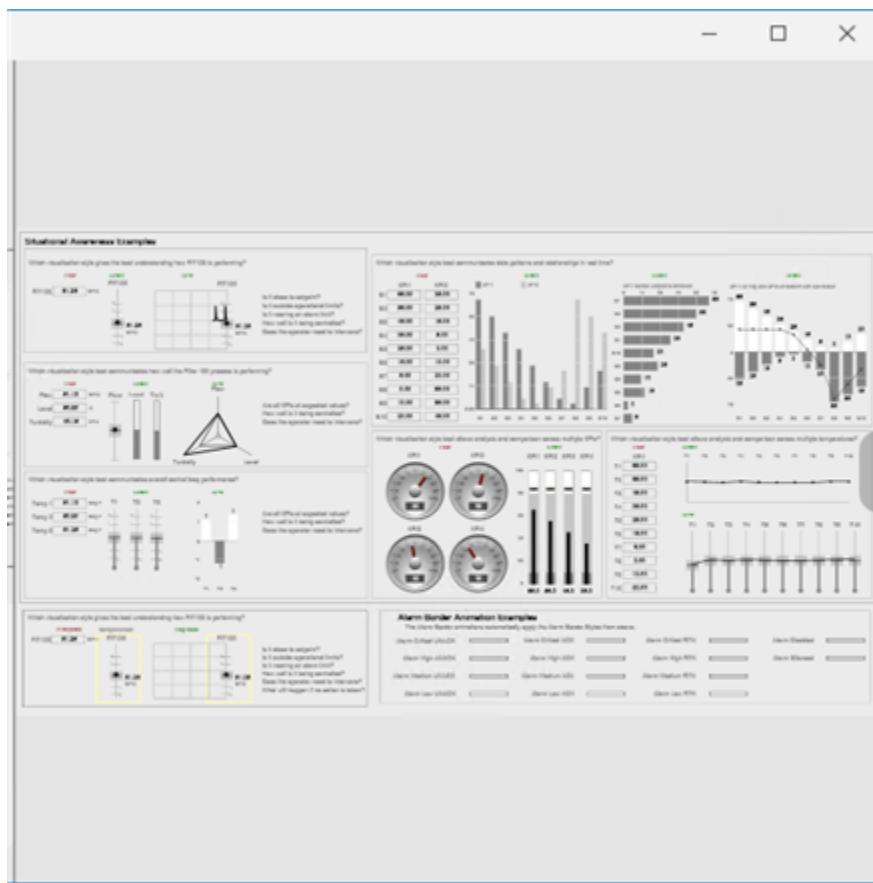
3. Drag the pane border to the desired position to change the size of the pane.

4. Save your changes to the layout.

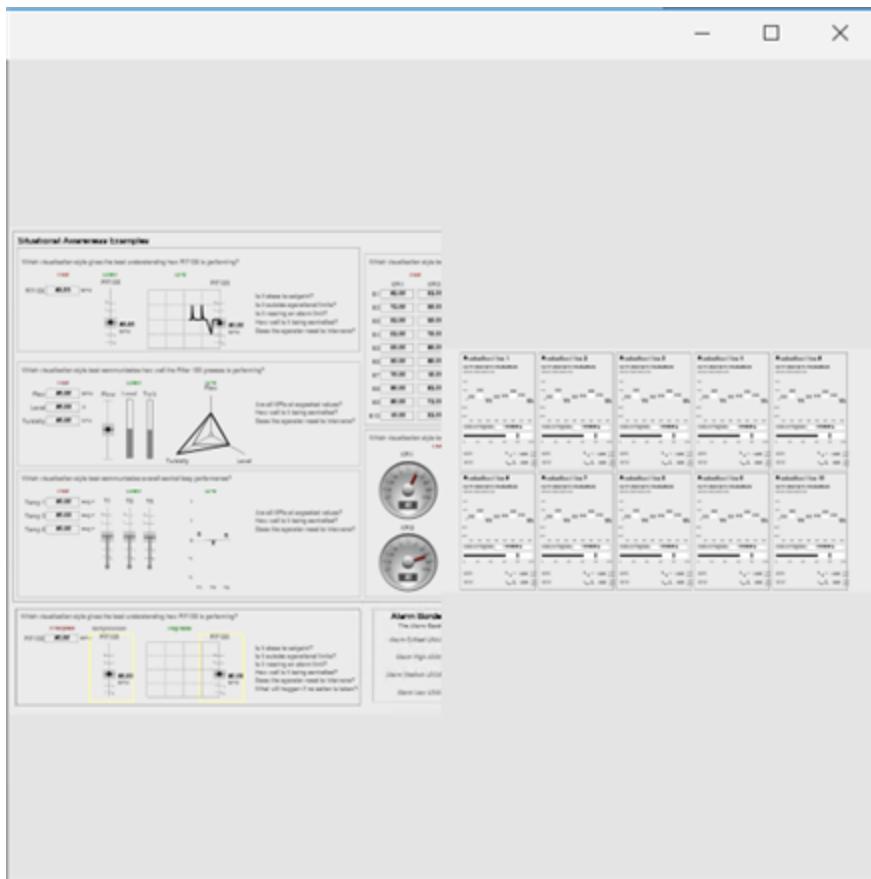
Show slide-in panes at runtime

During runtime, users can show a slide-in pane by one of the following methods:

- Click or tap on the slide-in pane tab.
- Press the keyboard shortcut Alt+Shift+<Arrow>, where <Arrow> is the arrow key in the direction away from the side with the slide-in pane. For example, if you have a slide-in pane at the left side of a layout, press the right arrow key.
- Perform a three-finger swipe gesture in the direction away from the side of the layout containing a slide-in pane.



The slide-in pane and its contents appear over the existing pane.



The background of the whole slide-in pane is the same as the layout's configured background color. If the layout's background color is transparent, the slide-in pane has an opaque white background behind it.

When the layout window is not the same size as the layout's configured size, the slide-in pane size changes to remain at the same proportion as the window size.

Users can hide a slide-in pane while the ViewApp is running by one of the following methods:

- Click or tap outside of the slide-in pane, including changing focus to another window.
- Press the Esc key.
- Perform a three-finger swipe gesture towards the side of the ViewApp window containing a slide-in pane.

Rename a pane

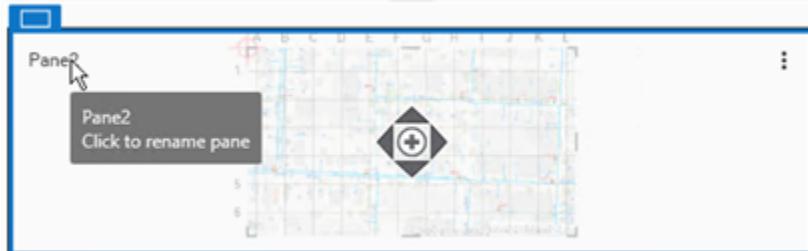
You can rename a pane within the Layout Editor by three methods.

Current Pane Name	Click or tap on the current pane name to type a new name.
Pane Options Menu	Click or tap the Pane Options help and select Rename Pane .
Shortcut Menu	Right-click within the pane and select Rename Pane .

To rename a pane

1. Open a layout from the **Visualization folder**.
2. Select the pane to rename.

When you place the cursor directly over the current pane name, a tooltip indicates the pane can be renamed.



3. Click the name of the pane.

A blue background behind the name of the pane indicates you are in edit mode.

4. Type a new name for the pane.

You can press Esc to cancel the rename request and keep the pane's current name.

5. After typing the new name, click outside of the name box or press Enter to commit the new name.

If you typed an invalid pane name, a red border appears around the edit box and a tooltip describes the problem.

6. Click **Save** to save the new pane name.

Remove a pane from a layout

You can remove a pane from a layout. If the removed pane is the default pane, the remaining pane with the largest space within the layout is assigned as the default pane. The freed space of the removed pane is assigned to the layout's remaining panes.

You can remove a pane from a layout by three methods.

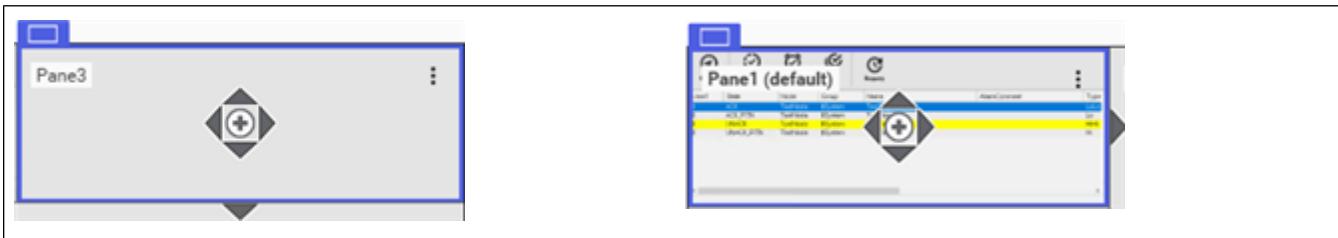
Delete Key	Select the pane to delete and press the Delete key.
Pane Options Menu	Click or tap the Pane Options button and select Delete Pane .
Shortcut Menu	Right-click the pane and select Delete Pane .

To remove a pane from a layout

1. Open a layout from the **Visualization folder**.
2. Select the pane you want to remove from the layout.
3. Select the **Pane Option** button to show a list of pane commands.
4. Select **Delete Pane** from the list.
The pane is removed immediately.
5. Click **Save** to save your changes to the layout.

Extend panes in a layout

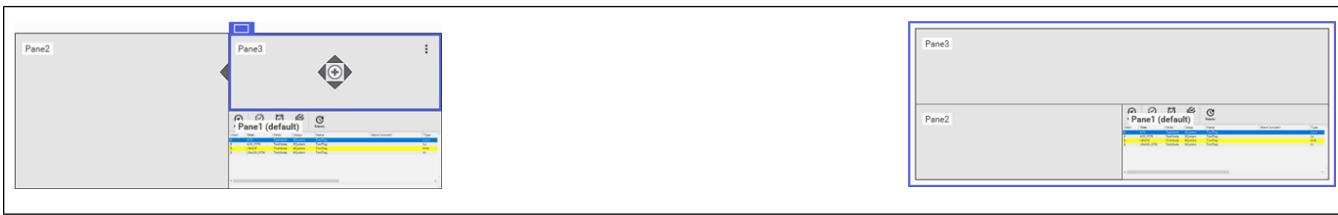
Extenders are triangular arrowheads that appear on vertical and horizontal dividers between panes of a layout.



An extender enables the source pane to extend into the area of a target pane. In the example below, Pane 2 is the source pane that will be extended into Pane 1, which is the target pane.



Extending a pane effectively rotates the orientation of the first parent common to both the source and target panes. In the following example, Pane 1 is the parent of Pane 2 and 3. When Pane 3 is extended into Pane 2, Pane 1's orientation changes from a column to a row.



For more information about the rules of extending panes, see [Extend panes in a layout](#).

To extend a pane in a layout

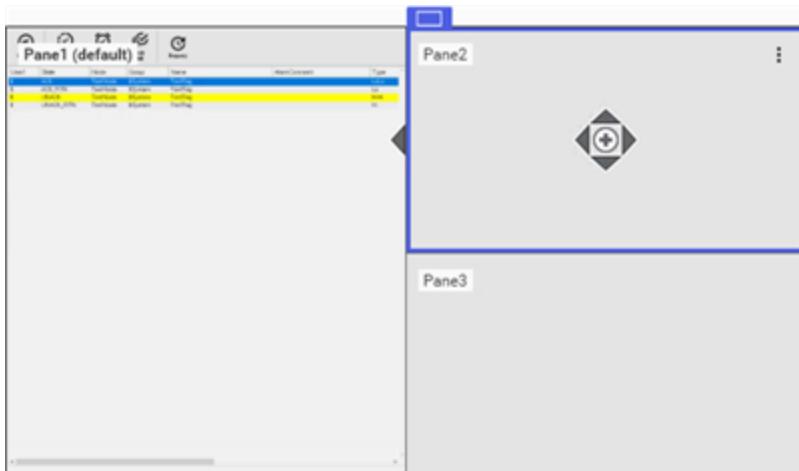
1. Open the layout from the **Visualization folder**.
The default pane of the layout is selected.
2. Select the pane to be extended.
An extender appears on a pane border that can be extended.
3. Tap or click the extender to extend the pane
4. Save your change.

Extend panes in a layout

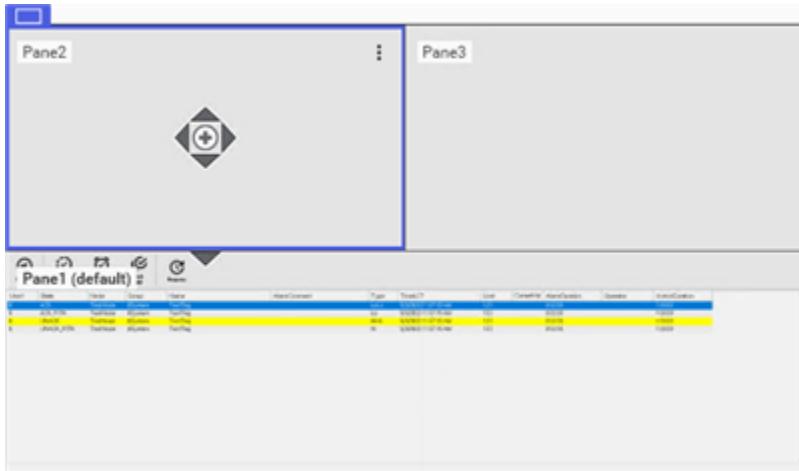
A layout shows panes arranged in vertical columns or horizontal rows. The Extend Pane command enables a source pane fitting the correct criteria to be extended Up/Down/Left/Right into a sibling target pane. In the following examples, Pane 1 is the parent pane.

A pane can be extended based on the following rules:

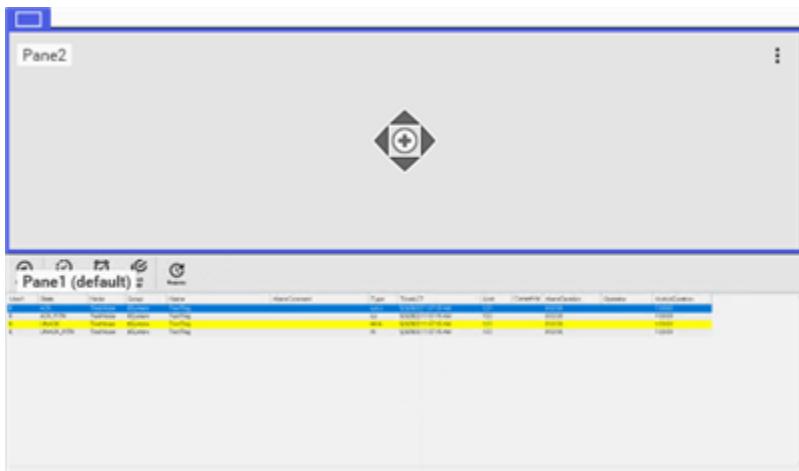
- A pane that is a row of its parent pane can be extended left or right



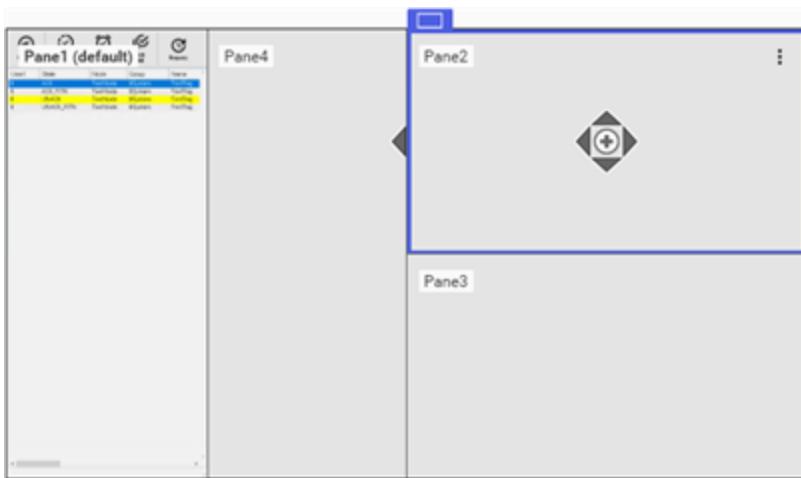
- A pane that is a column of its parent pane can be extended up or down



- A pane adjacent to the border of a layout window cannot be extended outwards beyond the layout's border.
- A pane that is an immediate child of its parent pane cannot be extended



- A pane can be extended into its parent pane's immediate siblings



- A pane cannot be extended if the number of panes will exceed the maximum of 50

Set the default pane of a layout

A layout includes a default pane, which should be the pane that is most likely to be updated with changed content during runtime. A default pane is commonly used with navigation actions when no pane is specified.

A layout containing only one pane is the default. If you add one or more panes, the original Pane 1 is designated as the default pane. You can assign a new default pane if you do not want to use Pane 1 as the default pane.

To select the default pane of a layout

1. Open the layout from the **Visualization folder**.
2. Select the pane that you want to assign as the default pane.
3. Right-click the selected pane to show the shortcut menu.
4. Select **Set as Default Pane**.
The name of the pane is shown in bold and the word **default** appears in parentheses to the right of the pane name.
5. Select **Save** to save your changes.

Set the pane presentation style

Content in a layout can be shown by three pane presentation styles

- **Single**

In Single mode, the pane hosts a single content item (Layout, Display, or Graphic) shown during runtime. Single is the default pane presentation style.

- **Multiple**

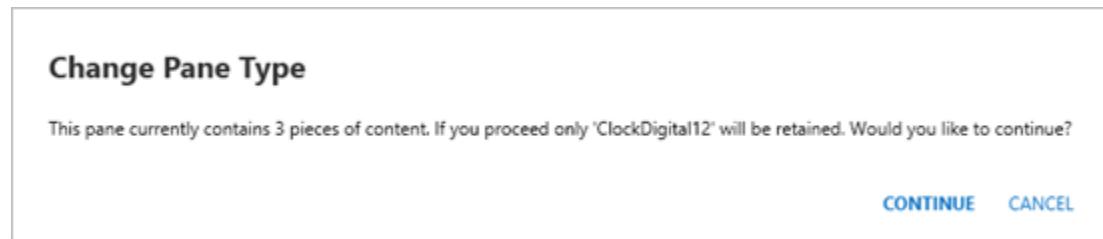
In Multiple mode, two or more content items are added to a multi-content pane by a script or a navigation API. Each content item is placed in a separate pane. During runtime, the user cannot select an item from Multiple panes. A script or navigation API controls the ordering of the panes within the stack. The content in the top most pane is visible during runtime.

- **Tabbed**

In Tabbed mode, two or more content items are added to a multi-content pane. Each content item is placed in a separate tabbed pane. During runtime, the user selects a tab to show its content in the pane.

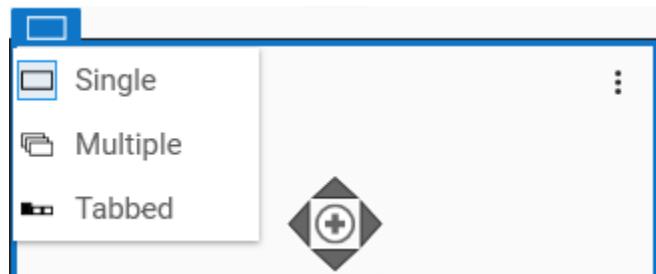


When you change a pane's presentation style from Single to either Multiple or Tabbed, any existing content in the pane is maintained. However, when you change the presentation style from Multiple or Tabbed to Single, you see a confirmation message that all content except the current selection in the action list will be removed from the pane.



To set the presentation style of a pane

1. Open the layout from the **Visualization folder**.
2. Select the pane to set a presentation style for.
3. Select the **Presentation Style** selector at the top left corner of the selected pane.



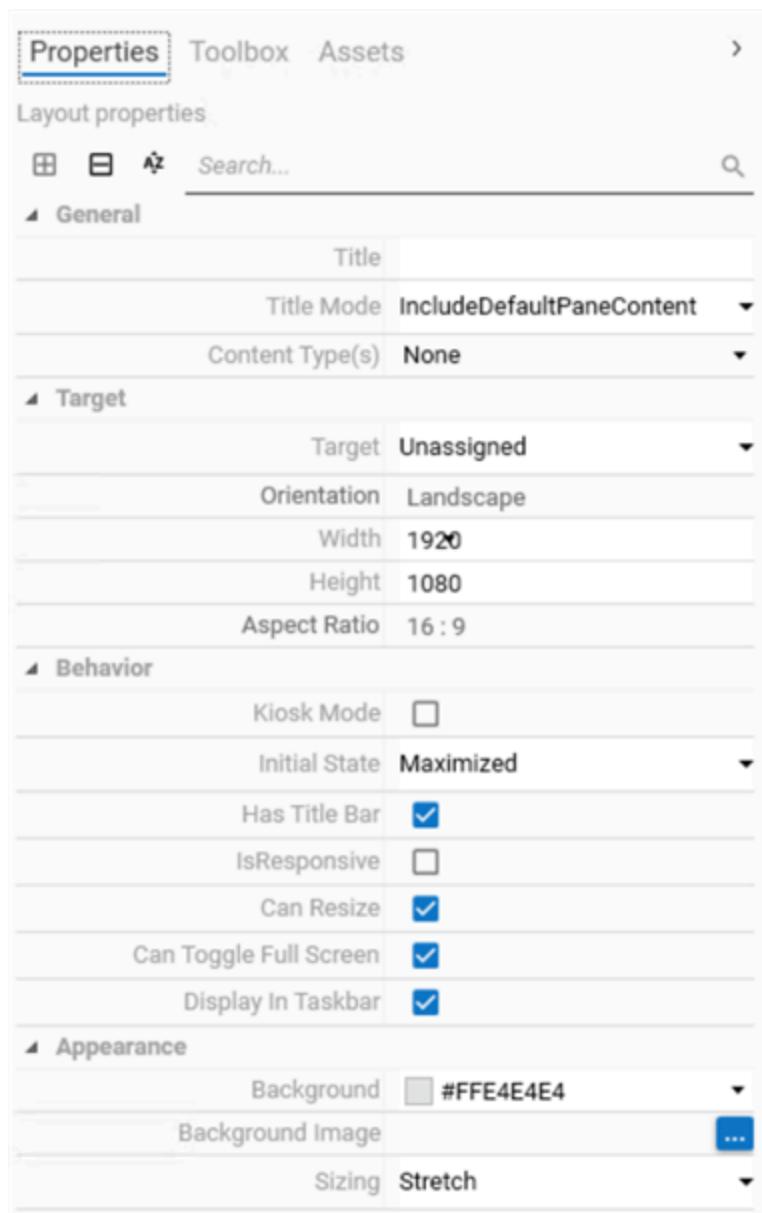
4. Select a presentation style from the list.
The icon shown in the **Presentation Style** selector changes to match the presentation style you selected.
5. Select **Save** to save your changes.

Set layout properties

Layout properties define the overall visual characteristics and runtime behavior of a layout. For information about pane properties, see [Set pane properties 1](#).

You can show layout properties from the Layout Editor by clicking or tapping outside of the displayed layout.

The following table shows the default values of layout properties. When no pane or divider is selected, layout properties show the properties of the overall layout.



Other property fields appear when some layout properties are assigned specific values. The following table lists any prerequisite property values in the **Prerequisite Property Values** column.

The following table lists all layout properties. The properties are organized into the categories shown in the **Properties** window of the Layout Editor. You can also view layout properties in alphabetical order by selecting the **Show Alphabetical** icon next to the **Search** field.

You type values or select values from drop-down lists to assign values to layout properties. A red background appears when you enter an invalid property value.

Short descriptions of the properties appear at the bottom of the **Properties** window for each selected property.

Property Category	Property	Default Property Value	Prerequisite Property Values	Description
General				

	Title	<Empty>	None	Text shown in the title bar of the layout's window during runtime.
	Title Mode	IncludeDefaultPaneContent	None	<p>Options specify whether the name of the default content shown in the default pane is appended to the title of the layout or not.</p> <p>TitleOnly: Shows only the title assigned to the layout.</p> <p>IncludeDefaultPane Content: The name of the content placed in the default pane is appended to the title of the layout window.</p>
	Content Type	<Empty>	None	Specifies the type of content represented by a layout. Select the arrow at the right of the Content Type field to show a drop-down list of content types. One or more content types can be selected from the list.
Target				
	Target	Unassigned	None	Specifies if the layout is intended for a screen, a pane of another layout, a pop-up, or is unassigned.

	Screen Profile	N/A	Target=Screen	Name of the screen profile containing a screen that the layout is intended to be displayed on. For more information about associating a screen profile to a layout, see Select a screen profile layout .
	Screen	N/A	Target=Screen	Name of the screen within a Screen Profile that the layout is designed to be displayed on.
	Layout	N/A	Target=Pane	Name of the layout containing the target pane that this layout is designed to be hosted in.
	Pane	N/A	Target=Pane	Name of the pane within the selected layout that this layout is designed to be hosted in.
	Orientation	N/A	Target=Screen or Target=Pane or Target=Popup	Orientation of the layout(Portrait or Landscape) determined by the selected screen profile screen or pane. Read only.
	Width	N/A	None	Width of the layout in pixels.
	Height	N/A	None	Height of the layout in pixels.
	Aspect Ratio	1.91:1	None	Calculated aspect ratio of the width to height size of a

				layout.
Behavior				
	Kiosk Mode	False	None	When set to True, the System Menu, and Window Controls (Maximize/Restore, Minimize and Close) buttons and their underlying functions are inaccessible during runtime.
	Initial State	Maximized	Kiosk Mode=False	Initial state of a layout's window during runtime. The initial state of the layout window can be one of the following: Maximized Full Screen Normal Minimized
	Has Title Bar	True	Kiosk Mode=False	When set to True, a title bar appears at the top of the layout's window, which is 33 pixels in height. As a result, the height of a layout designed for a screen resolution of 1920 x 1080 is adjusted to 1047 pixels to accommodate the height of the title bar.
	IsResponsive	False	Kiosk Mode=False	When set to True, the Layout editor enters responsive

				layout mode. For more information about Responsive Layouts, see About the responsive layout mode . See the following table for a set of layout properties that are unique to responsive layouts and only appear when the IsResponsive property is set to True.
	Can Resize	True	Kiosk Mode=False	When set to True, the user can resize the running View Application window by dragging the horizontal or vertical border. Also, selecting either the Minimize or Maximize Windows can be selected to resize the layout.
	Can Toggle Full Screen	True	Kiosk Mode=False	When set to True, the user can press the F11 key to toggle the running View Application window in and out of full screen mode.
	Display In Taskbar	True	None	When set to True, the layout's window will appear in the Windows taskbar.
Appearance				
	Background	#FFE4E4E4	None	Background color of a layout or a pane. Select the arrow at the right of the

				property field to show a color picker to set a background color.
	Background Image	<Empty>	Valid image file types / extensions: bmp, jpeg, jpg, gif, png, tif, tiff, ico, wmf, emf	Image file used as the background of a layout. The image spans all panes of a layout. Select the file picker button to open a file browser window, then navigate to an image file and select Open to set the background image. To remove a background image, select the X that appears next to the image name.
	Sizing	Stretch		Sets the method for resizing the Background Image file to fit the layout. Stretch: matches the height and width of the image file with the layout size. This can skew the image aspect ratio. Horizontal: matches the width of the image file to the width of the layout. This preserves the image aspect ratio, but can result in empty space above and below the image (letterboxing), or clipping the top and bottom of the image.

				<p>Vertical: matches the height of the specified image file to the height of the layout. This preserves the image aspect ratio, but can result in empty space to the right and left of the image (letter-boxing), or clipping the sides of the image.</p> <p>Tile: preserves the original height and width of the specified image file, but tiles (repeats) it to fill the entire layout.</p> <p>If any blank space appears around the image due to letter-boxing, the background color of the layout is shown.</p>
--	--	--	--	---

The following table shows layout properties that apply only to responsive layouts. The **IsResponsive** property must be set to True to show these properties.

Property Category	Property	Default Property Value	Prerequisite Property Values	Description
Extra Small				
	Column Span Extra Small	12	IsResponsive=True	Number of columns a pane can span when a responsive layout is displayed on an extra small screen.
	Column Push Extra Small	0	IsResponsive=True	Number of columns to push a responsive layout pane when displayed on an extra small screen.

	Column Pull Extra Small	0	IsResponsive=True	Number of columns to pull a responsive layout pane when displayed on an extra small screen.
	Pane Offset Small	0	IsResponsive=True	Number of columns to offset a responsive layout pane when displayed on a small screen.
	Pane Offset Extra Small	0	IsResponsive=True	Number of columns to offset a responsive layout pane when displayed on an extra small screen.
Small				
	Column Span Small	0	IsResponsive= True	Number of columns a pane can span when a responsive layout is displayed on a small screen.
	Column Push Small	0	IsResponsive= True	Number of columns to push a responsive layout pane when displayed on a small screen.
	Column Pull Small	0	IsResponsive= True	Number of columns to pull a responsive layout pane when displayed on a small screen.
	Show Pane Small	True	IsResponsive= True	When set to True, a pane can be shown on a small screen.
	Pane Offset Small	0	IsResponsive= True	Number of columns to offset a responsive layout pane when displayed on a small screen.

Medium				
	Column Span Medium	0	IsResponsive= True	Number of columns a pane can span when a responsive layout is displayed on a medium screen.
	Column Push Medium	0	IsResponsive= True	Number of columns to push a responsive layout pane when displayed on a medium screen.
	Column Pull Medium	0	IsResponsive= True	Number of columns to pull a responsive layout pane when displayed on a medium screen.
	Show Pane Medium	True	IsResponsive= True	When set to True, a pane can be shown on a medium screen.
	Pane Offset Medium	0	IsResponsive= True	Number of columns to offset a responsive layout pane when displayed on a medium screen.
Large				
	Column Span Large	0	IsResponsive= True	Number of columns a pane can span when a responsive layout is displayed on a large screen.
	Column Push Large	0	IsResponsive= True	Number of columns to push a responsive layout pane when displayed on a large screen.
	Column Pull Large	0	IsResponsive= True	Number of columns to pull a responsive layout pane when

				displayed on a large screen.
	Show Pane Large	True	IsResponsive= True	When set to True, a pane can be shown on a large screen.
	Pane Offset Large	0	IsResponsive=True	Number of columns to offset a responsive layout pane when displayed on a large screen.
Navigation				
	Enable Hierarchy Navigation	True	IsResponsive= True	When set to True, a responsive layout pane supports Natural Navigation by touch gestures or graphic buttons.
	Show Navigation Buttons	False	IsResponsive= True	When set to True, a responsive layout pane shows graphic buttons for Natural Navigation.

Set layout properties values

You set the properties of a layout by entering values in the data entry fields of the right window of the Layout Editor.

To set values of layout properties

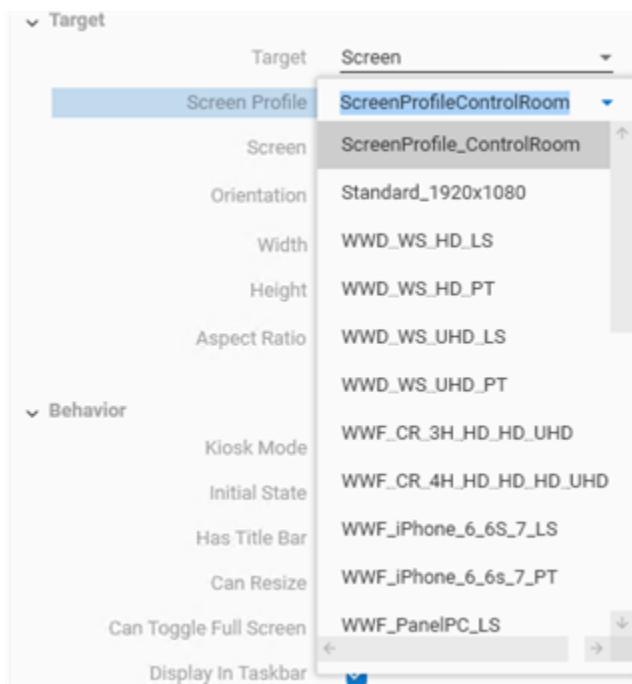
1. Open a layout from the **Visualization folder**.
2. Select outside of the layout in the edit area to ensure no pane or divider is selected.
3. Select the **Properties** grid of the Layout Editor to show the layout's current property values.

Not all layout properties may appear after selecting the **Properties** grid. Some properties only appear after other layout properties are set to specific values. For more information about each layout property, see [Set layout properties](#).

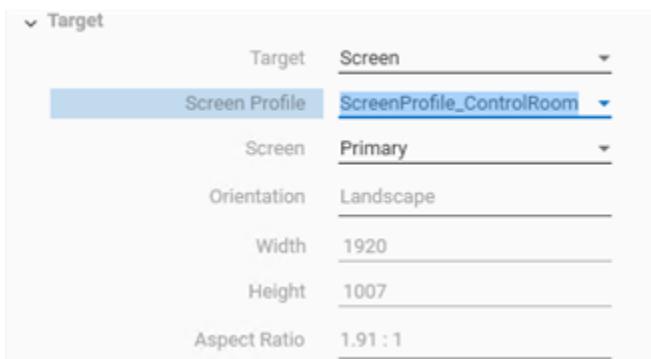
4. Assign values to layout properties.
- If you enter an invalid value in the property grid, the background color of property field changes to a light red. You can hover over the field with your mouse to show a tooltip error message describing why the entered value is invalid.
5. Select **Save** to save your changes to the layout properties.

Select a screen profile layout property

If the **Target** property is set to **Screen**, the **Screen Profile** property appears in the **Properties** window containing a drop-down list of screen profiles defined in the Galaxy.



When a screen profile is selected from the list, the **Screen** property appears containing a list of all screens defined in the selected screen profile.



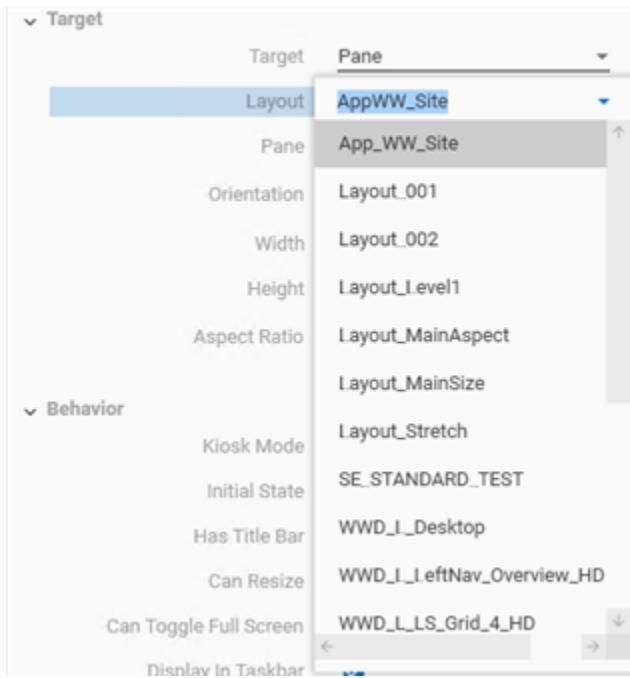
Based on the screen assigned as the value of the **Screen** property, the layout is updated to match the size of the target screen and sizing rules are enforced. The adjusted size of the target screen is based on the screen's resolution and the following layout properties:

- If the layout's **Initial State** property is **FullScreen**, the size is not changed
- If the layout's **Has Title Bar** property is **True**, the height is reduced by 33 pixels

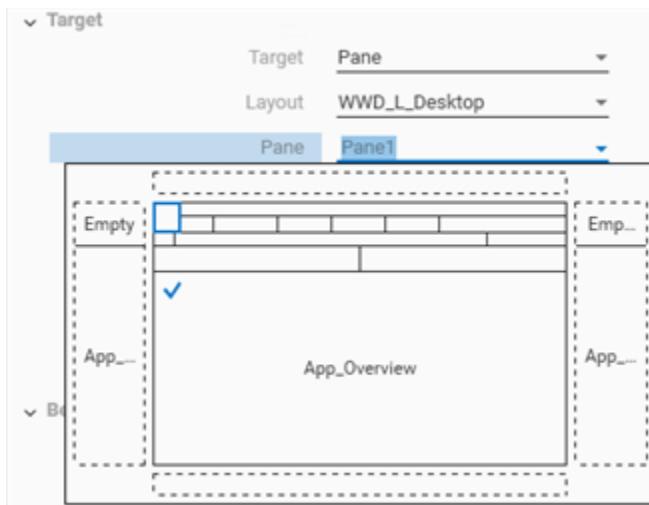
If the screen profile assigned to the **Target** property is deleted, then the **Screen Profile** property shows a red background in the entry field and an error message. If the screen profile assigned to the **Target** property is renamed, the updated screen profile name appears the next time the layout is opened in Layout Editor.

Select a pane layout property

If the **Target** property is set to **Pane**, the **Layout** property appears in the **Properties** window containing a dropdown list of layouts defined in the Galaxy.



When a layout is selected from the list, the **Pane** property appears. Selecting the arrow at the right of the **Pane** field shows a representation of the selected layout with its panes identified by name.



When a pane is selected the **Pane** property is updated with the selected pane's name. If the pane selected is the default pane then the **Pane** property is assigned the value of <Default>.

When the target pane is changed, the size of the layout being edited will be updated to match the size of the new target pane, and the sizing rules are enforced.

If the target pane is either deleted or renamed, then **Pane** property shows an error indication with red background and corresponding tooltip message – “Could not find pane ‘PaneName’ in target layout ‘LayoutName’.

If the selected layout is deleted or renamed, then the **Layout** property shows an error indication with red

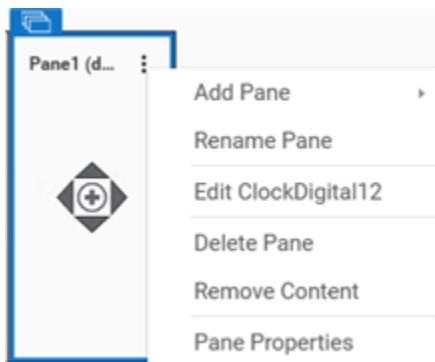
background and corresponding tooltip message – “Could not find target layout ‘LayoutName’.”

Set pane properties 1

Pane properties define the overall visual characteristics and runtime behavior of the selected pane within a layout. For information about layout properties, see [Set layout properties](#).

You show the properties of a pane within the Layout Editor by the following methods based on whether the pane contains content or not.

- When the pane is empty, click within the pane to show its properties,
- When the pane contains content, select the pane's **Pane Options** menu and select **Pane Properties**.



The following image shows the default property values of a pane that extends the full width of a layout. Other property fields appear when properties are assigned specific values or the state of the selected pane. The **Prerequisite Pane Properties** column of the table shown below list any other pane requirements to show a pane property.

The screenshot shows the 'Properties' tab selected in the top navigation bar. Below it, the 'Pane properties' section is expanded. It includes a search bar and several configuration sections:

- General:** Content Type(s) is set to 'None'. There is a checkbox for 'Use For Auto-Fill'.
- Size:** Width is set to '59 px (50 %)'. Width Sizing is 'Variable'. Minimum Width is '2'. Maximum Width is 'Unlimited'. A checkbox for 'Hide Column When Empty' is checked. Aspect Ratio is '2.03 : 1'.
- Appearance:** Background color is '#00FFFFFF'. Opacity is '100 %'. Padding is '0, 0, 0, 0'. Display Mode is 'Maintain Aspect'. Content Alignment is 'Center'. Interaction Mode is 'None'.
- Navigation:** Enable Hierarchy Navigation is checked. Show Navigation Buttons is checked.

You can search for a specific property by entering its name in the **Search** field. Any properties that contain your search string appear beneath the **Search** field.

The search field contains 'width'. The results show the 'Size' category with the following properties:

- Width: '183 px (50 %)' (highlighted in red)
- Width Sizing: 'Variable'
- Minimum Width: '2'
- Maximum Width: 'Unlimited'

You type or select values from drop-down lists to assign values to pane properties. A red background appears when you enter an invalid property value.

The following table lists all pane properties. The properties are organized into the categories shown in the **Properties** window of the Layout Editor. You can view pane properties in alphabetical order by selecting the **Show Alphabetical** icon next to the **Search** field. Short descriptions of the properties appear at the bottom of the **Properties** window for each selected property.

Category	Property	Default Value	Prerequisite Pane Properties	Description
General				
	Content Types	Any		Types of content intended to be displayed in a pane. Select the arrowhead at the right of the field to show a list of supported content types that can be selected. Any is the default content type. See Set pane properties: content types .
	Use For Auto-Fill	False		When True, pane content is filled automatically by Auto-Fill mode. For more information about Auto-Fill, see About auto-fill navigation .
Size				
	Width	N/A	Pane Width < Layout Width	Width of the pane in pixels and percentage of total layout width.
	Width Sizing	Variable	Pane Width < Layout Width	Specifies if pane width can be resized during runtime or remains at a fixed width.
	Minimum Width	2	Pane Width < Layout Width	Minimum width the pane can be resized to at runtime.
	Maximum Width	Unlimited	Pane Width < Layout Width	Maximum width the pane can be resized to at runtime.

	Hide Column When Empty	True	Pane Height < Layout Height	When True, a layout column is hidden if all panes within the column do not show content during runtime. See Set pane properties: hide when empty .
	Height	N/A	Pane Height < Layout.Height	The Height of the selected pane in pixels.
	Height Sizing	Variable	Pane Height < Layout.Height	Specifies if pane height can be resized during runtime or remains at a fixed height.
	Minimum Height	2	Pane Height < Layout.Height	Minimum height the pane can be resized at runtime.
	Maximum Height	Unlimited	Pane Height < Layout.Height	The maximum height of a pane when resized at runtime.
	Hide Row When Empty	True	Pane Width < Layout.Width	When True, a layout row is hidden if all panes within the row do not show content during runtime. See Set pane properties: hide when empty .
	Aspect Ratio	None		Calculated aspect ratio of the width to height dimensions of a pane.
Behavior				
	Can User Close	True	Presentation Mode = Tabbed	When True, users can close pane tabs at runtime.
	Maximum Items	20		The maximum number of items

				that can be open in the selected pane while the ViewApp is running.
Appearance				
	Background	#00FFFFFF		Background color of a pane. Select the arrow at the right of the field to show a color picker control to select a background color.
	Opacity	100%		Opacity percentage applied to a pane background and all content.
	Padding	0, 0, 0, 0		Number of pixels of blank space between a pane border and content within a pane.
	Display Mode	Maintain Aspect		Method used to change the size of content when a pane is resized during runtime. See Set pane properties: display mode .
	Show Zoom Control	True	InteractionMode = Pan Only or Pan and Zoom	When True, controls are shown during runtime for the user to pan or zoom a pane.
	Horizontal Scroll Bar Visibility	Auto	InteractionMode = Pan Only or Pan and Zoom	Specifies how the horizontal scrollbar is displayed when pane content exceeds the width of the pane.
	Vertical Scroll Bar Visibility	Auto	InteractionMode = Pan Only or Pan and Zoom	Specifies how the vertical scrollbar is displayed when

				pane content exceeds the height of the pane.
	Content Alignment	Center	DisplayMode = MaintainSize or MaintainAspect	Horizontal and vertical alignment of content within a pane.
	Interaction Mode	None		Specifies if the user can pan or zoom a pane during runtime to show content that exceeds the size of a pane.
Navigation				
	Enable Hierarchy Navigation	False		When set to True, gestures or buttons can be used to navigate with natural Navigation.
	Show Navigation Buttons	True	Enable Hierarchy Navigation = True	When set to True, the pane shows buttons for Natural Navigation.
	Enable Content Navigation	True	Pane Presentation Style = Multiple	When set to True, the pane supports content navigation using gestures or buttons.

Set pane properties 2

You set the properties of a pane by entering values in the data entry fields of the right window of the Layout Editor. If you enter an invalid property value, the background of the entry field turns red.

To set pane properties

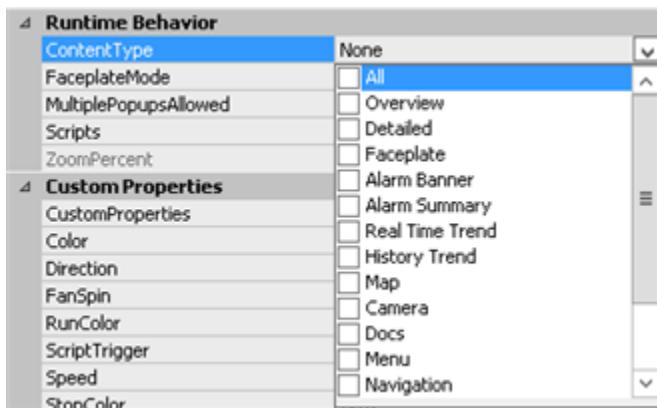
1. Open a layout from the **Visualization folder**.
2. Select the pane whose properties you want to change.
3. Select the **Properties** grid in the right column of the Layout Editor to show the pane's current property values.

Not all pane properties may appear after selecting the **Properties** grid. Some properties only appear after other properties are set to specific values.

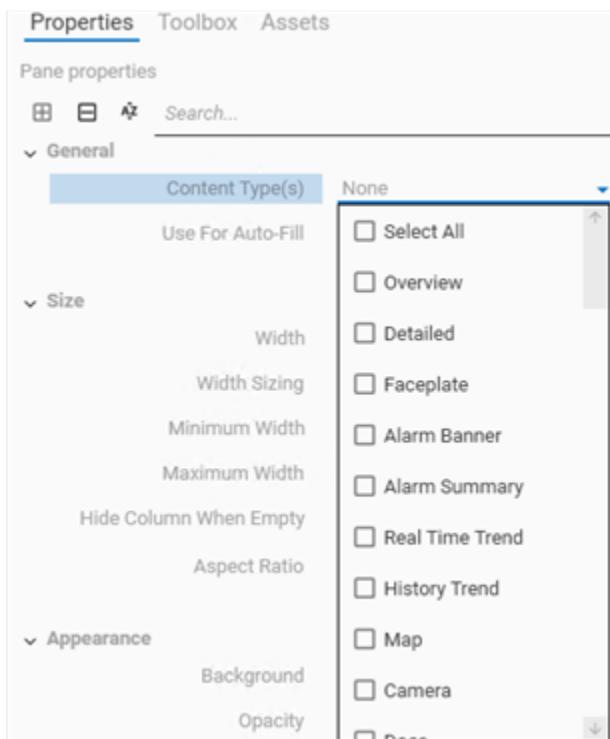
4. Accept the default value or enter new values in the data entry fields.
5. Select **Save** to save your changes.

Set pane properties: content types

A content type is a classification method to determine the types of content that can be shown in a layout pane during runtime. The Industrial Graphic Editor provides the **ContentType** property to define the content type of a graphic.



A corresponding layout pane property defines the type of content that can be added to a pane during runtime. When there is a match between graphic and pane content types, the content can be added to the pane during runtime and focus set to it.



You can select multiple content types as values for the **Content Types** pane property.

Set pane properties: display mode

The **Display Mode** property can be assigned three values that determine how content appears during runtime when the size of the containing pane changes size.

Display Mode Values	Description
Maintain Aspect	<p>The width and height ratio of content is maintained when a pane is resized larger or smaller. In the case when content exceeds the pane's width or height, the content is aligned within the pane based on the assigned Alignment and Padding property values. The content is truncated in the dimension that is larger than the pane's width or height.</p> <p>In the case when the entire content fits within a pane's dimensions, the content is positioned within the pane based on the assigned Alignment and Padding property values.</p> 
Maintain Size	<p>The dimensions of a piece of content remain constant when a pane is resized. When the content is smaller than the pane, the content is positioned within the pane based on the assigned Alignment and Padding property values.</p> <p>In the case when the size of a piece of content exceeds the dimensions of a pane, the content is centered in the pane and truncated in any dimension that exceeds the pane's width or height.</p> 

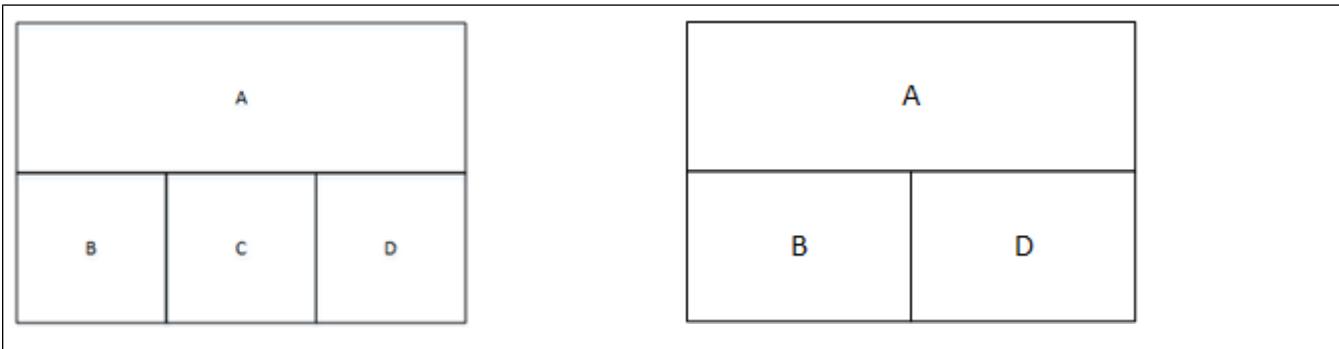
Stretch

The width and height of a piece of content adjust to fit the dimensions of a pane. In the case of a pane becoming smaller, the **Alignment** property value is ignored.

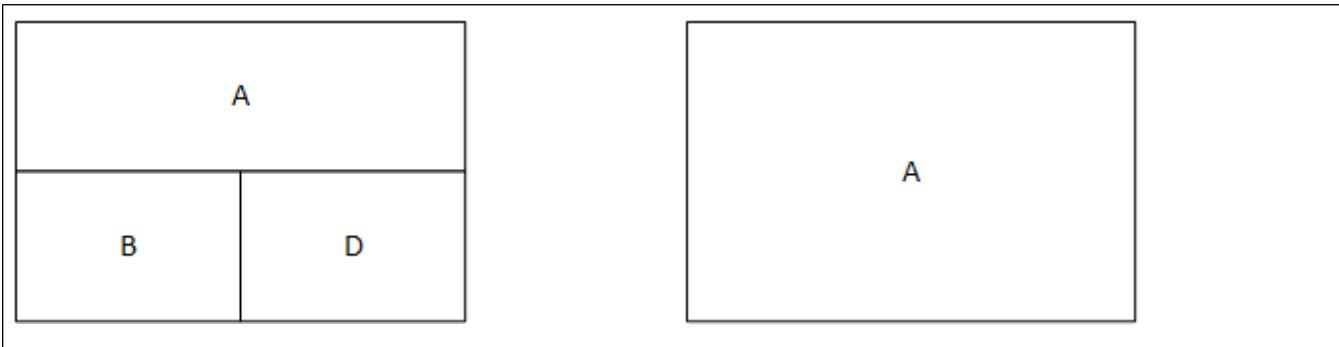
**Set pane properties: hide when empty**

If the **Hide Row When Empty** and **Hide Column When Empty** properties are set to True, a pane can be hidden during runtime when it is without content or the content is invalid.

In the following example, pane C has been configured with the **Hide column When Empty** property set to True, so it is not shown when it does not contain any content.



An entire row or column of panes can be hidden by configuring the appropriate property. In this example, if pane C is also configured with the **Hide Row When Empty** property set to True, this property propagates to panes B and D. When all three panes (B, C, and D) are empty, the entire row of panes is hidden in the layout.



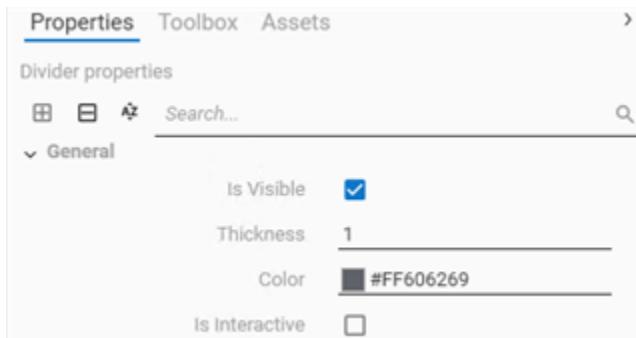
Set pane properties: auto-fill

The **Use for Auto-Fill** pane property determines if each navigation item can control whether the selected pane of a layout can be auto-filled or not. In Auto-Fill mode, the content shown in a layout's panes can come from different navigation items in the ViewApp's navigation model. Using hierarchy navigation in any pane will use the specific navigation item for the swiping behavior which placed that original content in the pane.

Auto-Fill has a set of options that control how content is automatically placed in panes without the user having to individually configure the content shown in each pane. For more information about Auto-Fill options, see [About auto-fill navigation](#)

Set layout divider properties

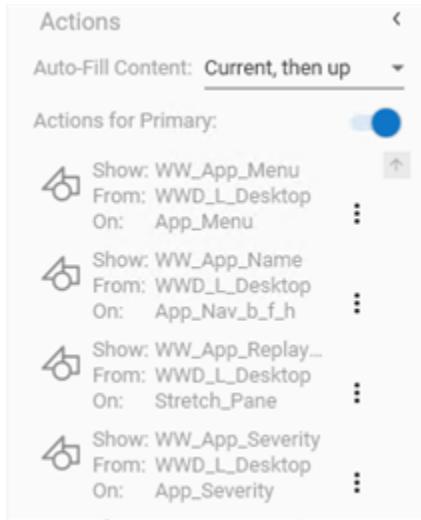
If you select a pane divider in a layout, divider properties that can be assigned values appear in the **Properties** area.



Property	Default Value	Prerequisite Pane Properties	Description
Is Visible	True	Divider selected	When True, the pane divider is visible while the ViewApp is running
Thickness	1	Divider selected and Is Visible = True	Thickness of a pane divider in pixels.
Color	#FF606269	Divider selected and Is Visible = True	Color of a pane divider. Select the arrow at the right of the field to show a divider color picker.
Is Interactive	False	Divider selected and Is Visible = True	When True, the user can move the divider bar to resize panes during runtime.

Add content to panes

You can drag and drop content to panes. Assigning content to a pane adds a Show Content action to the layout's navigation Action List, which coordinates the automatic filling of panes during runtime.

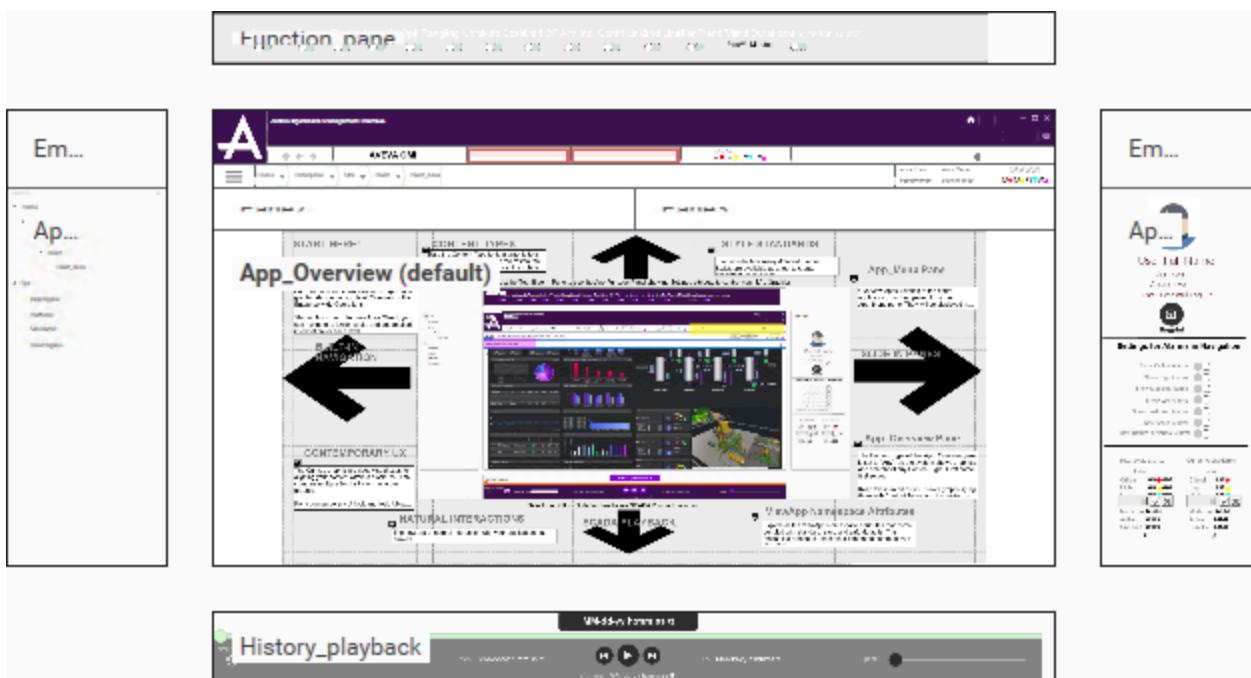


Content from the Toolbox or Assets tabs can be dragged into a pane of the layout. Content can include:

- Graphic
- Layout
- Control
- App Instance

While content is dragged over a layout, the background color of the target pane changes to indicate the pane will receive the content when it is dropped. The shape of a mouse cursor also changes to indicate the content can be dropped on the pane. After content has been dropped, a thumbnail appears within the pane representing its content.

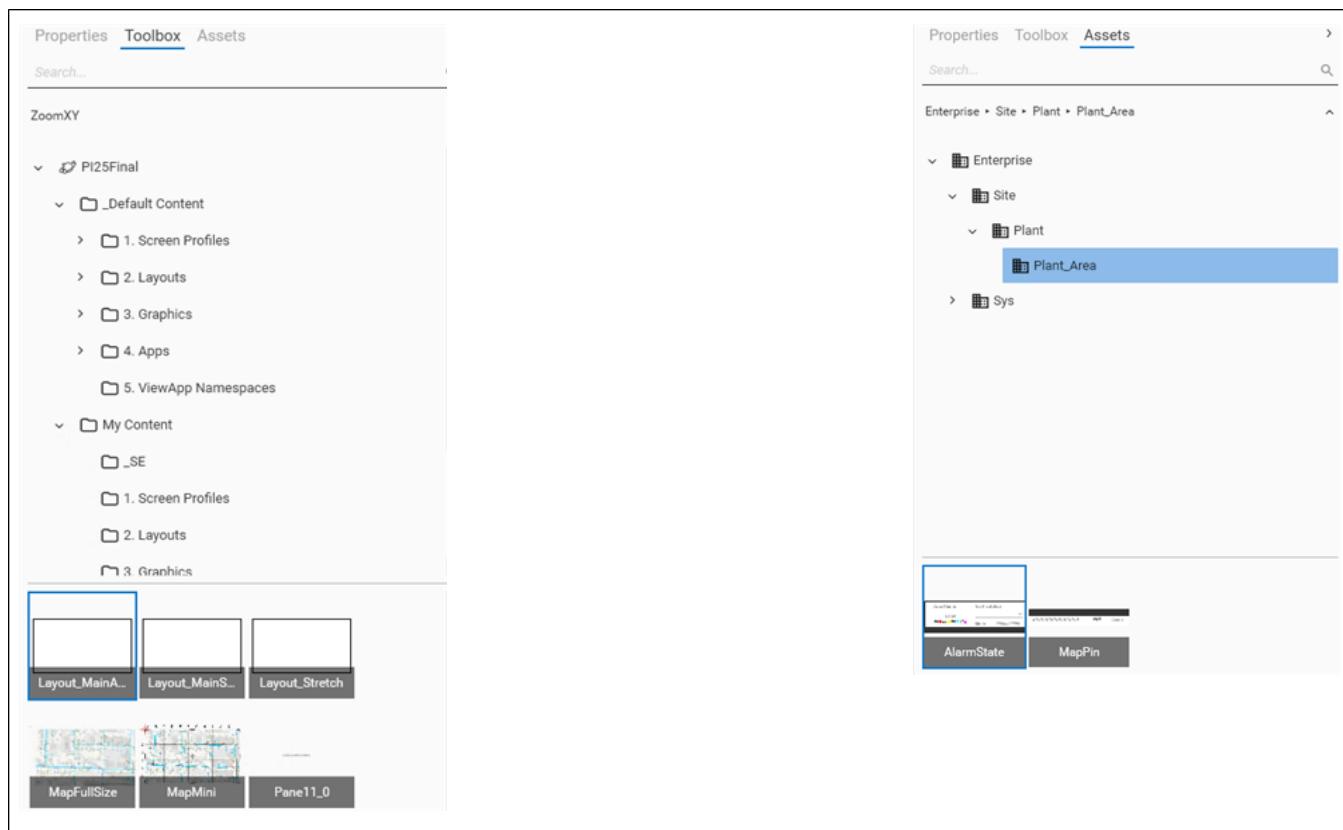
Note: Using a touch device, you can add content to a pane by selecting and holding the content to move it over the target pane.



Browse for content to add to layout

You can browse the **Assets** and **Toolbox** panes to find content that can be added to a layout. Each item shown in the **Assets** view represents an Area or an Asset as configured in the System Platform IDE. Each item shown in the **Toolbox** view represents either a folder, a graphic, or a control library. Selecting an item from a browse tree selects the item and displays any content belonging to that item in the content area beneath the browse list.

Placing your mouse cursor directly over a Toolbox or Assets content thumbnail shows a horizontal bar with the folder path to the selected content.



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

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

Search for content to add to layout

You can search within the currently displayed toolbox folders or assets by entering a term in the **Search** field. Search terms are case-insensitive and locate content or assets regardless of capitalization.

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

Asset Searches

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

Properties Toolbox Assets >

Filter

6 assets matching 'Filter':

Filters	WaterPlant
FilterSimulation_001	WaterPlant > SystemArea > Simulation
FilterSimulation_002	WaterPlant > SystemArea > Simulation
FilterSimulation_003	WaterPlant > SystemArea > Simulation
FilterSimulation_004	WaterPlant > SystemArea > Simulation
FilterSimulation_005	WaterPlant > SystemArea > Simulation

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

Placing the cursor over a content thumbnail or tapping it shows the folder in the Visualization folder where the item is located.

Toolbox Searches

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

Properties Toolbox Assets >

pump

11 content items matching 'pump':

application	pdf	Label
ISA_Pump1	ISA_Pump2	PumpCentrif...
PumpHydrDia...	PumpManual...	PumpOilField...
PumpSSCentr...	PumpVert	SA_MultiStag...
SA_Pump_Blo...	SE_Pictogram...	

Placing the cursor over a thumbnail in the search results list or tapping it shows the folder in the Visualization folder where the item is located.

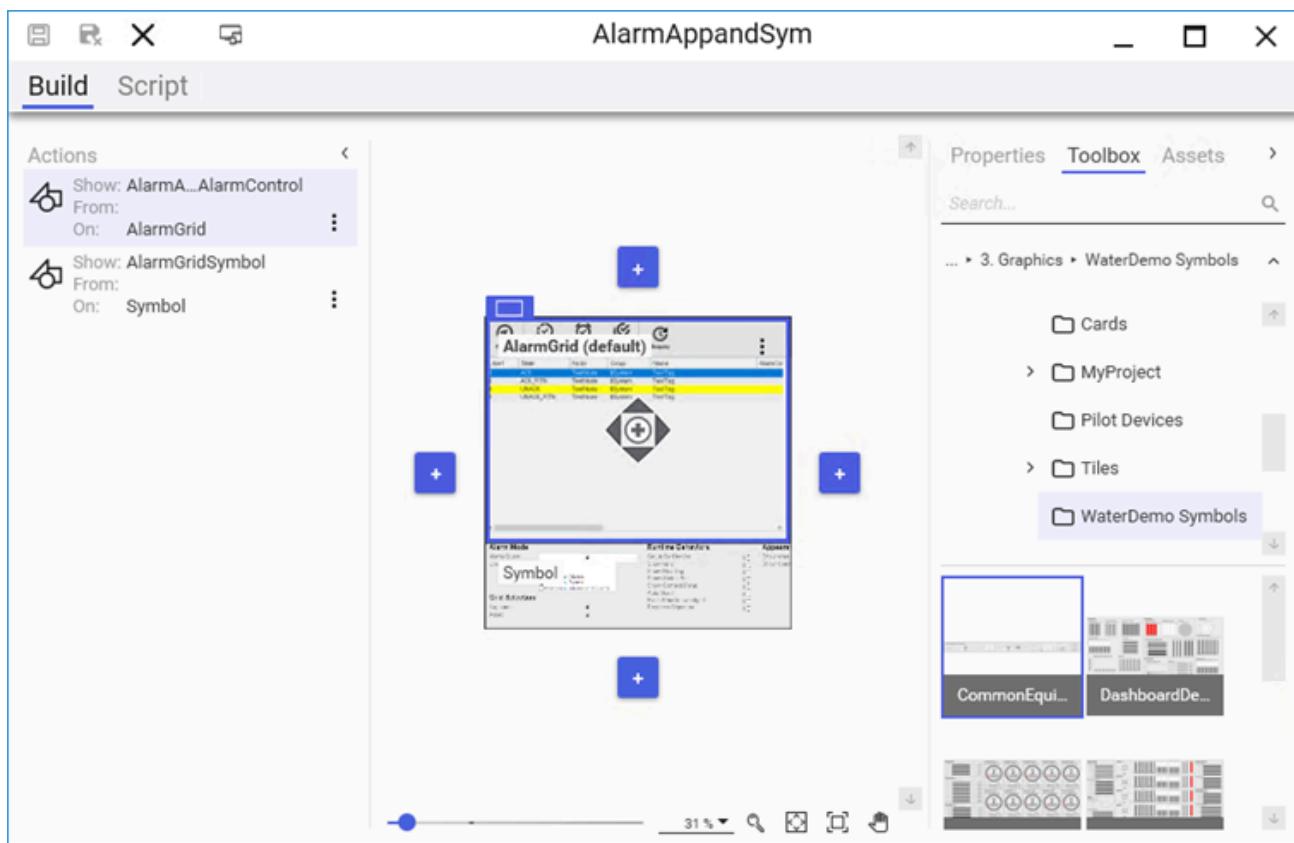


Search results items in the content area can be selected and dropped in a layout pane.

About adding content to an empty pane

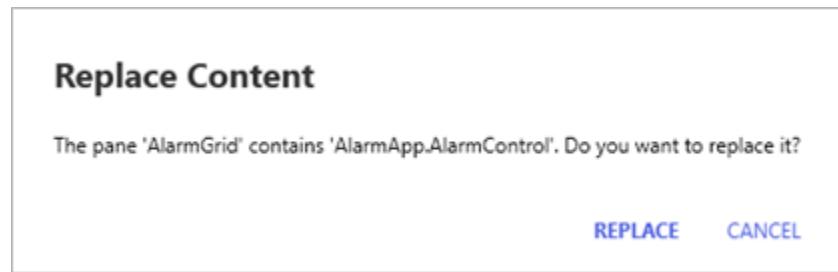
If you add content into an empty pane, then,

- The **Content** property is set to the name of the content added to the pane
- The **Pane** property is set to the name of the pane hosting the content.
- The **Actions** list updates to show an entry containing the name of the content and the name of the pane hosting the content
- A preview of the content appears in the pane



Note: If a layout is added to a pane within the same layout being edited, the cursor changes appearance to indicate the layout cannot be added.

If the target pane being receiving content is a single content pane and already contains content, you see a dialog box asking whether the current content should be replaced or not.



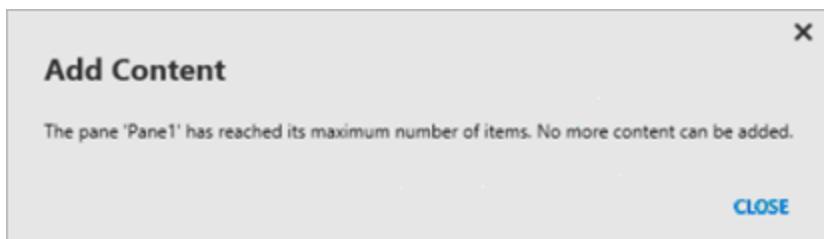
Select **Replace** to update the **Content** property. The thumbnail of the new content appears in the pane.

Select **Cancel** to cancel the add content process. The existing content remains in the pane.

Add content to multi-content panes

If a pane is defined as a multi-content pane, dropping a second content item adds the content and the pane is selected. The **Pane** property updates to show the name of the most recent content added to the pane.

If you attempt to drop a content item that will exceed the maximum limit set for the pane, the add content action will fail. A dialog box appears notifying you that the maximum number of content items has been reached and no more content can be added to the pane.



Move content to another pane

You can move content from one pane to another within the layout. The **Pane** property includes a thumbnail representation of the layout where you can select the target pane to move the content into.

The behavior of the Layout Editor is based on the presentation styles of the source and target panes and the number of content items currently in these panes.

- If content is moved from a pane specified with a Single presentation style to another Single presentation style pane, the source pane will become empty and the target pane will contain the moved content. New content can be added to the original source pane after its original source content has been moved.
- If content is moved from a pane specified with a Single presentation style to a multi-content presentation style pane, the source pane will become empty and the target pane will contain the moved content unless the maximum content limit has been exceeded.
- If content will be moved from a multi-content presentation style pane to another pane, the **Actions** list shows the default content to be moved with a highlighted background. If you want to move other content from the multi-content pane, select the item from the **Actions** list before selecting the target pane.

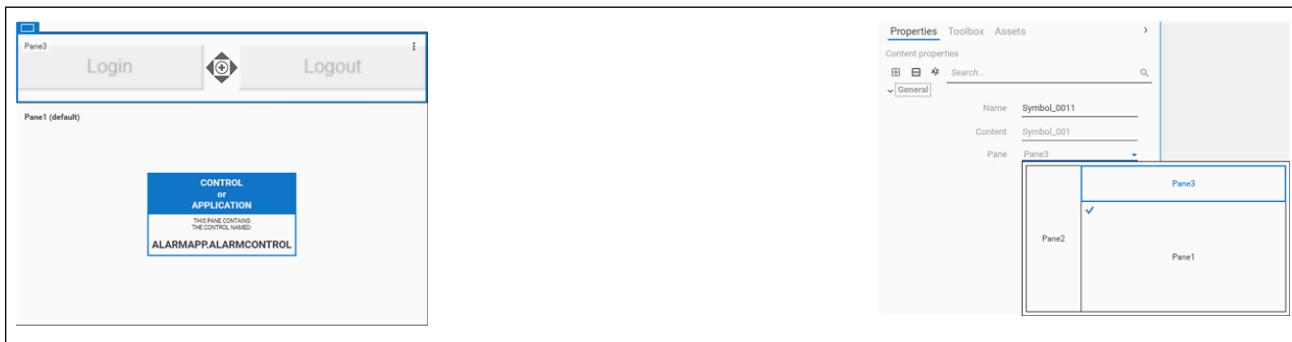
To move content to another pane within a layout

1. Select the pane containing content that you want to move to another pane.

If you select a multi-content pane, the top most content item shown in the **Actions** list is the selected default content to be moved.

2. Select the **Properties** grid to show the selected pane's properties.
3. Select the arrow at the right of the **Pane** property field to show a thumbnail of the layout.

The layout thumbnail identifies the selected source pane with a blue border and text.



If the source pane is a multi-content pane, the **Actions** list shows the pane's content as separate items.

4. If you are moving content from a multi-content pane, select a content item from the **Actions** list to be moved.

5. Select another pane in the thumbnail to move the content from the selected pane.

The **Actions** list updates to show the contents of each pane within the layout.

About the responsive layout mode

Video Tutorial: Introduction to responsive layouts in AVEVA OMI

https://player.vimeo.com/video/992306239?badge=0&autoplay=0&player_id=0&app_id=58479

Use the responsive layout mode if the layout you are designing is intended to be used in one or more ViewApps that will be viewed by a mix of devices, such as smart phones, tablets, and monitors. If the layout is only for use in a single ViewApp that will be viewed on a monitor, then using the fixed layout mode is appropriate.

For example, if you are viewing a ViewApp on a device that is 375 pixels wide, the Mobile layout is used, since the mobile device width is defined as no more than 375 pixels wide. If the device is 376 pixels wide, the next layout size (tablet) is used.

In the following descriptions, the nominal pixel dimensions (width x height) for each screen size are shown. The dimensions assigned to each screen size are fixed and cannot be overridden.

Four different screen resolutions (break-points) are defined for a responsive layout, as listed below. Each screen resolution represents a break-point, which represents the screen width at which a layout designed for that particular screen width takes effect. The layout design presented to the user is determined by the width, in pixels, of the viewing device. Screen height does not play any role in determining screen size for the responsive layout. Default layout settings are configured at the smallest screen size (mobile), and propagate to the larger screen sizes, unless overridden. See [About responsive layout propagation](#) for more information.

Note: The dimensions (width x height) shown for each screen size within the layout editor are nominal dimensions.

- **Mobile (extra small layout):** This layout is used when the viewing screen is 767 pixels or less. See [Display a web ViewApp on a mobile device](#) for information about ViewApp appearance at runtime.

The parameters set for the Mobile layout are used for each of the larger screen sizes, unless an override is set for larger screen size. An override set for Tablet, for example, would also apply to the Laptop and Desktop screen sizes, unless also overridden at one of those larger screen sizes.

- **Tablet (small layout):** This layout is used if the width of the viewing screen is between 768 and 991 pixels.
- **Laptop (medium layout):** This layout is used if the width of the viewing screen is between 992 and 1199 pixels.
- **Desktop (large layout):** This layout is used if the width of the viewing screen is 1200 pixels or more.
- **TV (extra large layout):** This layout is used if the width of the viewing screen is 1200 pixels or more.

Note: If you change the orientation of a device, the height of the screen becomes its width. For example, if you are viewing a layout on a mobile device and turn it sideways, the screen resolution becomes 812 x 375 (using the nominal dimensions in the Layout Editor). Therefore, the **Laptop** layout would be used instead, since it is used for screen widths between 768 and 991 pixels. You can demonstrate this by configuring a pane in a Mobile layout, and then select the **Rotate Screen**  icon. The layout automatically switches from the Mobile to the Laptop layout. You can demonstrate the same effect by dragging the right side of the layout to change its width.

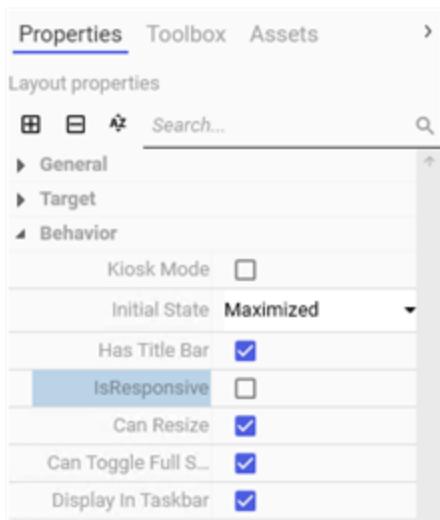
You can change the number of columns used for a pane at each of the defined screen resolutions. By default, new panes span the full width of the layout (12 columns). To create side-by-side panes, set the number of columns that each pane uses so the total number of columns for all the panes is 12 or less. If you have two 12-column panes configured for mobile devices, you may want to have the panes side-by-side when viewed in larger devices. In this case, you could set the width of each pane to 6 columns.

Build a responsive layout

To build responsive layouts, you first toggle the Layout Editor to responsive mode by doing one of the following:

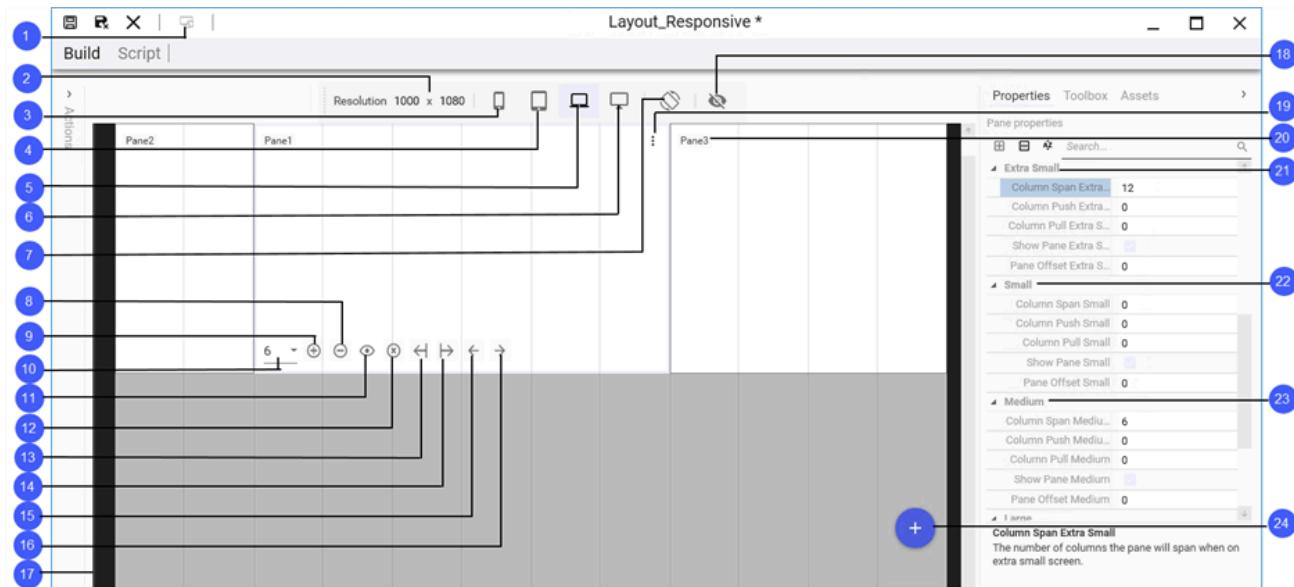
- Select the Fixed Layout/Responsive Layout button, located at the top left of the Layout Editor.

- Enable the **IsResponsive** property, located under Layout Properties. To see layout properties, the layout itself must be selected (not a pane).



Note: Anytime you switch between layout modes, all changes to the layout are discarded. There is no undo.

The figure below shows the various parts of the Layout Editor when Responsive mode is selected. For general information about using the Layout Editor, see [About the Layout Editor](#).



1	Fixed Layout/ Responsive Layout Button	Sets or clears the IsResponsive layout property. When IsResponsive is set, the build view changes from the default fixed layout view to the responsive layout view. Note that any configuration changes made in one view are not preserved when you switch between fixed and responsive views, since the two views cannot be reconciled. If you switch views after configuring the
---	--	---

		layout, all unsaved changes are lost.
2	Current screen resolution	Shows the layout height and width, in pixels, of the layout as it is currently displayed in the editor. Entering a height or width value changes the displayed layout to match the new dimension. If a width value is entered, the applicable screen size icon (mobile, tablet, etc.) that corresponds to the value is automatically highlighted, and the displayed layout changes responsively to match the configuration for that screen size.
3	Mobile icon	Selecting the Mobile icon changes the layout to match the settings entered for Extra Small layouts. Mobile settings apply to screens 767 pixels in width or less.
4	Tablet icon	Selecting the Tablet icon changes the layout to match the settings entered for Small layouts. Tablet settings apply to screens widths between 768 pixels and 991 pixels.
5	Laptop icon	Selecting the Laptop icon changes the layout to match the settings entered for Medium layouts. Laptop settings apply to screens widths between 992 pixels and 1199 pixels.
6	Desktop icon	Selecting the Desktop icon changes the layout to match the settings entered for Large layouts. Desktop settings apply to screens that are 1200 pixels in width or more.
7	Rotate layout	Changes the height of the currently-displayed layout to its width (and width to height). If this change in width exceeds the threshold value of the current layout size, the layout automatically changes to match the parameters

		of the new layout size. For example, rotating the layout when Mobile is selected changes its width to 812 pixels (the height setting), and thus the layout settings for Tablet devices take effect.
8	Increase Column Span	Each click increments the column span of the pane by one, up to the maximum of 12 columns.
9	Decrease Column Span	Each click decrements the column span of the pane by one, down to the minimum of 1 column.
10	Column Span	Sets the number of columns for the selected pane. The default is 12 (100% of layout width). You can type a number from 1 to 12, or click the arrow to select from a list.
11	Hide Pane	Sets or clears the Show Pane property for the currently-selected screen size only. Unlike other settings, the Hide Pane setting does not propagate to larger screen sizes. Hidden panes are not shown runtime, and are hidden at design-time unless Show Hidden panes (18) is selected. The default setting of Show Pane is true for each screen size. The ShowContent() method does not work with panes that are hidden.
12	Delete Pane	Deletes the pane from the layout. The selected pane is deleted from the entire layout, not just the currently-selected screen size. If there is only one pane in the layout, the pane cannot be deleted.
13	Insert Pane before selected	Adds a new pane to the layout. The new pane is inserted before the currently-selected pane.
14	Insert Pane after selected	Adds a new pane to the layout. The new pane is inserted after the currently-selected pane.

15	Move Pane Before	Moves the the pane in the layout to the position immediately before its current position. If the pane is the first in the layout, the Move Pane Before button is disabled. The new position for the pane applies to all screen sizes.
16	Move Pane After	Moves the the pane in the layout to the position immediately after its current position. If the pane is the last in the layout, the Move Pane After button is disabled. The new position for the pane applies to all screen sizes.
17	Resize bar (layout border)	Select and drag the resize bar (thick black border) from either side of the layout to increase or decrease currently-displayed layout width. To change the height of the layout, select and drag the resize bar from the bottom. To change width and height simultaneously, select and drag the resize handle (not shown) at the bottom-right corner of the resize bar.
18	Show/Hide hidden panes	Shows or hides any hidden panes for the currently-selected screen size only. Hidden panes for other screen sizes are not affected. If hidden panes are shown, this applies to design-time only, not runtime.
19	Pane Option menu	<p>Opens the context pane option menu. The options are:</p> <ul style="list-style-type: none"> • Rename Pane • Set as Default Pane • Delete Pane <p>If the pane contains content, additional options appear in the menu:</p> <ul style="list-style-type: none"> • Edit <Content Name>

		<ul style="list-style-type: none"> • Remove Content • Content Properties (toggles to Pane Properties when Content Properties are shown)
20	Pane name	This is an editable text field. Rename a pane by clicking on the name and entering a new one. You can also use the Rename Pane option in the Pane Option menu .
21	Extra small screen size properties	The default pane properties that are set for the screen size. These propagate to each of the larger screen sizes, unless overridden. Similarly, overrides serve as defaults for larger screen sizes, unless subsequently overridden at a larger screen sizes.
22	Small screen size properties	The set of pane property overrides that are set for the small (Tablet) screen size.
23	Medium screen size properties	The set of pane property overrides that are set for the medium (Laptop) screen size.
24	Add Pane Icon	Adds a new pane to the layout. The new pane is inserted after the last pane in the layout.

About responsive layout methods

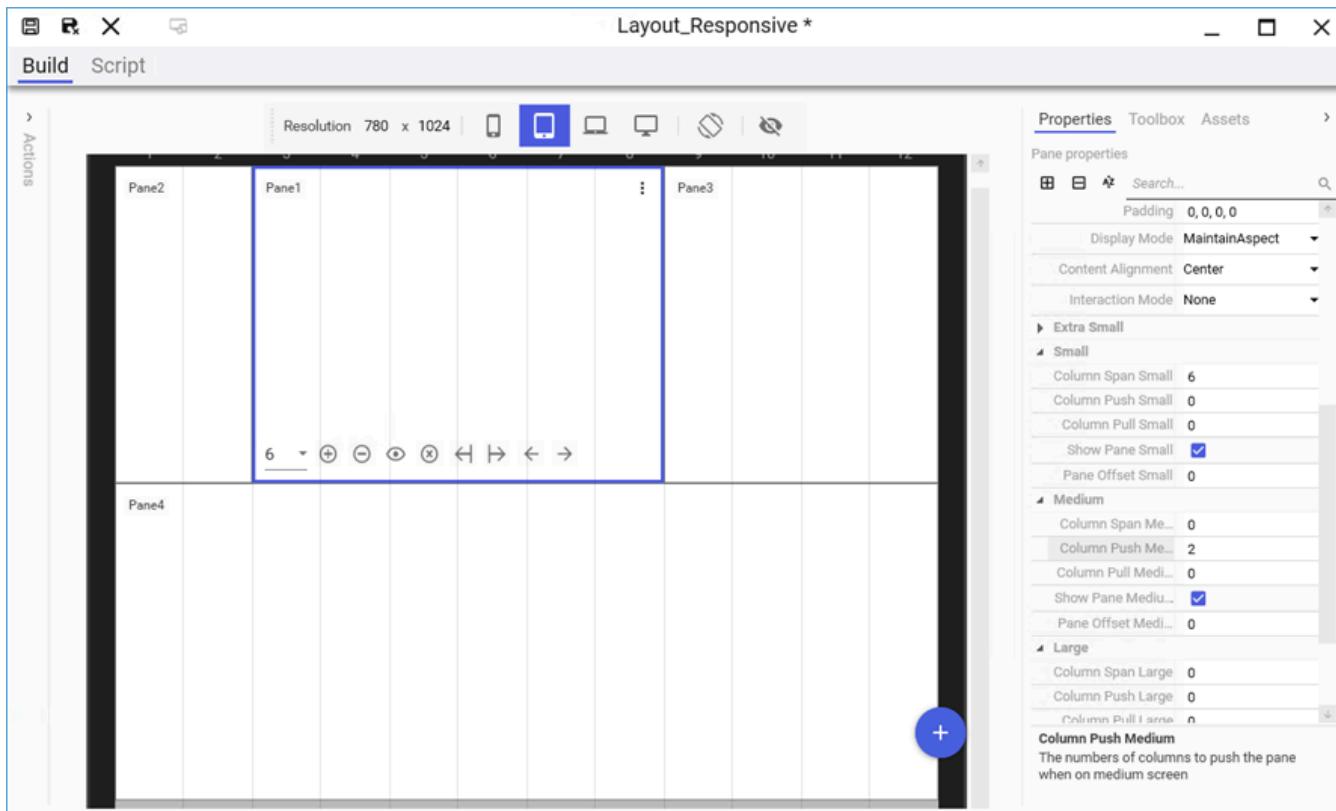
There are several ways to configure a responsive layout. These are:

- **Mostly Fluid:** this is the most basic and the simplest type of responsive layout to configure. The only layout property that requires configuration in a mostly fluid layout is the **Column Span** property, which can be modified to define a different width for a pane at each screen resolution. Even though responsive layouts are easy to configure using only the column span parameters, the Mostly Fluid method provides sufficient capability for most applications.
- **Column Drop:** in addition to setting different widths with the Column Span parameter, the Column Drop method makes use of the **Column Push** and **Column Pull** parameters. This allows content to flow differently at different screen sizes. One scenario for using column drop would be an application that has a navigation screen followed by an overview display. When viewed on a large screen, it makes sense for navigation to be shown above the overview display. However, when viewed on a mobile phone, a user would prefer to have the overview shown at the top of the screen, instead of having to scroll down to get to the main content.

- **Layout Shifter:** this is the most complex method for creating responsive layouts. The Layout Shifter method lets you show different content when the ViewApp is opened on different size screens. As with the Column Drop method, Layout Shifter uses the Column Push and Column Pull properties, and also uses the **IsVisible** property to selectively allow a different pane that contains different content to be shown at specific screen resolutions. For our example scenario, we will again consider a ViewApp with a navigation pane. In this case, we not only want to move the navigation pane up for viewing on larger screens, we also want to change its orientation from horizontal to vertical. To effect this, we create two panes: one with a vertical navigation app and the other with a horizontal navigation app. In addition to using Column Push and Column Pull, we will also use the **IsVisible** property so that the pane with applicable content is shown only when it should be.

Column Offset is another pane property that can be used, but this property is primarily used to align a pane as needed at different screen resolutions, rather than for enabling column shifts. For example, you can use Column Offset to center an "orphaned" pane at certain resolutions.

The figure below shows Pane 1 is selected and overrides have been entered for Small (Tablet) and Medium (Laptop) screen sizes. For Small screen sizes, **Column Span** is set to 6. This also becomes the setting for larger screen sizes, unless a new override is set at a larger size. For medium layouts, **Column Push** has been set to 2. This setting will also be in effect for larger screen sizes, unless another override is set. Pane 4 is offset by 2 columns. However, the setting is not shown because the pane is not selected.



About responsive layout propagation

Responsive layout properties, with the exception the Show Pane property, always propagate from smallest to largest (see below for more information about Show Pane). Therefore, you initially set the properties at the smallest screen size. These become the default parameters for the larger screen sizes. Each layout size has its own configuration parameters, and you can set the number of columns for a pane at each screen size. You can

also modify the display behavior of a pane at a particular screen size by using column push/pull and offset parameters. You can also set whether or not a pane will be visible at a particular screen size. Use the following parameters to set pane size and display behavior. These are configurable for each screen size. Always configure the smallest screen first, since the parameters propagate from smallest to largest.

- **Column Span:** Sets the width of the selected pane by number of columns. The minimum is one, and full layout width is 12 (maximum). The default is 12.
- **Column Push:** Moves the selected pane to the right by the number of columns specified. Only the selected pane moves. If there is a pane to the right, it covers the pushed pane. The default is 0.
- **Column Pull:** Moves the selected pane to the left by the number of columns specified. Only the selected pane moves. If there is a pane to the left, it is covered by the pulled pane. The default is 0.
- **Pane Offset:** Similar to column push, but offset is also pushes any panes to the right by the same number of columns. Unlike Column Push, an offset pane is not covered by the adjacent pane to the right. The default is 0.
- **Show Pane:** Determines if the pane is visible or hidden for a particular screen size. The default is visible (Show Pane is enabled). The Show Pane property only applies to the screen size for which it is set; it does not propagate. The ShowContent() method does not work with hidden panes.

The screenshot shows the 'Properties' tab selected in the top navigation bar. Under 'Pane properties', the 'Extra Small' screen size is currently active, displaying settings for Column Span, Column Push, Column Pull, and Show Pane. A blue circle labeled '1' is positioned next to the 'Extra Small' heading. The 'Small' screen size is expanded, showing its own settings for Column Span, Column Push, Column Pull, and Show Pane. A red box highlights the 'Column Push Small' setting, which is set to 1. A blue circle labeled '2' is positioned next to the 'Small' heading. Below the 'Small' section, 'Medium' and 'Large' screen sizes are listed with blue circles labeled '1' and '2' respectively.

Screen Size	Column Span	Column Push	Column Pull	Show Pane
Extra Small	12	0	0	Visible
Small	4	1	0	Visible
Medium				
Large				

1 Default pane settings, configured at smallest screen size (Extra Small).

2 Override settings for Column Span and Column Push, configurable for each larger screen size (in this case, Small). The overrides for Small apply to Small screens and larger. Settings with 0 (Column Pull and Pane Offset) indicate no override.

- You can enter overrides for one or all of the larger screen sizes. Any overrides set at one screen size become the default settings at larger screen sizes. For example, if you override settings for Laptop (medium) size, those overrides become the default settings for the Desktop (large) size as well.
- When a parameter is set to 0, it indicates that there is no override at that particular level of screen size, and that the default parameters set for a smaller screen have not been overridden.

Responsive Layout: Mostly Fluid

Video Tutorial: Create a layout in AVEVA OMI

https://player.vimeo.com/video/992308927?badge=0&autoplay=0&player_id=0&app_id=58479

Column Span Property

The Mostly Fluid method for configuring a responsive layout uses the **Column Span** property to set how much of the layout width a pane uses at a given screen size. Recall that Column Span can be any integer from 1 to 12, where 12 indicates that the pane will be 100% (12/12 columns) of the layout width. If Column Span is set to 6, the pane will be 50% (6/12 columns) of the layout width.

- The Column Span default for each pane is set by the Extra Small (Mobile) setting. The default is 12 and propagates to all screen sizes, if no overrides have been entered for a larger screen size.
- When the Column Span property is set to 0 for Small (Tablet) or larger screen sizes, it indicates that the parameter is using a setting that was configured for a smaller screen. If an override is set, the override propagates to larger screen sizes, unless a different number is set at a larger size.
- A number other than 0 indicates an override. It is not necessary to set overrides for every screen size. You only need to override a setting if you want to change the **Column Span** that was configured for a smaller screen setting.

To Configure a Mostly Fluid Responsive Layout

1. Open the layout in the **Layout Editor**.
 2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/Responsive** button  in the title bar.
 3. Add panes for each content item by clicking the **New Pane** button .
- Each new pane is added with default settings for width and height. The default width setting is 12 columns and the default height setting is 300 pixels; the height setting is constant for all screen sizes.
4. To resize a pane, select both the pane and the applicable screen size to which you want to apply overrides.
 - Pane width is set by the **Column Span** property. This is located under the applicable screen size area of the Pane Properties grid (Extra Small, Small, Medium, Large, Extra Large). The default width, which is displayed under the Extra Small (Mobile) screen size category, is 12 columns. This indicates that the pane will take up the full layout width.

For each of the larger screen size, Column Span is initially 0. This indicates that a previous setting for a smaller screen size will be used. Change the pane width as needed by setting a new Column Span. Screen sizes set to 0 will use the setting from a smaller size.
 - Pane height is set by the **Height** property (located in the **Size** properties area of the Layout Editor), and applies to all screen sizes. The default height is 300 pixels. Minimum height is 1 pixel. Change pane height as needed to accommodate the content that will be added.
- When the combined column span of two or more adjacent panes is 12 or less, the panes will be displayed side-by-side for the screen size(s) that match that configuration. For example, if **Column Span** for three adjacent panes is set to 4 in each pane, the panes will fit in a single row (3 panes x 4 columns = 12).
5. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, panes that have Column Span overrides will move. You can observe the same behavior in the ViewApp editor, once you have added the layout to a ViewApp. To check runtime behavior, use the **Preview** button in the ViewApp Editor, or deploy the ViewApp. As you resize the ViewApp, panes with Column Span overrides will move as the width of the ViewApp changes.
 6. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

Mostly fluid responsive layout example

1. Open the layout in the **Layout Editor**.
 2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/Responsive** button  in the menu bar.
 3. Add panes for each content item by clicking the **New Pane** button .
- Each new pane is added with default settings for width and height. You can modify these settings later.
4. Select the **Mobile** (Extra Small) screen size and check the pane properties. Each pane should show its default property settings.
 - Use the default setting as is (12 columns).
 5. Select the **Tablet** (Small) screen size.
 - For the **Tablet** screen size, change the **Column Span** property from 0 to 6 for both Pane 1 and Pane 2. These two adjacent panes will now be shown side-by-side. To display additional adjacent panes side-by-

side (but in a different row), change their **Column Span** properties as well to numbers that add up to 12 or less.

- You can leave the **Height** property at the default, or modify it as needed. Panes that are shown side-by-side do not need to have the same height setting, and the height setting remains the same for all screen sizes.

Note: Property settings propagate from smallest screen size to largest. When a parameter is set to 0, as in this case, it indicates that the parameter set for a smaller screen (in this case, Mobile), will be used the **Mobile** settings. It is not necessary to set overrides for every screen size. You only need to override a setting if you want to change the **Column Span** from the prior (smaller) screen setting.

6. Select the **Laptop** (Medium) screen size. For this size device

- For the **Laptop** screen size, change the **Column Span** property from 0 to 4 for Pane 1, Pane 2, and Pane 3. These three adjacent panes will now be shown in one row. To display additional adjacent panes side-by-side (but in a different row), change their **Column Span** properties as well to numbers that add up to 12 or less.

7. Select the **Desktop** (Large) screen size.

- For the **Desktop** screen size, change the **Column Span** property from 0 to 3 for Panes 1, 2, 3 and 4. These four adjacent panes will now be shown in one row. To display additional adjacent panes side-by-side (but in a different row), change their **Column Span** properties as well to numbers that add up to 12 or less.

8. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, panes that have Column Span overrides will move to allow panes that total 12 columns or less to next to each other.

- Once you have added the layout to a ViewApp, you can observe the same behavior in the ViewApp editor. To check runtime behavior, use the **Preview** button in the ViewApp Editor, or deploy the ViewApp.
- As you resize the ViewApp, panes with Column Span overrides will move as the width of the ViewApp changes.

9. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

If the TV screen size is added to the product:

1. Select the **TV (Extra Large)** screen size.
2. Change the **Column Span** property for Panes 1 through 6 from 0 to 2. These six adjacent panes will now be shown in one row.
3. To display additional adjacent panes side-by-side (but in a different row), change their **Column Span** properties as well to numbers that add up to 12 or less.

Configure a column drop responsive layout

Video Tutorial: Create a layout in AVEVA OMI

https://player.vimeo.com/video/992308927?badge=0&autoplay=0&player_id=0&app_id=58479

Column Push/Column Pull Properties

The Column Drop method for configuring a responsive layout uses the **Column Push** and **Column Pull** properties in addition to the **Column Span** property. See [Responsive Layout: Mostly Fluid](#) for information about the Column Span property.

- The **Column Push** property moves the selected pane to the right by the number of columns specified.
- The **Column Pull** property moves the selected pane to the left by the number of columns specified.

Column Push and Column Pull let you rearrange the order in which panes are presented at different resolutions. For example, when viewed on a mobile device, you probably want to have the main content, such as an overview graphic, shown first and followed less critical content, such as a navigation pane that the user scrolls down to view. However, when these panes are resized for viewing on the desktop, the navigation pane ends up on the right side of the screen instead of to the left.

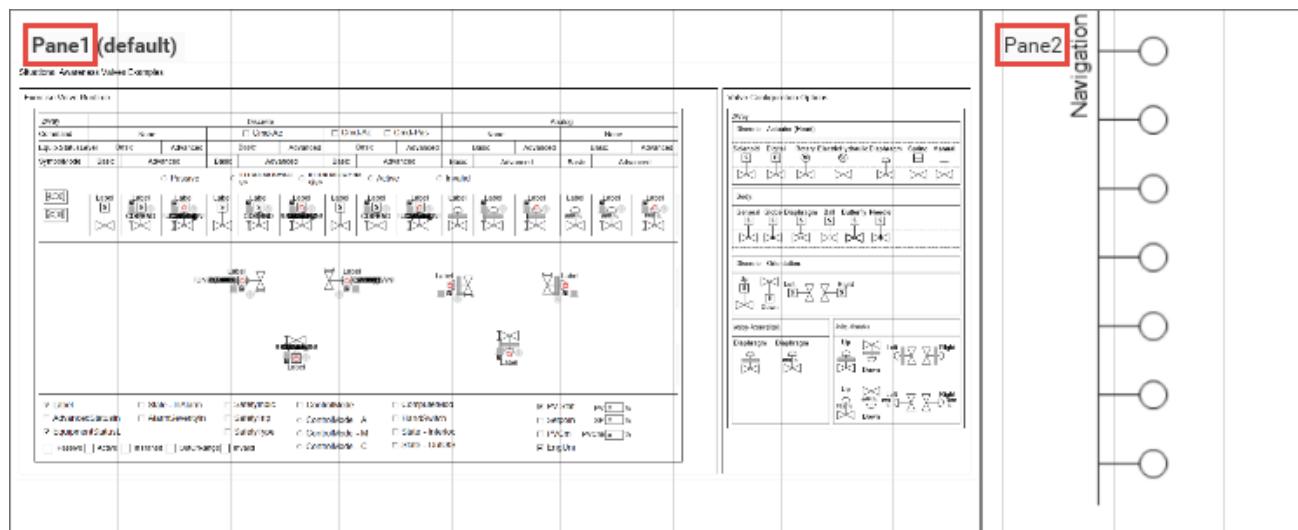
Use the Column Push and Column Pull properties to transpose the positions of two adjacent panes when using larger screen sizes.

To Configure a Column Drop Responsive Layout

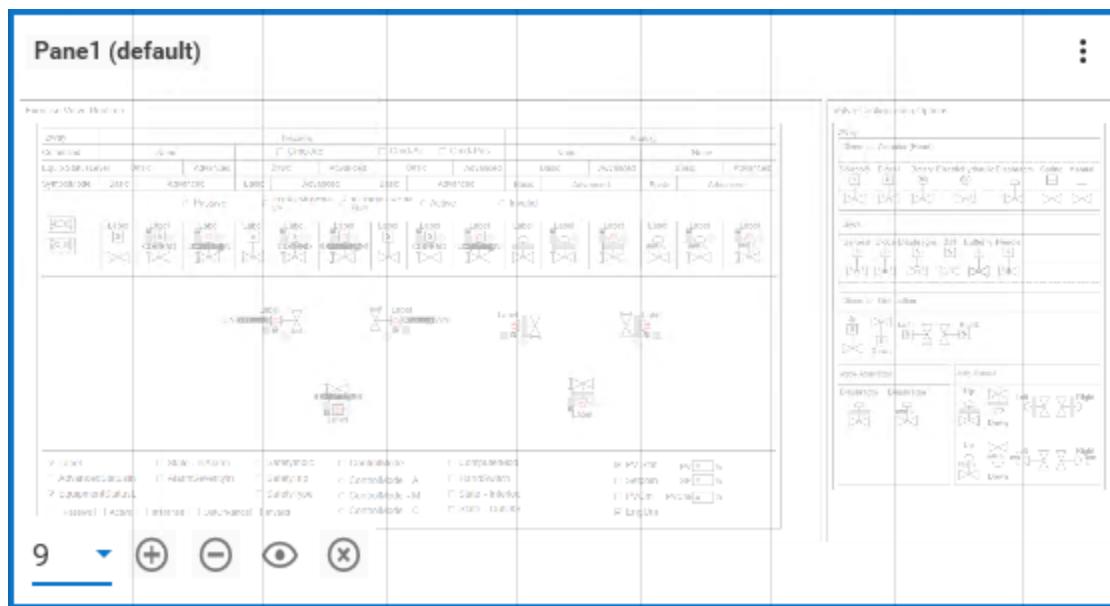
1. Open the layout in the **Layout Editor**.
2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/Responsive** button  in the menu bar.
3. Add panes for each content item, as described in [Responsive Layout: Mostly Fluid](#) .
4. Resize panes using the Column Span property, as described in [Responsive Layout: Mostly Fluid](#) .
5. To transpose two adjacent panes:

The follow screenshot shows two adjacent panes that have been resized to fit a single row when viewed on a

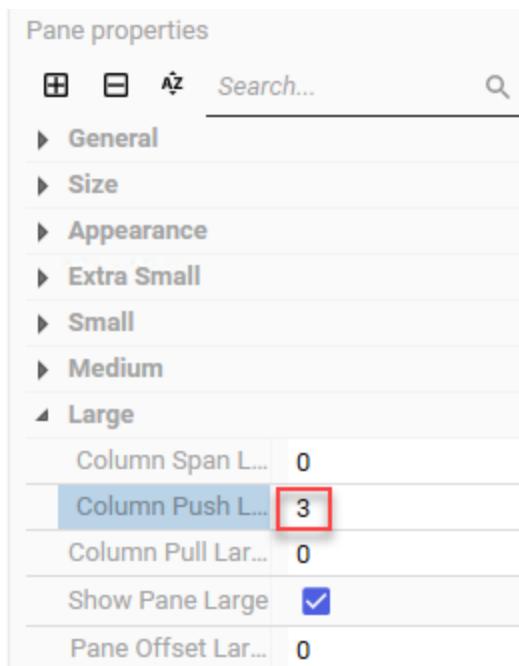
Desktop (large) screen. The main content of the layout is in Pane1 and navigation is in Pane2. For this procedure, Column Span for Pane1 is set to 9, and Column Span for Pane2 is set to 3.



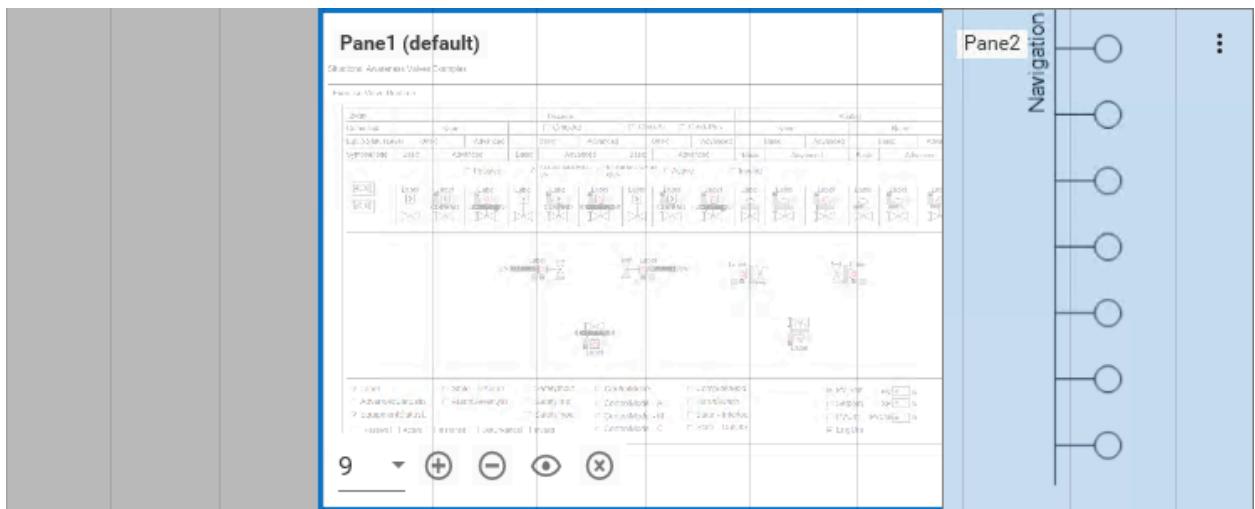
- Select the pane you want to move right (Pane1).



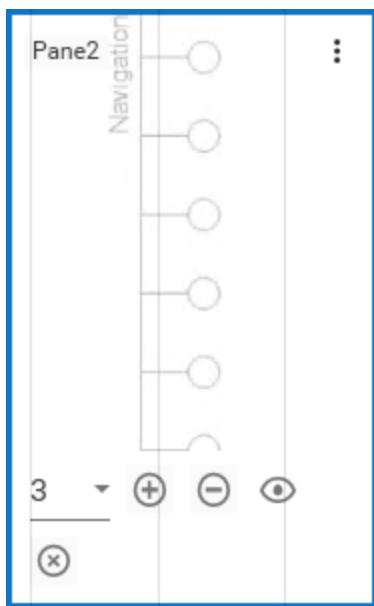
- Expand the screen size setting (Small, Medium, Large or Extra Large) that you are configuring, and set **Column Push** equal to the **Column Span** (3) of the pane to the right (Pane2).



Note that Pane1 moves three columns to the right and slides under Pane2.

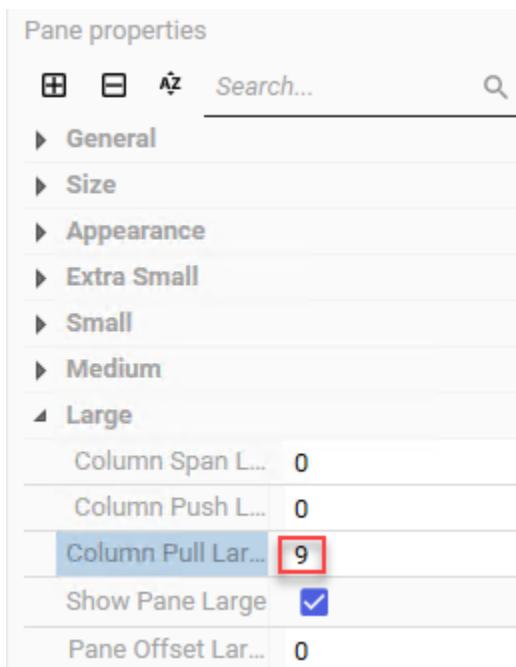


c. Select Pane2.



- d. Set **Column Pull** equal to the **Column Span** of the pane to the left (Pane1).

Pane 2 now slides to the left side of the layout.



6. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, Pane1 and Pane2 will switch positions.
7. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

To check runtime behavior, add the layout to a ViewApp. You can use the **Preview** button in the ViewApp Editor, or deploy the ViewApp. As you resize the ViewApp, the two panes will switch positions as the width of the ViewApp changes.

Configure a layout shifter responsive layout

Video Tutorial: Create a layout in AVEVA OMI

https://player.vimeo.com/video/992308927?badge=0&autoplay=0&player_id=0&app_id=58479

Show Pane Property

The Layout Shifter method for configuring a responsive layout uses the **Show Pane** property to allow different content to be displayed as a function of screen size. Layout Shifter also uses **Column Push** and **Column Pull** properties, introduce in the Column Drop method. **Column Span**, as the basic property that sets the width of a pane, is used in all responsive layout methods.. See [Responsive Layout: Mostly Fluid](#) for information about the Column Span property, and [Configure a column drop responsive layout](#) for more information about the Column Push and Pull properties.

The Show Pane property lets you show different content, contained in different panes, at different resolutions. Returning the navigation example presented in [Configure a column drop responsive layout](#), we may want to use vertically-oriented content at some screen resolutions, and horizontally-oriented content at other resolutions. With the Show Pane property, we can selectively show and hide content, depending on the width of the screen the user is using. The Show Pane property lets us conditionalize panes and their content, so they are shown only at specific screen sizes.

Note: The ShowContent() method does not work with hidden panes.

Use the Show Pane property along with other responsive layout properties to show/hide and change positions of selected panes.

For additional information about the responsive web design, including the Layout Shifter method, see "Learn Responsive Design" at [web.dev](#).

To Configure a Layout Shifter Responsive Layout

1. Open the layout in the **Layout Editor**.
2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/Responsive** button  in the menu bar.
3. Add panes for each content item, as described in [Responsive Layout: Mostly Fluid](#). As you add panes, think of the conditional
4. Resize panes using the Column Span property, as described in [Responsive Layout: Mostly Fluid](#).
5. Configure Column Push/Pull if needed for larger screen sizes, as described in [Configure a column drop responsive layout](#).
6. Determine which content is needed for each screen size, and the position in which the content should be displayed. For example:
 - For mobile devices, you want to show several panes that contain important graphics. You do not need to show a navigation pane, or you want the navigation pane to have its content oriented horizontally (such as in the Breadcrumb navigation app) and below the other content.
 - For desktop displays, you want the navigation pane to be the first pane and to have its content oriented vertically, such as in the NavTree navigation app.
7. For each applicable screen size for the conditional panes, select the pane and then enable or disable the **Show Pane** property.
8. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, panes will be shown or hidden.
9. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

To check runtime behavior, add the layout to a ViewApp. You can use the **Preview** button in the ViewApp Editor, or deploy the ViewApp. As you resize the ViewApp, the conditional panes will be shown or hidden as the width of the ViewApp changes.

Configure a layout shifter responsive layout example

The layout shifter pattern is the most complex of the responsive layout designs. Unlike a mostly fluid or column drop design, layout shifter actually presents users with different pieces of content when viewing the application at different screen sizes.

Show Content Property

In addition to the Column Span, and Push/Pull properties used for mostly fluid and column drop designs, respectively, the **Show Content** property allows different content to be shown or hidden at specific screen sizes.

The following example leverages Show Content to show and hide different navigation panes. For smaller screen sizes, we will show the navigation pane arranged horizontally across the screen, while for larger screen sizes, we will show a vertical navigation tree at the left of the screen. In the example, we will:

- Show the navigation pane with content arranged horizontally below the main content for mobile devices.
- Show the navigation pane with content arranged horizontally above the main content for tablets and laptops.

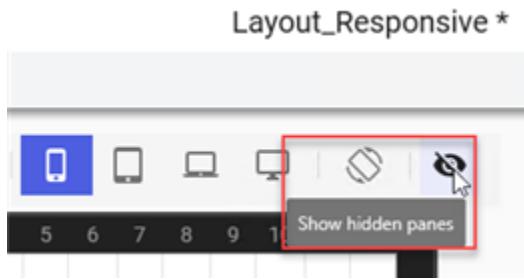
- Show the navigation pane with content arranged vertically to the left of the main content for large and extra large screens.

Note that this is a simplified example that is intended only to illustrate how you can use the Show Content property. Depending on your requirements, you can combine all the elements of mostly fluid, column drop, and layout shifter within a single layout.

1. Open the layout in the **Layout Editor**.
2. Switch to Responsive Layout view by enabling the **IsResponsive** layout property, or by clicking the **Fixed/Responsive** button  in the menu bar.

Pane 1 configuration (small device navigation)

3. Select **Pane 1** and set its height to 100 (the pane height property is under the **Size** group of pane properties and has a default value of 300). Retain the default settings for all other pane properties.
4. Click the **Show Hidden Panes** button  at the top of the layout to show panes regardless of how the **Show Pane** property is set. Showing hidden panes can help as you configure panes and their content.



5. With **Pane 1** selected, make sure that the **Show Pane Extra Small** and **Show Pane Small** properties are enabled (checked). Disable (uncheck) the following Show Pane properties:
 - **Show Pane Medium:** disabled
 - **Show Pane Large:** disabled
 - **Show Pane Extra Large:** disabled

Note: The **Show Pane** property, unlike the rest of pane properties, does not propagate and must be explicitly enabled or disabled for each screen size. The default state of the Show Pane property is enabled.

6. From the **Toolbox** tab, add the **NavBreadcrumbControl** navigation app to Pane 1. The NavBreadcrumbControl is a horizontally-oriented navigation control.

Pane 2 configuration (large device navigation)

7. Add a pane (Pane 2) by clicking the **New Pane** button .
8. If the **Toolbox** tab is still selected, reselect the **Properties** grid.
9. With **Pane 2** selected, make sure that the **Show Pane Medium** and **Show Pane Large** properties are enabled (checked). Disable (uncheck) the following Show Pane properties:
 - **Show Pane Extra Small:** disabled
 - **Show Pane Small:** disabled
10. Set the **Column Span Medium** property to 2 columns for **Pane 2**. This setting propagates to the larger screen sizes. Since Pane 2 is not shown for mobile (extra small) and tablet (small) devices, **Column Span** for these smaller screen sizes is irrelevant.
11. From the **Toolbox** tab, add the **NavTreeControl** navigation app to Pane 2. The NavTreeControl is a vertically-

oriented navigation control.

Pane 3 configuration (main content)

12. Add a pane (Pane 3) by clicking the **New Pane** button .
13. With **Pane 3** selected, configure the **Column Span Medium** property to 10 columns. This setting propagates to the larger device sizes, but the settings for **Column Span Extra Small** and **Column Span Small** do not change from 12 columns.
14. Add an overview graphic or other graphic to Pane 3. This will function as the main content for the layout.
15. To test the responsive behavior within the Layout Editor, drag the right or left edge of the layout to change its width. As the layout size exceeds or falls under the dimension that defines a particular layout size, the panes containing navigation content (vertical or horizontal) will be hidden or shown as applicable to the screen size.
 - Once you have added the layout to a ViewApp, you can observe the same behavior in the ViewApp editor. To check runtime behavior, use the **Preview** button in the ViewApp Editor, or deploy and launch the ViewApp.
 - As you resize the ViewApp, the Show Pane property determines which pane is shown as the width of the ViewApp changes.
16. Click the **Save and Close** button on the menu bar to save your changes and close the Layout Editor.

Responsive layout limitations

Limitations Inherent in a Responsive Layout

The following parameters and properties cannot be changed in a responsive layout:

- The values defined for responsive layout breakpoints cannot be changed. That is, the transition screen widths between layout sizes (extra small, small, medium, etc.) are fixed.
- The **AspectRatio** property is not supported for panes within a responsive layout.
- The **DisplayMode** property will not have any effect if a responsive layout is placed as the content of a pane in another layout.
 - Width: The responsive layout will occupy the full width of the pane in which it is placed.
 - Height: If the height of the responsive layout exceeds the height of the pane in which it is placed, a vertical scroll bar will appear.
- When a responsive layout is placed as the content of a pane in a fixed layout, the width of the fixed pane (NOT the width of the fixed layout) determines which breakpoint (screen size) the responsive layout will use. For example, consider a fixed layout that is 1280 pixels wide. It contains a pane that is 640 pixels wide that holds a responsive layout. In this case, the responsive layout uses the extra small (mobile) breakpoint, since the extra small size allows a layout up to 767 pixels wide. The overall size of the fixed layout is ignored.

InTouch Web Client Limitations

The following limitations apply when you view a ViewApp that uses a responsive layout in the InTouch Web Client:

- The InTouch Web Client does not support placing a responsive layout as the content of a fixed layout pane.
- The InTouch Web Client does not support placing a fixed layout as the content of a responsive layout pane.

- These limitations do NOT apply to the AVEVA OMI ViewApp at runtime.

Build layout scripts

The **Layout Editor** includes a tab for adding and editing scripts for execution at runtime. You can use different triggers, such as opening or closing the layout, to start the execution of a layout script, or setting an interval at which the script executes while the layout is open.

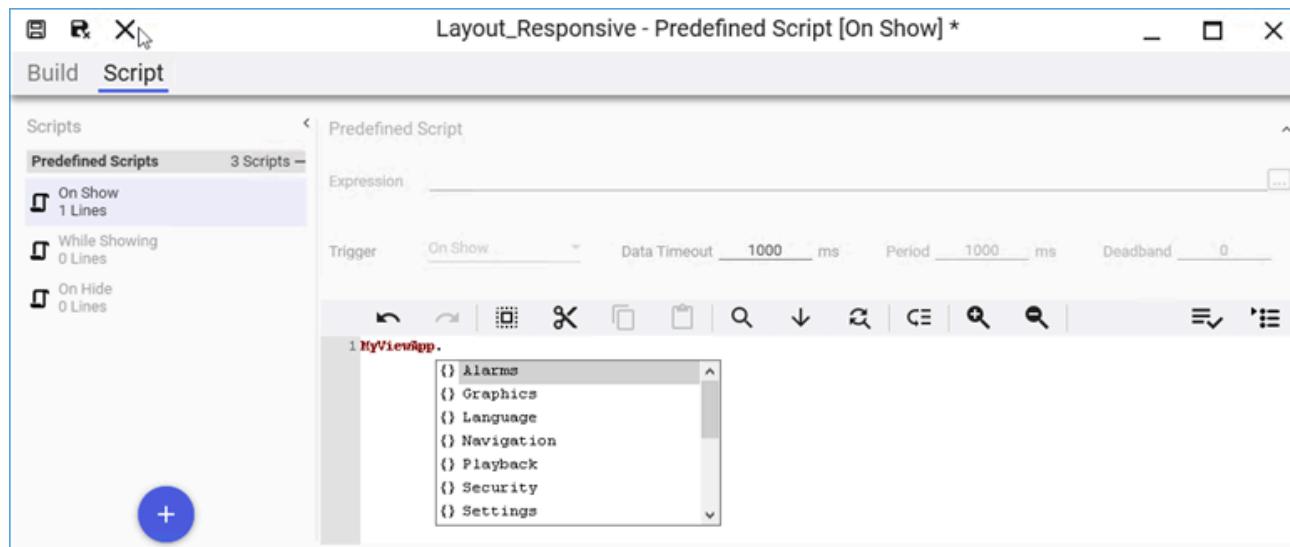
On Show layout scripts are executed before any graphic scripts contained within graphics in the layout, while On Hide layout scripts are executed after the graphic scripts. Other script types do not have a specific order of execution and may not execute in the same sequence each time. You can also add scripts triggered by events in an app or control, such as PropertyChange, MouseDown, etc.

Layout scripts work much like the graphic scripts that can be added to graphics to enable animation of the graphic or its elements. For more information about graphic scripts, see [About Symbol Scripts](#).

The layout script editor includes the following features:

- Autocomplete: as you type, the editor presents a dropdown list of namespaces, attributes, custom properties, exposed app properties, scripting methods, and other items that match the characters that you have entered. The items in dropdown list are filtered by context, so only valid items are listed in the dropdown.

Note: All public app properties will appear in the autocomplete list, even though not every public app property that appears in the list can be scripted. Properties that are not data-bound cannot be scripted. You can script public properties that contain their backing logic within the app itself. However, some apps which wrap around an underlying control, such as the Alarm App, contain pass-through properties which may require additional qualification. In a pass-through property, the backing logic is contained within the wrapped control. Therefore, to script a pass-through property, you must invoke the control namespace, even though autocomplete, by virtue of the property being public, will allow you to enter the property name with or without the control namespace. See [Alarm client control pass-through properties](#) for details.



The Layout Script Editor autocomplete feature works the same way as the Application Server QuickScript Editor and the Graphic Script Editor. For more information about Application Server scripting, see the Application Server Scripting Guide.

- Event handler: add scripts that are triggered by user interaction with an app, such as a mouse click or a menu

opening.

- Search function: includes options to find, find next, or replace a specified string.
- Undo/redo.
- Script validation: flags invalid entries in the script.

The script editor contains three separate areas:

- A pane at the left side of the editor lists all scripts within the layout. This list always includes the three built-in scripts (**On Show**, **While Showing**, and **On Hide**), and any named scripts and event-handler scripts that you add. Built-in scripts cannot be deleted or renamed, and are grayed-out until a script is added (regardless of the validity of the script). The number of lines in the script appears under the listed script name. If the script is not configured, the script shows that contains 0 lines. You can collapse and expand the script-list pane by selecting the chevron at the top right of the pane.
- The top pane of the script editor contains an area to enter an expression and script trigger type. Other fields in the top pane are for timeout, period, deadband, and whether to include quality changes. These fields are enabled only if they are applicable to the selected trigger.
- The main portion of the script editor has buttons for undo/redo, cut, duplicate, zoom in/out, and other editor functions. A button at the right of the pane lets you fully expand the editing pane. The cursor position is tracked by row and column numbers displayed at the bottom of the pane.

Types of layout scripts

The **Script** tab shows the scripts that have been configured for the layout:

- **Built-in scripts.** There are three built-in scripts that are automatically added to all layouts. These are listed in the **Script** tab and are always shown, whether or not they have been configured. The three built-in scripts are:
 - On Show
 - While Showing
 - On Hide
- **Named scripts.** You can add and configure an unlimited number of named scripts. These are expression-based scripts that can use any of the following triggers:
 - On True
 - On False
 - While True
 - While False
 - Data Change (this can also include a change in data quality)
- **Event-handler scripts.** These can be added and configured for apps and controls in the layout that contain public events. There are numerous types of events that can be scripted, and the events will be different between apps. Some examples of the types of events that can be scripted include:
 - Got Focus
 - Mouse Move
 - Mouse Double Click

- Drag Over
- Drop
- Key Up
- Context Menu Opening

About built-In layout scripts

Built-in scripts for layouts are similar in both concept and function to the Predefined Scripts for graphics that can be invoked through the **Industrial Graphic Editor**. For more information about the graphic built-in scripts, see [Predefined graphic script configuration](#). Built-in scripts are triggered by the state of the layout itself, and do not use an expression or reference.

- **On Show:** The script runs one time when the layout opens. You can specify a timeout period for the script in milliseconds to stop the script if a referenced attribute cannot be accessed.
- **While Showing:** The script runs at a specified interval (in milliseconds) while the layout is open.
- **On Hide:** The script runs one time when the layout closes. There are no configurable trigger parameters.

You can have only one built-in script of each type, for a maximum of three built-in scripts, per layout. See [About Built-in scripts](#) for more information.

About named scripts

You can add named layout scripts that are triggered by an expression or reference. The expression or reference is used to define the specified condition. There is no limit on how many expression-based scripts that you can include in a layout.

- **While True:** The script runs periodically, at a specified interval (in milliseconds), for as long as the expression remains true.
- **While False:** The script runs periodically, at a specified interval (in milliseconds), for as long as the expression remains false.
- **On True:** The script runs one time when the expression transitions to true.
- **On False:** The script runs one time when the expression transitions to false.
- **Data Changed:** The script runs whenever data referenced by the expression changes. You can specify a deadband to filter out inconsequential data changes. In addition, you can specify if quality changes will also be used to trigger the script to run.

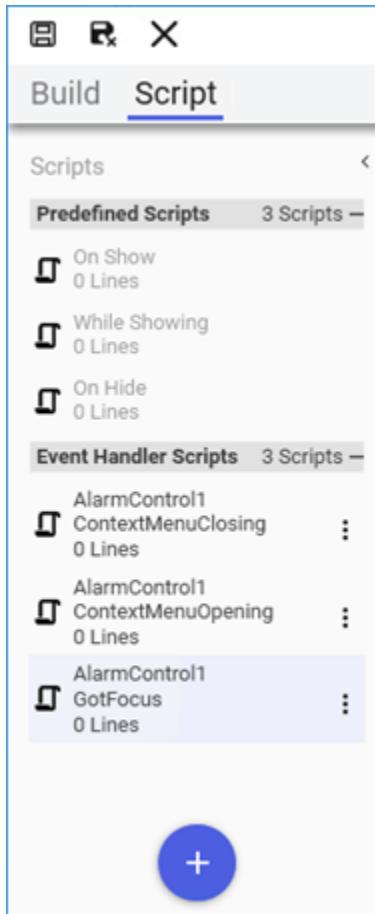
See [Add a named script](#) for more information.

About event-handler scripts

You can script against public events contained in apps and controls. When you add an app or control to the layout that contains public events, you can add event handler scripts from either the Build or Script tab that are triggered by selected events. The script is then triggered when, for example, the app detects a MouseDown or DataChange event.

Event-handler scripts are named for the app or control and the selected event, and cannot be renamed. They

appear in the Scripts pane at the left of the **Script** page, and are listed in alphabetical order, after build-in scripts and named scripts. Like the other two categories of scripts, event handler scripts are listed in their own folder. See [Layout event handler scripts](#) for more information.



Namespaces for use with layout scripts

Layout scripts can leverage both local and global namespaces, including the [About the MyContent namespace](#), which is local to each individual layout, and [About ViewApp namespaces](#), which are global across the Galaxy. The MyContent namespace can be referenced by layout scripts and expressions used within that layout. This provides the ability to isolate attributes within an individual layout, as opposed to the ViewApp namespace, which can be referenced by all layouts and ViewApps in the Galaxy.

Event-handler scripts, unlike named and built-in scripts, do not require the use of the MyContent namespace when scripting content properties and methods. To script a property or method, start by entering the name of the control instead of "MyContent." The autocomplete function will allow you to see the properties and methods contained in the control as you type.

About Built-in scripts

You can configure one of each type of built-in scripts for a layout. There is no requirement to configure built-in scripts. Unconfigured scripts are not used at runtime and do not generate errors or warnings.

Built-in scripts are named for their trigger type, and cannot be renamed or deleted. A built-in script is not

triggered by an expression, but instead uses the state of the layout to define when the script executes. The built-in scripts are::

- **On Show:** The script runs one time when the layout opens.
- **On Hide:** The script runs one time when the layout closes.
- **While Showing:** The script runs periodically for as long as the layout remains open.

For details about the Script Editor, see [Build layout scripts](#). See also [About the MyContent namespace](#) for information about referencing different content contained in the layout.

Note: The three **Built-in Scripts**, each named for a trigger type (**On Show**, **While Showing** and **On Hide**), cannot be deleted from the script list. Only one built-in script per trigger type can be configured for a single layout.

Configure built-in script

To configure a built-in script

1. Open the layout in the Layout Editor and select the **Scripts** tab.
2. Select one of the three built-in scripts from the script list in the left pane of the **Scripts** tab:
 - **On Show:** Use this to configure a script that runs one time when the layout opens.
 - **While Showing:** the script runs periodically while the layout is open.
 - **On Hide:** the script runs one time, when the layout closes.
3. Set any applicable timing parameters for the script.
 - **On Show:** the default **Data Timeout** is 1000 ms. See [Tips to ensure proper On Show script execution](#) for more information about configuring the timeout period.
 - **While Showing:** the default **Period** is 1000 ms. This value sets the interval at which the script will run while the layout is showing.
 - **On Hide:** There are no configurable timing parameters for an On Hide script. The script runs only once when the layout closes.
4. Enter your script in the main edit box. Autocomplete helps by displaying valid items, based on the initial characters you enter. The script syntax is the same as the syntax of AutomationObject scripting.
5. Enter your script in the main edit box. enter.Autocomplete helps by displaying valid items, based on the initial characters you enter. The script syntax is the same as the syntax of AutomationObject scripting.
 - You can validate your script for syntax errors prior to saving by pressing the **Validate Script**  button.
 - To clear all data entered for the the script, including the expression, press the **Clear Script**  button. There is no undo for this action.
6. To save the script, select the **Save**  or **Save and Close**  button.
7. The script is validated automatically. If there are warnings or errors generated, a dialog listing the errors is displayed. You have the following options:
 - Select **Yes** to save the script and preserve the errors.
 - Select **No** to discard the changes you made to the script. If you selected **Save and Close**, the Layout editor closes and you will lose any changes you made to the script. If you selected **Save**, the Layout Editor remains open and you can resume editing the script.
 - Select **Cancel** to resume editing the script. The Layout Editor remains open and you can resume editing

the script.

Tips to ensure proper On Show script execution

When an **On Show** script includes external references to ApplicationObject attributes, data from these attributes may not be immediately available when the **On Show** script runs. As a result, the script might not work properly.

To mitigate this possibility, you can enter a value in the **Data Timeout** field. The **Data Timeout** period sets a time limit to allow reference data to become available. At the end of the **Data Timeout** interval, or before the timeout expires if the reference data is available sooner, the **On Show** script is executed. Note that data quality is not checked, only that the data is available.

If the **Data Timeout** expires before all reference data is available, the **On Show** script is still executed. However, the script might not work properly.

You can reduce the possibility of linking to invalid data by having the script first test the quality and status of the attribute before using it in the rest of the script.

The default value in the **Data Timeout** field is 1000 ms (one second). The maximum data time-out value is 30,000 ms (30 seconds).

Note the following issues regarding **On Show** scripts and the **Data Timeout** function:

- The **Data Timeout** function is not available for the other trigger script types. It would be rare for external reference data to not be available in time for those scripts.
- The execution of the **On Show** script is not delayed if there is an invalid reference (that is, the reference's quality is Bad).
- Named scripts are blocked until the **On Show** script has completed, so some could be missed. For example, the named script **OnDataChange** might not run for the first few updates.
- Delayed **On Show** scripts within nested embedded graphics might run out of order for the different nested levels. If the outer-most level is delayed but the inner levels are not delayed and are executed immediately, the order of execution will be changed.

Note: You can increase the effective data time out period for an **On Show** script by using the **Additional settings** feature when editing an OMI ViewApp. See [Set additional properties for a ViewApp](#) for more information.

Add a named script

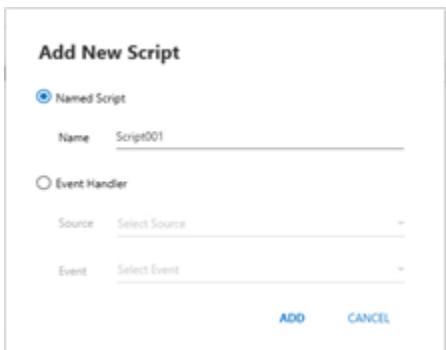
In addition to the built-in scripts that are triggered by the status of the layout, you can add scripts that are triggered by expressions, or by changes in values and/or data quality. These expression-based scripts are called "named scripts." You can configure a named script to run:

- One time when an expression changes from false to true or true to false, or when a value or its quality changes. Scripts that use the **On True**, **On False**, and **Data Change** triggers will run once when the specified event occurs.
- Periodically, at the specified interval, when the trigger is set to **While True** or **While False**.

Each added script can contain only one trigger type.

To add a named script to a layout

1. Open the layout in the Layout Editor and select the **Script** tab.
2. Click the **Add Script**  button.
3. Select the **Named Script** choice from the pop-up window.



4. To rename the script from the default "Script00n," click or touch the script name in the **Name** field and edit the name.
5. Click **Add**. The new script is created in the Scripts list.

To rename the script after it has been added to the Scripts list, click or touch the script name and edit the name.

To configure a named script

1. In the **Expression** field, enter an expression, value or reference. As you type, an autocomplete menu will present any valid items that start with the characters you are typing.
The expression acts as data source for the script trigger.
2. In the **Trigger** field, select the script trigger:
 - **While True** triggers the script to run periodically when the expression is true. The default **period** for **While True** scripts is 1000 ms. To change the period, enter a new value in the field.
 - **While False** triggers the script to run periodically when the expression is false. The default **period** for **While False** scripts is 1000 ms. To change the period, enter a new value in the field.
 - **On True** triggers the script to run one time when the expression transitions to true from false.
 - **On False** triggers the script to run one time when the expression transitions to false from true.
 - **Data Change** triggers the script one time when the value of the expression or its quality changes. You can specify a **deadband** value to filter insignificant changes. To trigger the script for changes to both value and quality, select the **Quality Changes** check box.
3. If you are writing/editing a **While True** or **While False** script, specify the frequency that the script will be executed by entering an interval (in milliseconds) in the **Period** field.

The **Period** field specifies the length of the interval from the start of one script execution cycle to the next, while the layout remains open. If the **Period** field is set to a very short interval, system performance may decrease. Enter an interval value in milliseconds.

- Default: 1000 ms (one second)
- Maximum: 360,000 ms (six minutes)

Note: **Data Timeout**, **Deadband**, and **Quality Changes** fields are not applicable to, and are disabled for **While**

Showing scripts.

4. Enter your script in the main edit box. Autocomplete helps by displaying valid items, based on the initial characters you enter.
 - You can validate your script for syntax errors prior to saving by pressing the **Validate Script**  button.
 - To clear all data entered for the script, including the expression, press the **Clear Script**  button. There is no undo for this action.
5. To save the script, select the **Save**  or **Save and Close**  button.
6. The script is validated automatically. If there are warnings or errors generated, a dialog listing the errors is displayed. You have the following options:
 - Select **Yes** to save the script and preserve the errors.
 - Select **No** to discard the changes you made to the script. If you selected **Save and Close**, the Layout editor closes and you will lose any changes you made to the script. If you selected **Save**, the Layout Editor remains open and you can resume editing the script.
 - Select **Cancel** to resume editing the script. The Layout Editor remains open and you can resume editing the script.

About the ShowContent() function

Note: The Show/Hide Content() script functions are for use with AVEVA OMI ViewApps only. These functions cannot be used with InTouch HMI applications.

The Show/Hide Content() script functions let you write scripts for AVEVA OMI layouts, Industrial Graphics, Application Server objects to display a graphic, layout, or external content item (for example, a web page, video, or text file) inside a specific pane within an AVEVA OMI ViewApp. The Show/Hide Content() script functions are complementary to the Show/Hide Graphic script functions and the Show/Hide Graphic animation feature.

- Use the Show/Hide Graphic script functions to load or hide a graphic in a modal or modeless popup, in an InTouch HMI application or an AVEVA OMI ViewApp.
- Use the Show/Hide Content script functions to load or hide a graphic, layout, or an external content item into a pane in an AVEVA OMI ViewApp. When a Layout linked to an asset is added to a pane in another layout, any empty panes in the linked Layout that have the "Use for Autofill" flag enabled will attempt to fill the empty panes with content from the linked asset, using the "Current Only" autofill setting.

The Show/Hide Content functions let you specify:

- The specific content to show or hide in a pane.
- The pane in which to show or hide the content.
- The screen that contains a pane in which to show or hide the content.
- The type of content you want to show or hide.
- Any property overrides to apply when showing the content.
- Whether to confine the search for a pane in which to show or hide the content to the source screen, the primary screen, or to search all screens, and whether to search in the top level layout or a nested (embedded) layout.

Configure the Show/Hide Content script functions

Use the Show/Hide Content functions in layout scripts, Industrial Graphic scripts, or Application Server object scripts to populate a pane with a specific graphic, layout, or external content object in an AVEVA OMI ViewApp. You can call the same script multiple times to refresh content, or to show the content in a different pane. If you choose to display the same content in different panes, you can alter its settings through the ParameterOverride parameter.

If ShowContent calls the same content item while it is open, ShowContent searches for open content items that match the specified parameters in the call. Any matching items are closed and then reopened. If the content has changed, the latest changes are shown when the content is redisplayed.

Important: The Show/HideContent functions can be used in a graphic or layout action script, named script and pre-defined script. Although the system allows you to include it in a server script, such as Start Up, On Scan, Off Scan, Shut Down and Execute, you will not be able to execute the function at runtime.

To include a script that contains the Show/Hide Content functions within a script

1. Open the System Platform IDE.
2. Open a new or existing graphic or layout.
 - To add a layout script, click the **Script** tab.
 - To add a graphic script, create a graphic and open it for editing. Double-click it to open the **Edit Animations** page, then open the script editor and click the **Display Script Function Browser** icon. The **Script Function Browser** appears. Click the ShowContent() script function in the **Graphic Client** list, and then click **OK**. The following code snippet is added:

```
Dim contentInfo as aaContent.ContentInfo;
contentInfo.Content = "<ContentName>";
ShowContent( contentInfo );
```

Note: You can click **Help** to view the Help after you have selected any Graphic Client script function.

3. Modify the script as needed. ContentInfo is a predefined structure that contains data members listed below. Content is the only required parameter. For more information about the Show/Hide Content, see the AVEVA Scripting Guide.
 - **Content:** name of the content (graphic, layout, or external content object) to be loaded into the pane. Content is a string and is the only required parameter.
 - **Name:** name property of the content. This is automatically created when the content is added to the layout.
 - **ScreenName:** name of the screen that contains the pane in which to load the content. ScreenName is a string.
 - **SearchScope:** when ScreenName is not specified, **SearchScope** determines which screens within the Screen Profile are searched for a pane in which to place the content. SearchScope is an enum that can be set to one of the following values:
 - **Self:** This is the default. Searches for matching content within the panes of the layout from which the ShowContent call was made. Self will search from a nested (embedded) layout, if the call was made from that layout. The remaining SearchScopes search ONLY the top level if there are nested layouts.
 - **AllScreens:** Searches the top level layout on all screens, starting with the source screen, then the primary screen, and then any remaining screens in alphabetical order. The search stops as soon as a matching pane is found. If a matching pane is not found, Content is placed in the default pane of the source screen.

- **SourceScreen:** Searches the top level layout only in the source screen (the screen that made the ShowContent call).
- **PrimaryScreen:** Searches the top level layout only in the primary screen, as designated in the Screen Profile.
- **PaneName:** name of the pane in which to load the content. PaneName is a string.
- **ContentType:** specifies the type of content to be loaded. The ContentType can be matched against the Content Type designation that was set for a pane. ContentType is a string.
- **PropertyOverrides:** specifies overrides for custom properties. This parameter is only valid if a graphic is the designated content. It is not valid if a layout has been designated. PropertyOverrides is a ValuePair structure.
- **OwningObject:** specifies an automation object as an owning object.

Example

```
Dim contentInfo as aaContent.ContentInfo;
Dim cpValues [2] as aaContent.PropertyOverrideValue;
cpValues[1] = new aaContent.PropertyOverrideValue("CP1", "20", true);
cpValues[2] = new aaContent.PropertyOverrideValue("CP2", "Pump.PV.TagName", false);
contentInfo.Content = "S1";
contentInfo.ContentType = "Overview";
contentInfo.OwningObject = "Enterprise";
contentInfo.PaneName = "Pane1";
contentInfo.ScreenName = "Wall";
contentInfo.PropertyOverrideValues = cpValues;
contentInfo.SearchScope = aaContent.SearchScope.PrimaryScreen;
ShowContent ( contentInfo );
```

About the best pane match algorithm

When some or all of the optional parameters are not provided or are invalid, ShowContent follows a set of rules to determine in which pane the specified content should placed. In addition, the SearchScope parameter can be used to limit which screens will be searched for a pane.

- If SearchScope is set to SourceScreen, only the screen that made the ShowContent call is searched. If no matching pane is found, Content is loaded into the default pane of the SourceScreen.
- If SearchScope is set to PrimaryScreen, only the primary screen in the Screen Profile is searched. If no matching pane is found, Content is loaded into the default pane of the PrimaryScreen.
- If SearchScope is set to AllScreens, every screen contained in the Screen Profile can be searched. This is the default behavior. Searches progress as follows and stop at the first match:
 - SourceScreen
 - PrimaryScreen
 - Any additional screens are searched in alphabetical order.
 - If no matching pane is found, Content is loaded into the default pane of the SourceScreen.
- If SearchScope and ScreenName are not specified and ShowContent is called from an embedded layout:
 - ShowContent searches locally for a matching pane within the embedded layout.
- If SearchScope or ScreenName is specified and ShowContent is called from an embedded layout:

- ShowContent searches for a pane globally and uses the SearchScope and/or ScreenName settings to determine which screens within the Screen Profile associated with the ViewApp will be searched.

Order of precedence for determining content placement

Content is the only required parameter. Three values are also checked according to the following order of precedence to determine placement of the content:

1. PaneName: optional parameter
2. ContentType: optional parameter
3. Type of content: this is part of the Content definition and is not specified in the ShowContent call.

Search steps (SearchScope not specified or set to AllScreens)

Scenario 1: PaneName PaneName is specified	Assumes SearchScope not set or set to AllScreens. Otherwise, search is confined to the named screen. <ol style="list-style-type: none">1. Search SourceScreen for PaneName.2. If not found, search PrimaryScreen for PaneName.3. If not found, search any additional screens in alphabetical order for PaneName. The search stops as soon as a matching PaneName is found.
Scenario 2: ContentType PaneName is not specified OR matching PaneName not found AND ContentType is specified	Assumes SearchScope not set or set to AllScreens. Otherwise, search is confined to the named screen. <ol style="list-style-type: none">1. Search SourceScreen for a pane that supports the ContentType.2. If not found, search PrimaryScreen for pane that supports the ContentType.3. If not found, search any additional screens in alphabetical order for a pane that supports the ContentType. The search stops as soon as a pane that supports the ContentType is found.
Scenario 3: Type of content PaneName is not specified OR matching	Assumes SearchScope not set or set to AllScreens. Otherwise, search is confined to the named

PaneName not found AND ContentType not specified or not found

screen.

1. Evaluate type of content for the specified Content.
2. Search SourceScreen for a pane that supports the type of content.
3. If not found, search PrimaryScreen for pane that supports the type of content.
4. If not found, search any additional screens in alphabetical order for a pane that supports the type of content.

The search stops as soon as a pane that supports the type of content is found.

Scenario 4: No matching pane

No matches found from previous searches (scenarios 1 through 3)

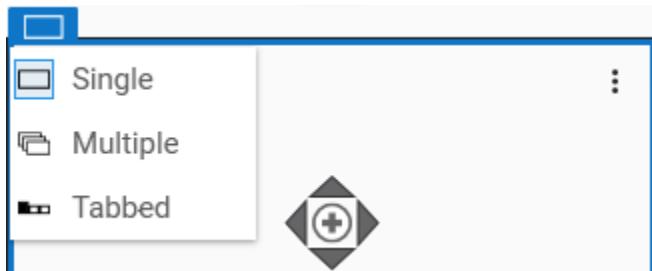
When SearchScope is not set or set to AllScreens:
If no match is found using PaneName, ContentType, or type of content, Content is loaded into the default pane of the SourceScreen.

When SearchScope is set to SourceScreen:
Only the SourceScreen is searched and if no match is found, Content is loaded into the default pane of the SourceScreen.

When SearchScope is set to PrimaryScreen:
Only the PrimaryScreen is searched, and if no match is found, Content is loaded into the default pane of the PrimaryScreen.

Content display rules

The way content is displayed differs slightly between panes that are configured as single content panes versus those configured as multi-content panes. The differences are described below. Pane configuration (single or multi-content) is set by opening the **Presentation Style** selector at the top left of the pane in the **Layout Editor**. See [Set the Presentation Style of a Pane](#) in the *System Platform Help* for additional information about single and multiple content modes.



If there are multiple ShowContent() calls, but with different parameters, each ShowContent() call opens a

different pane or tab. The content in each pane or tab is configured with the parameters that are specified in the call.

Content display rules – single content pane

If the pane already contains content, the existing content is closed and the content specified by ShowContent() then is loaded into the pane. This occurs even if the existing content is the same as the specified content.

Content display rules – multi-content or tabbed pane

For panes that are configured as multiple-content or tabbed panes, the exact behavior will depend on the structure of the pane.

- If the specified Content is already open in the pane or a tab, the pane or tab remains open and the existing Content is replaced with a new instance of the same Content.
- If the pane or tab has been split (that is, contains child panes), or contains a nested layout, existing panes and/or tabs remain open, and the specified content is loaded into a new tab or pane and the focus is set to the new tab or pane.
 - This is true as long as the number of tabs or child panes is less than the maximum available. The default setting allows up to 20 panes or tabs, but you can increase this to 50.
 - If the maximum number of tabs or child panes has been reached, the pane that has been updated last (oldest content) is closed and a new tab or pane is opened in its place.

Layout event handler scripts

The Layout Editor lets you add event-handler scripts to a layout if you have already added an app or control to a pane within the layout that contains public events. There are two ways to add an event-handler script after dropping an app or control onto a pane:

- From the **Build** tab
- From the **Script** tab

When you add an event-handler script, triggered by a specific event that occurs in the selected app, the script is added to the scripts that are listed under "Event Handler Scripts" on the **Script** tab of the Layout Editor. You can then select and configure the script in the script editor.

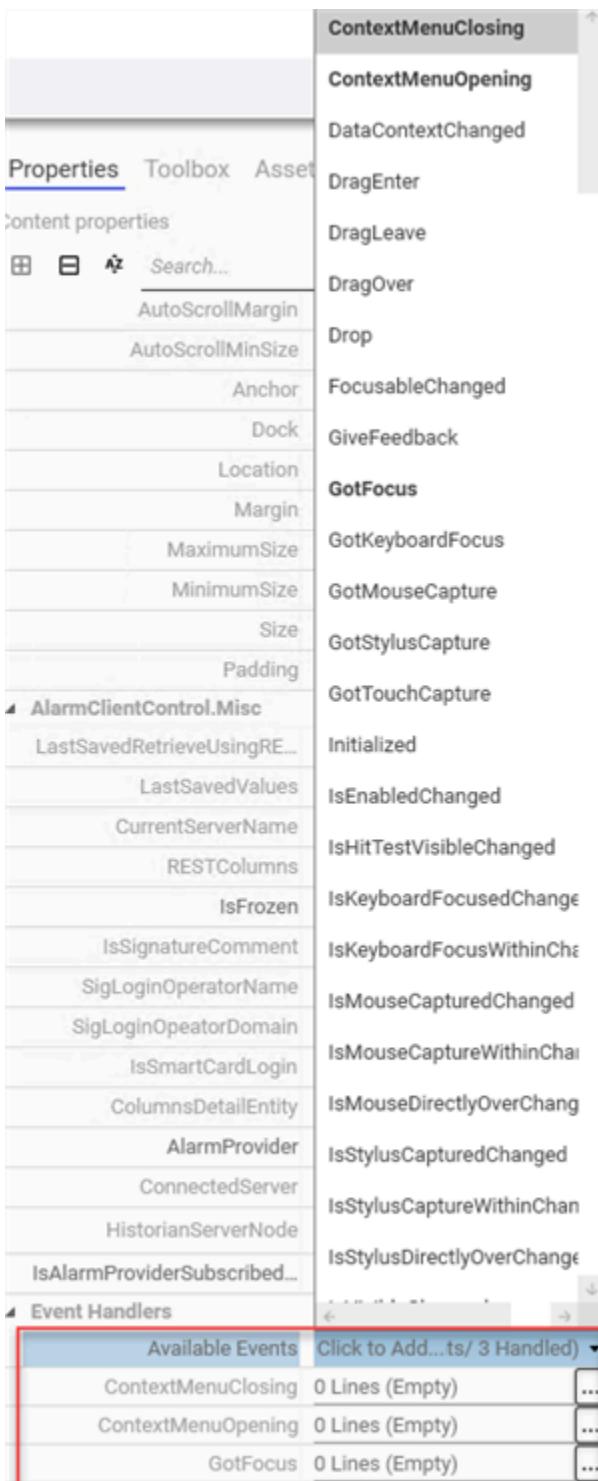
Note: Event-handler scripts, unlike named and built-in scripts, do not require the use of the MyContent namespace when scripting content properties and methods. To script a property or method, start by entering the name of the control instead of "MyContent." The autocomplete function will allow you to see the properties and methods of the control as you type. However, apps based on Windows Presentation Foundation (WPF) or .NET Controls may not provide the related event parameters in autocomplete. See [Layout event handler parameters](#) for more information.

Add an event-handler script to a layout from the Build tab

To add an event-handler script to a layout from the Build tab

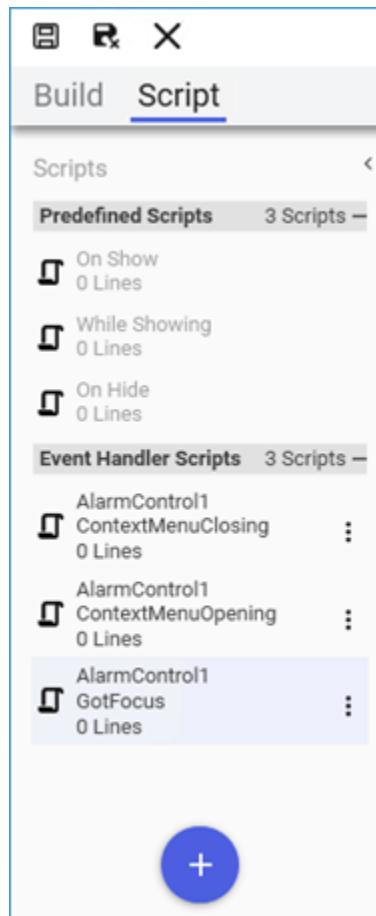
1. Select the pane that contains the app or control that you are scripting.
2. Select **Properties** grid and scroll to the bottom of the properties list.
3. If necessary, expand the **Event Handlers** section to show **Available Events**.

4. To add a script to an event, click on **Click to Add Event...**
5. The list of public public events and methods contained in the app or control is shown.
6. Select the event name for which you want to add a script. After you select the event name, it is shown in **bold** type in the list, and listed below the **Click to Add Event...** option.



7. Switch to the **Script** tab. The new Event Handler script that you just added is listed alphabetically, by name of app and event, under **Event Handler Scripts** in the left column of the editor.

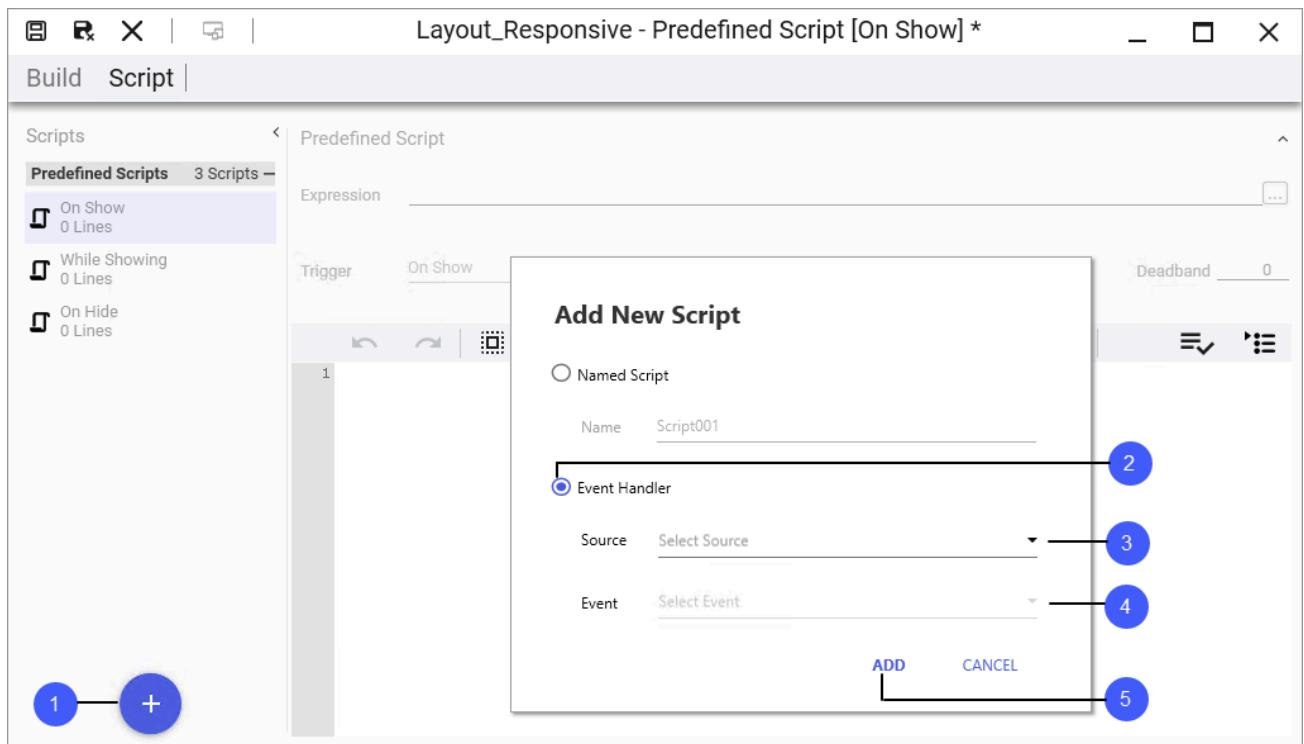
The first line shows the name of the app, the second line shows the event, and the third line shows the number of lines in the script.



Add an event-handler script to a layout from the Script tab

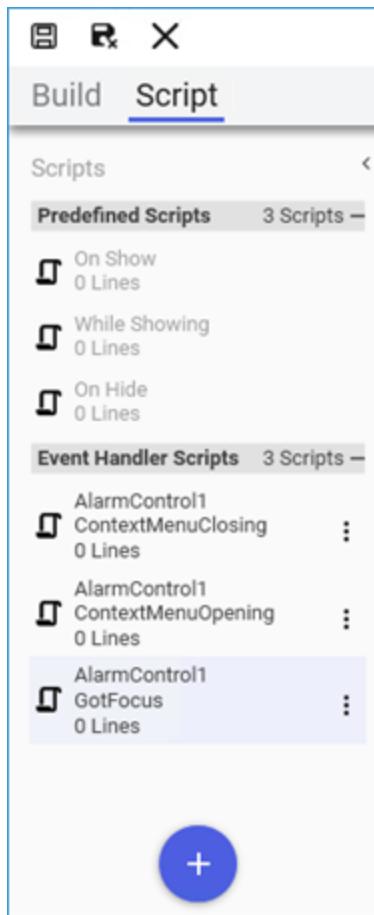
To add an event-handler script to a layout from the Script tab

1. Click the **Add Script**  button.
2. Select the **Event Handler** choice from the pop-up window.
3. Select the app/control name from the **Source** pulldown list.
4. Select the event name from the **Event** pulldown list.
5. Click **Add**.



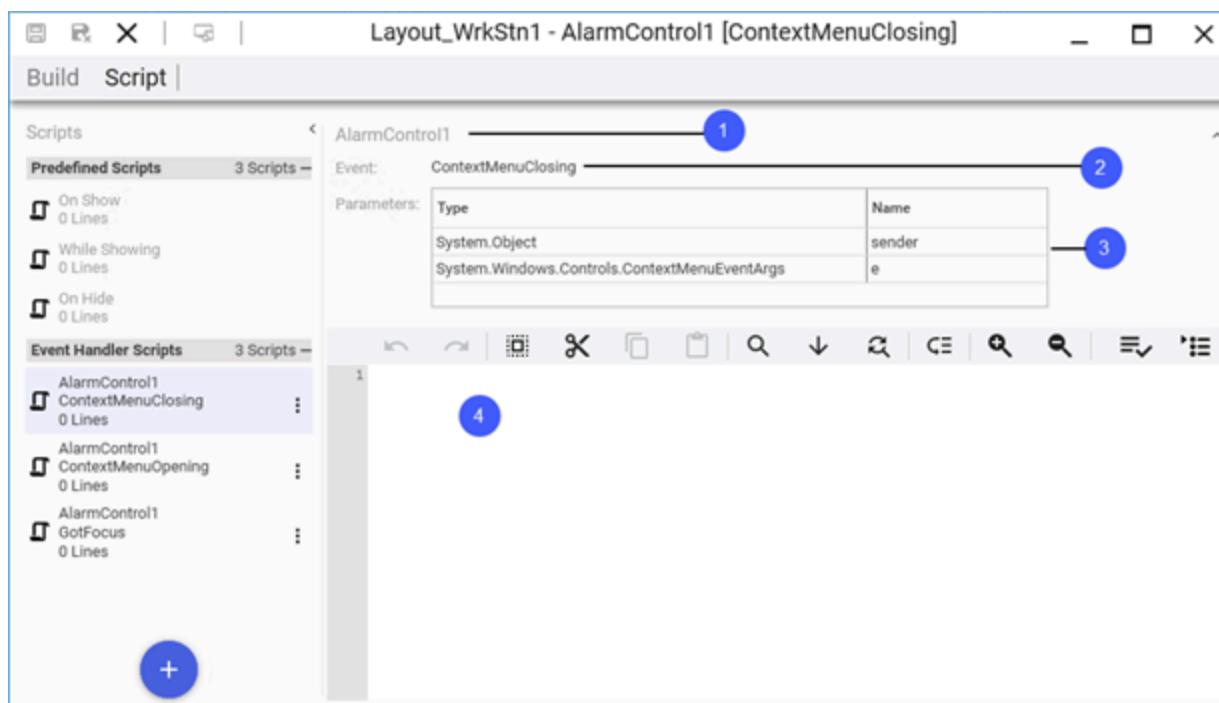
6. The new Event Handler script that you just added is listed alphabetically, by name of app and event, under **Event Handler Scripts** in the left column of the editor.

The first line shows the name of the app, the second line shows the event, and the third line shows the number of lines in the script.



Configure an event handler script

When you add an Event Handler script and select the script name from the list of scripts on the **Script** tab, the app name, event, and parameters associated with the event are shown in a grid at the top of the page, above the script editing area.



1	App/Control Name
2	Event Name
3	Parameter Grid
4	Script Editing Area

In the parameter grid, you can change the order in which the parameters are shown by clicking on either of the column headers. Parameters can be shown in ascending or descending order by name or type.

All public events, methods, and .NET dependency properties are exposed for apps that are compiled against the current version of the SDK. If an app was compiled against a previously released version of the SDK, only .NET dependency properties are exposed.

When you create an Event Handler script, the event name appears in bold in the list of available events.

To configure an event handler script

1. To change a parameter name, select its name in the parameter grid and enter a new name.
2. Enter your script in the main edit box. You can directly access properties and methods that belong to the app/control, without using the MyContent namespace. Autocomplete helps by displaying valid items, based on the initial characters you enter.
 - You can validate your script for syntax errors prior to saving by pressing the **Validate Script** button.
 - To clear all data entered for the script, including the expression, press the **Clear Script** button. There is no undo for this action.
3. To save the script, select the **Save** or **Save and Close** button.

4. The script is validated automatically. If there are warnings or errors generated, a dialog listing the errors is displayed. You have the following options:

- Select **Yes** to save the script and preserve the errors.
- Select **No** to discard the changes you made to the script. If you selected **Save and Close**, the Layout editor closes and you will lose any changes you made to the script. If you selected **Save**, the Layout Editor remains open and you can resume editing the script.
- Select **Cancel** to resume editing the script. The Layout Editor remains open and you can resume editing the script.

Event handler script limitations

The events for existing custom apps that were compiled against an older version of the AVEVA OMI SDK are filtered, and only .NET dependency properties will be exposed for event scripting. This ensures that public properties that may not have been intended to be used for scripting cannot be used this way.

Do not create event handlers for events that occur during, or shortly after, the creation of the app instance or short. For example, do not create an event handler for the "Initialized" event.

Do not create event handlers for touch events. These will not be executed at runtime.

Layout event handler parameters

Apps based on Windows Presentation Foundation (WPF) or .NET Controls may not provide the related event parameters in autocomplete as you are entering an event handler script in the Layout Script Editor. Typically, this occurs for events, such as mouseover, doubleclick, etc., and other events related to WPF types.

To enable autocomplete for event parameters within your event handler script, import the corresponding DLL where the event parameter type is defined. Use the System Platform IDE import function to import the DLL as a script library.

Layout execution order

In general, graphic scripts run before layout scripts run. This is to ensure that any references within the layout script will exist when a layout script calls the reference. The exception to this rule is a layout On Hide script, which, when triggered by the layout closing, will run before any graphic scripts that are still waiting to execute. Layout scripts thus have the following precedence with regards to graphic scripts:

- The On Hide layout script has precedence over graphic scripts.
- Graphic scripts have precedence over the On Show layout script.
- Named layout scripts can run in any order (not necessarily in the listed order), and may be interspersed with graphic scripts.

Any named script that is triggered by the Data Change trigger type runs the first time when the reference is subscribed to. This behavior is different than the Data Change trigger behavior of Application Server scripts and can take considerable time in intermittent networks.

Data binding in a layout

As you add apps and graphics to a layout, the exposed properties of the apps and the public properties of graphics are added to the MyContent Namespace. For more information, see [About the MyContent namespace](#). You can view and edit the property values of a selected app or graphic you select the **Properties** grid of the **Layout Editor**. The properties can be used in runtime, both directly by setting references and through scripting.

A unique content name is generated and displayed in the **Layout Editor** at the time you drag and drop an app or graphic onto a layout pane. The generated name uses the original graphic or app name and appends a number. For example, if you drop a graphic named "MyGraphic" on the pane, the unique generated name will be "MyGraphic1" for the first usage of the graphic; if you drop the same graphic on another pane the generated name will be "MyGraphic2." The syntax for referencing a custom property in MyGraphic1 is:

```
MyContent.Mygraphic1.CustomPropertyName
```

The autocomplete function of the Layout Editor works with the MyContent namespace and presents a dropdown list of apps, graphics, custom properties, and exposed app properties contained within the namespace. The items in dropdown list are filtered by context, so only valid items are listed in the dropdown.

In addition to exposed app properties, any WPF dependency properties contained within apps are also exposed, and you can bind to them as well.

If you attempt to bind to a property that does not exist, an error is displayed in the **Layout Editor**. If you do not correct the error before saving and/or closing the layout, a warning is displayed that the reference could not be resolved, and if you save the layout, it will be shown to be in a warning state in the **Visualization folder**.

About the MyContent namespace

The MyContent namespace lets you create references in scripts to tie together the content added to a single layout. The layout content can consist of graphics, apps, other layouts, and controls. The MyContent namespace encompasses everything the layout contains, including:

- Public custom properties of graphics associated with the layout
- Exposed properties of apps contained in the layout
- WPF dependency properties of apps contained in the layout

The exposed properties of each app are listed in the AppManifest.xml file for the app.

Note: Event-handler scripts, unlike named and built-in scripts, do not require the use of the MyContent namespace when scripting app or control properties and methods. To script a property or method, start by entering the name of the control instead of "MyContent." The autocomplete function will allow you to see the properties and methods of the control as you type.

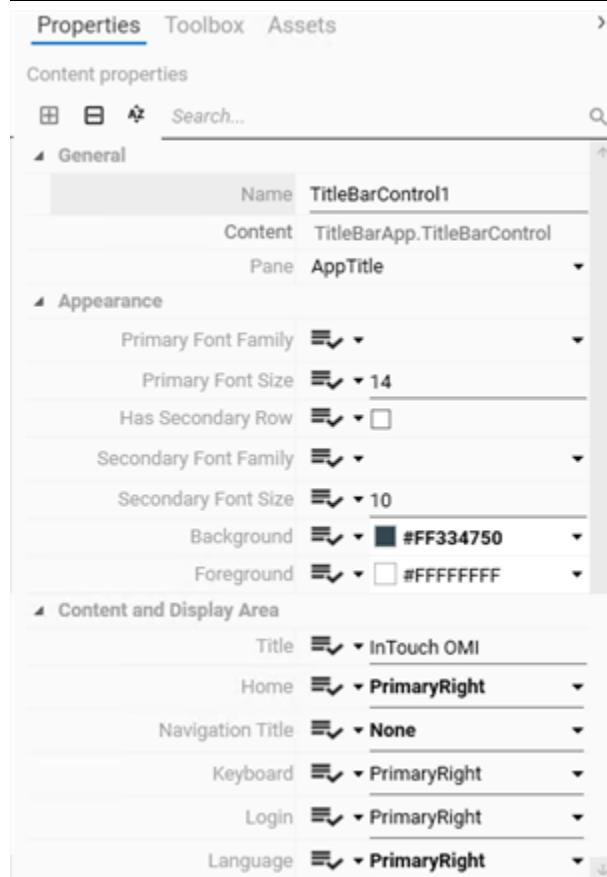
Properties designated as private are not accessible in the MyContent namespace. Each layout has its own MyContent namespace and you cannot create references to properties in another layouts via this namespace. If you need to reference properties in a different layout, use the ViewApp namespace and create attributes as needed for the references. See [About ViewApp namespaces](#) for more information.

Note: MyContent Namespace properties cannot be referenced from an object script. References from object scripts will validate successfully within the script editor, but a configuration error message will be displayed when you save the object. Objects that contain a script that references a MyContent Namespace property will remain in a warning state and the reference will not resolve.

Add property binding

Public properties for both **AVEVA OMI apps** and Industrial Graphics that contain custom properties are visible and configurable in the Layout Editor. Configuration consists of overriding default settings such as enabling or disabling a property, setting the binding mode of a property, or changing a string value. You can view the properties of an app or graphic by selecting the pane that contains the app or graphic and then selecting the **Properties** grid. Private properties are not exposed and not configurable through the Layout Editor.

Note: Although the Industrial Graphic Editor enables you to view graphic custom properties that are set to private, the properties cannot be overridden, either in the Industrial Graphic Editor or Layout Editor, unless you first set the property to public in the Industrial Graphic Editor. You cannot view graphic custom properties or app properties set to private from the Layout Editor. If you want to override an app property that is set to private, you must change it programmatically. There is no graphical interface for changing app properties from private to public.



App property binding can be set to static, dynamic/read-only, dynamic/write-only, or dynamic/read-write, as follows:

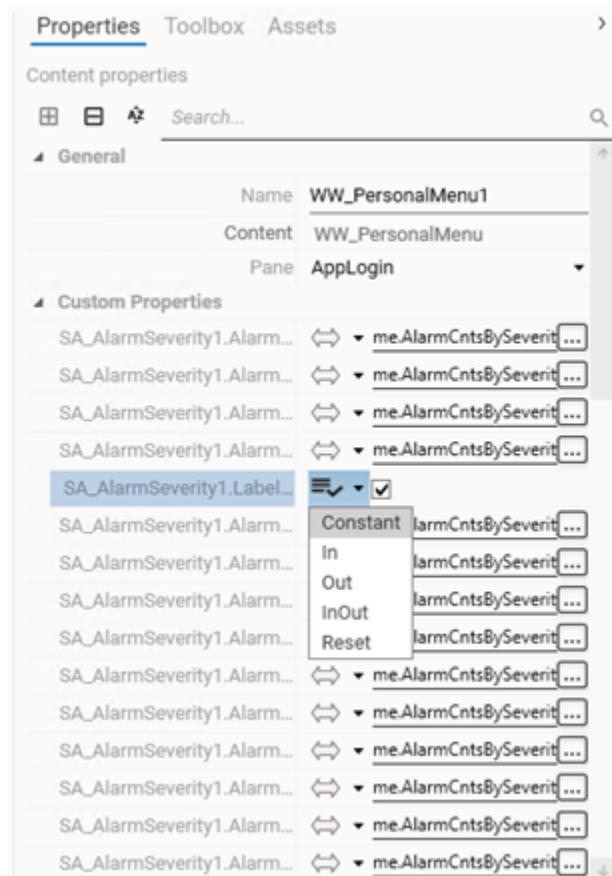
Constant	Static binding to the specified property value.
In	Dynamic property value with read only binding by the control to a reference.
Out	Dynamic property value with write only binding by the control to a reference.

InOut	Dynamic property value with read/write binding by the control to a reference.
Reset	Reset to the default property value.

When you override an app custom property, a "revert" symbol  appears to the right of the property. This signifies that the default value of the property has been overridden. You can revert to the default value of the property by clicking on or touching the revert symbol. The revert symbol does not appear when you override other app properties, such as general or appearance properties.

Set custom property binding type

Public app properties are visible and configurable in the Layout Editor. Similarly, graphic custom properties that are public are also visible and configurable in the Layout Editor. Configuration consists of overriding default settings such as enabling or disabling a property, setting the binding mode of a property, or changing a string value. You can view the custom properties by selecting an embedded graphic and then selecting the **Properties** grid on the Layout editor.



Custom property binding can be set to static, dynamic/read-only, dynamic/write-only, or dynamic/read-write, as follows:

Constant	Static binding to the specified property value.
In	Dynamic property value with read only binding by the

	control to a reference.
Out	Dynamic property value with write only binding by the control to a reference.
InOut	Dynamic property value with read/write binding by the control to a reference.
Reset	Reset to the default property value.

Use a layout as content

You can link a layout to an asset or template, in the same way that you link a graphic. Similarly, you can link layouts to an object wizard. See [Link to shared content in the visualization folder](#) for more information.

OwningObject Property

When you embed a graphic in another graphic, or add a graphic to a layout, the OwningObject property is exposed. The OwningObject property references the object that hosts the graphic or layout, and can be modified at runtime or via a bound value to the property.

If the OwningObject property of the layout is modified, all pre-existing subscribed references are removed and the layout is reloaded in its original state. If the layout contains empty panes set to use AutoFill, the new OwningObject property is used to build the relative references to the content.

AutoFill Limitations

In a multi-galaxy environment, AutoFill is unable to fill the panes if the owning object is set to an object in the remote galaxy.

Multi-Content Pane Limitations

A layout should not be added to multi-content **stacked** panes because navigation buttons will not be visible. If a layout is added to a multi-content **tabbed** pane, navigation is possible by using the tabs. See [About ViewApp navigation](#) for more information.

Effects of autofill on a layout linked to an asset

When a Layout linked to an asset is added to a pane in another layout, any empty panes in the linked Layout that have the "Use for Autofill" flag enabled will attempt to fill the empty panes with content from the linked asset, using the "Current Only" autofill setting. For more information, see [About auto-fill navigation](#).

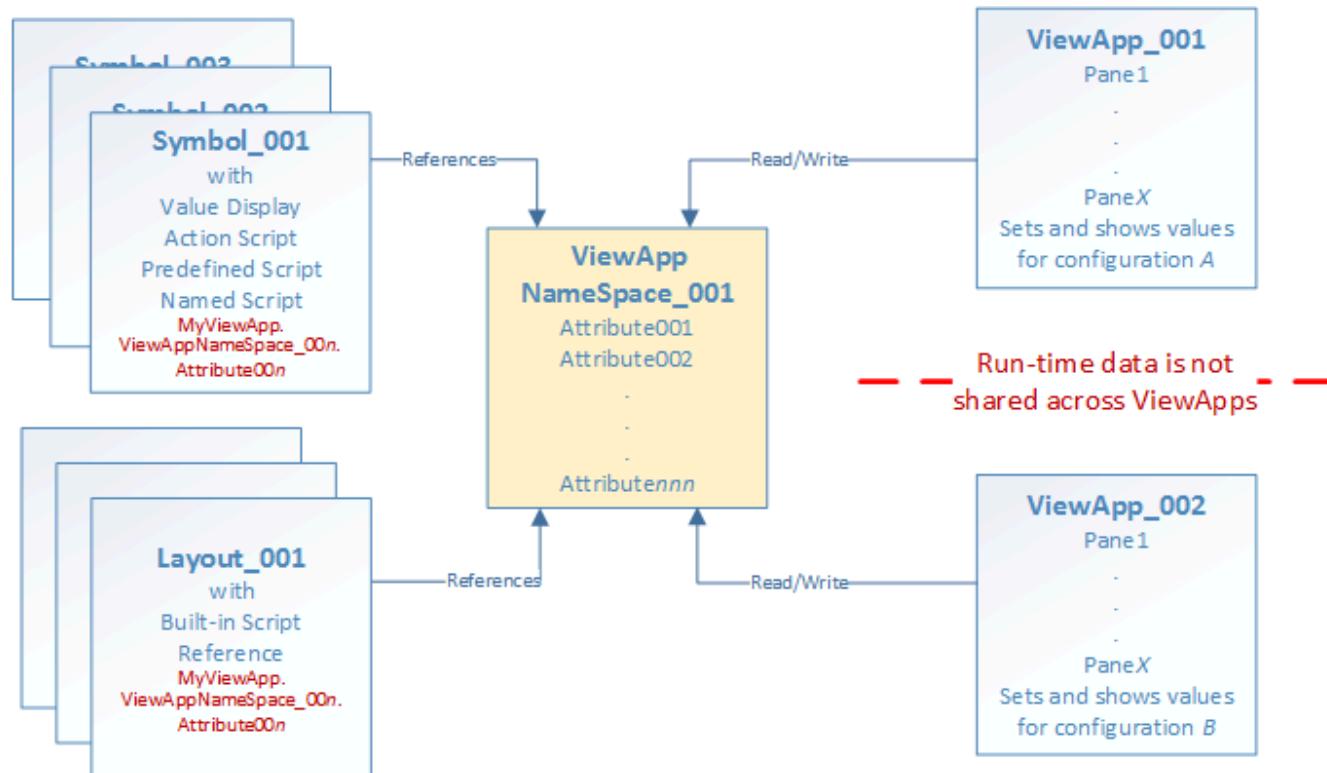
About ViewApp namespaces

ViewApp Namespaces are repositories for custom ViewApp attributes. A ViewApp Namespace can be thought of

as a tag bucket, and the ViewApp Namespace attributes as individual tags. ViewApp Namespace attributes are functionally similar to graphic custom properties, but are not bound to individual graphics. To invoke autocomplete for ViewApp namespaces and attributes when creating a graphic or layout script in the Script Editor, enter the keyword "MyViewApp." For more information, see [About custom properties for graphics](#).

ViewApp Namespaces are owned by the Galaxy in which they are created. You can reuse ViewApp Namespaces across multiple Viewapps, but while configuration data is shared across all ViewApps, runtime values can only be shared within a single ViewApp. Each ViewApp maintains its own copy of the ViewApp Namespace data. ViewApp Namespace attributes can be used in graphic scripts and references, animation scripts, and layout scripts and references. When you create references to ViewApp attributes in graphic or layout scripts and animations, attribute values can be shared across different panes of a running AVEVA OMI ViewApp. The ViewApp Namespace attributes are referenced much like static ViewApp attributes for alarms, navigation, language, etc. For more information, see "My ViewApp Attributes" in the *AVEVA OMI SDK Help* for more information about the static ViewApp attributes.

Note: ViewApp Namespace attributes cannot be referenced from an object script. References from object scripts will validate successfully within the script editor, but a configuration error message will be displayed when you save the object. Objects that contain a script that references a ViewApp Namespace attribute will remain in a warning state and the reference will not resolve.



ViewApp Namespace Editor

The ViewApp Namespace Editor lets you perform the following functions:

- Add, duplicate, and remove ViewApp attributes
- Rename ViewApp attributes
- Set ViewApp attribute data types

- Configure initial ViewApp attribute values
- Set ViewApp attributes as read-only and/or retentive
- Add descriptions to ViewApp attributes

ViewApp Namespace Elements

Each ViewApp Namespace attribute consists of the following elements:

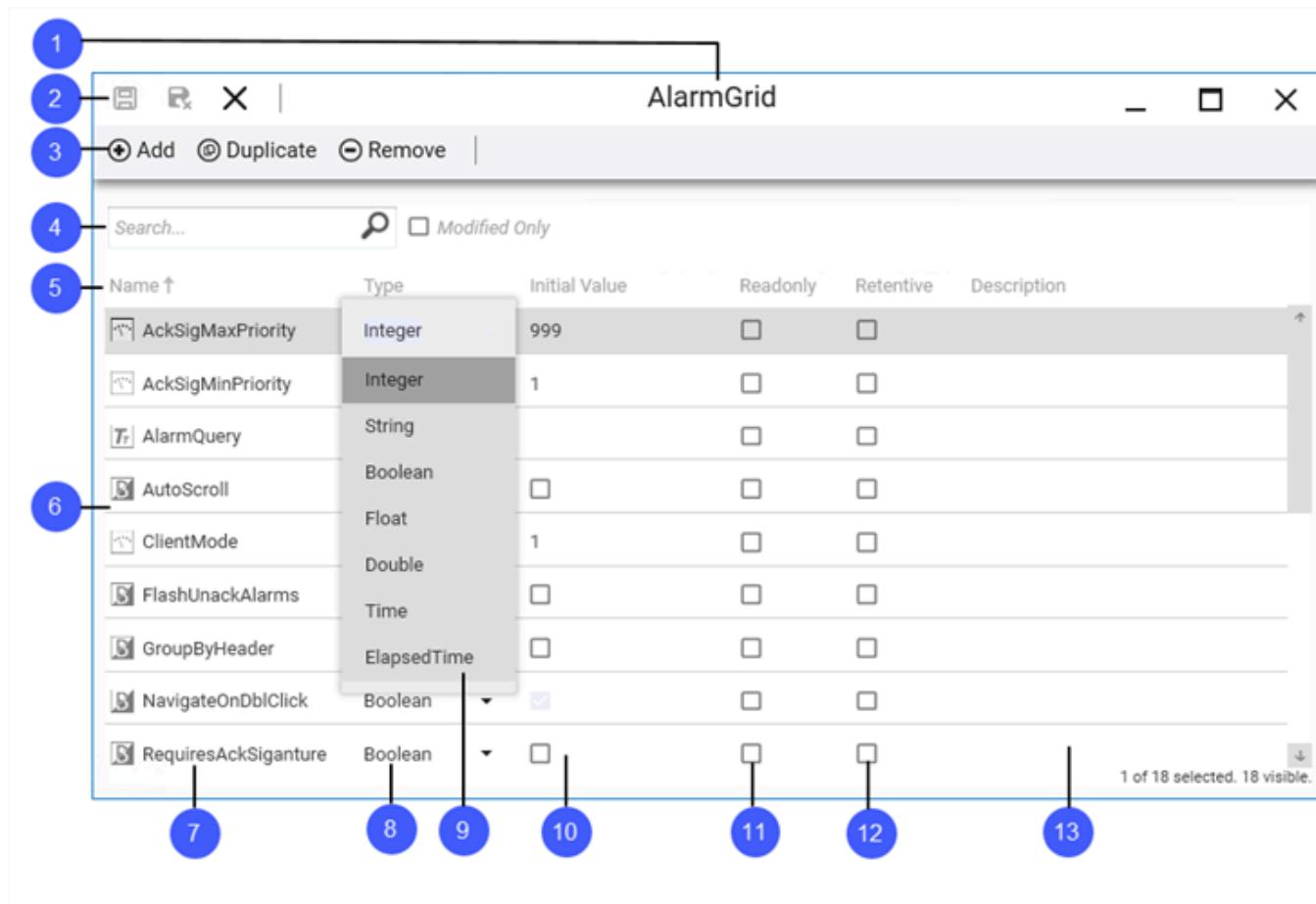
- Name: this is a string and must match galaxy naming rules
- Description: the description is shown when browsing for attributes in the Galaxy Browser
- Data type: supported data types and default values are:
 - Boolean – false (unchecked)
 - Double – 0.0
 - Elapsed time – 00:00:00.0000000
 - Float – 0.0
 - Integer – 0
 - String – empty string ("")
 - Time – current time, in the format 2/18/2018 11:36:14:385 AM

Note: Wherever possible, the initial value of the attribute persists when the attribute type is changed. The most notable exception to this behavior occurs when changing an attribute type from Boolean to String. In this case, the initial value is set to an empty string.

- Read-only option: when selected, the value will not change at runtime. Default value is unchecked (false).
- Initial value or expression: expressions can be used as values
- Retentive value: default value is unchecked (false). The last runtime value of a retentive attribute is saved during a clean shut-down of the ViewApp, and will be restored the next time the ViewApp is launched. An attribute cannot be both read-only and retentive.

About the ViewApp namespace editor

The figure below shows the various components of the **ViewApp Namespace Editor**. Use the ViewApp Namespace Editor to create and configure custom attributes that can be used by all ViewApps in the Galaxy.



1	ViewApp Namespace Name	Name assigned to the ViewApp Namespace. An asterisk (*) after the name indicates that the ViewApp Namespace has unsaved changes.
2	Menu Bar	Commands to save, save and close, and close the ViewApp Namespace in the editor.
3	Attribute Command Bar	Commands to add, duplicate, or remove attributes. You can use the Ctrl or Shift key to select multiple attributes for duplication or removal.
4	Search Box	Enter text to filter the attribute list. The text can appear in any part of the attribute name. If the "Modified Only" checkbox has been selected, only attributes that have changed since the last save are shown.

5	Active Sort Column	An up or down arrow next to the column name indicates that attributes are currently sorted by that column and the sort direction.
6	Attribute List	The attributes contained in the ViewApp Namespace are displayed in the attribute list. Attributes with unsaved changes are shown in bold type.
7	Name Column	Names of the attributes in the ViewApp Namespace. Unsaved name changes can be reverted by pressing the Esc key, provided that you have not navigated away from the changed name field.
8	Data Type Column	Data types of the attributes in the ViewApp Namespace. The data type is also indicated by an icon that precedes the attribute name.
9	Data Type Drop Down	Assign a data type from the drop down to configure the selected attribute.
10	Initial Value	Initial value setting for the attribute that is shown at runtime. Unsaved value changes can be reverted by pressing the Esc key, provided that you have not navigated away from the changed value field.
11	Read-Only CheckBox	Select the box to designate an attribute as read-only. Default value is false (Read/Write). An attribute cannot be read-only and retentive.
12	Retentive CheckBox	Select the box to designate an attribute as retentive. Default value is false (non-retentive). An attribute cannot be retentive and read-only.
13	Attribute Description	Add a description that will appear as a tooltip.

Create a ViewApp namespace

Video Tutorial: Create a ViewApp namespace in AVEVA OMI

https://player.vimeo.com/video/992306078?badge=0&autoplay=0&player_id=0&app_id=58479

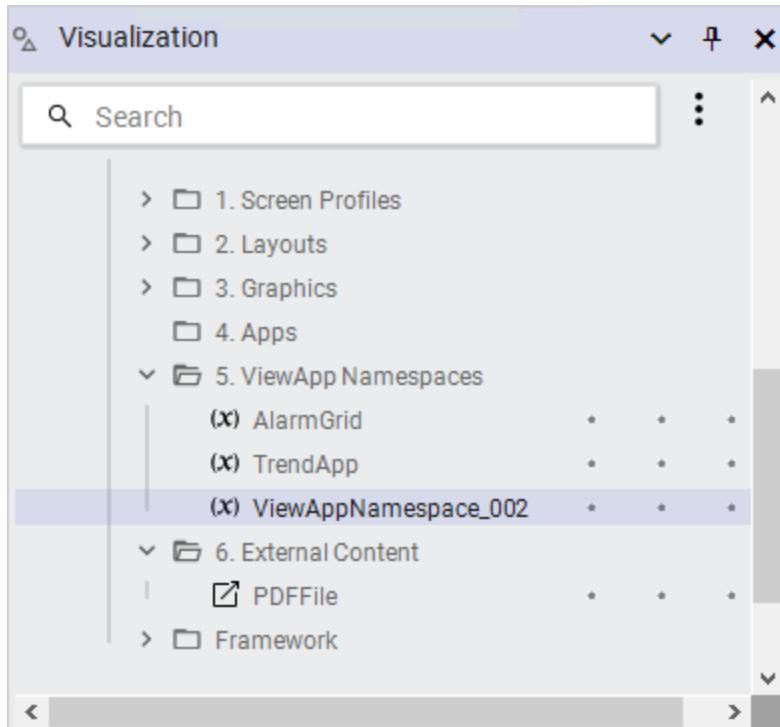
When you first create a new Galaxy, the Visualization folder does not contain a default ViewApp Namespace. To create a ViewApp Namespace, you can:

- On the **Home** ribbon, select **Namespace** in the **Create** area.
- Use the keyboard shortcut **Ctrl + Shift + V**.
- Right-click in the **Visualization folder** to bring up the shortcut menu. Then, select **New > ViewApp Namespace**.

Any of these actions creates a new ViewApp Namespace in the Visualization folder with the default name **ViewAppNameSpace_001**. After you create the ViewApp Namespace, you can:

- Rename, duplicate, delete, check out, or export the NameSpace.
- Open the ViewApp Namespace in the editor and add or edit attributes. After creating ViewApp Namespace attributes, you can reference them from graphic and layout scripts.

Note: ViewApp Namespace attributes cannot be referenced from an object script. References from object scripts will validate successfully within the script editor, but a configuration error message will be displayed when you save the object. Objects that contain a script that references a ViewApp Namespace attribute will remain in a warning state and the reference will not resolve.



Add and configure custom ViewApp namespace attributes

Video Tutorial: Configure A View App Namespace

https://player.vimeo.com/video/992306078?badge=0&autoplay=0&player_id=0&app_id=58479

This topic describes the general steps to add and configure custom ViewApp Namespace attributes into a ViewApp using a simple example.

The **ViewApp Namespace Editor** includes menu bar commands to **Add**, **Duplicate**, and **Remove** custom attributes.

Add	Adds a custom attribute to the ViewApp Namespace. The default name for the first attribute added is Attribute001. The number within the attribute name is incremented by one each time you add a new attribute. If you change the name of an attribute, the next added attribute uses the new name and increments the number contained within the name by one (the number can appear anywhere within the name). If the new name does not contain a number, "001" is appended to the name of the duplicated integer.
Duplicate	Duplicates the selected attribute and all of its properties. The only exception is that the attribute number contained in the name is incremented by one. If the selected attribute does contain a number, "001" is appended to the name of the duplicated attribute. You can select multiple attributes using the Shift or Ctrl key, and duplicate all selected attributes in a single operation.
Remove	Removes the selected attribute. You can select multiple attributes using the Shift or Ctrl key, and remove all selected attributes in a single operation.

The following procedure describes how to use these commands to create and configure ViewApp Namespace attributes.

To add custom attributes to a ViewApp Namespace

1. In the **Visualization folder**, open a ViewApp Namespace for editing.
2. Select **Add** from the menu bar of the **ViewApp Namespace Editor** to add a custom attribute.
3. Rename the attribute if necessary. To revert a pending name change, press **Esc**.
4. Configure the properties of the attribute.
 - a. Select **Type** from the dropdown menu and then choose the type of attribute you are adding: Integer, String, Boolean, Float, Double, Time, or Elapsed Time.
 - b. Set the initial value.
 - a. To revert a pending change to an initial value (other than the initial value of a Boolean), press **Esc**.
 - b. For Booleans, a checkbox sets the value. The default is unchecked for false.

Note: Wherever possible, the initial value of the attribute persists when the attribute type is changed. The most notable exception to this behavior occurs when changing an attribute type from Boolean to

String. In this case, the initial value is set to an empty string.

- c. Set whether the attribute is read-only.
 - d. Set whether the attribute is retentive. See [About the attributes page](#) for more information.
-

Note: An attribute cannot be set to both retentive and read-only. Since the value of a read-only attribute cannot be changed at runtime, there is no purpose in setting the value to persist across restarts.

- e. Enter a description for the attribute.

To duplicate custom attributes in a ViewApp Namespace

1. Open the ViewApp Namespace in the **ViewApp Namespace Editor**.
2. Select one or more attributes to duplicate. Use the **Shift** or **Ctrl** key to select multiple attributes.
3. Select **Duplicate** from the menu bar of the **ViewApp Namespace Editor**. This will create copies of the selected attributes. Duplicated attributes will have the same root names, types, initial values, read-only/retentive selections, and descriptions as the originals.
4. Rename the attributes as needed.
5. Configure attribute properties as needed. To revert a pending change to an attribute name or initial value (other than the initial value of a Boolean), press **Esc**.

Export a ViewApp namespace

ViewApp NameSpaces can be exported as standalone, reusable objects for use in other Galaxies. For example, you could create a ViewApp NameSpace that is broadly applicable across a variety of Galaxies for multiple locations. Once the ViewApp NameSpace has been imported into another Galaxy, it can be customized as needed.

A ViewApp NameSpace will be bundled automatically with other objects you are exporting, if one of the linked layouts or graphics you are exporting contains a reference to the ViewApp NameSpace. When you select a ViewApp for export, a dialog will ask if you want to include ViewApp NameSpaces with the export (along with navigation hierarchy assets and automation objects that host graphics). If you select the checkbox to include these items, then all ViewApp NameSpaces will be bundled with the exported ViewApp and its associated navigation hierarchy assets and automation objects.

Use the same method to export ViewApp NameSpaces that you use to export other objects. See [Export objects](#) for additional information.

Import a ViewApp namespace

To reuse one or more ViewApp NameSpaces that were previously created in another Galaxy, you can export the ViewApp NameSpaces you want to use into an .aaPKF (or .aaPDF file, if you need to save the existing configuration data). Then, import the file into the Visualization folder of the new Galaxy. See [Import objects](#) for additional information.

Once you have imported a ViewApp NameSpace into the Visualization folder, you can open and configure it with the **ViewApp Namespace Editor**.

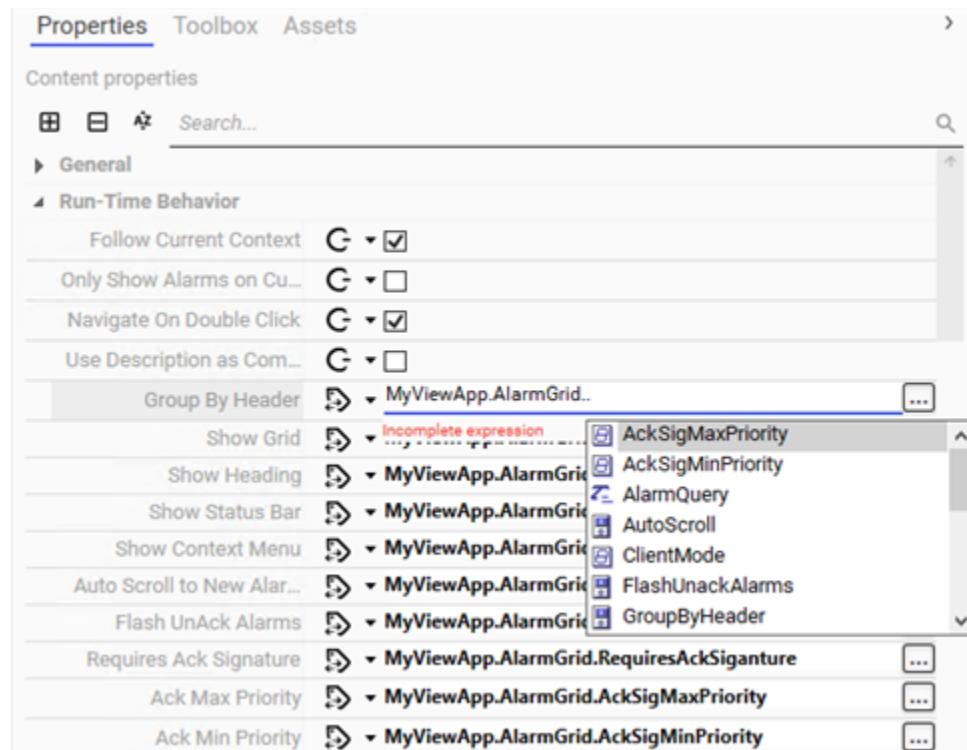
Create references to ViewApp namespace attributes

In order for a ViewApp NameSpace to have value, you must create references to its attributes. You can create references to ViewApp NameSpace attributes from:

- Graphic scripts
- Graphic animations
- Layout scripts

Note: ViewApp Namespace attributes cannot be referenced from an object script. References from object scripts will validate successfully within the script editor, but a configuration error message will be displayed when you save the object. Objects that contain a script that references a ViewApp Namespace attribute will remain in a warning state and the reference will not resolve.

References can also be configured on application properties that support data binding. This is supported in the **Properties Grid** of the **Layout** and **ViewApp** editors, as shown below. See [Configurable Properties in Archestra Apps](#) for additional information.



Each ViewApp NameSpace attribute reference must be prefixed with the reserved system name "MyViewApp." For example, to reference "Attribute001" in "ViewAppNameSpace_001," the reference would be:

MyViewApp.ViewAppNameSpace_001.Attribute001

Autocomplete will display items in the autocomplete list box to help you as you begin typing references in the script editor or animation editor. Just double-click the item you want from the list box (for example, "MyViewApp") or select the item and press the **Enter** key. Each part of the attribute name must use a dot (period) as a separator.

About retentive attributes

Retentive attributes for ViewApp NameSpaces differ in how they behave from other Application Server objects and from InTouch retentive tags. The table below compares the runtime behavior of retentive attributes/tags for:

- ViewApp NameSpace retentive attributes
- Application Server retentive attributes. This includes retentive attributes for application and system objects such as:
 - User Defined objects
 - Analog Device/Discrete Device objects
 - Area objects
 - WinPlatform objects
- InTouch retentive values

ViewApp NameSpace Retentive Attribute Values	Application Server Objects Retentive Attribute Values	InTouch Tag Retentive Values
Retentive attribute values are saved when the ViewApp is closed. The saved value is loaded to the ViewApp when it is reopened. If more than one instance of a ViewApp is open, values for the last ViewApp that is closed will overwrite all previous values for the other ViewApps.	Values for retentive attributes are saved to disk every 10 seconds. Retentive attribute values are preserved across deployments. When an object is undeployed, the last saved value is restored at the next deployment	Retentive tag values are retained when the application stops, and are loaded when the application starts again. WindowViewer uses the retained value as the initial tag value when it restarts, but does not write the value to the I/O device. I/O values are only updated after the I/O Server scans the device.

About alarms

You can create AVEVA OMI applications that generate alarms and events to provide the status of a running business process.

- Alarms warn about process conditions that can potentially cause problems. Typically, you set up an alarm to become active when a process value exceeds a defined limit. For example, you can set an alarm for a pump that warns when no fluid pressure is detected.

An alarm is an abnormal condition that requires immediate attention. An operator usually acknowledges an alarm. Application Server objects handle the real-time reporting of alarms and provides special clients for viewing them.

- Events represent normal system, application, or user occurrences that produce status messages. A typical event occurs when an operator logs on to an application at the beginning of a work shift. Application Server can detect events, store them as historical data, and report them to client applications.

Application Server objects include built-in event and alarming reporting capabilities. You must configure alarms for each object in the IDE to use the event and alarm functions.

Event definition

Events represent normal system, application, or user occurrences that produce status messages. A typical event occurs when an operator logs on to an application at the beginning of a work shift. Application Server can detect events, store them as historical data, and report them to client applications.

An event indicates a significant occurrence that is detected, reported, and saved as historical data. Events can be detected by any Application Server component including automation objects and the OCMC.

Events provide a means for any software component to log information about a system, application, or operator action. Application Server components use an event API to send event messages to Alarm/Event distributors and the Historian. Also, events are sent to client applications like AVEVA OMI to be shown in real-time trends or reports.

Types of events

Application Server creates several types of runtime event messages, which are saved as historical data.

- System events

Event messages are logged when a Platform, Engine, Area, DI Network or DI Device start or stop. Any system action that affects a large number of objects is logged as an event.

A System event message contains the following information:

- Event type, which is System.
 - Tagname, which is the name of the object generating the event.
 - Tag description, which is a short description of the object generating the event.
 - Area, which is the name of the area that contains the object generating the event.
 - Event description, which describes the system action and can be either Started or Stopped.
 - Timestamp, which is the current system time.
- Security-audit (operator change) events

Event messages are logged to the Alarms and Events subsystem when an operator logs on to or logs off from an application, and when a logon attempt fails (for example, due to an incorrect password). Also, security-audit events are logged when an operator changes actions by means of User sets.

A Security-audit event message contains the following information:

- Event type, which is operator change.
- Timestamp, which is the date and time when the operator change event occurred. The timestamp of the event is the current AppEngine scan time.
- Tagname, which is the name of object generating the event.
- Prim.attr, which is the reference string of the attribute being changed.
- Tag description, which is a short description of the object. For Secured and Verified writes, this will contain the following:
 - Type of write (Secured or Verified)
 - Comment from the user, if any
 - Reason description, if any

- FieldAttribute description, if the attribute is a FieldAttribute and has a description; otherwise, this is the Object description.
 - Area, which is the name of the area that contains the object generating the event.
 - UserEngine, which is the name of the view engine or other user engine requesting the operator change.
 - Operator1, which is the full name of the primary operator requesting a change. The full name is an attribute of the UserProfile.
 - Operator2, which is the full name of the secondary operator validating the change, if any.
 - Old value, which is the previous value of an attribute.
 - New Value, which is the new value of an attribute.
- Application (or process) related events

Application event messages are generated by application objects in response to process actions. For example, application event messages are created when a process pump starts or stops.

An Application event contains the following information:

- Event type, which is data change.
- Timestamp, which is the date and time when the application event occurred. The timestamp assigned to the event is the timestamp of the attribute associated with the event, if available. Otherwise, the event timestamp is the current AppEngine scan time.
- Tagname, which is the name of the object generating the event.
- Prim.attr, which is the reference string of the attribute being changed.
- Tag description, which is a short description of the object.
- Area, which is the name of the area that contains the object generating the event.
- Old value, which is the previous value of an attribute.
- New Value, which is the new value of an attribute.

Alarm definition

Alarms warn about process conditions that can potentially cause problems. Typically, you set up an alarm to become active when a process value exceeds a defined limit. For example, you can set an alarm for a pump that warns when no fluid pressure is detected.

An alarm is an abnormal condition that requires immediate attention. An operator usually acknowledges an alarm. The System Platform framework handles the real-time reporting of alarms and provides special clients for viewing them.

Application and system objects can detect and generate an alarm. To detect an alarm, a system or application object sets a Boolean Attribute flag to indicate whether the alarm condition of the object is currently true or false.

To report an alarm, the object must contain an alarm feature. The alarm feature makes a reference to the object's Boolean flag to determine whether the alarm condition is true. It then combines this information with the current alarm mode to determine whether to report a this as an active or inactive alarm state. An alarm feature is dedicated to reporting a single alarm condition's state. The alarm feature sends alarm notification messages to System Platform alarm and event distributors.

Every alarm notification includes a set of fields containing data that describes the alarm. Some alarm notification data is saved as historical data. The following list describes all fields sent with an alarm notification.

- TagName: The name of the object generating the alarm. Saved as historical data.
- Name: The name of the alarm. Saved as historical data.
- InAlarm: A Boolean value that indicates whether the object's alarm state is currently active or inactive. Saved as historical data.
- Quality: The current quality of the data upon which the alarm is based. Saved as historical data.
- OnTimeStamp: The time that the attribute value transitioned to an alarm state. The value timestamp of the attribute is used, if available. If not, the timestamp is the AppEngine scan time. Saved as historical data.
- OffTimeStamp: Time that the alarmed attribute value returns to normal. The value timestamp of the attribute is used, if available. If not, the OffTimeStamp is the AppEngine scan time. If an active alarm is disabled, it forces a return to normal and the timestamp is the current time. Saved as historical data.
- Category: An integer between 1 and 14 that identifies the type of alarm and source of the alarm. These values are associated with Internationalized category labels.

Alarm category labels can be localized to other languages. Application Server uses the default Galaxy language to retrieve these strings and send them to InTouch. The alarm category labels appear in InTouch and InTouch history as the default Galaxy language strings. Saved as historical data.

- Priority: An integer value from 1 to 999 indicating the severity of the alarm. An alarm priority of 1 is most urgent and 999 least urgent.
- TargetValueReference: An optional field that makes a reference to the target value of the alarm. Not saved as historical data.
- ActualValueReference: An optional field that makes a reference to the actual attribute value for the alarm condition. Not saved as historical data.
- TargetValue Snapshot: An optional field containing the target value of the attribute at the time when the alarm became active. Saved as historical data.
- ActualValueSnapshot: An optional field containing the actual value of the attribute at the time when the alarm became active. Saved as historical data.
- EngUnitsReference: The reference to the engineering units string for the condition. Saved as historical data.
- AcknowledgedFlag: Indicates whether the alarm is acknowledged or not. If this flag is FALSE, the alarm is still unacknowledged. Saved as historical event data.
- AcknowledgeTime: Indicates the time when the alarm was acknowledged if the AcknowledgedFlag is TRUE. Saved as historical data at the time of acknowledgement.
- AcknowledgeUserId: The string containing the name of the user who acknowledged the alarm. Saved as historical data at the time of acknowledgement.
- AlarmMode: Indicates if the alarm mode is Enabled, Silenced or Disabled. Saved as historical data at the time when the alarm mode changes.
- Message text describing the alarm, which can be statically or dynamically constructed. The message typically contains the alarm description, the exceeded limit value, and possibly the variable value. For an alarm feature, this message is retrieved by means of a reference to a string/international string attribute in the object. The reference is setup at ObjectDesigner time for alarms. If none is specified, then the common feature short description attribute is utilized. This field is also provided in the alarm feature section and can be dynamically generated and scripted. Saved as historical data.
- Area: The name of the area that contains the object generating the alarm. Saved as historical data.

Types of Alarms

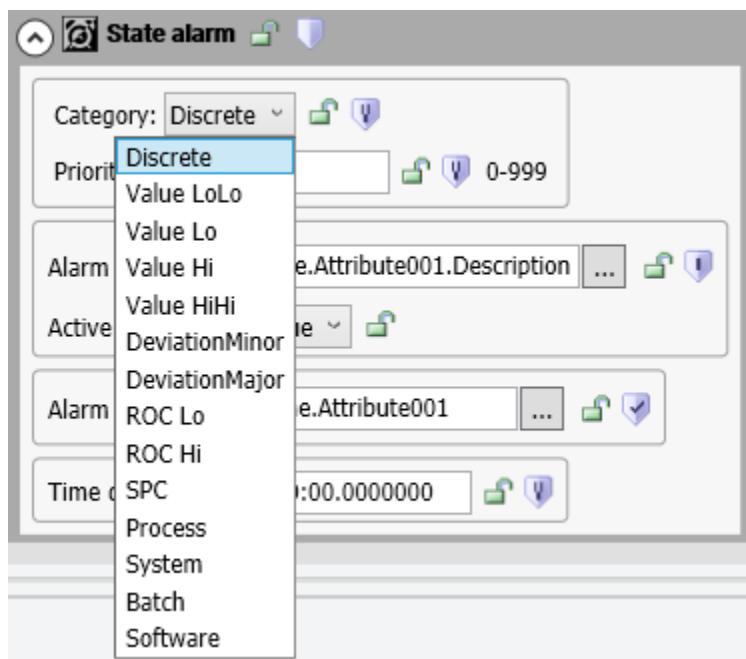
Application Server supports the following types of alarms:

- State alarms, which are also known as Boolean alarms. See [The State Alarm Feature](#) for more information.
- Limit alarms. See [The Limit Alarms Feature](#) for more information.
- Target deviation alarms. See [The Deviation Alarms Feature](#) for more information.
- Rate of change alarms. See [The Rate of Change \(ROC\) Alarms Feature](#) for more information.
- Bad value alarms. See [The Bad Value Alarms Feature](#) for more information.

The types of alarm that you can configure are based on the data type of the attribute's value.

Configure State Alarms

A state alarm set on an attribute of Boolean data type corresponds to a discrete tag with two possible states. When you create a state alarm, you configure whether the active alarm state corresponds to the TRUE or FALSE state of the attribute.



You can set an alarm category. Valid categories are:

Category	Description
Discrete	A discrete value event or alarm, such as a change of state.
Value LoLo	A continuous value is significantly below the acceptable range.
Value Lo	A continuous value is below or is approaching the low

	acceptable range.
Value Hi	A continuous value is above or approaching above the high acceptable range.
Value HiHi	A continuous value is significantly above the acceptable range.
DeviationMinor	A value has a minor deviation (plus or minus) from the target or setpoint.
DeviationMajor	A value has a major deviation (plus or minus) from the target or setpoint.
ROC Lo	The rate of change for a value is too slow.
ROC Hi	The rate of change for a value is too fast.
SPC	A value deviates from the SPC target/range.
Process	An alarm or event associated with the physical process/plant has occurred.
System	An alarm or event associated with the automation system has occurred.
Batch	An alarm or event associated with a batch process has occurred.
Software	An alarm or event associated with a software operation or logic (such as divide by zero in script) has occurred.

You can set an alarm message and Priority for a state alarm. The time deadband sets the length of time that an attribute value must continuously remain in an alarm or unalarmed state. The time deadband filters out rapid, transitory value spikes.

The timestamp when a state alarm becomes active or inactive is the most current timestamp of the corresponding input value. If there is no timestamp associated with the alarmed value, the AppEngine timestamp is used instead.

You can only configure an external alarm source for use with Historian Clients, not InTouch clients.

To get the alarm history records for an externally detected alarm (from within a PLC), create a reference to the source_processvariable property, as follows:

1. Create a boolean user-defined attribute.
2. Enable I/O and State Alarm features for the boolean.
3. Bind the IO feature for the boolean to a tag in the PLC.
4. Set the I/O to read the PLC alarm bit where the alarm is being detected in the PLC. Note that there is nothing in the boolean attribute definition to indicate it represents an alarm associated with another attribute (like a ".PV" attribute).

5. To establish the alarm association, associate the boolean (created in step 1) with another attribute. At runtime, this association will set the “source_processvariable” in the alarm history event. This allows clients of History Event and Alarms records to correlate externally detected alarms with the attributes against which they are being reported.

See the sample configuration, below, for additional details.

Sample external alarm source configuration

The screenshot shows the configuration interface for an attribute named "Pump001.PV". The top section displays basic metadata: Name (Pump001.PV), Description (Process variable), Data type (Boolean), Writeability (User writeable), Initial value (False), and available features (I/O, History, State alarm, Bad value alarm, Statistics, Log change). The "State alarm" feature is selected. The bottom section provides detailed configuration for the state alarm, including options for Read, Read/Write, or Write, and a connection to a PLC tag ("Some_PLC_tag"). It also includes settings for the alarm message ("me.Pump001.PV.Description"), priority (500), active state (True), and time deadband (00:00:00.0000000).

Attribute Name: Pump001.PV

Type: Boolean

IO: Enabled

Points to a tag within the PLC.

State Alarm: Enabled

Alarm For Attribute: me.Pump001.PV

Attribute Name: Pump.PV.Hi

Type: Boolean:

IO: Enabled:

Points to an alarm tag in the PLC.

This tag will be true if there is a high alarm generated by Pump.PV.

State Alarm: Enabled

Alarm For Attribute: me.Pump001.PV

Attribute Name: Pump.PV.HiHi

Type: Boolean:

IO: Enabled:

Points to an alarm tag in the PLC.

This tag will be true if a high-high alarm is generated by Pump.PV.

State Alarm: Enabled

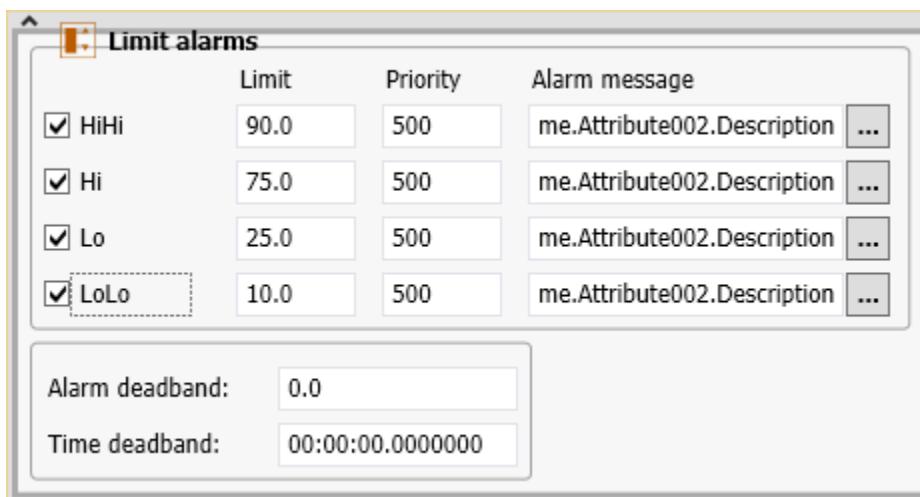
Alarm For Attribute: me.Pump001.PV

Configure Limit Alarms

A limit alarm compares the current value to one or more predetermined alarm limits within the attribute's full range of values. If the value exceeds a limit, an alarm occurs.

Run-time graphic showing limit alarm values for LoLo, Lo, Hi, and HiHi alarms

You can individually select and configure values and priorities for the LoLo, Lo, Hi, and HiHi alarm limits. You can set individual messages for each alarm limit.



You can also configure alarm and time deadbands for limit alarms. The alarm deadband is an absolute value, in engineering units, that the attribute value must exceed the configured HiHi, Hi, Lo, or LoLo limit before the alarm is triggered, or that the attribute value must revert to before the limit alarm resets to its inactive state.

For example:

- The HiHi alarm limit is set to 90.
- The alarm deadband is set to 5.

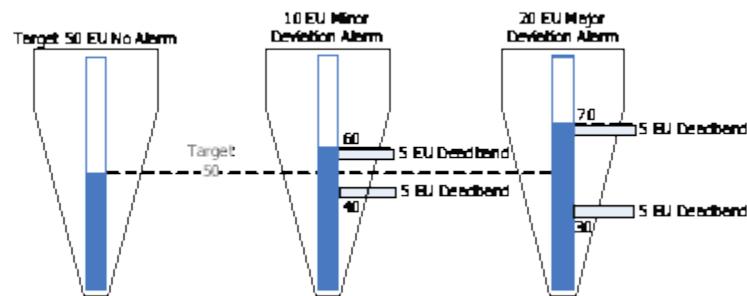
- The attribute value must exceed 95 (limit plus deadband value) to trigger the HiHi alarm.
- The attribute value must fall below 85 (limit minus deadband value) to reset the HiHi alarm to an inactive state.

The time deadband sets the length of time that an attribute value must continuously remain in an alarm or unalarmed condition. The process variable must remain above or below the indicated limit for at least the indicated deadband time before the application object updates the status of the alarm CONDITION Boolean. Then, standard Alarm feature logic determines whether to take that updated alarm condition and report changes to the alarm state or not.

The timestamp when a limit alarm becomes active or inactive is the most current timestamp of the corresponding input value. If there is no timestamp associated with the alarmed value, the AppEngine timestamp is used instead.

Configure Target Deviation Alarms

A target deviation alarm compares the current attribute value to a target Engineering Units value. Then, the absolute value of the difference is compared to one or more alarm deviation limits expressed in EngineeringUnits.



You can individually select and configure values and priorities for the minor deviation limit and the major deviation limit. You can set individual messages for the minor and major deviation alarm limits.

Deviation alarms			
	Tolerance	Priority	Alarm message
<input checked="" type="checkbox"/> Minor	10.0	500	me.Attribute002.Description <input type="button" value="..."/>
<input checked="" type="checkbox"/> Major	15.0	500	me.Attribute002.Description <input type="button" value="..."/>
Target: <input type="text" value="50.0"/> Deviation deadband: <input type="text" value="0.0"/> Settling period: <input type="text" value="00:00:30.0000000"/>			

The deviation alarm's settling period is the time allowed for the attribute value to reach an expected target value after a device starts. No alarm can occur during the settling period.

You can also configure a value for a deviation deadband, which is expressed in Engineering Units. The deadband range sets a threshold that an attribute value must change from a deviation limit to reset the alarm to the inactive state.

The timestamp when a deviation alarm becomes active or inactive is the most current timestamp of the

corresponding input value. If there is no timestamp associated with the alarmed value, the AppEngine timestamp is used instead.

Configure Rate of Change Alarms

A rate of change alarm identifies when an attribute value is changing too quickly over time. For example, you can set a rate of change alarm for a tank level that indicates when the pump inlet pressure increases too quickly.

Alarm limits are expressed in the Engineering Units of the attribute's value over an interval, which can be per second, minute, hour, or day.

Run time display showing changes in value over a one minute interval

Rate of change is the calculated slope, which is the absolute difference between the current and previous attribute values divided by a specified interval. When the slope (positive or negative) exceeds a specified value, a rate of change alarm occurs.

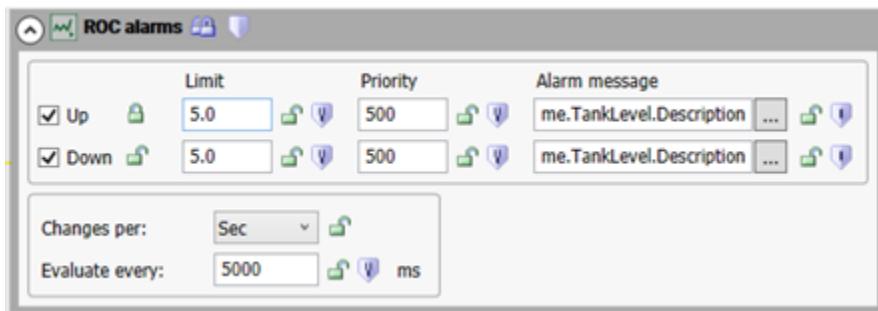
As described above (see [The Rate of Change \(ROC\) Alarms Feature](#)) the change limits, direction of change (up or down), time interval for the change, and the evaluation interval are configurable.

Example alarm configuration:

- An ROC alarm limit is set to issue an alarm if the volume increases more than 5 liters (up limit) per second ("Changes per" <unit of time>).
- The evaluation interval is set at 5000 ms (5 seconds).

Alarm condition based on the example alarm configuration:

- After 5 seconds (the evaluation interval), the attribute value has changed from 17 to 45 liters, resulting in an increase of 28 liters ($45-17=28$).
- The calculated ROC over the evaluation period is 5.6 liters per second ($28/5=5.6$). Since this is more than the configured ROC limit (5 liters per second), an alarm condition exists.



You can select and configure the value and priority for the upward and downward ROC limits. You can set individual messages to be shown when the upward or downward ROC limit is exceeded.

The timestamp when a rate of change alarm becomes active or inactive is the most current timestamp of the corresponding input value. If there is no timestamp associated with the alarmed value, the AppEngine timestamp is used instead.

Statistical alarms

A statistical alarm is one in which a statistic is calculated, based upon an attribute. If the statistic exceeds some

pre-set limit, the object flags the alarm condition as TRUE.

Set alarm state with object attributes

The Application Server alarm enable/disable mechanism includes four attributes to set an object alarm mode and report alarm status:

- [AlarmModeCmd attribute](#)
- [AlarmInhibit attribute](#)
- [AlarmMode attribute](#)
- [AlarmModeEnum attribute](#)

AlarmModeCmd attribute

AlarmModeCmd is a writable attribute that sets the current commanded alarm mode for the object. You set the AlarmModeCmd to enabled, silenced, or disabled with a script, user input, or from an attribute configured to read from an input source.

AlarmInhibit attribute

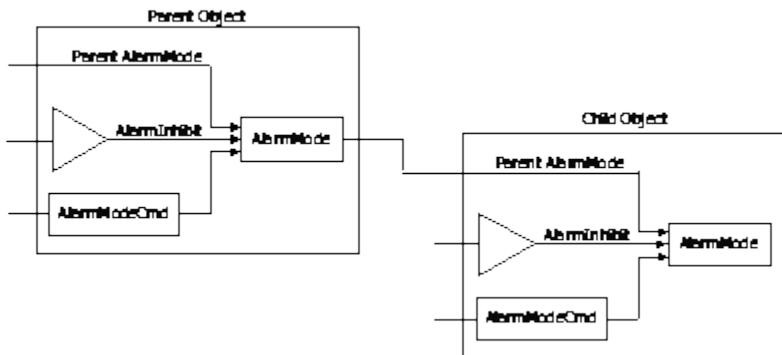
The AlarmInhibit attribute disables one or more alarms when set to TRUE. The value of the AlarmInhibit attribute is typically set by a script, manually by the user, or from an attribute configured with an input feature. If the AlarmInhibit attribute is set TRUE, all alarms of the object and of any contained objects are disabled.

When the AlarmInhibit attribute is set to FALSE, alarms are not inhibited and the object AlarmMode and parent object AlarmMode determine whether alarming is enabled, silenced, or disabled.

AlarmMode attribute

The AlarmMode is a calculated attribute that identifies the object alarm mode and is based upon the current values of an object's:

- [AlarmModeCmd attribute](#)
- [AlarmInhibit attribute](#)
- Parent object AlarmMode attribute



Application Server checks the AlarmModeCmd and AlarmInhibit attributes of an object and the AlarmMode status of the parent object. Application Server then updates the object's AlarmMode attribute to reflect the most restrictive setting.

All individual alarms use the object's AlarmMode status to determine whether they are enabled, silenced, or disabled.

AlarmModeEnum attribute

The _AlarmModeEnum attribute can be used by any object to obtain an enumeration of permissible settings for the AlarmModeCmd attribute.

Alarms and buffered data

An object with alarms enabled detects the alarm condition and sets an associated alarm condition attribute to true. If the object detects the alarm condition based on an attribute with buffered data (HasBuffer property set to true), it enables buffer support for the condition attribute and sets the condition attribute's Buffer property to a VTQ buffer of true/false values representing alarm conditions.

Alarm detection

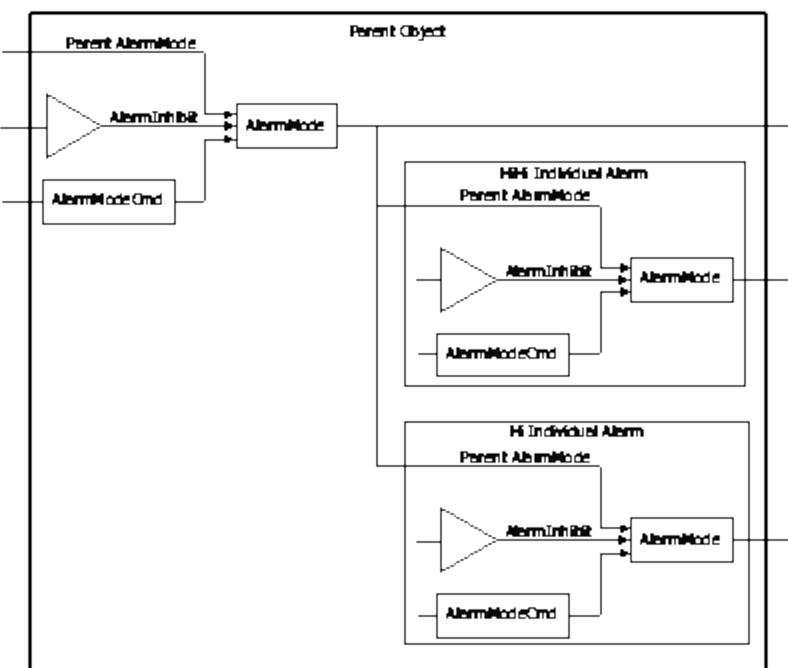
The Buffered Data feature will pass buffered alarm VTQs to the alarm subsystem in the same manner as with buffer data not enabled. When enabled, the Buffered Data feature can generate multiple alarm messages in the same scan.

For more information on alarm detection, see [Alarms and buffered data](#).

Set the alarm state for individual alarms

You can set individual alarms within an object for each type of alarm. For example, you can set alarms for each of the limits of a level alarm.

The following figure shows an object's individual alarms for the HiHi and Hi alarm limits.



The calculated AlarmMode attribute value of an individual alarm uses the same inputs as an object alarm. The parent AlarmMode attribute is from the object itself. As with object alarms, the individual alarm mode is set to the most restrictive input state. For example, if the object's AlarmMode state is disabled and the individual alarm's AlarmInhibit is FALSE, the individual alarm is disabled.

Each individual alarm is autonomous from other individual alarms in an object. The AlarmMode of an individual alarm is not propagated to other alarms. Unlike inhibit for the entire object, inhibit of an individual alarm does not affect the alarms of any contained objects. You can selectively enable, silence, or disable an individual alarm and not set other alarms to the same value within the object hierarchy.

Enable, silence, and disable alarms

Alarms can be enabled, disabled, or silenced while an application is running. An object's alarm state can be set at the Area level, at the container object level, or at the individual object. In addition, individual alarms within a single object can be enabled, silenced, or disabled.

- Enabled: All alarms for an object are reported to client applications and saved as historical data. The enabled state is less restrictive than the silenced or disabled alarm states.
- Silenced: All alarms for an object are detected and logged in the Historian alarm and event history database. Silenced alarms are not logged to the InTouch alarm database, and are not shown in alarm clients displaying current alarms or recent history alarms. The silenced alarm state is more restrictive than the enabled state, but less restrictive than the disabled state.
- Disabled: No alarms for the object are detected. The alarm is return-to-normal until the alarm is re-enabled. The disabled state is more restrictive than the silenced and enabled alarm states. A disabled alarm does not require acknowledgement.

When an alarm state changes from silenced to enabled, the following applies:

- Only the last occurrence of the alarm appears in alarm clients.

- Only the last occurrence of the alarm is logged into the alarm database.
- Only the last occurrence of the alarm is saved to history.

The following figure shows how the different alarm modes affect the different phases of an alarm. In the case of Alarm Suppressed, selected alarms can be filtered out of the InTouch AlarmViewer display by an operator, or programmatically by a script.

Alarm condition, alarm state, and alarm notification settings affect the alarm display at runtime

To ensure that alarmed attributes always have current data, the alarm feature always registers a reference to the alarmed attribute. This guarantees that Message Exchange never suspends updates for this attribute. Even if alarms are disabled for a particular attribute, the alarmed attribute cannot enter an Advanced Communication Management Suspended state.

The object hierarchy and alarm states determine the final alarm condition of an object.

- An Area object's alarm state determines the alarm state for all alarms of objects that belong to the Area.
Diagram showing that the alarm state of an area object determines the alarm state for all objects that belong to the area
- The most restrictive setting within an object hierarchy determines the object's alarm state.
The most restrictive setting within an object hierarchy determines the object's alarm state
- When an individual object alarm is silenced or disabled, it applies only to that alarm and not to other alarms belonging to the object.
A silenced or disabled alarm applies only to that alarm and not to other alarms belonging to the object
The alarms on any contained object are not affected. The disabled or silenced state of an individual alarm does not propagate downward through the object hierarchy to the alarms of any contained or assigned object.

Enable alarms

To enable an object's alarms, you must ensure that the AlarmModeCmd and AlarmInhibit attributes are enabled for the object, its container, and its area. An event, including the user's name, is generated indicating the object's alarms are enabled.

When object alarms are enabled, you can enable, silence, or disable an individual alarm.

Silence alarms

When object alarms are silenced, an individual alarm that is enabled or silenced is forced to be silenced. When object alarms are silenced, an individual alarm can be disabled.

Disable alarms

When object alarms are disabled, an individual alarm that is enabled or silenced is forced to be disabled.

When object alarms are enabled and an individual alarm is enabled or silenced, the individual alarm can be inhibited. This forces the individual alarm to be disabled.

When object alarms are silenced and an individual alarm is enabled or silenced, the individual alarm can be inhibited. This forces the individual alarm to be disabled.

When object alarms are inhibited, an individual alarm that is enabled or silenced is forced to be disabled.

Shelve alarms

You can shelve alarms to temporarily hide them from displays for a fixed period. Alarms continue to be historized, even when they are shelved.

Shelving typically is used to reduce alarm "noise", or to temporarily suppress alarms that might be triggered during system modifications or repairs, allowing you to focus on correcting other more urgent alarms.

Shelving is similar to silencing an alarm, but shelved alarms differ from silenced alarms in the following ways:

- Shelved alarms have a built-in associated time-out period. Shelved alarms are automatically unshelved when the configured shelving period expires. You can also manually unshelve alarms and return them to an active, displayed state.
- Alarm shelving must be enabled at an area level, but shelving applies only to individual alarms. You cannot shelve a hierarchy of alarms for an entire area or for an entire object. You cannot propagate alarm shelving through the Model View hierarchy.
- The system enforces role-based limitations on permission to shelve alarms, alarm severity levels that can be shelved, and the total number of alarms a user can shelve.

The system tracks who shelved the alarm, from what workstation, the reason for shelving the alarm, when shelving began, and when shelving will expire. Shelved alarms aggregate in similar fashion to silenced alarms. For information about alarm aggregation, see [Aggregated alarm information](#).

A set of seven attributes provide runtime alarm shelving information and control:

AlarmShelveCmd	User writeable. Use this attribute to shelve or unshelve an alarm. Default values: Duration = 0, Reason = ""
AlarmShelved	Read-only, Boolean value. Shows True if alarm is shelved, False if alarm is unshelved. Default value: False
AlarmShelveStartTime	A read-only date/time stamp indicating when alarm shelving began, based on the engine time when the shelving request was received, Default value: Blank
AlarmShelveStopTime	A read-only date/time stamp equal to the AlarmShelveStartTime plus the duration for which the alarm is to be shelved. Default value: Blank
AlarmShelveReason	A read-only string value providing the reason for which the alarm was shelved or un-shelved by the Alarm Shelve command. Default value: Blank (See AlarmShelveCmd attribute.)

AlarmShelveUser	Read-only, the name of the user who most recently shelved or un-shelved the alarm with the Alarm Shelve command. Default value: Blank
AlarmShelveNode	Read-only, the name of the computer node from which the user most recently shelved or un-shelved the alarm with the Alarm Shelve command. If the node is hosted in a Terminal Server client session, the node and the TSE ID are both identified. Default value: Blank

Enable alarm shelving

Alarm shelving is configured from the **Galaxy Configuration** menu, and is enabled on the Area object.

To enable alarm shelving

1. Enable at least one security role to configure alarm shelving. Associate the relevant user with that role, if not already associated, before proceeding to step 2.
2. Select the **Galaxy** menu item, then select **Configure**.
3. Select **Galaxy**, then select **Alarms**. The **Alarms** page appears.

The screenshot shows the 'Configure' screen of the Galaxy Configuration interface. On the left, there is a sidebar with various configuration options like Security, Galaxy, Styles, Alarms and events (which is selected and highlighted in blue), Languages, Integrated development environment, Communication, and System. The main area is titled 'Alarms and events' and contains two tabs: 'Alarms' (selected) and 'States'. Under the 'Alarms' tab, there is a table titled 'Alarms' with columns: Severity, Description, Shelf, Historize, From priority, To priority, and Image. The table rows represent severity levels 1 through 4 with descriptions Critical, High, Medium, and Low respectively. The 'Shelf' column contains checkboxes, with levels 3 and 4 having checked boxes and levels 1 and 2 having unchecked boxes. A red box highlights the 'Shelf' column header. Below the table are sections for 'Modes' and 'Events'. The 'Modes' section has three entries: Inhibited/Disabled, Silenced, and Shelved. The 'Events' section lists event types: System, Application, and User, each with a checked 'Historize' checkbox. At the bottom right are 'Cancel' and 'Save' buttons.

Severity	Description	Shelf	Historize	From priority	To priority	Image
1	Critical	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	250	
2	High	<input type="checkbox"/>	<input checked="" type="checkbox"/>	251	500	
3	Medium	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	501	750	
4	Low	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	751	999	

4. Select the severity level in the **Shelf** column. By default, severity levels 3 (medium) and 4 (low) are enabled.

- In the IDE, open the relevant Area object editor. Click the **Shelve Alarms** checkbox to enable shelving for that area. Shelving is enabled by default.

Shelf alarms at runtime

Use a runtime client to shelve and unshelve alarms. From Application Server, you can use Object Viewer to monitor and control shelved alarm attributes. Shelved alarms can also be monitored and controlled from an embedded Alarm Control and at least one animation to write to the AlarmShelveCmd attribute. For information about using the Alarm Control to shelve alarms, see *Shelve and Unshelve Alarms at Run Time* in the AVEVA OMI help.

To shelve or unshelve an alarm in Object Viewer

- In Object Viewer, select the object in the **Console Tree** (left pane) which contains the alarm you want to shelve. The object attributes display in the **Details Pane** (right pane).
- In the Attributes list, find the attribute configured with the alarm you want to shelve.
For example, "Inlet_001.Discrete_001".
- In the Attributes list, below the attribute in step 2, find the **AlarmShelveCmd** attribute.
For example, "Inlet_001.Discrete.001.AlarmShelveCmd".
- Double-click the .AlarmShelveCmd attribute. The **Modify String Value** dialog opens.

Attribute Name	Value	Timestamp	Quality	Status	Security...	Category
StateRunning.RateDeadBand	0.0		C0:Good	Ok	Operate	Write
PV.Bad.InAlarm	false	4/21/2022 4:06:...	C0:Good	Ok	ReadOnly	Calculated
PV.Bad.TimeAlarmOn			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.TimeAlarmOff			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.TimeAlarmAcked			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.Priority	900		C0:Good	Ok	Tune	Write
PV.Bad.Category	Discrete		C0:Good	Ok	Tune	Write
PV.Bad.AckMsg			C0:Good	Ok	FreeAccess	Write
PV.Bad.Acked	true		C0:Good	Ok	ReadOnly	Calculated
PV.Bad.DescAttrName			C0:Good	Ok	FreeAccess	Write
PV.Bad._refAlmAttrKey			C0:Good	Ok	ReadOnly	System
PV.Bad._CategoryEnum			C0:Good	Ok	ReadOnly	Configurable
PV.Bad.ActiveAlarmState			C0:Good	Ok	ReadOnly	Configurable
PV.Bad._AlarmModeEnabled			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.AlarmMode			C0:Good	Ok	FreeAccess	Write
PV.Bad.AlarmModeCmd			C0:Good	Ok	FreeAccess	Write
PV.Bad.AlarmInhibit			C0:Good	Ok	FreeAccess	Write
PV.Bad.AlarmShelveCmd			C0:Good	Ok	FreeAccess	Write
PV.Bad.AlarmShelved			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.AlarmShelveStatus			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.AlarmShelveStart			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.AlarmShelveStop			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.AlarmShelveReason			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.AlarmShelveUsual			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.AlarmShelveNormal			C0:Good	Ok	ReadOnly	Calculated
PV.Bad.Condition	false	4/21/2022 4:05:...	C0:Good	Ok	ReadOnly	Calculated

Modify String Value

Reference: Equipment_Agitator_001.PV.Bad.AlarmShelveCmd

Duration=0; Reason="";

Cancel Apply OK

- To shelve an alarm, enter a duration in hours or in decimal fractions if the duration is to be less than one hour. The duration cannot be blank.

- b. Enter a reason between the quotation marks for shelving or unshelving. The reason cannot be blank. Click **OK**.
- c. To unshelve a shelved alarm, enter a value of "0".

The six read-only alarm shelving attributes will display the current shelving status information.

Throttle alarms

Alarm throttling prevents network and message queues from flooding during periods of high alarm activity. Throttling sets a maximum number of transitions into and out of an alarm state within a defined period. Alarm transitions that exceed the throttling limit are not reported.

The WinPlatform object includes tuning parameters that specify the maximum alarm rate per second on an engine. The Alarm throttle limit attribute must be used in conjunction with the scan period to determine the maximum number of alarms that can occur in a scan cycle. You can set the alarm throttle limit when you configure the WinPlatform object. For more information about setting the scan period and alarm throttle limit, see [Configure WinPlatform object alarms](#).

Alarm messages can be throttled for alarms going into alarm and out of alarm state. Alarm acknowledgement messages are not throttled. Users can still acknowledge alarms even when alarm throttling is active. Users can still disable alarms for objects or areas when the alarm rate causes throttling to occur. Also, the alarm inhibit and the disable/enable/silence messages are not throttled.

If an active alarm is disabled, the going out of alarm and disabled messages are sent to the notification distributor. If alarms are being throttled, the going out of alarm message can be throttled. The disabled message is accepted regardless of the throttling status. The going out of alarm message is not raised again. This can result in never logging a message for that alarm indicating it went out of alarm. The final alarm list in the Notification Distributor still shows the correct alarm state.

Propagate timestamps with alarms and events

An alarm feature always registers a reference to the alarmed attribute. This registered reference guarantees that Message Exchange never suspends updates for the alarmed attribute. Even if alarms are disabled for a particular attribute, the Advanced Communication Management feature cannot suspend the attribute.

Be aware that the time stamp propagates to all masked bits of an integer attribute, even if only one of the bits changes.

For example, you have an Integer address in a PLC that represents 16 different alarm states. You assign ObjA.Attr_Integer to point to the PLC address. You then split the bits to different alarm attributes, adding one attribute for each alarm and naming them ObjA.Attr_Alarm00 to ObjA.Attr_Alarm15. Each attribute has an input source that refers to a different bit of ObjA.Attr_Integer. For example, ObjA.Attr_00.InputSource -> me.Attr_Int.00, and so on. At runtime, when bit 00 is changing in the PLC, all of the attributes (ObjA.Attr_Alarm00 to ObjA.Attr_Alarm15) get a new time stamp, as all the bits changed. This can cause incorrect time stamps for alarms.

Alarms or events become active

For alarms or events, if an attribute has no timestamp, the current AppEngine scan time is used instead. If an attribute has a timestamp, the timestamp of the value that caused the alarm to occur is used as the value for the

object's AlarmOnTime attribute.

When an alarm is silenced or disabled, and an alarm condition becomes TRUE, when the alarm is later enabled, the timestamp for AlarmOnTime is the timestamp of the most recent attribute change, not the time at which the alarm was enabled.

For an event, if an attribute has a timestamp, the timestamp of the value that caused the event to be reported is used for the EventTime. A System Event timestamp is the current system time.

Alarms become disabled

If an active alarm becomes disabled, the alarm is forced to return to normal. The timestamp corresponds to the current time when the alarm became disabled, not the timestamp of the attribute nor of the alarm condition. Otherwise, the assigned timestamp is the AppEngine scan time.

Alarms revert to normal

If an attribute value has a timestamp, the timestamp of the value that caused the alarm to revert to normal is used for the AlarmOffTime.

If alarm is based upon several attributes or upon several values of a single attribute, the most recent timestamp is used when assigning values to the AlarmOnTime or AlarmOffTime.

Alarm acknowledgement

Alarms are acknowledged by users who view unacknowledged alarms from their client applications like InTouch HMI or AVEVA OMI. Only alarms of certain priority levels need to be acknowledged.

The basic workflow to acknowledge an alarm consists of the following general steps:

- The alarm is detected and reported to any subscribed alarm clients. The alarm is unacknowledged unless it is of a priority level that does not require acknowledgement.
- An authorized user attempts to acknowledge the alarm from the client. The user can type an optional comment when acknowledging the alarm.
- The user's acknowledge request is sent to the detecting object's alarm feature. The acknowledgment must pass through standard security checks first. The acknowledge request contains the user's name and any alarm comment.

For the alarm utility, the alarm is acknowledged within the alarm feature immediately. The user name, comment, and acknowledged time are also saved. Alarm comments can be localized into any supported language.

The acknowledge is considered an alarm state change, which is sent to all subscribed clients. When an alarm is acknowledged, the current AppEngine timestamp is used as the acknowledgement time.

After an alarm is acknowledged for the first time, any additional (extra) acknowledgement attempts for the same alarm are rejected and an error is returned.

Acknowledge alarms when signature is required

You can use the SignedAlarmAck() script function in Industrial Graphics to require a signature to acknowledge alarms. See "Signature Security for Acknowledging Alarms" in *Creating and Managing Industrial Graphics User*

Guide for information about adding the script function to a graphic. Then, embed the graphic in the WindowMaker window or an AVEVA OMI layout.

When you acknowledge the alarms at runtime, the script function checks for a required signature. If a signature is required, it checks Galaxy security settings and alarm priority. If any alarm in the list falls within the configured conditions, you must provide your signature to acknowledge the alarms. You must sign in to acknowledge the alarms if no user is logged on to the InTouch or OMI application.

If any alarm in the list is requires a signature, then a signature is required for the entire list of alarms.

Note: The SignedAlarmAck() script function supports only System Platform alarms.

By inspecting the Alarm Viewer or the Alarm Control client, you can determine if the alarm acknowledgements were successful.

Configure alarms

Alarming capabilities are a part of object templates, but they are not implemented until you configure the object in the IDE. After alarms are configured, you can view Application Server alarms in a client visualization application like InTouch HMI or the AVEVA OMI Alarm App.

You can add latched alarms as an additional alarm state for the Galaxy to provide an extra level of visibility and oversight to alarm conditions. See Alarm Latching for more information.

Configuring an object to be an alarm provider includes the following general sequence of steps:

1. Decide whether alarm notification is needed for each possible alarm condition of an object. For example, a command time-out alarm for a valve if the output command fails to move the valve.
2. Edit the object and set an attribute that specifies alarming.
3. Edit the object from the IDE and assign values to alarm attributes.
4. Configure the alarm properties. Typically, the fields that require configuration are Category, Priority and Description.
5. Configuring any limit fields to set an alarm. For example, the feedback time-out time limit.

You can add alarm detection and reporting capabilities to objects that were not originally developed to detect alarms. You do this by configuring alarm features for the object's attributes.

Configure alarms for system objects

To enable alarming for your application, you need to configure your Galaxy's WinPlatform and AppEngine objects as alarm providers. Both system objects report their own alarms.

Client applications subscribe to application object alarms by the area containing the objects. Client applications can also subscribe to WinPlatforms and AppEngines directly. These are called "pseudo-areas." They do not need to be assigned to an area for a client to see the alarms, although the user may want to assign them to an area, such as for simplifying the alarm query in the InTouch Alarm Viewer or Alarm DB Logger.

The following list shows the alarms for each system object.

- WinPlatform
 - Excess CPU load alarm
 - Low disk space alarm

- Excessive page faults alarm
 - Low memory alarm
 - Engine failure alarm
 - Engine checkpoint failure alarm
 - Object quarantined condition
 - Subscription folding condition
 - Scheduler scan overrun condition
- A WinPlatform object includes a general communication alarm when it loses contact with the areas to which it is subscribed.
- AppEngine
 - Checkpoint failure alarm
 - Object quarantined condition
 - Subscription folding condition
 - Scheduler scan overrun condition
 - Redundancy failover alarm
 - Redundancy Standby unavailable
 - Redundancy Standby not ready
 - ViewEngine
 - Checkpoint failure alarm
 - Object quarantined condition
 - Subscription folding condition
 - Scheduler scan overrun condition

WebViewEngine, Area and InTouchViewApp system objects do not include any alarms.

Configure WinPlatform object alarms

You select the areas of your Galaxy to monitor for alarms from the WinPlatform object. Also, you can select specific alarms for the status of the WinPlatform itself.

You must specify that the WinPlatform object is an InTouch alarm provider to subscribe to alarms from the various areas (and pseudo-areas) of the Galaxy and report them to the InTouch Alarm Manager.

Configure alarm priority for communication failure

InTouch Alarm Provider reports any communication outage to the InTouch Alarm Manager. Communication alarms are displayed and must be acknowledged. By default, the priority of a communication failure is "1", the highest priority.

There can be instances when a communication failure is not a high-priority alarm, and should not require the attention of a high-priority alarm. Shutting down or undeploying a platform as a normal operation can result in communication failure alarms, for example.

To indicate whether the alarm is of high or low importance, the communication failure alarm priority is configurable at design time as an integer of range 1–999. InTouch Alarm Provider must be enabled in the

WinPlatform object editor, **General** tab.

Register using "Galaxy_<GalaxyName>" instead of "Galaxy" option

When you select the **Register using "Galaxy_<Galaxy name>" instead of "Galaxy" option**, you enable ITAlarmProvider contained in WinPlatform to register the ITAlarmSubsystem using a provider name of Galaxy_GalaxyName.

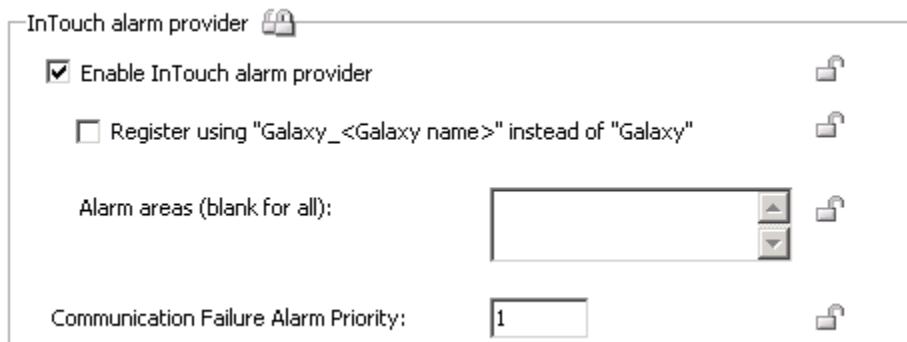
For example, if your galaxy is named Andromeda, the ITAlarmSubsystem registers as Galaxy_Andromeda.

This configuration can be useful if you are using an Alarm Client to display alarms from multiple galaxies. For more information about multiple galaxies, see [Working with Multiple Galaxies](#).

- This means that all InTouch applications which query alarms from this platform should be modified to use Galaxy_GalaxyName!AreaName.
- Select this option when the Alarm Control, the embedded alarm client (EAC), or the AlarmViewControl in InTouch queries alarms from different galaxies.

To configure a WinPlatform object to be an alarm provider

1. Open the WinPlatform object in the Object Editor.
2. Click the **General** tab.



3. Select the **InTouch alarm provider** check box.
4. Select the **Register using "Galaxy_<GalaxyName>" instead of "Galaxy"** check box to enable Galaxy_<GalaxyName> registration for alarm comment language switching. An information box appears. Click **OK** on the information box to continue.
5. In the **Alarm Areas** box, type the names of areas to subscribe to for alarms.
If you leave the **Alarm Areas** box blank, the WinPlatform subscribes to all areas in the Galaxy.
If you want to subscribe to only selected areas within the Galaxy, insert a space between each area name.
For example:
Area1 Area2 Area3
6. Enter a number from 1 to 999 to configure the **Communication Failure Alarm Priority**.
7. Click the **Engine** tab to show the **Alarm throttle limit** box. Either accept the default throttle limit of 2000 alarms per second, or enter another value. A value of 0 disables alarm throttling. For more information about alarm throttling, see [Throttle alarms](#).
8. Click the **Alarms** tab to show platform and engine, and scheduler alarms that can be set for the WinPlatform

object.

[General](#) [Engine](#) **Alarms** [Platform History](#) [Scheduler History](#)**Platform** Report excess CPU load alarm

Alarm limit:

80.0



Priority:

500



Value deadband:

0.0

 Report low disk space alarm

Alarm limit:

10000.0

MB



Priority:

500

 Report excessive page faults alarm

Alarm limit:

NaN



Priority:



Value deadband:

0.0

 Report low memory alarm

Alarm limit:

1000.0

MB



Priority:

500



Value deadband:

0.0

 Report engine failure alarm

Priority:

500

**Engine** Report checkpoint failure alarm

Priority:

500

 Report object quarantined condition

Priority:

500

 Report subscription folding condition

Priority:

500

**Scheduler** Report scan overrun condition

Consecutive scan overrun limit:

1



Priority:



Page 464

9. Select the check box next to each alarm that you want to enable for the WinPlatform object.
10. Set the limit, value deadband, and priority for each alarm you selected.
11. Save and close the Object Editor.
12. Check the object in to the Galaxy.
13. Deploy the object in an on scan state.

Configure alarms for an AppEngine object

You can set AppEngine attributes that determine whether alarms are enabled for an engine, scheduler, and redundant failover engine.

To configure AppEngine object alarms

1. Open the AppEngine object in the Object Editor.
2. Click the **Alarms** tab to show engine, scheduler, and redundancy alarms that can be set for the AppEngine object.

The screenshot shows the 'Alarms' tab selected in the top navigation bar. The interface is organized into three main sections: 'Engine', 'Scheduler', and 'Redundancy'. Each section contains a list of alarm types with checkboxes and priority input fields.

- Engine:**
 - Report checkpoint failure alarm
Priority:
 - Report object quarantined condition
Priority:
 - Report subscription folding condition
Priority:
- Scheduler:**
 - Report scan overrun condition
Consecutive scan overrun limit: Priority:
- Redundancy:**
 - Report alarm when a failover occurs
Priority:
 - Report alarm when Standby becomes unavailable
Priority:
 - Report alarm when Standby becomes not ready
Priority:

3. Select the check box next to each alarm that you want to enable for the AppEngine object.
4. Set the priority for each alarm you selected.
5. Save and close the Object Editor.
6. Check the object in to the Galaxy.
7. Deploy the object in an on scan state.

Configure alarms and events for application objects

The following table shows the different types of alarms that can be specified for application objects. The table shows the application objects containing native alarm attributes.

You can also set alarms for an object's attributes. The types of alarms available depend on the attribute type. For more information about setting alarms for attributes, see [Set alarms on the attributes page](#).

Application Object	Alarm Types				
State	Limit	Target Deviation	Rate of Change	Statistics	
AnalogDevice	X	X	X	X	
DiscreteDevice	X				X
FieldReference					
Sequencer	X				
SQLData					
UserDefined	X	X	X	X	

The following list shows the types of alarms for each application object in more detail.

- AnalogDevice
 - Level alarms (HiHi, Hi, Lo, LoLo) [limit alarms]
 - Rate of Change alarms (Up, Down)
 - Target Deviation alarms (Minor, Major)
 - PV Bad Value alarm [state alarm]
- DiscreteDevice
 - Uncommanded change alarm [state alarm]
 - Command time-out alarm [state alarm]
 - Active1 state alarm [state alarm]
 - Active2 state alarm [state alarm]
 - Fault state alarm [state alarm]
 - Active1 state duration alarm [statistical alarm]
 - Active2 state duration alarm [statistical alarm]
- Sequencer
 - Execution halted [state alarm]
 - Condition trigger failure [state alarm]
 - OnEntry output failure [state alarm]
 - OnExit output failure [state alarm]
- UserDefined (Attributes can be alarmed)
 - PV State alarm [state alarm]
 - PV Bad Value alarm (that is, bad quality) [state alarm]
 - Attribute alarm features
 - State alarms
 - Limit alarms (HiHi, Hi, Lo, LoLo)

- Rate of Change alarms (Up, Down)
- Deviation alarms (Minor, Major)
- Bad Value alarm (that is, bad quality)

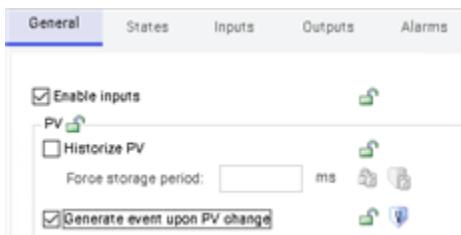
There are no built-in alarms for these application objects:

- FieldReference
- SQLData

You can also configure your application objects to generate an event each time the object's PV value changes. As mentioned above, you can also configure an alarm on any object for any attribute. See [Set alarms on the attributes page](#) for more information.

To configure alarming and events for application objects

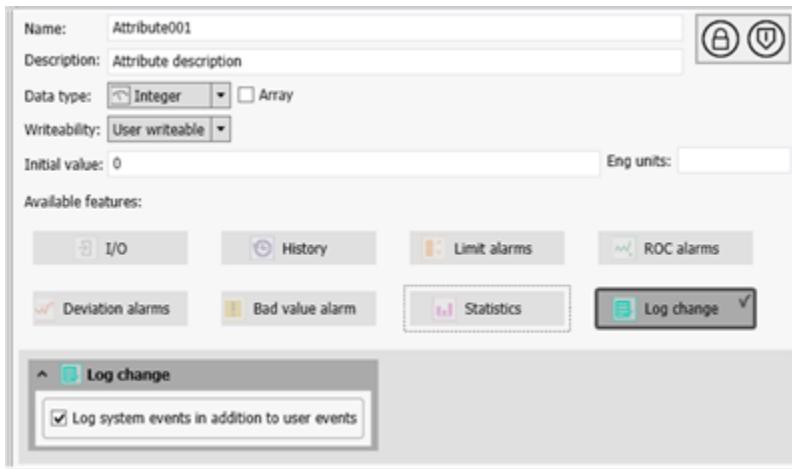
- Open the application object with the Object Editor.
- For a DiscreteDevice, click the **General** tab, enable inputs, and then enable **Generate event upon PV**.



For objects that do not include this check box, use the **Attributes** page.

Note: If you are using field attributes instead of Attributes, use the **Field Attributes** page.

- Enable the **Log change** feature.
- Enable **Generate event upon change**.



- Select or clear the check box based on whether you want to generate an event each time the object's PV value changes.
- Click the tab that lists alarm attributes.
 - For the AnalogDevice object, click **Alarms**.
 - For the DiscreteDevice object, click **Alarms**.

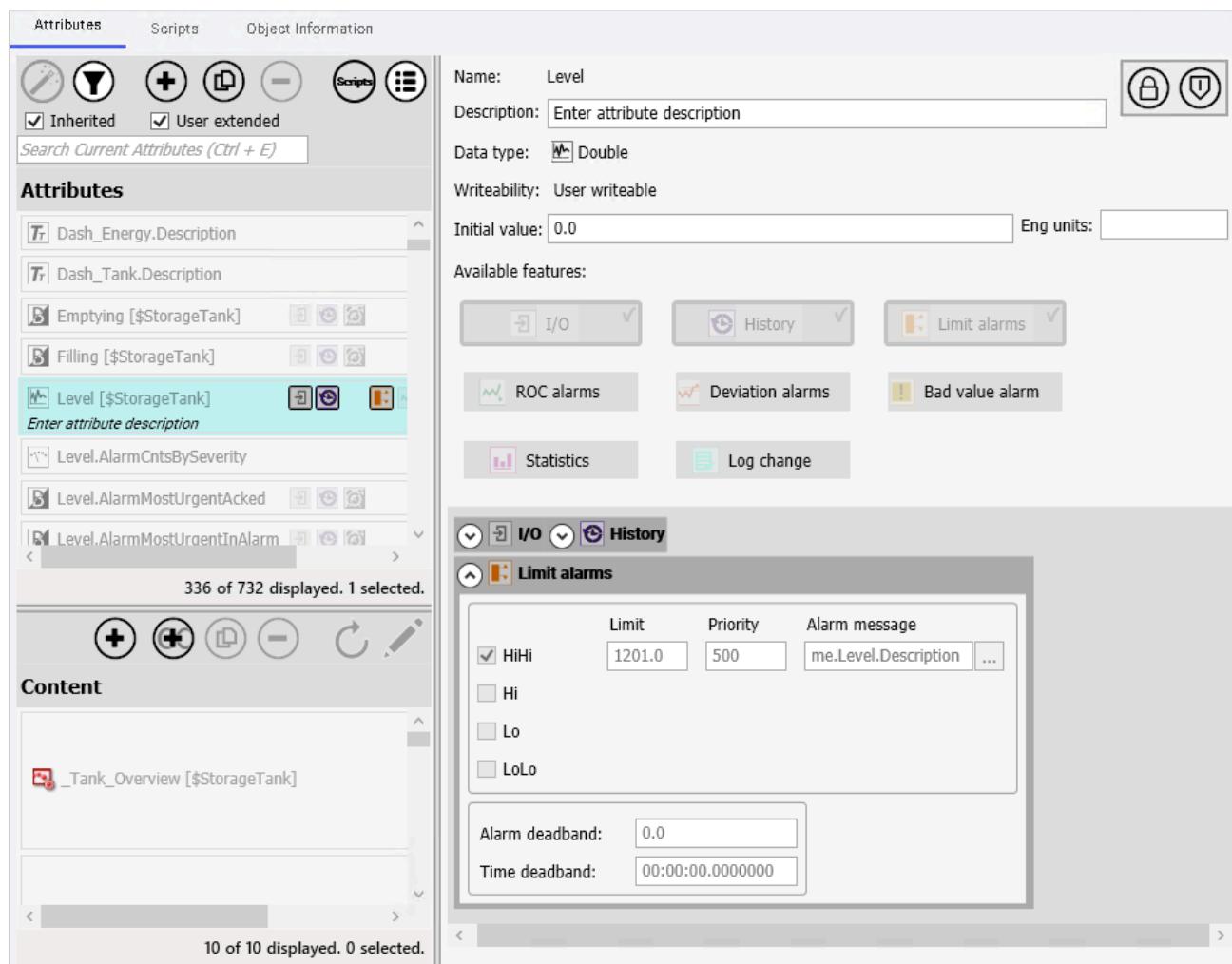
- For the Sequencer object, click **Settings**.
 - For the UserDefined object, click **Attributes** (or **Field Attributes**, if you are using field attributes). See [Set alarms on the attributes page](#) for further directions.
5. Select the check box that enables alarming for the object.
 6. Assign values to the attributes for the type of alarm you selected by completing the following steps:
 - a. Assign values to the alarm limits based on the type of alarm.
 - b. Assign an alarm priority (1-999) for each limit you set.
 - c. Accept the default alarm message or include another message for each alarm limit.
 - d. Assign values to the remaining attributes based on the type of alarm you selected. For more information about other alarm attributes, see [Types of Alarms](#).
 7. Save your object changes and close the Object Editor.

Set alarms on the attributes page

You set alarms for all Application Server objects in the Attributes page in the Object Editor.

To specify alarms for an object's attributes

1. On the **Attributes** page of the Object Editor, select an attribute from the **Attributes List**.



2. Select the alarm features you want to configure for the attribute. For Boolean data types, you can select:

- **I/O Alarm**
- **State alarm**
- **Bad value alarm**

For Integer, Float and Double data types, you can select:

- **Limit alarms**
- **ROC alarms**
- **Deviation alarms**
- **Bad value alarms**

Note: Depending on the object type, some of these alarm types may not be available.

3. For each alarm feature that you activate, enter the settings for each alarm.
 - a. Assign values to the alarm limits based on the type of alarm.
 - b. Assign an alarm priority (1-999) for each limit you set.
 - c. Accept the default alarm message or include another message for each alarm limit.
 - d. Assign values to the remaining attributes based on the type of alarm you selected. For more information about other alarm attributes, see [Types of Alarms](#).

4. To configure event logging for an object, select the **Log change** feature and click the checkbox **Log system events in addition to user events**.
5. Save your object changes and close the Object Editor.

Distribute alarms and events

After you configure object instances for alarm detection, deploy the instances and put them On scan. The instances begin checking for alarm conditions.

When an alarm is detected, or an event occurs, a notification is reported to its alarm and event distributor, which is running on the same AppEngine.

These alarm and event distributors include:

- **Area objects:** Area objects report detected alarms through the Area, which distributes them to alarm and event clients.
- **WinPlatform objects:** Report their own alarms and events.
- **AppEngine objects:** Report their own alarms and events.
- **Device Integration objects:** Report their own alarms and events

The Area object plays a key role in alarm and event distribution. All objects belong to an Area. Areas can contain sub-Areas. Alarm and event clients are configured to subscribe to a set of Areas.

Areas provide a key organizational role in grouping alarm information and assigning it to users who use alarm and event clients to monitor their Areas of responsibility.

WinPlatforms, AppEngines and Device Integration objects do not report their alarms and events to Area objects even though they belong to Areas. This allows alarm clients to receive alarm notifications without any dependencies on Area objects. For example, a deployed and running WinPlatform can report alarms even though its Area is not deployed and running.

Alarm-event distributor objects maintain a list of all currently active alarms and inactive but unacknowledged alarms. They do not maintain a list of events, which are routed to clients that are currently subscribed at the time of the event.

You can configure a WinPlatform to act as an InTouch Alarm Provider in the runtime environment.

The WinPlatform sends an alarm through the InTouch Distributed Alarm System to InTouch clients when the WinPlatform loses communication with an Area that it subscribes to. This condition typically occurs during a network outage with computers hosting those Areas.

In a network outage, the WinPlatform InTouch Alarm Provider sends an alarm for each disconnected Area that it subscribes to, including all of its alarm distribution hierarchy. Each of these alarms is a high priority alarm that contains the name of the Area to which communication is lost. These communication problem alarms must be acknowledged.

Although they still appear in the historical record, any current alarms from the disconnected Area drop from the InTouch client's summary list. They can no longer be acknowledged.

When communication to the disconnected Areas is restored, any unacknowledged alarms generated in those Areas are sent to the alarm client.

Subscribe to alarms and events from a client

Clients indicate interest in alarms and events by subscribing to an Area. When subscribing to an Area, the subscription is actually to all notification distributors within that Area.

For example, if an Area contains sub-Areas, those sub-Areas are subscribed to. If WinPlatforms, AppEngines or Device Integration objects belong to an Area, those objects are also directly subscribed to.

When a notification distributor receives an alarm and event subscription from a client, the notification distributor provides the client with the following:

- A list of all current alarm conditions, including unacknowledged return-to-normal conditions.
- An alarm condition state change. A state change includes transitions into or out of alarm (return to normal) and change in acknowledged flag.
- An event occurrence.

Alarm and event subscription requests do not include filters, for example, only show alarms greater than a specific priority value. All alarm and event messages received by the notification distributor are sent to all subscribed clients. Filtering is provided as a display option by clients.

Using InTouch HMI as the Alarm and Event Client

InTouch runtime clients subscribe to event reports from a Galaxy. Application Server reports alarms to the InTouch Distributed Alarm System, which subscribes to alarm and event reports from a Galaxy.

An InTouch client application can visualize Application Server components. An InTouch alarm client can show alarm information for new, unacknowledged alarms, including all required fields.

The new alarm is in the unacknowledged state. An operator can view alarms, acknowledge alarms, disable alarms, and enable alarms from the client application running in InTouch WindowViewer.

Syntax of alarm queries

InTouch alarm queries subscribe to alarm and event information from objects within a Galaxy. The alarm and event queries can be in the form of user input or a script.

Alarm query syntax must be in one of the following forms:

\Provider!Area

or

\Node\Provider!Area

You can have one or more references in a query separated by spaces.

You can also optionally append a tagname filter at the end, separated by another exclamation mark:

\Provider!Area!Filter

\Node\Provider!Area!Filter

The filter can have a wildcard * character at the beginning or at the end, but not both.

The \Node at the beginning is only important if you want to query for alarms from a provider on another computer. Otherwise, you can leave it off and the reference is assumed to be a provider on the local computer. The provider name Galaxy refers to alarms and events that get reported by the WinPlatform configured as an

InTouch alarm provider on that computer node.

Alarm query syntax when 'Register Using Galaxy_<GalaxyName>' is enabled

In the WinPlatform object, when you enable InTouch alarm provider, you can enable **Register using Galaxy_<GalaxyName> instead of Galaxy**. This option will register the platform to the alarm subsystem using the Galaxy name prefixed by "Galaxy_" instead of just the word "Galaxy". This allows an InTouch application to monitor alarms from multiple Galaxies and avoid name conflicts.

Syntax changes slightly when Galaxy_GalaxyName is enabled:

- Use \\ for computer name.
- Use \ for Galaxy or Galaxy_<GalaxyName>.
- Use ! for Area.

For example: \\Galaxy\MyGalaxy!Area001.

If Galaxy_GalaxyName is not enabled in WinPlatform, then the default behavior described in the previous section applies.

You can determine if Galaxy_<GalaxyName> has been enabled by monitoring the runtime attribute of the platform ITAlarmProvider.ProviderNameAsGalaxyNameEnabled.

Alarm query examples

- You can submit a query to get all alarms from Area1 and all other alarms within Area1, as reported by the WinPlatform object on the local computer.

```
\Galaxy!Area1
```

The query returns all alarms and events from all objects directly contained in Area1 and any sub-areas contained by Area1. This hierarchy is determined by what is configured in the Model View in the IDE.

- If Area1 and Area2 are two separate mutually exclusive areas, you can submit a query for alarms from both areas.

```
\Galaxy!Area1 \Galaxy!Area2
```

- If you're on NodeA and the WinPlatform is on NodeB, you can submit a query for the alarms from the remote computer.

```
\\\NodeB\Galaxy!Area1
```

- You can submit a query for all alarms from objects whose name begins with "Tank" in the TankFarm1 area.

```
\Galaxy!TankFarm1!Tank*
```

The trailing wildcard character matches alarms from all objects with names that begin with "Tank" like Tank001, Tank002, TankUpper, or TankLower.

- You can submit a query for specific alarm types. For example, you can submit a query for all HiHi alarms in the TankFarm1 area.

```
\Galaxy!TankFarm1!*.*.HiHi
```

- You can submit a query for all types of alarms from a specific object within an area.

```
\Galaxy!TankFarm1!Tank752.*
```

The trailing wildcard character matches all alarm types for Tank752.

Alarm requirements for InTouch client applications

For Application Server alarming to function, the following conditions must be met:

In Application Server:

- One or more Area objects are deployed and running.
- The source object is on scan.
- The source object's Area is on scan.
- Alarming must be enabled for the target object.
- An InTouch alarm provider on any WinPlatform in the Galaxy.

In InTouch:

- The InTouch client application is running in WindowViewer.
- An InTouch alarm ActiveX control is placed in a window and configured as an alarm consumer for the Galaxy.
- The user has logged into InTouch with security enabled and is authorized to acknowledge alarms for the object that is in the alarm state. If the user only wants to view alarms, security authorization is not required.

Application Server validates the user has sufficient security privileges to acknowledge the alarm.

If the user does not have privileges to acknowledge alarms, the user can attempt to acknowledge the alarm, but the Galaxy rejects the acknowledgment request. The alarm remains unacknowledged in the InTouch Alarm display.

The rejected alarm acknowledge event is recorded in InTouch Event History if the user attempting the acknowledgement has a valid Galaxy user account. Otherwise, the rejected acknowledgement is not recorded as an event.

You can acknowledge multiple alarms by providing a valid signature. At runtime, the Industrial Graphics SignedAlarmAck() script function checks a set of alarms and conditions to find out if any alarm from the list requires a signature in order to acknowledge it. If so, you must provide your user name, password and domain or your Smart Card details and PIN to authenticate yourself and acknowledge the alarms. To be able to provide the Smart Card details, your computer must be configured for Smart Card authentication.

Note: Smart Card authentication is not supported in multi-galaxy environments for read/write operations to remote galaxies.

If even one alarm in the list requires a signature, you must provide a signature to acknowledge the alarms. The SignedAlarmAck() function will acknowledge ALL the alarms in the list.

With the SignedAlarmAck() function, you can provide credentials and acknowledge an alarm or group of alarms even if you are not the logged-on operator.

Alarms and event implementation in application server and runtime applications

Application Server and its runtime applications (AVEVA OMI and InTouch HMI) implement alarms and events. The following table shows the similarities and differences between the products.

Item	InTouch	Application Server/OMI
Alarm configured or detected by	Within a tag	Within an object
Alarm Classes (client column)	Only certain classes of alarms are supported or detected: DSC, VALUE, DEV, ROC, SPC.	No system-wide distinction for classes. Alarms are tied to a Boolean that can be triggered from any logic.
Alarm Type (Sub-class) (client column)	Discrete, LoLo, Lo, Hi, HiHi, MinorDev, MajorDev, ROC, SPC. Client column.	No sub-class. The Alarm feature name is the closest concept. For example, ".PVHiAlarm". Mapped from Category.
Priority (client column)	1-999 (1 most urgent)	Priority 0-999. 0 most urgent. 0 is mapped to 1 in InTouch.
Name (client column)	Alarm name = Tag name.	Object.attribute
Comment (client comment)	Separate alarm comment, which is different from the tag comment.	Object short description or alarm message where available.
Group	Alarm group allows client-side filtering. Sub-groups must be on same InTouch.	No alarm group. But Area provides mappable concept. Sub-Areas can be on different nodes.
State	<p>No behavioral differences between Application Server/OMI and InTouch HMI.</p> <p>Four states, which are combinations of ACK/UNACK and ALARM/RTN. Optionally, a fifth state, Latched, can be enabled for the Galaxy.</p> <ul style="list-style-type: none"> • UNACK/ALARM (usually displayed as UNACK) • ACK/ALARM (usually displayed as ACK) • UNACK/RTN (usually displayed as UNACK_RTN) • ACK/RTN (usually displayed as ACK_RTN) • LATCHED (optional) <p>These states have a 1:1 correspondence with states of the Alarm feature, which keeps track of whether the alarm is InAlarm and IsAcked.</p> <p>Alarms in the state ACK/RTN are not shown in the SUMMARY alarm display because they do not need any further action from the operator unless Latched is enabled for the Galaxy. But, all four states appear in the HISTORY display, and in the Alarm Database.</p> <p>If the Latched state is enabled, ACK/RTN alarms remain visible in the</p>	

Item	InTouch	Application Server/OMI
alarm display until the operator sends a Dismiss command.		
Value, CheckValue	Only static values sent with alarm message.	Static values and dynamic references are provided.
Ack	No behavioral differences between Application Server/OMI and InTouch HMI. All alarms are sent to the client and require acknowledgement regardless of priority.	
Dismiss	If the Latched alarm feature has been enabled for the Galaxy, all alarms in LATCHED state require a dismiss command.	
History	No behavioral differences between Application Server/OMI and InTouch HMI. Alarm state changes are logged to event history and shown on historical client.	

Configure plant state-based alarms

You can map alarm modes on a per-Galaxy basis to different plant operational states to control how alarms are reported. Five plant states are pre-defined, and have default alarm states associated with them:

Plant state	Default alarm state	Available alarm states
Running	Enable	Enable
Maintenance	Disable	Enable / Silence / Disable
Startup	Silence	Enable / Silence / Disable
Shutdown	Disable	Enable / Silence / Disable
Testing	Silence	Enable / Silence / Disable

You can define new plant states, rename plant states, or remove existing plant states, except the "Running" state (you can, however, rename "Running"). The alarm state for Running is read-only and cannot be changed from Enable.

After you have defined plant state-based alarm configurations for the Galaxy, you can assign plant state-based alarming to area objects in the Galaxy. This is done by setting the PlantState attribute for each area object that will use plant state-based alarming. The area object will automatically update its PlantAlarmMode attribute to match the alarm state that is set for the PlantState currently assigned to it.

Attribute	Definition
PlantState	The name of the currently assigned plant state (Running, Maintenance, etc.).
PlantStateAlarmMode	The alarm state of the assigned plant state (enable, silence, or disable). This attribute is read-only at runtime.

Mapping alarm modes to plant states

Define plant states and map them to alarm modes in the IDE. To access the configuration dialog:

1. From the ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then select **Alarms**. The **Alarms** page appears.

Note: Settings in this dialog are not shared across Galaxies in a multi-Galaxy environment. Each Galaxy in the environment will have its own alarms configuration.

To use the default values, you do not have to set anything in this dialog. Simply enable plant state-based alarms for each area object that is to use this feature. See [Configure the state-based alarm mode on an area object](#) for additional information.

Alarms and events

Alarm adorner option

Severities	Alarm states
<input checked="" type="checkbox"/> Critical	<input checked="" type="checkbox"/> UNACK (Un-Acknowledged)
<input checked="" type="checkbox"/> High	<input checked="" type="checkbox"/> ACK (Acknowledged)
<input checked="" type="checkbox"/> Medium	<input checked="" type="checkbox"/> RTN (Return to normal)
<input checked="" type="checkbox"/> Low	

Options for aggregated alarm counters by severity

States and modes included	
<input type="checkbox"/> Inhibited	<input type="checkbox"/> Disabled
<input checked="" type="checkbox"/> Silenced	<input checked="" type="checkbox"/> Shelved

Alarm plant state

Description	AlarmMode
Running	Enable
Maintenance	Disable
Startup	Enable
Shutdown	Silence
Testing	Silence

Alarm logic

Alarms Latching	
<input type="checkbox"/> Enable Latching	

Change alarm modes and modify plant states

- To change the AlarmMode value, click the **AlarmMode** associated with the Plant State that you want to change. Then, select the new value (enable, silence, or disable).

Note: **AlarmMode** for Running is read-only and cannot be changed.

- To add a new Plant State, Click the **Add (+)** button. A new row is created at the bottom of the table.

Enter the **Description** of the new plant state. The maximum length is 64 characters.

Select a value for the **AlarmMode** (enable, silence, or disable).

- To delete an existing Plant State, click the **Remove (-)** button. The selected plant state and its associated value are removed from the table.

Note: You cannot delete the first row of the **Alarm plant state** table. Running is the default PlantState **Description** in the first row.

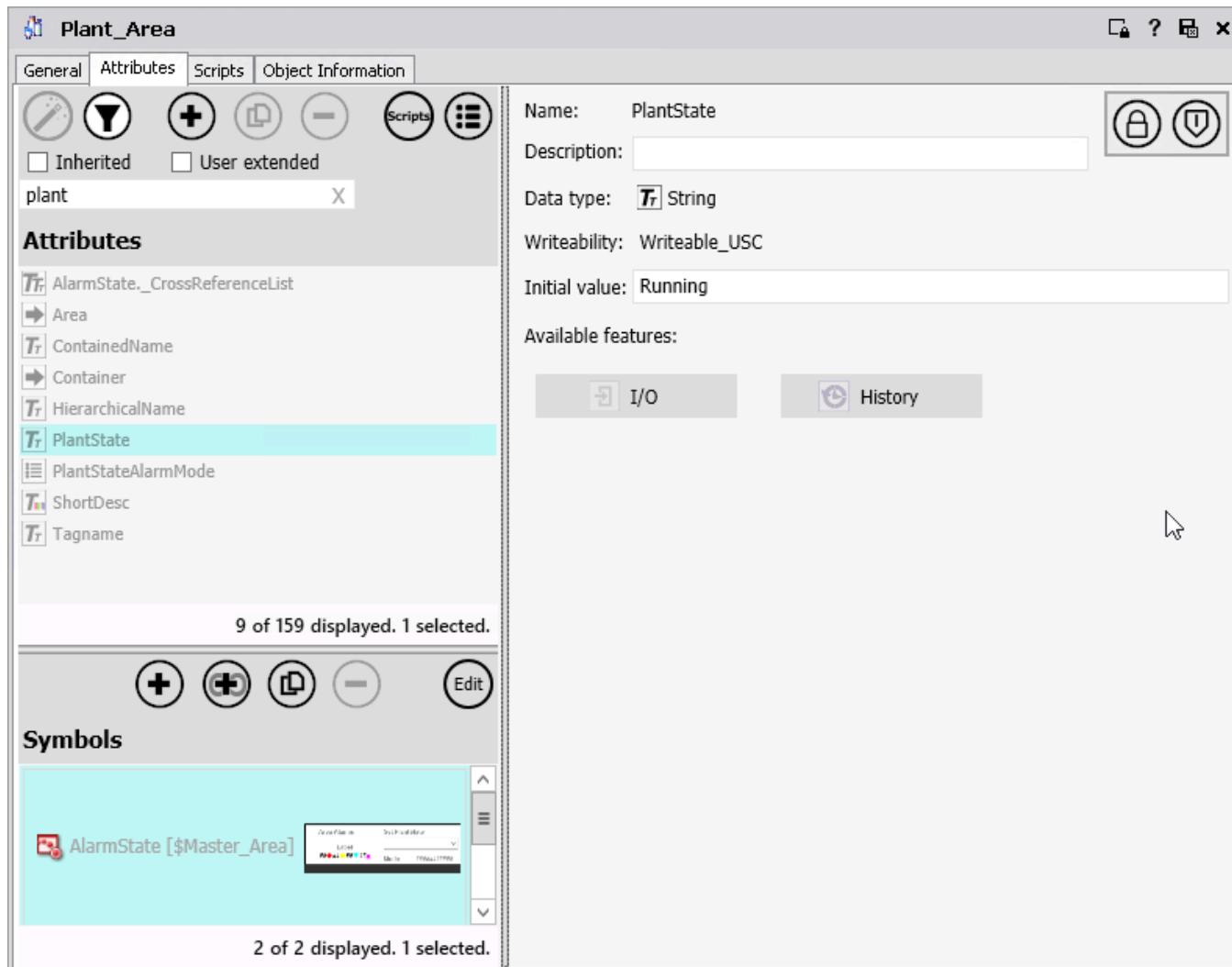
- To change the description of an existing PlantState, double click the plant state you want to rename, then enter the new description. The maximum length is 64 characters.

Note: All Plant States, including Running, can be renamed.

Configure the state-based alarm mode on an area object

State-based alarm modes apply to String attributes, with writeability set to Writeable_USC (these cannot be

changed). To use state-based alarming for one or more areas, you must configure the PlantState attribute for each area that will use plant state-based alarming.



Configure plant state alarming

1. Open the area object and click the Attributes page.
2. Select the PlantState attribute.
3. Enable the **I/O** feature for PlantState and select Read, Read/Write, or Write.
4. Enable the **History** feature and configure it.
5. Select the PlantStateAlarmMode attribute.
6. Repeat steps 3 and 4 to configure the **I/O** and **History** features for PlantStateAlarmMode.

View plant state-based alarms at runtime

You can view plant state-based alarming at runtime through clients such as Object Viewer and InTouch applications in WindowViewer.

User access and security at runtime

Operators with the appropriate permissions can change the PlantState of an area through a runtime client. An operator may have to change PlantState from Startup to Running, for example. Scripts can also be used to implement modifications.

To change a PlantState, the operator enters a string to match one of the defined PlantState values (for example, "Running"). Before a change to PlantState can be implemented, the system checks that the user belongs to a role that has the permission "Can Modify Plant State of an Area" and is therefore authorized to make the change. If the operator enters a string that does not match a defined PlantState, the change is rejected.

At runtime, the AlarmMode attribute is read-only. Therefore, the AlarmMode for a PlantState cannot be changed through a runtime client (for example, changing from enable to silence). It can only be changed through the IDE.

If an area is assigned to a PlantState and the PlantState is deleted through the IDE, the area will remain in that PlantState until it is changed. If AlarmMode for the deleted PlantState is anything other than enable, the AlarmMode will change to enable.

Note: Operators with appropriate permissions can use the AlarmModeCmd attribute to change AlarmMode of an area at runtime. However, the AlarmModeCmd can only be used to set a more restrictive condition than the AlarmMode of the area's corresponding PlantState. For more information about changing alarm modes, see [Enable, silence, and disable alarms](#).

Configure priority ranges for alarm historization, mapping, and shelving

The **Alarms and Events Configuration** dialog lets you:

- Map alarm severities to priority ranges.
- Enable alarm shelving by priority range.
- Enable or disable historization of both alarm severities and event types. All alarms are historized by default.

Default Alarm Mapping and Historization Values					
Severity	Description	Shelve	Historize	From Priority Range	To Priority Range
1	Critical	N	Y	1	250
2	High	N	Y	251	500
3	Medium	Y	Y	501	750
4	Low	Y	Y	751	999

Shelving, historization and priority ranges are configurable.

Mapping alarm severity to priority

- Severity 1 starts at priority 1 by default. Severities can be mapped to priorities in ascending or descending order. For example, severity 1 can map to priority range 1–250 or it can map to priority range 999–751.

- Severity 4 ends at 999 by default, but 999 is not required to be the end-of-range number. For example, severity 4 can end at 900, leaving a priority range above 900 unmapped to any severity level.
- A priority outside the configured priority ranges will return severity 0, not mapped. Since the priority is unmapped, it cannot be historized.

Note: Mapped alarm severities are not shared across galaxies in a multi-galaxy environment. Each galaxy in the environment has its own priority to severity mapping.

Configure historization for alarms and events

Alarms and events

Alarms
States

Alarms

Severity	Description	Shelve	Historize	From priority	To priority	Image
1	Critical	<input type="checkbox"/>	<input checked="" type="checkbox"/>	1	250	1
2	High	<input type="checkbox"/>	<input checked="" type="checkbox"/>	251	500	2
3	Medium	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	501	750	3
4	Low	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	751	999	4

Modes

Description	Image
Inhibited/Disabled	
Silenced	
Shelved	

Events

Types	Description	Historize
1	System	<input checked="" type="checkbox"/>
2	Application	<input checked="" type="checkbox"/>
3	User	<input checked="" type="checkbox"/>

Historization is configurable for both events and alarms. Event types and the default historization setting for each type are:

Default event type historization values		
Types	Description	Historize
1	System	Y

Default event type historization values

2	Application	Y
3	User	N

Note: In prior versions of Application Server, historization of alarms and events was configured through the Alarm DB Logger Manager (an InTouch component) and utilized the SQL database "WWALMDB." The method described here replaces the Alarm DB Logger method.

To configure alarm and event mapping

1. From the ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then select **Alarms**.
The **Alarms** page appears.
3. Configure alarms:
 - a. Edit the priority range for each severity level, or accept the default values.
 - b. Accept the default (enabled) historization or click to clear the check box and disable historization for each severity level.
4. Configure events: Accept the default historization values, or click to clear the check box and disable historization for each event type.

If you want to change the default Alarm Border indicator images shown in the **Alarms and Events Configuration** dialog, see "Changing Alarm Border Indicator Icons" in the *Creating and Managing Industrial Graphics User Guide*.

Monitor alarm severities at runtime

Monitor alarms by severity, expressed as integers 1–4, using clients such as Object Viewer and AVEVA OMI ViewApps. You can monitor individual alarms and you can monitor alarms aggregated by containment level. For more information about aggregating alarm states, see [Aggregated alarm state information](#).

Typically, alarm priorities and priority to severity mapping are not changed during runtime, but it is possible to change alarm priority configuration and severity mapping without closing the client application.

Alarm ranking at runtime

The most important alarm has the lowest severity number, but other criteria are taken into account when ranking alarms by urgency at run time.

Alarm Mode	Enabled is more urgent than silenced.
Alarm Shelved	FALSE (not shelved) is more urgent than TRUE (shelved).
InAlarm	TRUE (InAlarm) is more urgent than FALSE (normal).

Acked	FALSE (unacked) is more urgent than TRUE (acked).
Severity Level	1 is most urgent; 4 is least urgent.

Aggregated alarm information

Alarm aggregation lets you see all the active alarms within an area object, as well as all the active alarms on all the area's contained objects, including contained area objects. This provides an efficient way at runtime to identify if there are any active alarms (in the InAlarm state), and the location and status of alarms, for example, unacknowledged and unacknowledged - return to normal. This makes it possible to follow a trail from one level to the next to find the underlying cause of a complex object's alarms.

To obtain aggregated alarm severity status information:

1. Map alarm severity levels to priority ranges. For more information, see [Configure priority ranges for alarm historization, mapping, and shelving](#).
2. Configure alarms on objects. For more information, see [Configure alarms](#).
3. Enable alarm aggregation by Area. For more information, see [Configure alarm state aggregation](#)
4. View aggregated alarm status information by means of runtime clients and applications. For more information, see [Monitor alarm state information at runtime](#).

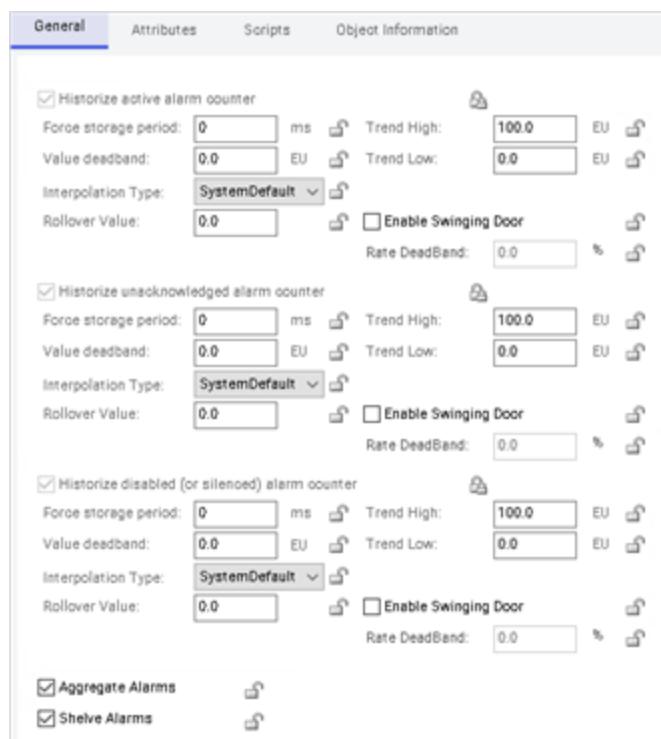
Configure alarm state aggregation

Configuring alarm state aggregation consists of normal alarm configuration procedures plus an added step of enabling the aggregation feature on each relevant Area object.

Note: Alarm aggregation is enabled by default on Area objects. Alarm aggregation cannot be disabled on application objects, such User Defined objects.

To configure alarm state aggregation

1. Accept the default settings or configure alarm severities for each of the four severity levels you want to aggregate. This is a global configuration.
2. Configure an object with one or more alarms. All alarms configured on all objects will be aggregated to the Area object that contains the objects.
3. Set the AlarmModeCmd of the object to enabled or silenced, but not to disabled.
4. Set the AlarmModeCmd of at least one alarm on the object to enabled or silenced, but not to disabled.
5. Set the AlarmModeCmd of the Area object to enabled or silenced, but not to disabled.
6. Check the **Aggregate Alarms** checkbox on the Area object editor to enable alarm aggregation. This sets the value of the AlarmAggregationStateCmd attribute to True.



Aggregated alarm state information

Runtime clients and applications, such as Object Viewer, AVEVA OMI ViewApps, and InTouch HMI applications, can show alarm aggregation statuses.

You can use animations such as the alarm border animation with Situational Awareness Library graphics or with Industrial Graphics in both Managed InTouch applications and AVEVA OMI ViewApps, and you can add alarm aggregation displays to navigation items in AVEVA OMI ViewApps.

Alarm aggregation functionality can be described for an object, for an area, and for an attribute.

Object	<p>Aggregation represents all alarms on the object, on all contained objects, and on their descendants down to the lowest level of the model view.</p> <p>Alarm aggregation values on child objects are added to the values of the parent object or objects.</p> <p>All objects have this alarm aggregation functionality.</p>
Area	<p>Aggregation represents the alarms on the Area object itself, on all objects assigned to the area, and on all sub-Areas, down to the lowest level of the model view.</p> <p>If the Area's AlarmMode is silenced, all alarms on all Objects in that Area will be silenced. For more information about setting alarm modes, see the following topics:</p>

	<ul style="list-style-type: none">• Alarm definition• Set alarm state with object attributes• Enable, silence, and disable alarms.
Attribute	Aggregation represents all the alarms on the attribute itself. This is the lowest level of aggregation.

Alarms are aggregated if they are in one of three states:

- UNACK_ALM
- ACK_ALM
- UNACK_RTN

Alarms in the ACK_RTN state are not aggregated.

Alarms in silenced mode are aggregated, even though they do not appear in a runtime client.

A set of five attributes provide runtime aggregated alarm status information:

AlarmMostUrgentSeverity	Displays the severity as an integer 1–4 of the most important current alarms on an object and its descendants. If no alarms are in the InAlarm state or waiting to be acknowledged, the value is 0.
AlarmMostUrgentMode	Displays the mode (enabled or silenced) of the most important current alarm. Alarms in disabled mode are not aggregated.
AlarmMostUrgentInAlarm	Displays the InAlarm status (true or false) of the most important current alarm.
AlarmMostUrgentAcked	Displays the acknowledgement status (true or false) of the most important current alarm.
AlarmCntsBySeverity	In AVEVA OMI and InTouch HMI runtime client applications, this attribute is used to show the number of alarms by status and severity. In the Object Viewer, this attribute displays an array that shows: <ul style="list-style-type: none">• The total number of active alarms (UNACK_ALM + UNACK_RTN + ACK_ALM) at each severity level.• The UNACK_ALM alarm count.• The UNACK_RTN alarm count.• A decimal representation if any active alarm bits are set on the local object.

Note: By default, **AlarmCntsBySeverity** includes shelved and silenced alarms, but does not include hidden or disabled alarms. You can change this by changing the **AggregatedAlarmStatesIncluded** filter. For more information, see [AggregatedAlarmStatesIncluded attribute](#)

AlarmCntsBySeverity attribute

AlarmCntsBySeverity is a 13 element array, divided into three groups of four, plus one additional element to indicate local alarm severity and status. The first group of four elements shows the number of aggregated active alarms by severity (S1 through S4), the next group shows the aggregated number of unacknowledged alarms (UNACK_ALM) by severity, and the third group shows the aggregated number of return to normal (UNACK_RTN) alarms by severity. The final element of the array is a bit value that defines which, if any, of the alarms are on the selected (local) object. See [Local alarm display in the Object Viewer](#) for additional information.

You can determine the number of active, acknowledged alarms (ACK_ALM) by subtracting the number of unacknowledged alarms shown in the second group of elements from the active alarm count shown in the first group of elements.

Note: By default, **AlarmCntsBySeverity** includes shelved and silenced alarms, but does not include hidden or disabled alarms. You can change this by changing the **AggregatedAlarmStatesIncluded** filter. For more information, see [AggregatedAlarmStatesIncluded attribute](#)

AggregatedAlarmStatesIncluded attribute

AggregatedAlarmStatesIncluded is a four-element array that controls whether shelved, hidden, silenced, and/or disabled alarms are included in the values of the **AlarmCntsBySeverity** attribute. It is a galaxy-wide attribute. It is the same for all engines, areas, and objects.

The elements of **AggregatedAlarmStatesIncluded** are:

Element	Alarm condition	Default
1	Shelved	True
2	Silenced	True
3	Disabled	False
4	Hidden	False

You can change the **AggregatedAlarmStatesIncluded** settings using the IDE. Follow the procedure below.

To configure the alarm states included in **AlarmCntsBySeverity**:

- From the IDE main menu, select **Galaxy**, then **Configure > Galaxy - Alarms and events**. This opens the **Alarms and events** dialog.
- Select the **States** tab.
- In the **Options for aggregated alarms counters by severity** section, select the checkboxes for the alarm

states to include in the **AlarmCntsBySeverity** totals and clear the checkboxes for those not to include.

Alarms and events

Alarms States

Alarm adorner option

Severities	Alarm states
<input checked="" type="checkbox"/> Critical	<input checked="" type="checkbox"/> UNACK (Un-Acknowledged)
<input checked="" type="checkbox"/> High	<input checked="" type="checkbox"/> ACK (Acknowledged)
<input checked="" type="checkbox"/> Medium	<input checked="" type="checkbox"/> RTN (Return to normal)
<input checked="" type="checkbox"/> Low	

Alarm plant state

Description	AlarmMode
Running	Enable ▾
Maintenance	Disable ▾
Startup	Silence ▾
Shutdown	Disable ▾
Testing	Silence ▾

Options for aggregated alarm counters by severity

States and modes included
<input type="checkbox"/> Inhibited
<input type="checkbox"/> Disabled
<input checked="" type="checkbox"/> Silenced
<input checked="" type="checkbox"/> Shelved

Alarm logic

Alarms Latching

Enable Latching

Local alarm display in the Object Viewer

The Object Viewer shows thirteen values, separated by commas, for the **AlarmCntsBySeverity** attribute. The first twelve values (logically divided into groups of four) show how many alarms of each severity and type have been aggregated for the selected object or area:

Attribute Name	Value	Timestamp	Quality	Status	Security...
Tagname	Platform	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
ShortDesc	The Platform represents ...	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
ScanStateCmnd	true	7/25/2017 9:28:28.225 AM	C:Good	Ok	Operate
ScanState	true	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
SecurityGroup	Default	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
Area	Sys	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
Container	BL07251	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
Host	BL07251	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmMode	Enable	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmModeCmnd	Enable	7/25/2017 9:28:28.225 AM	C:Good	Ok	FreeAccess
AlarmInhibit	false	7/25/2017 9:28:28.225 AM	C:Good	Ok	FreeAccess
InAlarm	true	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
ConfigVersion	7	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
ContainedName		7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
ExecutionRelatedObject		7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
ExecutionRelativeOrder	None	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
HierarchicalName	Platform	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmMostUrgentSeverity	2	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmMostUrgentMode	Enable	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmMostUrgentAcked	1	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmCrtSBySeverity	0,1,0,0,0,1,0,0,0,0,0,0,34	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmMostUrgentInAlarm	true	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmMostUrgentShelved	false	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
AlarmCrtSBySeverityEnableShelved	0,0,0,0	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly
Engine.Historian.ConnectStateCmd	true	7/25/2017 9:28:28.225 AM	C:Good	Ok	Tune
Engine.Historian.ConnectState	false	7/25/2017 9:28:28.225 AM	C:Good	Ok	ReadOnly

1. Number of active alarms, by severity.
2. Number of unacknowledged active alarms, by severity.
3. Number of unacknowledged alarms that have returned to normal, by severity.
4. Bit value that shows which alarm severity and status applies to the local object.

This value is a sum of the bit values shown in the following table:

	ß (LSB) Active (1)				UnAck_ALM (2)				UnAck_RTN (3) (MSB) à			
	S1	S2	S3	S4	S1	S2	S3	S4	S1	S2	S3	S4
Bit value	1	2	4	8	16	32	64	128	256	512	1024	2048

In the Object Viewer example shown above, the local object bit value (4) is 34. This is the sum of the active S2 bit value plus the UnAck_ALM S2 bit value. Other examples:

- If the local object has one or more active S1 alarms that are unacknowledged, the bit value is 17 (S1 Active Alarm = 1; S1 UnAck_ALM = 16).
- If the local object has one or more active S4 alarms that are UnAck_RTN, the bit value is 2056 (S4 Active Alarm = 8; S4 UnAck_RTN = 2048).
- If there are no local alarms, the bit value is 0.
- If there are multiple local alarms with different severities and unack statuses, the bit value is the sum of each bit that is set. For example, if there are two unacknowledged S1 alarms, but one has returned to normal, the bit value is 273 (two S1 Active Alarms = 1; one S1 UnAck_ALM = 16; one S1 UnAck_RTN = 256). Note that

each bit that is set is only counted once, even if more than one alarm exists at that severity or status.

When alarm aggregation is shown on a navigation item in an AVEVA OMI ViewApp, the bit value is used by the navigation item to determine the alarm location (selected object or a contained object). See [Aggregated alarm display in OMI ViewApp navigation](#) for additional information.

Note: By default, **AlarmCntsBySeverity** includes shelved and silenced alarms, but does not include hidden or disabled alarms. You can change this by changing the **AggregatedAlarmStatesIncluded** filter. For more information, see [AggregatedAlarmStatesIncluded attribute](#)

Aggregated alarm display in OMI ViewApp navigation

The **Alarms and events** dialog is used to configure which aggregated alarms are displayed on a Galaxy-wide basis. You can set whether or not display alarm information in individual AVEVA OMI ViewApps by enabling/disabling the **ShowAlarms** attribute. AVEVA OMI ViewApps display alarm indicators and alarm counts for aggregated alarms on NavigationApp controls (NavigationTree and BreadCrumbControl), if the **ShowAlarms** attribute is enabled in the ViewApp. To enable this attribute, select the **Show Alarms** option when you configure the control in the **Layout Editor**.

Note: Alarm indicators ("adorners") used in AVEVA OMI navigation are FrameworkElement Adorners, part of the Microsoft .NET Framework used in creating and configuring apps for AVEVA OMI ViewApps.

The opacity of the alarm indicator shows if the alarm is on the selected navigation node or on a contained object. When enabled, the following alarm severities and alarm statuses are shown:

- If the color of the alarm indicator is opaque (solid), the alarm is on the selected (local) object. If enabled, the severity (S1 to S4) and status of the alarm (UNACK_ALM, UNACK_RTN, ACK_ALM) is shown graphically, along with a count for each alarm severity that has been enabled.
- If the color of the alarm indicator has some transparency, the alarm is on a contained object.
- The color of the blinking line under the object name shows the highest-severity (most urgent) alarm.
- An indicator on the NavigationTree or BreadCrumbControl shows if the data quality of the object is bad. Bad quality may result if communication has been lost with the object.

You can also determine which navigation node or asset contains the alarm by selecting the alarm icon. This displays a tooltip that indicates the node(s) or asset(s) sending the alarm(s).

Set the **Alarm adorner option** in the **Alarms and events** dialog (**Galaxy > Configure > Galaxy - Alarms and events**, then select the **States** tab). This configures which aggregated alarms are displayed on NavigationApp controls and any custom apps that subscribe to the aggregated alarm count during runtime.

Alarms and events

Alarms States

Alarm adorner option

Severities	Alarm states
<input checked="" type="checkbox"/> Critical	<input checked="" type="checkbox"/> UNACK (Un-Acknowledged)
<input checked="" type="checkbox"/> High	<input checked="" type="checkbox"/> ACK (Acknowledged)
<input checked="" type="checkbox"/> Medium	<input checked="" type="checkbox"/> RTN (Return to normal)
<input checked="" type="checkbox"/> Low	

+
-

Description	AlarmMode
Running	Enable ▾
Maintenance	Disable ▾
Startup	Silence ▾
Shutdown	Disable ▾
Testing	Silence ▾

Options for aggregated alarm counters by severity

States and modes included	
<input type="checkbox"/> Inhibited	<input type="checkbox"/> Disabled
<input checked="" type="checkbox"/> Silenced	<input checked="" type="checkbox"/> Shelved

Alarm logic

Alarms Latching	
<input type="checkbox"/> Enable Latching	

The following **Alarm Adorner Options** are set through the dialog. The corresponding attribute settings are propagated to runtime nodes (all attribute names are prefixed by MyViewApp.Alarms). For additional information about these attributes, see [Alarm adorner attributes](#).

Alarm severities to display	Attribute name (alarms namespace)	Description
Critical	ShowCriticalAlarms	When enabled, the ViewApp displays critical alarms
High	ShowHighAlarms	When enabled, the ViewApp displays high alarms
Medium	ShowMediumAlarms	When enabled, the ViewApp displays medium alarms
Low	ShowLowAlarms	When enabled, the ViewApp displays low alarms
Alarm states to display	Attribute name (alarms namespace)	Description
UNACK (Unacknowledged)	ShowUnacknowledgedAlarm	When enabled, the ViewApp

		displays UNACK alarms
ACK (Acknowledged)	ShowAcknowledgedAlarms	When enabled, the ViewApp displays ACK alarms
RTN (Return to Normal)	ShowReturnToNormalAlarms	When enabled, the ViewApp displays RTN alarms
LATCHED	ShowLatchedAlarms	Latched alarm state must be enabled for the Galaxy for Latched alarms to show. When enabled, the ViewApp displays Latched alarms.

Alarm indicator appearance

The appearance of alarm indicators is configured through the **Alarm Element** tab of the **Galaxy Styles** page.

Standard_Style (Default)

The screenshot shows the 'Galaxy Styles' page with the 'Alarm Element' tab selected. The page is organized into a grid of 16 smaller grids, each representing a different alarm indicator style. Each small grid contains a preview of the indicator, a 'Sample text' button, and a set of edit icons. The styles are grouped into four rows:

- Row 1:** Alarm_Critical_ACK (red), Alarm_Critical_RTN (red with diagonal lines), Alarm_Critical_UNACK (red), Alarm_High_ACK (yellow).
- Row 2:** Alarm_High_UNACK (yellow border), Alarm_Inhibited (grey), Alarm_Low_ACK (magenta), Alarm_Low_RTN (magenta with diagonal lines).
- Row 3:** Alarm_Med_RTN (cyan with diagonal lines), Alarm_Medium_ACK (cyan), Alarm_Medium_UNACK (cyan), Alarm_Shelved (orange).
- Row 4:** AlarmBorder_Critical (grey), AlarmBorder_Critical (grey), AlarmBorder_Critical (grey), AlarmBorder_High (orange).

Alarm adorner attributes

Alarm Adorner attributes are available only for use in AVEVA OMI ViewApps. All Alarm Adorner attributes operate in the `MyViewApp.Alarms` namespace. Attributes for the Alarm Adorner are specified with the `MyViewApp.Alarms` prefix in the form `MyViewApp.Alarms.attribute_name`. Alarm Adorner attributes cannot be

used in an InTouch application because the MyViewApp.Alarms prefix is regarded as a configuration error.

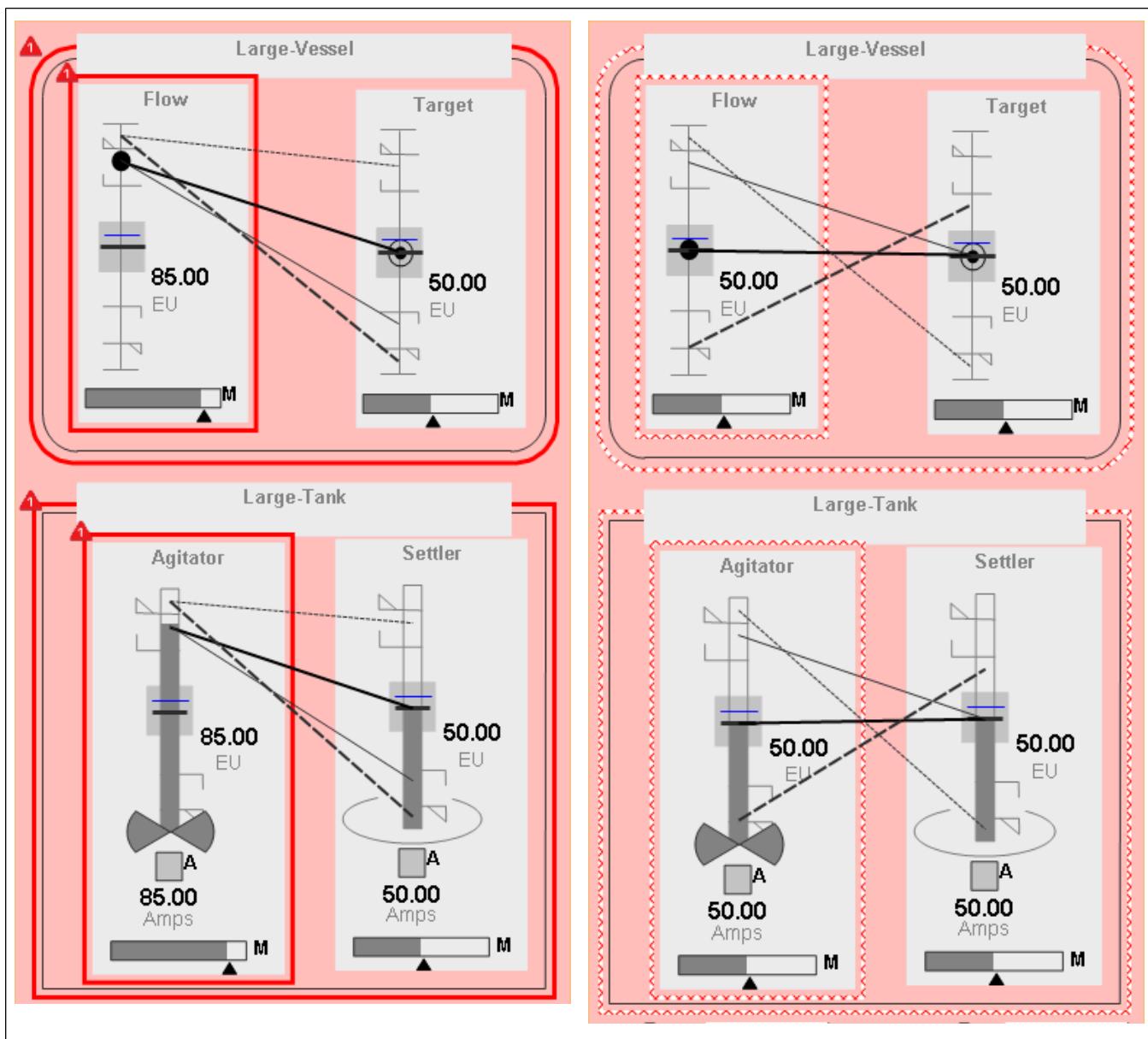
Attribute name	Data type	Attribute type	Default value	Description
ShowCriticalAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays critical alarms
ShowHighAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays high alarms
ShowMediumAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays medium alarms
ShowLowAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays low alarms
ShowUnackedAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays UNACK alarms
ShowAckedAlarms	Boolean	Read/ Write	True	If True, the ViewApp displays ACK alarms
ShowReturnToNormal Alarms	Boolean	Read/ Write	True	If True, the ViewApp displays RTN alarms

Monitor alarm state information at runtime

You can obtain runtime alarm state information, including aggregated alarm information, using the Object Viewer or InTouch Tag Viewer. You can visualize the objects in an AVEVA OMI ViewApp or InTouch HMI application using the AlarmApp for AVEVA OMI or Alarm Client for InTouch HMI, along with the alarm border animation to represent the aggregated alarms at each level of containment. You can also enable aggregated alarm information for the NavTree

Note: AlarmApp is not supported in InTouch HMI applications. Similarly, AlarmClient is not supported in AVEVA OMI ViewApps. Use AlarmApp for AVEVA OMI ViewApps to allow alarm aggregation to follow selected objects and areas.

An alarm border animation applied to a group of graphics can indicate a priority 1 alarm with a red border, flashing or solid for different alarms. The alarm border animation can also indicate a return to normal (RTN) for the same graphics.



Alarm clients configured in InTouch applications will display the severity of each alarm in the **User1** field. You can change "User1" to a more descriptive heading, such as "Severity".

For information about configuring alarm animations, see "Configuring an Alarm Border Animation" in the *Creating and Managing Industrial Graphics User Guide*.

About graphic elements

This topic explains how to work with the common features of graphic elements.

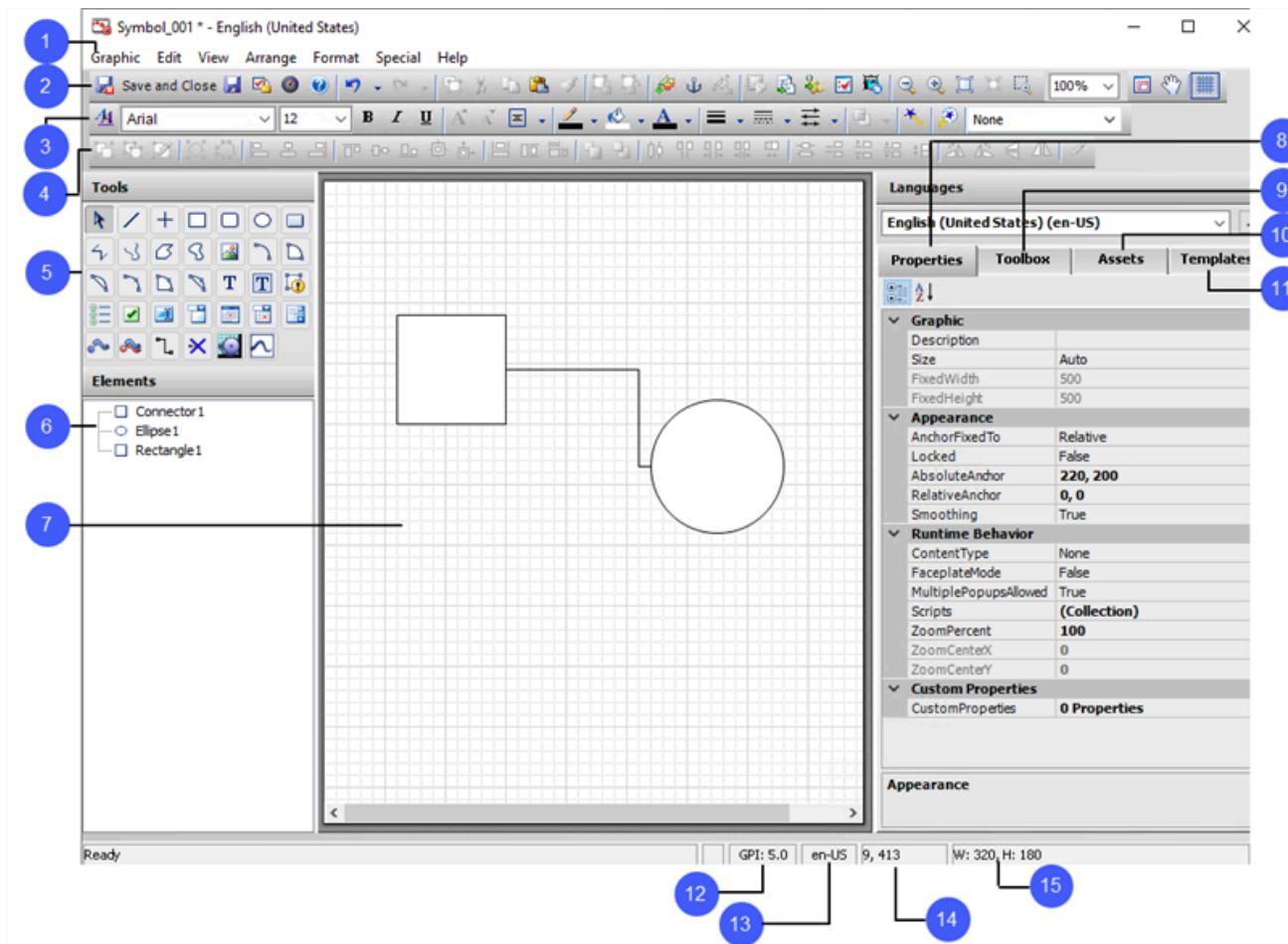
Graphic elements are basic shapes and controls you can use to create a graphic. You can:

- Draw an element by selecting an element from the **Tools** panel, placing it on the canvas, and then configuring its properties.
- Select one or more elements on the canvas with the mouse or from the **Elements** list.

- Edit certain elements in a special way called inline editing.
- Copy, cut, paste, and duplicate elements.
- Move elements around on the canvas.
- Align elements to each other.
- Change the spacing between elements.
- Resize elements.
- Change the z-order of elements to change which elements appear on top of others when they overlap.
- Rotate elements.
- Change the origin of elements to specify around which point the elements are rotated.
- Flip elements on their horizontal or vertical axis.
- Lock elements to stop them being moved or changed.
- Undo and redo any number of changes made previously to the graphic.
- Create groups of elements to bind them together.
- Create a path graphic from multiple open line elements.
- Import an SVG (Scalable Vector Graphic) graphic.

About the Graphic Editor

The figure below shows the various components of the Industrial Graphic Editor to create and configure graphic elements and graphics. You check a graphic and then double-click it to begin editing. The Industrial Graphic Editor opens and shows the graphic you selected on the canvas area.



1	The Menu Bar shows options with drop-down lists of commands to modify the graphic on the canvas.
2	The Command bar includes commands to save, save and close, and close the Industrial Graphic Editor.
3	The Configuration bar includes options to configure the visual appearance of text and lines that are part of the graphic on the canvas.
4	The Placement bar includes options to align selected graphic elements or graphics on the canvas.
5	The Tools pane contains a set of icons that represent graphic elements. A graphic element is selected from the Tools pane and then dragged and dropped onto the canvas for editing.
6	The Elements area lists the current set of graphic elements or graphics on the canvas in a hierarchical view.

7	The Canvas area shows graphic elements or other content that belong to the graphic currently being edited.
8	The Properties grid lists the Graphic, Appearance, Runtime Behavior, and Custom Properties of the graphic element or graphic selected on the Canvas. Each property contains a data entry field to assign a value to the property.
9	The Toolbox tab includes a set of folders containing screen profiles, layouts, graphics, apps, and ViewApp namespaces.
10	The Assets tab shows the derived instances of objects listed in the Model view of a ViewApp.
11	The Templates tab shows the contents of the Template folder .
12	The GPI field shows the calculated Graphic Performance Index of the graphic currently being edited. The GPI is a value from 0 to 5 that indicates load-time performance of each piece of content that belongs to a graphic.
13	The Language field shows the languages configured for the selected graphic on the canvas.
14	Current cursor position by pixel width and height from the origin of the canvas (0,0), which is the top left corner of the canvas area.
15	Width and height in pixels of the selected graphic element on the canvas.

About the graphic element property commonalities

Some properties are common to most graphic elements, such as fill, line styles, and visibility. You can:

- Edit the name of an element.
- Edit the fill properties of an element.
- Edit the line properties of an element.
- Edit the text properties of an element.
- Set the style.
- Set the transparency level of an element.
- Tweaking colors and style for an element's gradient style.

- Enable and disable elements for runtime interaction.
- Change the visibility of an element.
- Change the tab order of an element.
- Use the Format Painter to format elements.
- Edit the general properties of a graphic.

About graphic-specific and element-specific properties

You can configure graphic-specific and element-specific properties.

You can configure:

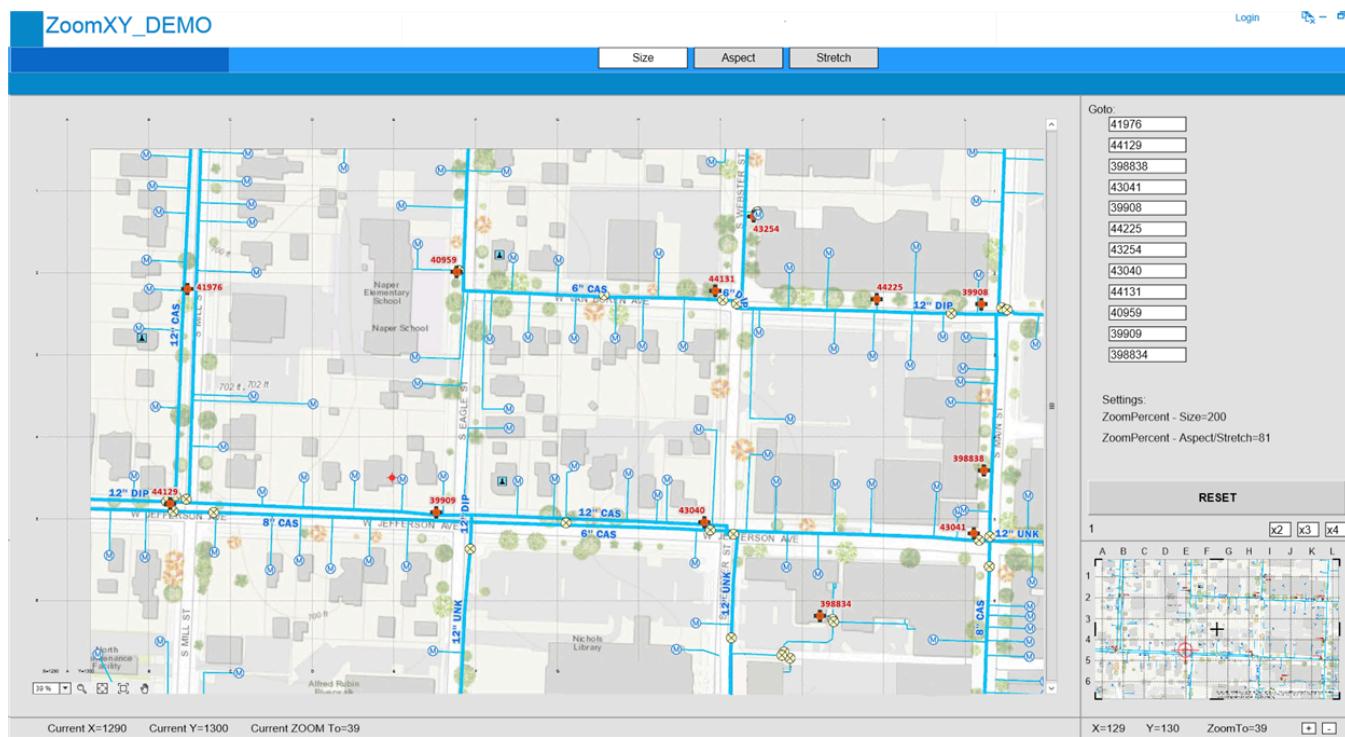
- General properties of a graphic.
- Radius of rounded rectangles.
- Shape and end appearance of lines and H/V lines.
- Auto-sizing and word-wrapping in text boxes.
- Image-specific properties.
- Button-specific properties.
- Control points and tension in curves.
- Angles in pies, chords, and arcs.
- Status elements.
- Windows common controls.

For more information about these topics, see [Edit graphic-specific and element-specific properties](#) in the Industrial Graphic Editor help.

About zoom percentage

Graphics shown in a running ViewApp can be zoomed in to show more detail using scripting. Instead of zooming in from the existing center point of the pane, a script can specify a zoom area center point within the boundaries of a graphic. The ability to place the zoom area center point anywhere within the boundaries of a graphic restricts the zoom area to a specific portion of the graphic to show more detail.

The following graphic shows a graphic of a map within a running ViewApp. The red crosses represent the center points of the different zoom areas of the graphic. The buttons shown at the right of the graphic include action scripts that set XY coordinate center points of the zoom areas. When an operator selects a button, the graphic is moved within the pane to set the selected zoom area to the center of the pane. The map can be zoomed in or out.



Set the zoom percentage

There are several distinct tasks that must be completed to enable programmatic zooming in a ViewApp. The following list shows the recommended order these tasks should be completed.

1. Determine the paired X and Y coordinate values that will serve as the center points of the programmatic zoom areas of the graphic.
2. Assign values to layout pane properties that affect programmatic zooming. See [About Layout pane property effects on zoom behavior](#).
3. Create a set of attributes with the Namespace Editor that will be used in scripts and animation. See [Set zoom percentage attributes](#).
4. Write a set of scripts and assign animations to enable programmatic zooming. See [Minimal set of scripts and animations to enable programmatic zooming of graphics](#)

Specify Zoom Center Point Coordinates of a Graphic

Before you create scripts and animations to set programmatic zooming of a graphic, you must determine the different X,Y coordinates within a graphic that you want to assign as the center points of the zoom areas.

To determine the zoom center point coordinates of a graphic

1. Open the graphic in the Graphic editor.
2. Place the mouse cursor directly over the point in the graphic that you want to set as the center point of a zoom area.

The current X and Y cursor values of the cursor location are shown in the status bar beneath the canvas area of the Graphic editor.

498, 1120

3. Save the coordinates of each location to specify them in action scripts.

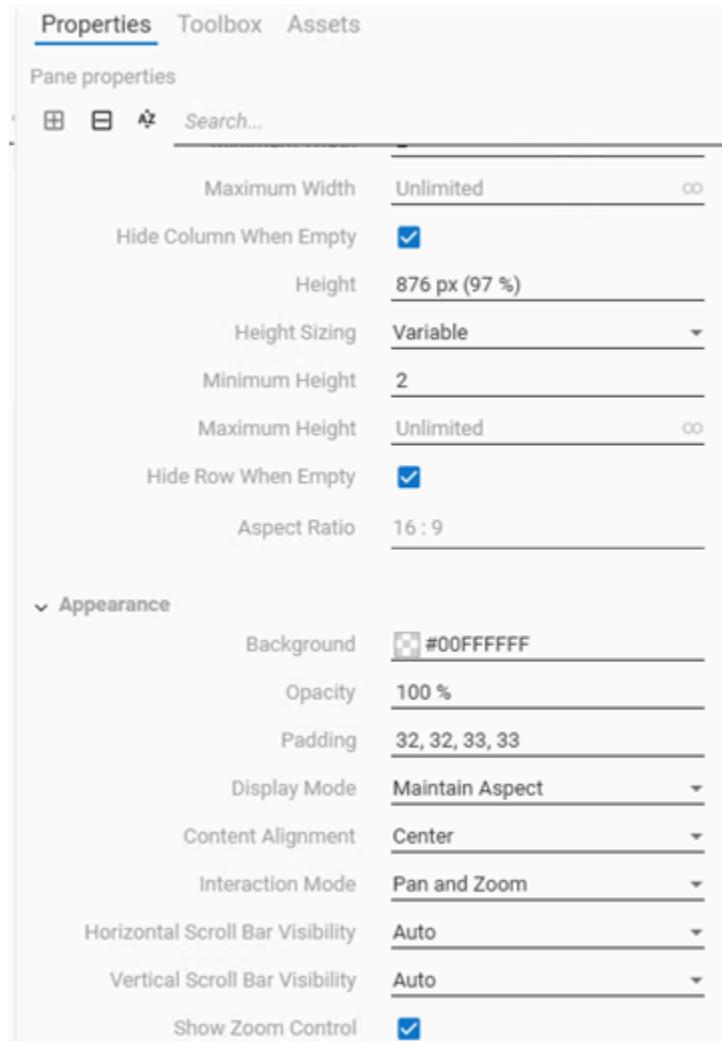
About Layout pane property effects on zoom behavior

Layout pane properties specified during design time determine some aspects of zoom behavior during runtime.

- The values assigned to the **Display Mode** property determine the placement of the graphic within a pane when a graphic is zoomed larger than the dimensions of a pane and whether a graphic retains its original aspect ratio.

Important: The zoom percent range varies based on the value assigned to the **Display Mode** property. When **Display Mode** is set to **Maintain Size**, the zoom percent range is from 10 percent to 5000 percent. When **Display Mode** is set to **Maintain Aspect** or **Stretch**, the zoom percent range is from 100 percent to 5000 percent.

- The values assigned to the **Content Alignment** property determines the placement of graphic within a pane when zoomed larger than a pane in at least one dimension.
- Panning and zooming should be enabled for a pane and show the pan and zoom control icons during runtime.



The following table lists important pane properties that will affect programmatic zoom control of graphics.

Pane Property	Description
Display Mode	<p>Assigned value determines how a graphic is resized to fit the current pane size. For more detail about the different Display Mode values, see Set pane properties: display mode.</p> <ul style="list-style-type: none"> • Maintain Aspect is the default value to retain the aspect ratio of the graphic when it is zoomed larger or smaller. A graphic's left and right boundaries will align with the pane or its top and bottom boundaries will align with the pane. A graphic is truncated in any dimension that is larger than the pane's width or height. • Maintain Size places a graphic within the pane based on the assigned Alignment and Padding property values when the content is smaller than the pane. When the size of a graphic exceeds the pane's boundary, the graphic retains its aspect ratio and is centered in the graphic and truncated in any dimension beyond the pane boundary. • Stretch resizes the graphic regardless of its aspect ratio to completely fit within the dimensions of the pane.
Content Alignment	<p>Assigned value determines the alignment of the graphic within a pane area. For more information about the different Content Alignment values, see Set pane properties: display mode. Center is the default to place the center of the graphic at the center of the pane area.</p>
Interaction Mode	<p>Assigned values determines whether the user can pan or zoom the pane area. Pan and Zoom is the default to enable panning and zooming in a pane during runtime.</p>
Show Zoom Control	<p>Assigned value determines if the pan and zoom controls are shown in a pane during runtime. The recommended value is to show both controls during runtime.</p>

Set zoom percentage attributes

The following figure shows an example set of attributes created with the Namespace editor to control zoom behavior with scripts. In this example, the MyZoom Galaxy namespace has been created to contain the zoom attributes.

These attributes can be used in action scripts and a variety of animations to select a zoom level and center point of a zoom area during runtime.

TrendApp						
		Search... <input type="text"/> <input type="button"/>				
Name ↑	Type	Initial Value	Readonly	Retentive	Description	
CurrentX	Integer	0	<input type="checkbox"/>	<input type="checkbox"/>	Current ZoomX in full size	
CurrentY	Integer	0	<input type="checkbox"/>	<input type="checkbox"/>	Current ZoomY in full size	
CurrentZoom	Integer	0	<input type="checkbox"/>	<input type="checkbox"/>	Current Zoom level in full size	
Mode	Integer	0	<input type="checkbox"/>	<input type="checkbox"/>	1=Size,2=Aspect,3=Stretch	
X	Integer	0	<input type="checkbox"/>	<input type="checkbox"/>	Request ZoomX	
Y	Integer	0	<input type="checkbox"/>	<input type="checkbox"/>	Request ZoomY	
ZoomAspectStretch	Integer	81	<input type="checkbox"/>	<input type="checkbox"/>		
ZoomSize	Integer	200	<input type="checkbox"/>	<input type="checkbox"/>		
ZoomTo	Integer	-1	<input type="checkbox"/>	<input type="checkbox"/>	Request Zoom level from mini	

0 of 10 selected. 10 visible.

The following table describes a suggested set of attributes that provide programmatic zoom levels based on the example attributes shown in the figure above.

Attribute	Script or Animation	Purpose
CurrentX	Value Display animation	An integer value that shows the current X value of the center point of a zoom area.
CurrentY	Value Display animation	An integer value that shows the current Y value of the center point of a zoom area.
CurrentZoom	Value Display animation	An integer values that shows the current zoom level.
Mode	Action Script	<p>An integer value that specifies the pane's display mode.</p> <ul style="list-style-type: none"> • Maintain <code>Size=MyViewApp.MyZoom.Mode ==1</code>

		<ul style="list-style-type: none"> Maintain <code>Aspect=MyViewApp.MyZoom.Mode==2</code> Stretch=<code>MyViewApp.MyZoom.Mode==3</code>
X	Action Script	An integer value that sets the X coordinate value of the center point of the zoom area.
Y	Action Script	An integer value that sets the Y coordinate value of the center point of the zoom area.
ZoomAspect Stretch	Action Script	An integer that sets the zoom percentage when the value of the Mode attribute is set to Maintain Aspect or Stretch .
ZoomSize	Scripts and User Input animation	An integer that sets the zoom percentage when the value of the Mode attribute is set to Maintain Size .
ZoomTo	Scripts and User Input animation	<p>An integer that sets the zoom percentage.</p> <p>The zoom range minimum or maximum percentage values can be specified by using ZoomTo in an animation expression.</p> <p><code>MyViewApp.MyZoom.ZoomTo<10</code> <code>MyViewApp.MyZoom.ZoomTo>4999</code></p> <p>Zoom in and zoom out buttons are disabled when the ZoomTo attribute expression is true.</p>

For more information about using these attributes in scripts and animation, see [Minimal set of scripts and animations to enable programmatic zooming of graphics](#).

Minimal set of scripts and animations to enable programmatic zooming of graphics

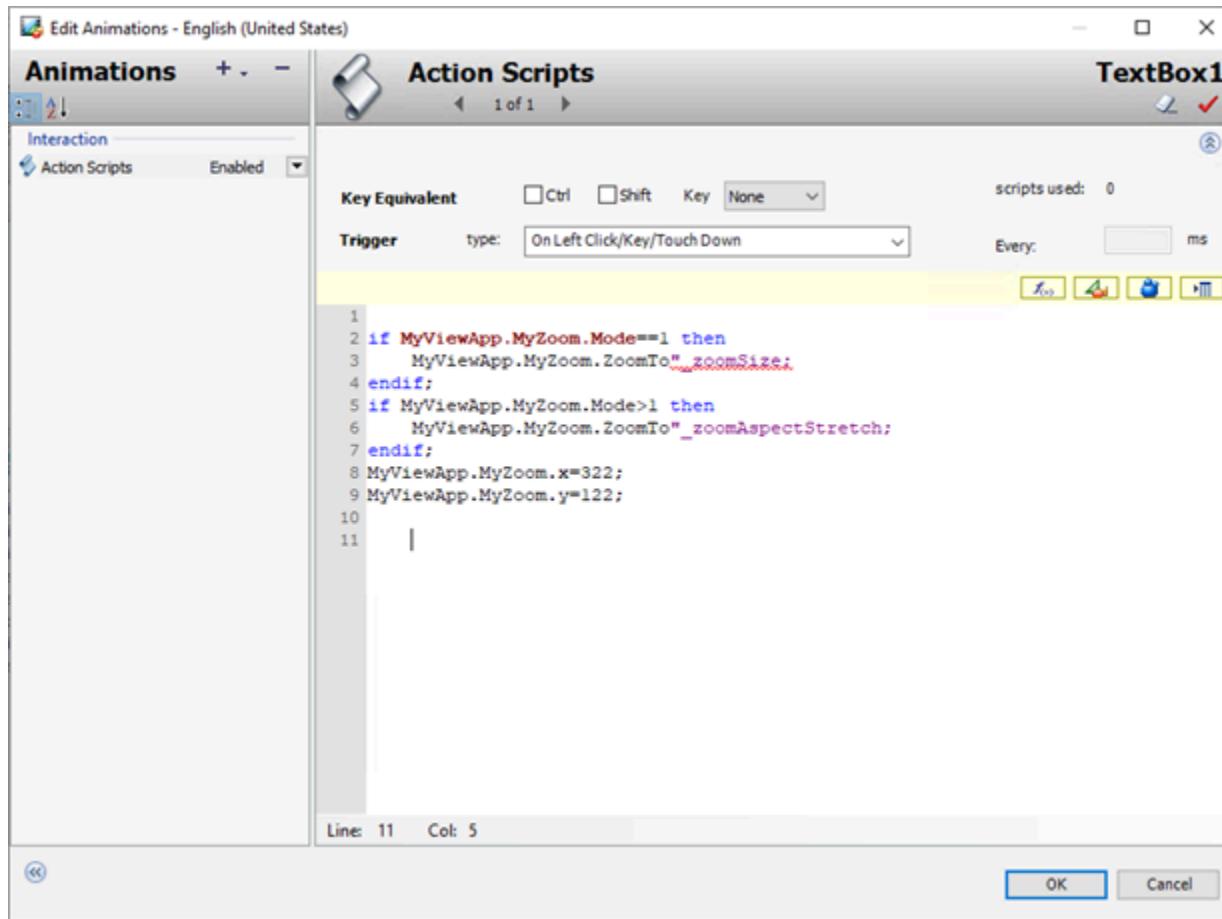
As part of building a ViewApp that incorporates user controls to resize graphics during runtime, scripting and animation must be associated with the user controls.

Note: This topic describes a minimal set of scripts and animations to enable programmatic zooming of graphics. More complex scripting can be used to provide additional zoom features or displays.

Create an Action Script for Each Zoom Center Point

The following figure shows an example of an action script to set the center point of a zoom area on a graphic. In this example, the script determines the current display mode of the pane and then sets a zoom percentage. The script then sets the X and Y coordinates of a center point of the zoom area that has been previously mapped on the graphic.

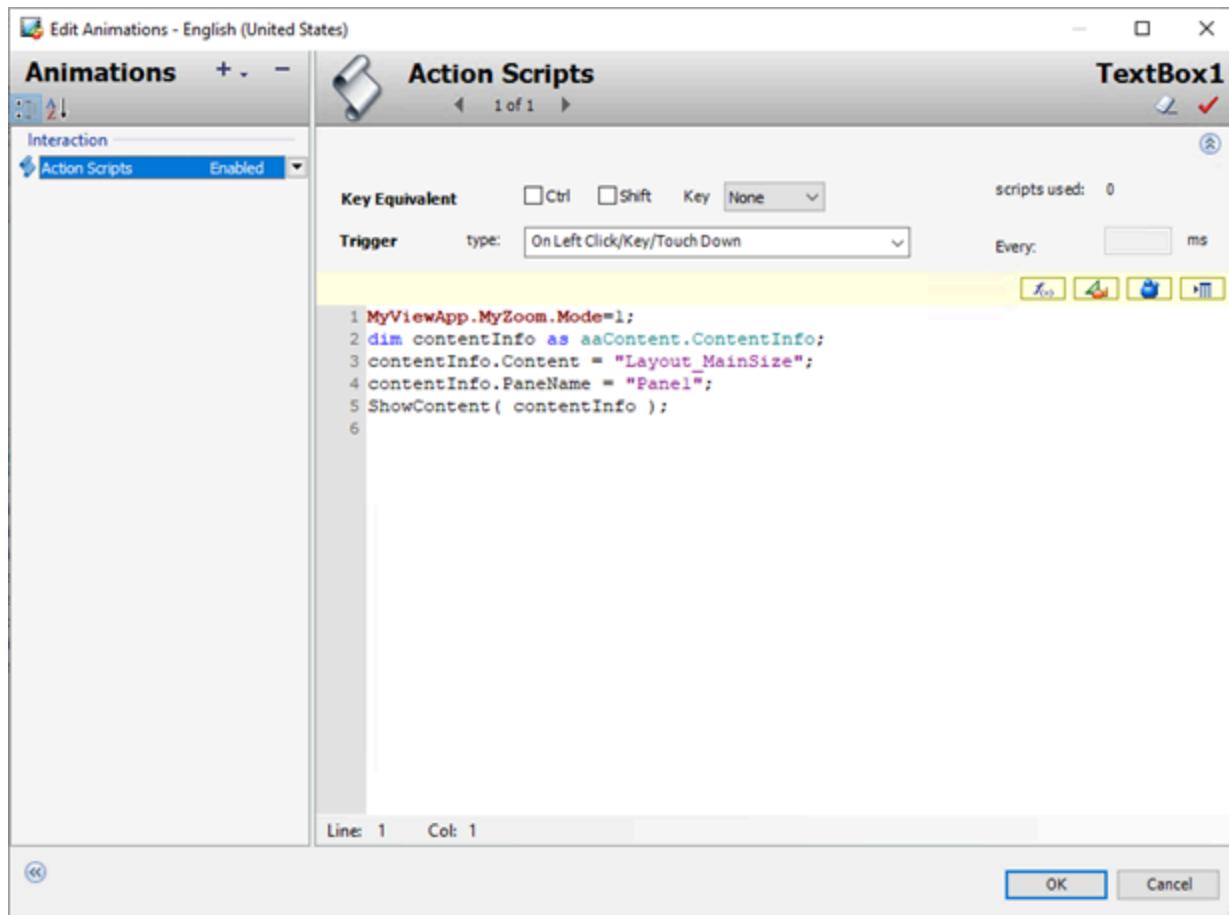
Typically, the action script is associated with a button that the user selects to move to the center point of the zoom area of the graphic.



Important: In scripts, the attribute that sets the zoom percentage value must be declared before the attributes that specify the X and Y coordinates of the zoom area. The X and Y coordinates can be specified in any order.

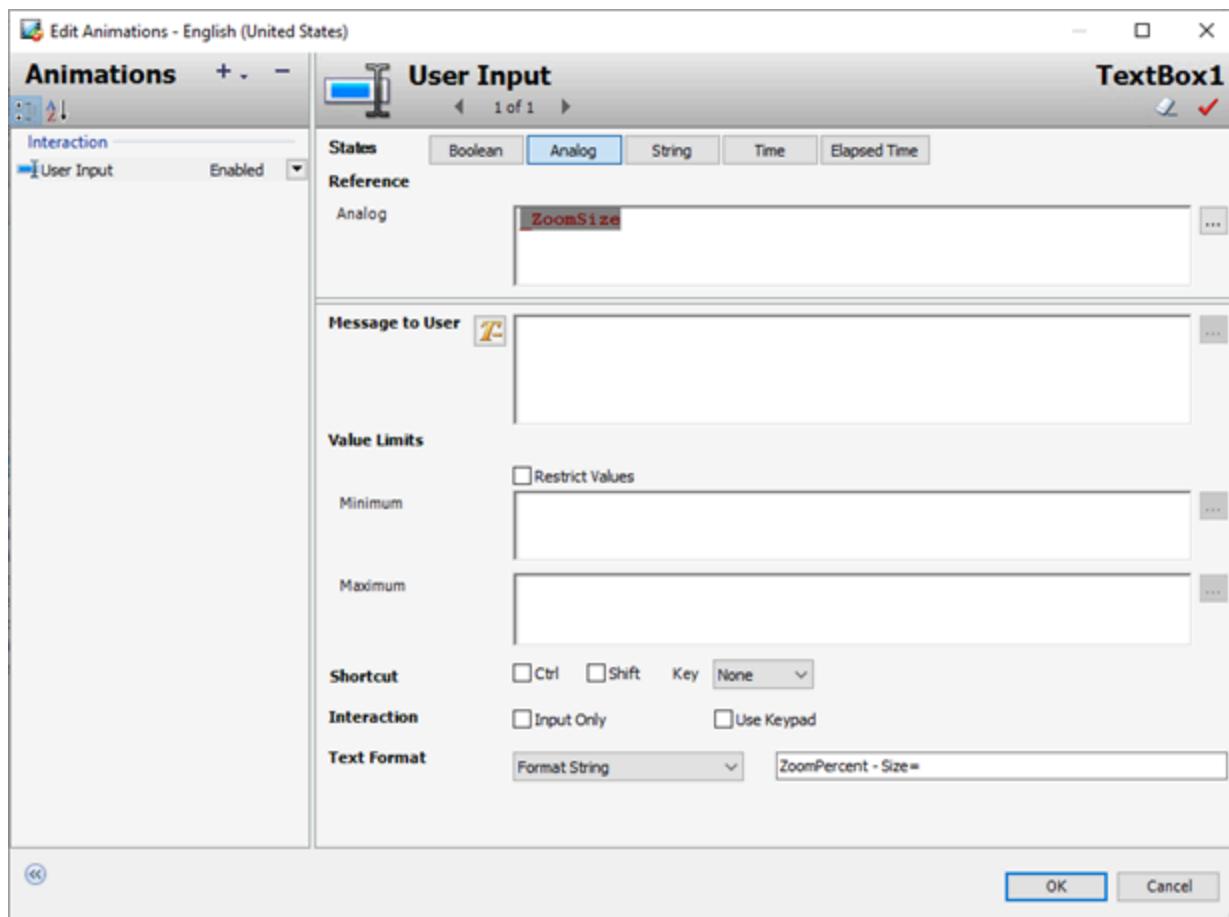
Create an Action Script to Select Different Layouts Based on the Display Mode Property Value

This type of script is optional. If you want to change layouts containing different graphics during runtime based on a selected Display Mode, create an action script that includes the ShowContent method. In this example, a different layout is used for each possible Display Mode property value, which are associated with buttons.



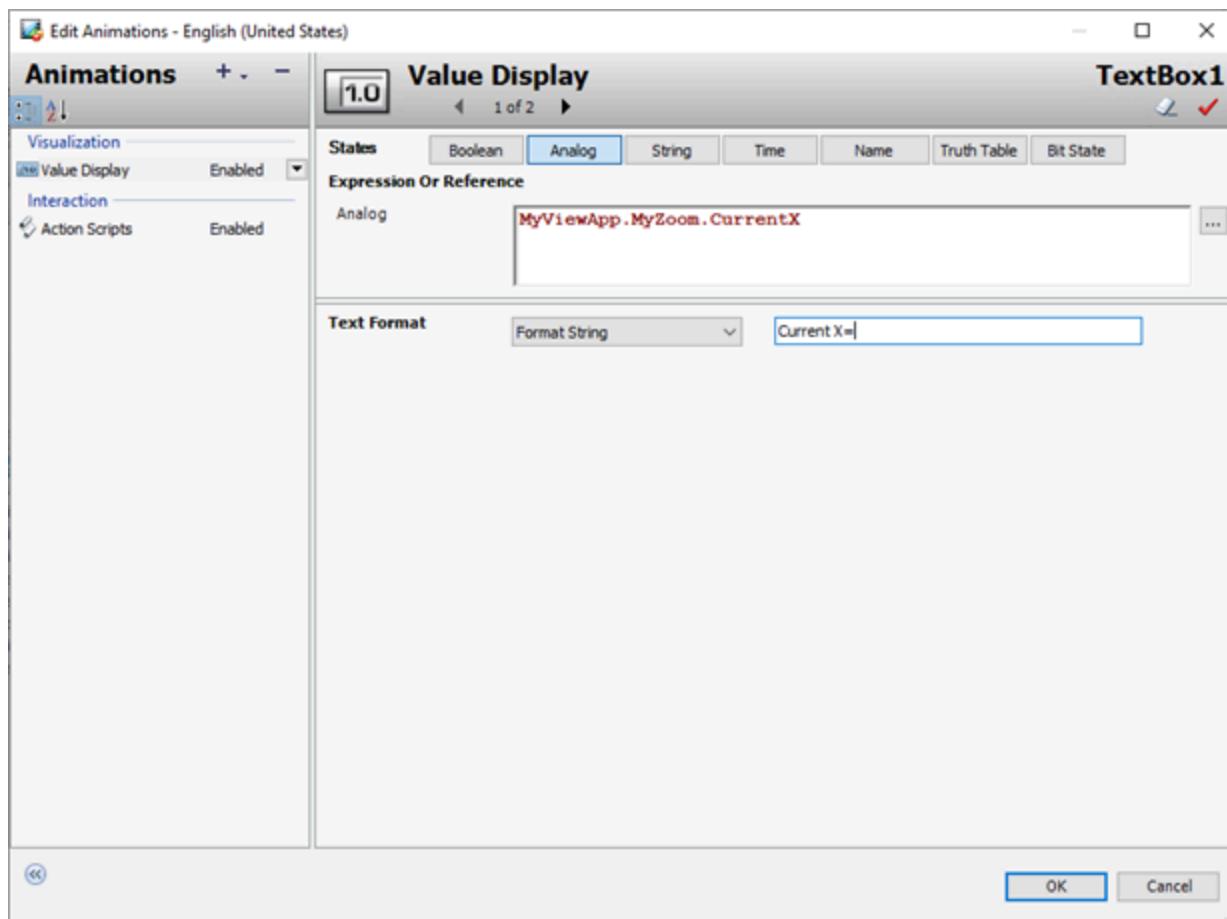
Create User Input Animation to Set Zoom Values

The following figure shows an example of User Input animation to set the zoom level of a graphic. In this example, the user selects a data entry field during runtime.



Create Value Display Animation to Show Current Graphic Coordinates

You may want to show the current zoom area coordinates of your graphic during runtime. The following figure shows an example of Value Display animation that shows the current X coordinate value of the center point of a zoom area. You can also use Value Display animation to show the current zoom level of the graphic.



Monitor and show quality and status

You can configure your graphic to show non-good status and quality of attributes or tags in different ways:

- A status element shows a specific icon depending on the quality and status of configured attributes, tags or elements.
- The text, fill, or line appearance of elements is overridden depending on the quality and status of the attributes and tags they reference.
- Elements are drawn with an outline depending on the quality and status of the attributes they reference.

Quality and status elements might not be supported by all HMIs. Refer to "Working with the Industrial Graphic Editor" in your HMI help for more information.

Use status elements

Status elements show a specified graphic depending on the quality and status of:

- Attributes and tags configured for specific animated elements.
- One or more specified attributes or tags.

You can assign status elements to an animation in three steps:

1. Draw the status element on the canvas.
2. Associate the status element with animated elements on the canvas and/or attributes that provide the quality and status data to be monitored.
3. If needed, configure the appearance of the status element.

Draw the Status Element on the Canvas

You can easily place a status element on the canvas to show an icon that indicates quality and status of attributes or tags contained in selected animated elements and/or specified attributes and tags.

You do this as you would with any other element. For more information, see [Drawing and Dragging Elements](#).

Configure the Status Element

You can associate the status element with:

- Animated elements that use attributes and tags that provide the quality and status that is to be monitored.
- Attributes and tags that provide the quality and status to be monitored.

In both cases, the appearance is set by the settings in your application. For example, in AVEVA ViewApps it is set in the **Quality and Status** tab of the **Configure Galaxy Style Library** dialog box.

For more information on how to configure this animation, see [Configuring Animation for a Status Element](#).

Set the Appearance of a Status Element

You can set the appearance of a status element depending on the quality and status of its referenced attributes and tags and/or attributes and tags used in its referenced elements.

You can also preview the appearance of a status element. For more information, see [Previewing all Status Appearances](#).

You can reset the appearance of a status element to its default. For more information, see [Reset an Override Appearance to its Default](#).

Override Element Appearance in Accordance with the Quality and Status of its Attributes

You can configure any animated element to appear differently depending on the quality and status of its associated attributes and tags if your HMI application supports it.

For animated elements, you can:

- Override the appearance of the text font, style, and blinking.
- Override the appearance of the fill style and blinking.
- Override the appearance of the line style, weight, pattern, and blinking.
- Preview all status appearances in one dialog box.
- Reset the status appearances to their defaults.

- Use an outline to indicate a specified status or quality.

Note: Instead of overriding the appearance of elements on the canvas, you can use a status element. The status element shows an icon representing quality and status of monitored attributes and tags.

Override the Text Appearance of Elements to Indicate Non-Good Status or Quality

If your application supports it, you can configure the text appearance of animated elements with attributes that have non-good status or quality are overridden with a specific text appearance.

Override the Fill Appearance of Elements to Indicate Non-Good Status or Quality

If your HMI application supports it, you can configure the fill appearance of animated elements with attributes that have non-good status or quality are overridden with a specific fill appearance.

Override the Line Appearance of Elements to Indicate Non-Good Status or Quality

You can configure a Galaxy so that the line appearance of animated elements with attributes that have non-good status or quality are overridden with a specific line appearance.

To override the line appearance of elements specified by a Status element

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select **Enable Quality and Status Display**.
5. Select the edit icon  for a status or quality.
6. Click the **Line** tab.
7. To override the line pattern, select the **Line Pattern** checkbox and select a line pattern from the adjacent list.
8. To override the line weight, select the **Line Weight** checkbox and type a new line weight in the adjacent box. Alternatively, you can increment or decrement the line weight by clicking the adjacent boxes.
9. To override the line color:
 - a. Select **Line Color** checkbox.
 - b. Click the color box.
 - c. Select a line color from the **Select Line Color** dialog box. For more information, see [Setting Style](#).
10. To override the line blink behavior:
 - a. Select **Blink**.
 - b. Select a blinking speed from the **Speed** list (Slow, Medium, or Fast).
 - c. Click the color box.
 - d. Select a line blink color from the **Select Blink Color** dialog box. For more information, see [Setting Style](#).
11. Click **Apply**.

Add Outlines to Elements to Indicate Non-Good Status or Quality

You can configure the Galaxy so that animated elements with attributes that have non-good status or quality are shown with an outline.

To add outlines to elements to indicate non-good status or quality

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select **Enable Quality and Status Display**.
5. Click the **Outline (OI)** tab.
6. Select the edit icon  for a status or quality.
7. Select **Show Outline**.
8. To set the line pattern, select **Line Pattern** and select a line pattern from the dropdown list.
9. To set the line weight, select **Line Weight** and enter a line weight. Alternatively, you can increment or decrement the line weight by clicking the adjacent boxes.
10. To set the line style:
 - a. Click the color box next to **Line Color**.
 - b. Select a line color from the **Select Line Color** dialog box. For more information, see [Setting Style](#).
11. To set the line blink behavior:
 - a. Select **Blink**.
 - b. Select a blinking speed from the **Speed** list (Slow, Medium, or Fast).
 - c. Click the color box.
 - d. Select a line blink color from the **Select Blink Color** dialog box. For more information, see [Setting Style](#).
12. Click **Apply**.

Preview all Status Appearances

You can preview the appearance of all status overrides by showing the **Status Legend** dialog box.

To preview all override appearances

1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Click **Preview Legend**. The **Status Legend** dialog box appears.
5. Click **Close**.

Reset an Override Appearance to its Default

For any status, you can reset the default appearance:

- Text, fill, and line overrides.

- Outline settings.
- Status element settings.

To reset a status or quality to its default appearance

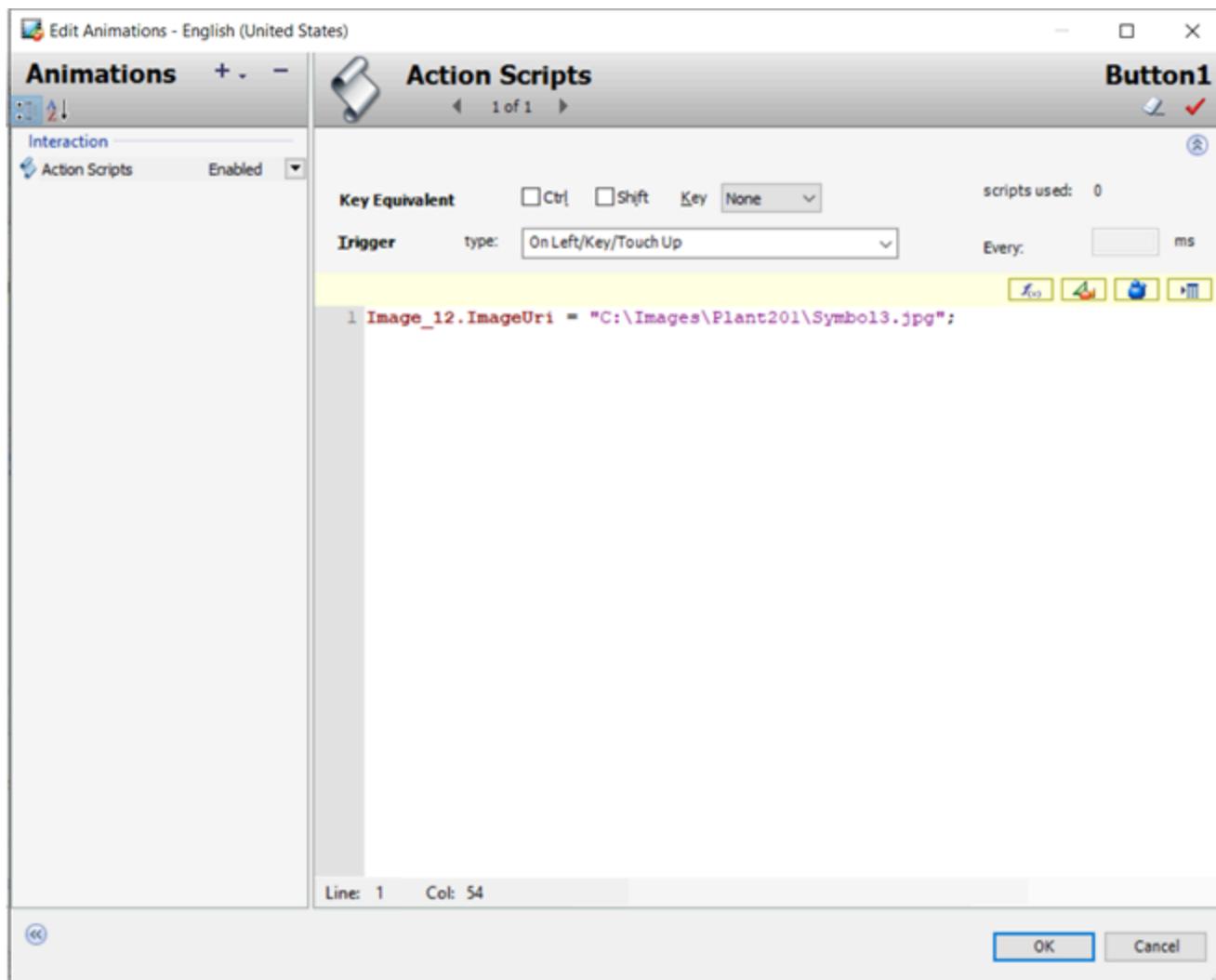
1. On the IDE ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Styles**. The **Style Library** opens.
3. Select the **Quality and Status** tab.
4. Select the edit icon  for a status or quality.
5. Click **Reset to default**. All text, fill, and line overrides, status element icons, and outline settings are reset to their defaults.

Change images at runtime

You can use the `ImageUri` property in an image scripts to change an image at runtime through scripting. This property is only available for image scripts. The syntax is:

`<owningImage>.ImageUri = <path to image to be shown>`

See the example below.



Use animation

You can use animations to change the appearance of graphic elements at runtime. Animations are driven by data that comes from attribute values and expressions as well as element properties.

You can use:

- **Visualization animations** such as visibility, fill style, line style, text style, blinking, percent fill horizontal, percent fill vertical, horizontal location, vertical location, width, height, orientation, value display or tooltip.
- **Interaction animations** such as disable, user input, horizontal slider, vertical slider, pushbutton, action script, show graphic or hide graphic.
- **Element-specific animations** for the Status element and Windows common control elements.

Each element in your Industrial graphic can have one or more animations. You can disable and enable individual animations. You can also cut, copy and paste animations between elements. Only animations supported by the target element are pasted.

You can also substitute references and strings in animations.

For detailed information and instructions on animations, see "Animate graphic elements" in the Industrial Graphic Editor help.

Note: Not all animations are available for all element types. Some animations do not make logical sense, such as line style with a text element. You cannot select or copy these animation combinations.

Create and manage graphics

You can configure Industrial Graphics and Situational Awareness Library graphics to visualize data in an AVEVA OMI ViewApp.

You use the Industrial Graphic Editor to create graphics from basic elements, such as rectangles, lines, and text elements. You can also use the Industrial Graphic Editor to embed and configure Industrial Graphics from the Visualization folder library of graphics .

After you create a graphic, you can embed it into another graphic or place it in a layout pane to be displayed at runtime.

You can embed a graphic in a template or instance of an object, or into an InTouch window using WindowMaker, providing several ways to visualize object-specific information quickly and easily. Embedding a graphic in a template means that you can update one graphic and cascade the changes throughout your ViewApp.

Depending on your development requirements, you can select where and how to store Industrial Graphics.

- Store graphics in the Visualization folder if you want to define them as a standard that you can re-use, such as a generic valve graphic. You can store Industrial Graphics here if you want to use them only in the InTouch HMI or in AVEVA OMI ViewApps.
- Store graphics as AutomationObject templates if you want to use them in multiple instances at runtime. For example, you can create a valve graphic contained in an AutomationObject template that represents the functionality of a valve type on your plant floor.
- Store graphics as AutomationObject instances if you want to use them in only one specific object instance. For example, an AutomationObject instance that can be assigned a very specific piece of machinery as a graphic.

The topics in this section discuss some general topics and describe some actions which have aspects specific to OMI. For general information and procedures about creating and managing graphics, see [Create and manage industrial graphics](#) in the Industrial Graphic Editor help.

Create a graphic

You can create a new graphic:

- In the Visualization folder, for generic graphics that you frequently use in different situations. For example, a valve graphic.
- From a template. Do this if the graphic applies to any object instance created from that template. An example is a graphic representing a type of tank when your facility has multiple tanks of that type.
- From an object instance. Do this if the graphic is particular to that object and you are unlikely to re-use it in any other situation.

Creating a new graphic adds it to your graphics library. You can then manage and use the graphic. For example, you can:

- Edit the graphic
- Rename it
- Copy it
- Create new folders -- groups of related graphics, such as ones to use in a particular application or that represent related devices
- Move graphics between folders
- Rename folders

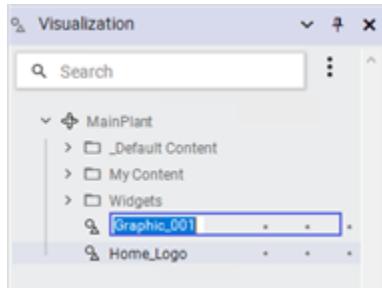
Create graphics in the visualization folder

You can create a new graphic in the Visualization folder. It is then listed in the Visualization folder with a proposed default name. You can:

- Rename the graphic.
- Move the graphic.
- Edit the graphic with the Industrial Graphic Editor.

To create a new graphic from the IDE

1. On the **Home** ribbon, in the **Create** area, select **Graphics**. The **Visualization folder** appears and a new graphic is listed.



Note: You can also press **CTRL + SHIFT + H** to create a new Industrial graphic or right-click in the **Visualization folder** and select **New** and **Graphic** from the shortcut menu.

2. Rename the graphic.

Names must be unique within the entire Visualization folder hierarchy. Valid characters for graphic names include alphanumeric characters, #, and _ (underscore). Graphic names cannot contain spaces and the graphic name cannot begin with the \$character.

3. Double-click the graphic name. The Industrial Graphic Editor appears.
4. Draw the graphic.

Create graphics in AutomationObject templates

You can create a graphic from the Object Editor **Attributes** tab in an AutomationObject template. Creating a

graphic this way automatically associates the new graphic with the AutomationObject.

To create a new graphic for an AutomationObject template

1. Open the AutomationObject template. Click the **Attributes** tab. Any local and inherited graphics are listed.
2. In the **Content** area, click the **Add (+)** icon.
3. Give the new graphic a name. Names must be unique. Valid characters for graphic names include alphanumeric characters, \$, #, and _ (underscore). Graphic names cannot include spaces and cannot begin with the \$ character.
4. If needed, type the description of the graphic in the **Description** box.
5. Make sure the graphic is selected and click the **Edit** icon. The Industrial Graphic Editor opens.
6. Draw the graphic.

Create graphics in AutomationObject instances

You can create a graphic from the **Attributes** tab in an AutomationObject instance. Creating a graphic this way automatically associates it with the AutomationObject instance.

Note: AutomationObjects can also inherit graphics from their parent template. You can only view an inherited graphic in read-only mode. Inherited graphics can only be removed or edited in the template where they are defined.

To create a new graphic for an AutomationObject instance

1. Open the AutomationObject instance. Click the **Attributes** tab. Any local and inherited graphics are listed.
2. In the **Content** area, click the **Add (+)** icon.
3. Give the new graphic a name. Names must be unique. Valid characters for graphic names include alphanumeric characters, \$, #, and _ (underscore). Names cannot include spaces and cannot begin with the \$ character.
4. If needed, type the description of the graphic in the **Description** box.
5. Make sure the graphic is selected and click the **Edit** icon. The Industrial Graphic Editor opens.
6. Draw the graphic.

Graphic Change Propagation

If the graphic is hosted by the Visualization folder and edited, all graphics hosted by AutomationObject templates and instances that contain embedded instances of this graphic are also updated.

If the graphic is hosted by an AutomationObject and edited, all graphics hosted by derived AutomationObjects are also updated.

Organize graphics

Use the library available to your HMI/SCADA software to organize your graphics by creating a folder hierarchy as you would with files and folders in Windows Explorer. You can move graphics around within the folder hierarchy and copy or rename them as needed.

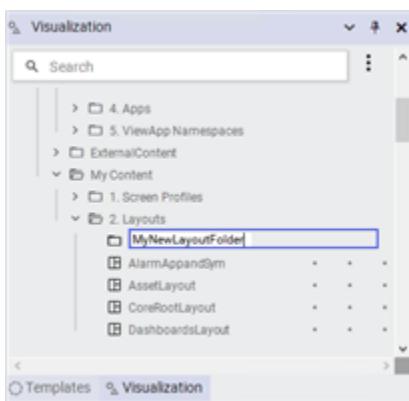
For more information about organizing graphics, see the help file for your HMI/SCADA software.

Create folders in the visualization folder

You can create folders in the **Visualization folder** to organize your graphics. For example, you can create a folder called "Valves" to store different valve graphics.

To create a folder in the Visualization folder

1. Open the **Visualization folder**.
2. Select the folder under which you want to create a new folder. Select the Galaxy name if there are currently no folders.
3. On the **Home** ribbon, in the **Create** area, select **Folder**, then **Visualization folder**. A new folder is created with a default name.
4. Rename the new folder as needed.



Use graphic thumbnails

When you place your mouse cursor over an Industrial Graphic in the Visualization folder, a thumbnail view of the graphic's displays. This allows you to preview the graphic without opening it.

The graphic thumbnail updates automatically when a change is made. However, thumbnails of graphics that contain embedded instances of the base graphic will not automatically update. You must manually do so by right clicking the graphic and selecting **Update Thumbnail**.

You can update graphic thumbnails from any of the following levels:

- individual graphic
- folder level
- update all graphics from the root of the Visualization folder

The time it takes to refresh a folder or the Visualization folder is proportional to the number of graphics each contains. This may take several minutes.

Note: If you have imported a newer version of a client control already embedded in a graphic, restarting WindowMaker and updating the graphic thumbnail will not update the contents of the newly imported control. To update the graphic with the contents of the new control, edit and save the graphic. The graphic thumbnail will then reflect the contents of the new control.

Move graphics between folders

You can move graphics from one folder in the Visualization folder to another. Moving graphics between folders does not affect their functionality.

To move graphics between folders

1. Open the Visualization folder.
2. Locate the graphic you want to move.
3. Do either of the following:
 - Drag the graphic to the folder you want to place it in. The graphic moves to the new folder.
 - To move the graphic to the top level in the Visualization folder hierarchy, drag the graphic to the Galaxy name icon.

Rename graphics

You can rename a graphic at any time. Renaming a graphic does not affect its functionality.

Graphic names must be unique within the entire hierarchy of the Visualization folder.

To rename a graphic

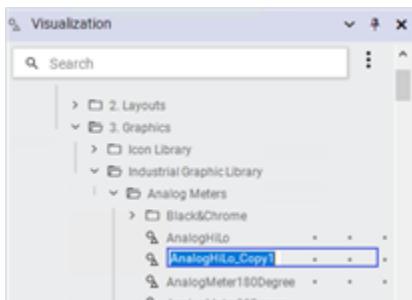
1. Open the Visualization folder.
2. Select the graphic to rename.
3. In the **Edit** area of the **Home** ribbon, click **Rename**. The graphic name is in edit mode.
4. Type a new unique name for the graphic and click **Enter**.

Copy graphics

You can create copies of graphics in the Visualization folder. The copies are suffixed with "_Copy1", "_Copy2", and so on.

To create a copy of a graphic

1. Select the graphic to copy.
2. In the **Edit** area of the **Home** ribbon, click **Duplicate**. A copy of the graphic is created.



3. If needed, type a new name for the graphic.

Delete a graphic

You can delete a graphic that you no longer need. Deleting a graphic removes it completely from the Industrial Graphic Editor.

- When you delete a graphic, you are shown where the graphic is used. This allows you to understand the impact of deleting the graphic before you actually delete it.
- You cannot delete a graphic that someone else has open for editing or left checked out.
- If you delete a graphic that is embedded in another graphic, it shows a Not Found message.

To delete a graphic

1. Select the graphic or graphics that you want to delete.
2. On the **Home** ribbon, in the **Edit** area, click **Delete**.

The **Confirm Delete** dialog box appears, showing the number of objects that reference each graphic you are deleting.

3. To see the places that a graphic is being used, click the down arrow for that graphic at the right of the **Referenced** column.
4. To delete the graphic(s), click **Delete**.

View a graphic in read-only mode

You can view a graphic in read-only mode if you don't want to edit it, or if it is checked out by somebody else. If you open a graphic in read-only mode, you have access to all functions in the Industrial Graphic Editor that don't change the graphic.

To view a graphic in read-only mode

1. Select the graphic that you want to view in read-only mode.
2. Click **Open Read-Only**. The selected graphic opens in the Industrial Graphic Editor.

Rename graphic folders

You can rename a graphic folder at any time. Renaming a folder does not affect the functionality of any graphics it contains.

To rename a folder in the Visualization folder

1. Open the Visualization folder.
2. Select the folder to rename.
3. On the **Home** ribbon, in the **Edit** area, click **Rename**. The folder name is in edit mode.
4. Type a new unique name for the folder and click **Enter**.

Delete graphic folders

You can delete a graphic folder in the Visualization folder at any time.

You can only delete folders that do not contain any graphics. Move the graphics to another folder or delete them before deleting the folder.

To delete a folder from the Visualization folder

1. Open the Visualization folder.
2. Select the folder to delete.
3. On the **Home** ribbon, in the **Edit** are, click **Delete**. When a confirmation message appears, click **Yes**.

Move graphic folders

You can move a graphic folder within the Visualization folder. If you move a folder, all graphics and folders it contains are also moved.

To move a folder in the Visualization folder

1. Open the Visualization folder.
2. Locate the folder to move.
3. Do either of the following:
 - Drag the folder to the folder you want to place it in. The folder is moved to the new location.
 - To move the folder to the top level in the hierarchy, drag it to the Galaxy name icon.

Customize graphic folders

You can hide or show graphic folders after creating them. You can do this only for first-level folders--those directly beneath the Galaxy in the hierarchy.

To customize graphic folders

1. On the ribbon, select **Galaxy**.
2. Select **Configure**, then **Integrated development environment**, then **Folders**.
3. In the **Folders** pane, click **Visualization**.
4. Do one or more of the following:
 - To hide or show all folders, clear or select the checkbox for **Folder list**.
 - Clear the checkbox for any folder you want to hide.
 - If a folder is hidden and you want to display it, select its checkbox.
5. When you are done with your changes, select **Save**.

Folders are shown or hidden, depending on your settings.

Create multiple configurations of a graphic

The Symbol Wizard Editor is a feature of the Industrial Graphic Editor to create multiple configurations of a graphic. A graphic configuration represents different visual or functional variations of a graphic.

Graphic configurations are created using layers containing associated graphic elements, custom properties, and named scripts. Based on graphic properties and possible values of these properties, rules are applied that specify when a layer is part of a graphic configuration.

For more information about using Symbol Wizards to create multiple configurations of a graphic, see [Using the Symbol Wizard Editor](#).

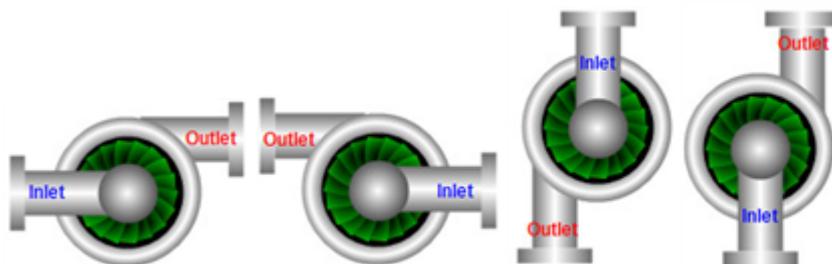
Understand visual and functional graphic configurations

Standard Industrial Graphics show reasonably realistic views of process objects. These graphics can be modified with the Symbol Wizard to incorporate multiple visual configurations in a graphic.

Situational Awareness Library graphics are a special set of Industrial Graphics that are available for use in your HMI/SCADA software, and are designed using the Symbol Wizard Editor. Their design cannot be changed. However, you can select Wizard Options from the Symbol Wizard Editor to select the configurations that are incorporated into each graphic's design.

Visual graphic configurations

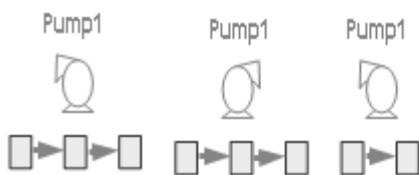
Using an example of a centrifugal pump with separate inlet and outlet pipes, there are four practical visual configurations. The pump's blade housing is common and appears in all possible configurations. But, the pump's inlet and outlet pipes can be placed at the left or right in a horizontal direction or at the top or bottom when the pump is oriented vertically.



Orientation is the visual property that identifies the different configurations of a pump graphic. The attributes associated with the Orientation property are left, right, top, and bottom.

Functional graphic configurations

Situational Awareness Library graphics include functional properties in addition to visual properties. For example, a multi-stage pump graphic includes a Wizard Option to select either a five-stage, three-stage, or single stage pump in addition to a visual Orientation property to select left or right pump configurations.



About graphic scaling

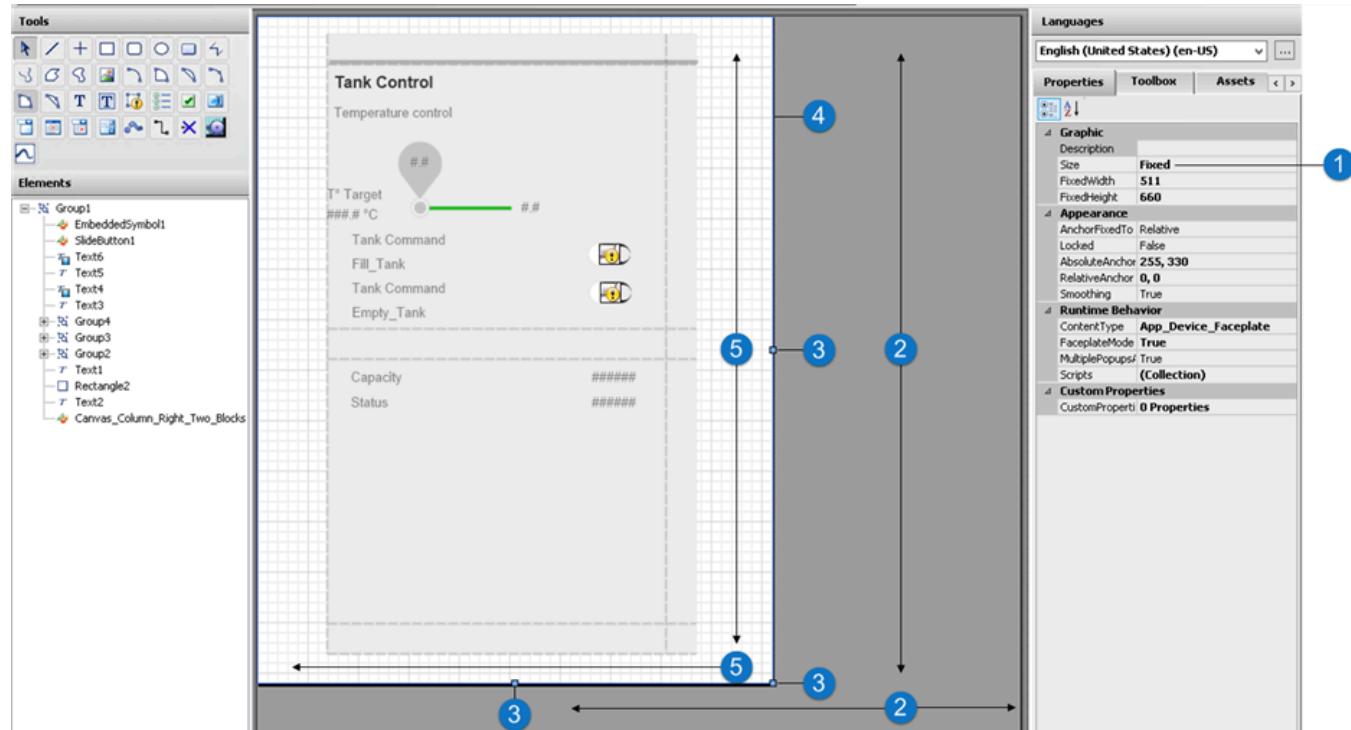
By default, a graphic attempts to scale to the size of its hosting pane and maintain the graphic's horizontal and vertical aspect ratio. The size of a graphic created using the Industrial Graphic Editor is determined by a bounding rectangle around all graphic elements and embedded graphics that compose the graphic.

When a graphic is placed in a layout pane, if the graphic's bounding rectangle exceeds the size of its hosting pane, the bounding rectangle becomes smaller by reducing the original size of its contents. If the graphic's bounding rectangle is smaller than the pane size, the bounding rectangle increases and enlarges the original size of its contents.

Graphics that have a defined fixed size render in runtime identically to graphics without a fixed size. A graphic's size adjusts to the size of the pane while maintaining its aspect ratio as appropriate. Most graphics that have a configured fixed size are typically placed in a specific target pane of a known size that enables them to be rendered exactly as designed in the Industrial Graphic Editor without any required resizing during configuration or runtime.

About the Industrial Graphic Editor

After placing a graphic or graphic elements on the canvas of the Industrial Graphic Editor, you can set several properties to specify a fixed size for the graphic. The following screen shot shows how to adjust the fixed size of a graphic.



1	<p>The Size property is set to Auto by default, which automatically sizes graphics. Set Size to Fixed to create fixed size graphics. When set to Fixed, the FixedWidth and FixedHeight properties appear, which show the dimensions of the fixed area of a graphic in pixels.</p> <p>You can enter the graphic's width and height directly in the FixedWidth and FixedHeight fields.</p> <ul style="list-style-type: none">• FixedWidth<ul style="list-style-type: none">• Default value = 500• Maximum value = 7680• Minimum value = 0• FixedHeight<ul style="list-style-type: none">• Default value = 500• Maximum value = 4320• Minimum value = 0 <p>You can also change the width and height of the graphic's fixed size area by moving the boundary with its grab handles.</p>
2	<p>The gray area represents that part of the Industrial Graphic Editor canvas that is outside the boundary of the fixed size area of the graphic.</p>
3	<p>Grab handles on the fixed size boundary can be moved to change the size of the fixed area of a graphic. The FixedWidth and FixedHeight properties update to show the current width and height when you move the boundary.</p> <p>Grab handles are placed at the following locations on the fixed size boundary:</p> <ul style="list-style-type: none">• Top right corner• Center right side• Bottom right corner• Bottom center
4	<p>The thick black line shows the boundary between the Industrial Graphic Editor canvas within the fixed size area of the graphic set by the graphic's FixedWidth and FixedHeight properties and the canvas area outside the fixed dimension area.</p>

5

The white area of the Industrial Graphic Editor canvas is within the fixed size boundary of the graphic.

You can place graphic elements from the Industrial Graphic Editor's **Tools** section or embed graphics as you would normally do when the **Size** property is set to **Auto**. As long as the graphics elements or embedded graphics are placed inside the fixed size boundary, the graphic will fit the size of a pane perfectly. Graphics within the graphic maintain the same size, scale, spacing, and white spaces around the graphics on the canvas specified during design time.

Create a fixed-size graphic for a ViewApp pane with a known size

This topic describes the general sequence of steps to create a fixed size graphic designed to fit a ViewApp pane with a known size. This workflow is typical of creating content that will fit exactly within a pane while editing a ViewApp with the ViewApp Editor.

Important:

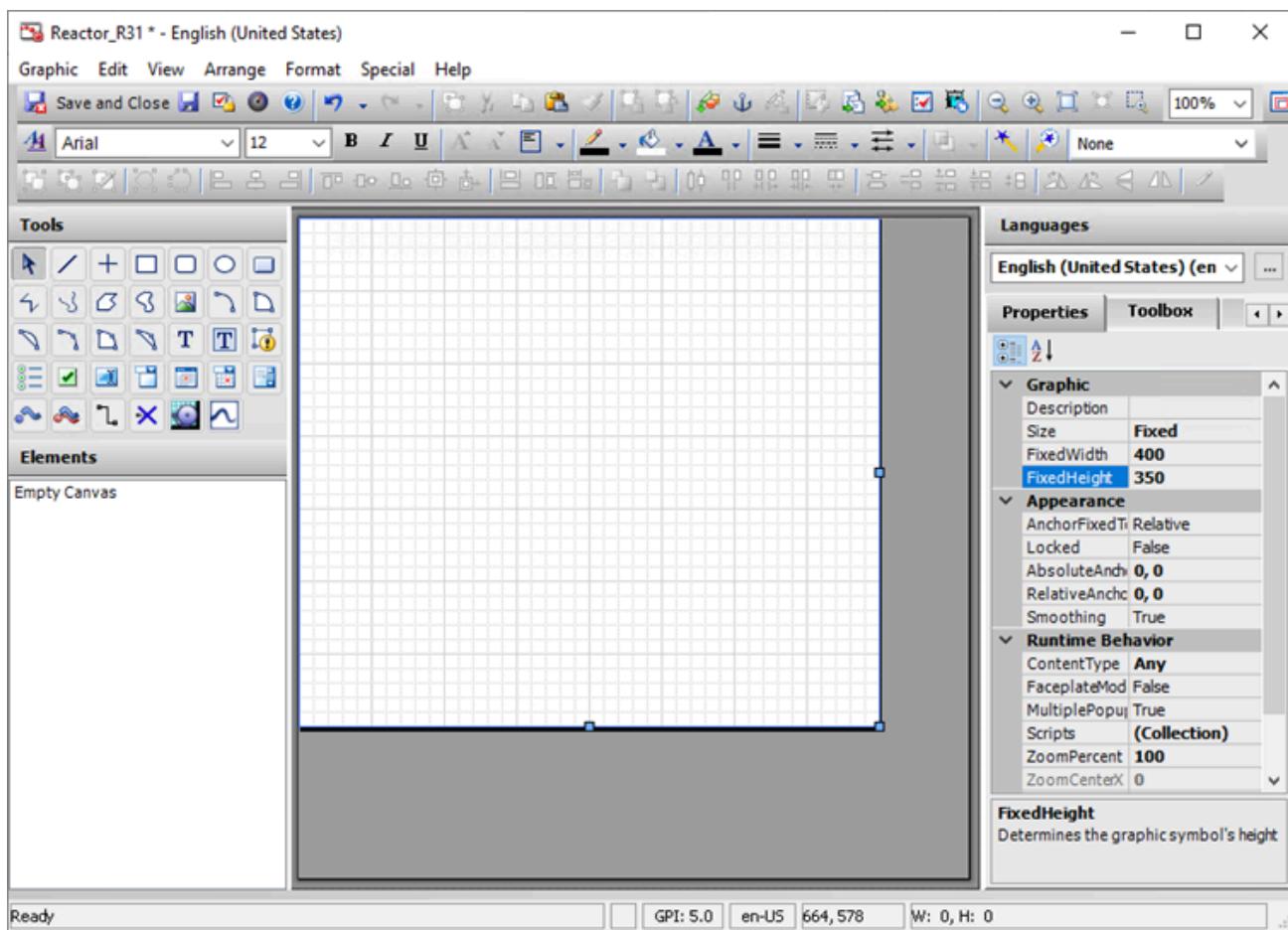
The size of a graphic in runtime is affected by the resolution of the screen that will show the running ViewApp and the specified width and height of the layout pane in which the graphic is placed. Also, the layout **Display Mode** property includes a set of options that determines how content is resized during runtime to fit within the size of a layout pane.

The following procedure describes the steps to create a fixed size graphic that will fit within a pane of a known size. You should create a fixed size graphic whose width and height matches the resolution of the screen profile. You should set the dimensions of the hosting pane to the same size as the graphic.

To create a fixed size graphic that fits within a pane of a known size

1. Open the ViewApp in the ViewApp Editor that will contain the fixed size graphic.
2. Right-click the pane that will contain the fixed size graphic and select **Create Graphic**.
A dialog box appears requesting a name for the graphic to be created.
3. Assign a name to the graphic.

The Industrial Graphic Editor opens, and the white portion of the canvas shows the dimensions of the pane where you created a graphic. The **Size** property is set to **Fixed** and the **FixedWidth** and **FixedHeight** properties show the dimensions of the pane in pixels.



4. Place graphic elements and graphic to create a graphic.
5. Save and close the graphic to return to the ViewApp Editor.

The graphic you created appears in the pane where you created the graphic. The graphic should fit into the pane at the same size as it was designed in the Industrial Graphic Editor.

6. Save the ViewApp and deploy it.
7. Open the deployed ViewApp

In runtime, the ViewApp will show the graphic you created at the same size as it was originally designed.

The rendered graphic shows:

- The graphic fits in the pane at the same size as it was designed in the Industrial Graphic Editor.
- The graphic's size does not change in runtime.
- Any blank spaces around the graphic are maintained and appear in the graphic during runtime.

Create a fixed-size graphic for a ViewApp pane size

This topic describes the general procedure to create a fixed size graphic that will fit a pane of a known size even if the pane does not exist when the fixed size graphic is created.

This workflow is typically the case where two different employees work on a ViewApp to update it to reflect changes in a production environment. The first employee is usually a user who creates a fixed size graphic. A second employee will drag and drop the graphic in a pane while editing the ViewApp with the ViewApp Editor.

To create a fixed size graphic that will fit a pane with a known size

1. The first employee creates a new graphic and opens it in the Industrial Graphic Editor.
The entire canvas area of the Industrial Graphic Editor is enabled to place graphic elements and graphics.
2. Configure the graphic to a known pane size.
 - a. Select the **Graphic** property **Size** and set it to **Fixed**.
The **FixedWidth** and **FixedHeight** properties appear after the **Size** property is set to **Fixed**.
 - b. Update the values of **Fixed Width** and **Fixed Height** to the pixel dimensions of a pane with a known size.
You can also move the grab handles on the fixed size boundary to set the dimensions of the graphic.
3. Place the graphic elements and other graphics within the fixed size area of the graphic.
4. If you want white space around the graphic, adjust the width and height of the graphic to show a white margin around the graphic elements and graphics that compose your graphic.
5. Save and close the graphic.
6. The second employee creates a layout to be used in the ViewApp with a pane that is the same size as the fixed size graphic.
7. Open the ViewApp with the ViewApp Editor.
8. Assign a Screen Profile and the layout containing the pane to the ViewApp.
9. Select a screen that contains the layout with the pane of a known size.
10. Select the fixed size graphic created by the first employee and drag and drop it in the pane.

The configured graphic appears in the selected pane of the ViewApp when it is edited by the ViewApp Editor.
The graphic preview shows:

- The graphic fits in the targeted pane with the same size as it was designed in the Industrial Graphic Editor.
 - If necessary, the graphic can scale larger or smaller to fit the pane.
 - Any white space placed around the graphic while it was created is maintained when the graphic is placed in a pane.
11. Save the ViewApp and deploy it
 12. Open the running ViewApp.

The running ViewApp shows the fixed size graphic in the target pane at the same size as it was created in the Industrial Graphic Editor and configured in the ViewApp Editor. The rendered fixed size graphic shows:

- The graphic fits in the pane with the same size as it was designed in the Industrial Graphic Editor
- Any white space placed around the fixed size graphic is maintained.

About fixed-size graphics when pane is resized

Graphics created for a layout pane obtain size dimension from the pane's size by default. The values assigned to the **FixedHeight** and **FixedWidth** properties automatically match the height and width of the pane. Graphics that are designed with its elements placed within the dimension boundaries will fit in the pane.

When the dimensions of a graphic are changed while editing in the ViewApp Editor, the following behaviors are observed:

Fixed Size Graphic Size is Greater Than Pane Size

- When the graphic size is larger than hosted pane size, elements placed outside the pane are included in the configuration view. The graphic's size is reduced to fit it entirely within the dimensions of the pane.
- When the **FixedWidth** and **FixedHeight** properties of a graphic match the pane size, but there are graphic elements beyond the graphic's fixed width and height boundaries, the graphic's size is reduced to fit the graphic elements outside the fixed size area within the dimensions of the pane.

Fixed Sized Graphic Size is Smaller Than Pane Size

When a graphic's fixed size is smaller than the hosted pane size, the graphic's size is increased to fit the pane size. The sizes of all of the graphic's graphic elements are increased.

Fixed Size Graphic Embedded in Another Graphic

- The embedded fixed size graphic maintains its size configured at design time.
- There will not be any clipping of any white space provided around primitives or embedded graphics present in the graphic.
- There will not be any scaling of graphics. An exception to this is if the embedded graphic was resized at design time, the graphic will look scaled.

About fixed-size graphics during runtime

The following list summarizes the key behaviors of fixed size graphics during runtime.

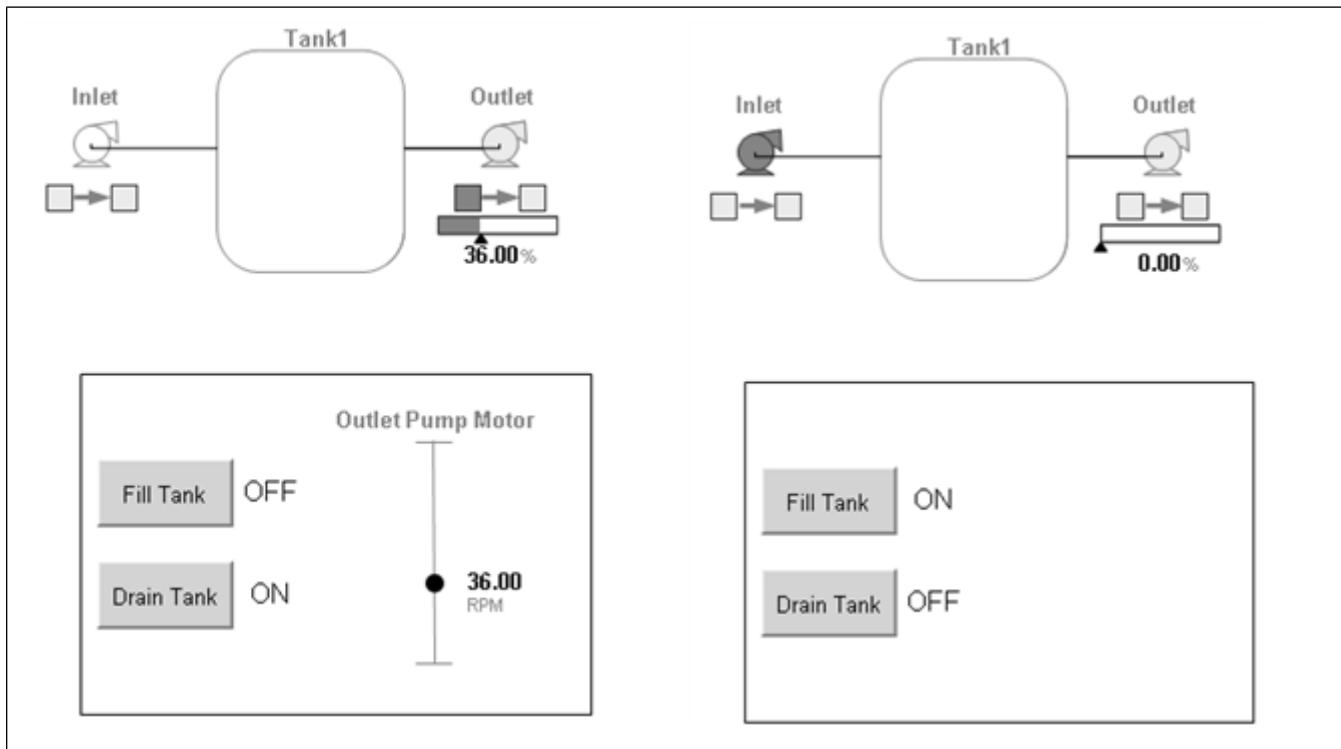
- Graphics that have the **Size** property set to **Fixed** render all graphic elements placed within the fixed size boundary at their configured fixed size at runtime.
 - Graphics do not change size or aspect ratio within the pane that is hosting the graphic.
 - Graphics configured to fit exactly within a hosting pane render to fit exactly within the size of a pane. The graphic elements in the graphic render at the same size and placement as designed in the Industrial Graphic Editor.
 - The white space around primitives or embedded graphics present in the graphic maintains its size and placement within the graphic.
- Fixed size graphics change size or aspect ratio to fit the hosting pane during runtime in the following cases:
 - When the graphic's size is larger or smaller than the hosting pane.
 - When graphicsize is the same as the hosting pane but there are graphics that are placed outside the fixed size boundary configured for the graphic. This includes graphics that are placed on negative coordinates.
 - When graphic size is the same as the hosting pane, but graphics that are placed on the boundary of the graphic marked by the size configured for the graphic.
- Embedded graphics follow the same behavior as standard fixed size graphics.

Create faceplate graphics

A faceplate graphic typically shows visual elements for operators to monitor and manage equipment represented by graphics in a running application. Operators can select buttons in a faceplate graphic to turn equipment on or off. A meter or other type of graphic can be embedded in a faceplate to show the current measured value of a selected piece of equipment during runtime.

Graphics that have been enabled for faceplate compatibility mode enable operators to manage a set of similar graphics using a single faceplate. Instead of creating a separate faceplate for each graphic, a faceplate can be adapted for different graphic configurations by showing or hiding graphic elements or embedded graphics in the faceplate itself.

For example, you can create a faceplate graphic to fill or drain a tank. The faceplate graphic shows an embedded RPM meter when a SA_MultiStagePump graphic is configured as an outlet pump with a variable speed motor controller. The same faceplate can be used without a meter when another SA_MultiStagePump graphic represents a fixed speed inlet pump that runs without an output controller.



In the example of the embedded RPM meter in the faceplate graphic to fill or drain the tank shown above, two conditions must be met to hide the RPM meter:

- The embedded RPM meter graphic's **FaceplateMode** property must be set to true.
- The embedded RPM meter graphic must have a custom property with a reference to an invalid value.

Important: Only AVEVA OMI ViewApps support faceplate compatibility mode. Managed and traditional InTouch applications do not support faceplate mode.

About faceplate graphics creation

This section describes the general steps to create a faceplate graphic, which consist of the following:

- Determine the graphics on a faceplate that should be shown or hidden to support different graphic configurations.
- Determine the valid and invalid reference states that will show or hide graphics on a faceplate.
- During design time, set faceplate compatibility mode at the root level or canvas of the graphic.

Determine the Graphics that Run in Faceplate Compatibility Mode

Consider the graphics that will represent the equipment to be controlled and monitored from a faceplate.

Determine the unique qualities of each piece of equipment that will need to be shown on the faceplate by a graphic. These unique qualities are the object attributes, such as a PV attribute or ON/OFF attribute. The presence of these attributes based on the configuration of the graphic determine how the faceplate is built.

Determine the Valid and Invalid Reference States that Show or Hide Faceplate Graphics

Determine the reference states when faceplate graphics need to be shown or hidden for all configurations of equipment represented by the graphics.

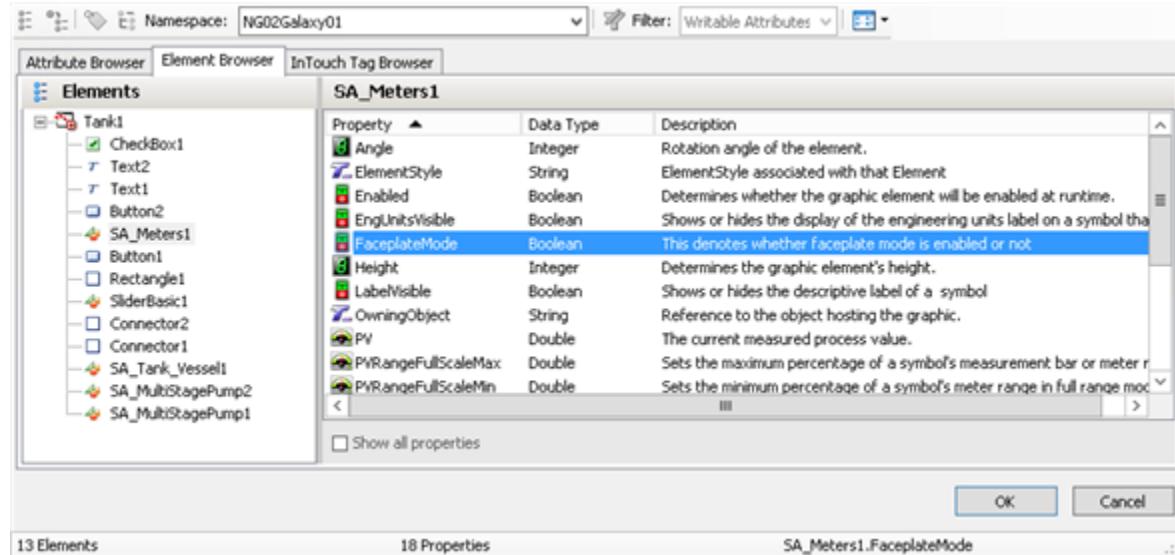
In the example of using a faceplate to manage fixed-speed and variable speed pumps, a meter should be hidden when an operator is managing a fixed-speed pump. When running a fixed speed pump, a reference to a meter's PV custom property is invalid and will hide the meter when it is running in faceplate compatibility mode.

Set Faceplate Compatibility Mode

Set faceplate compatibility mode for those graphics that appear on a faceplate that must be hidden to support a specific equipment configuration. Faceplate compatibility mode can be set on graphic groups and the root level or canvas of a graphic. For more information about the inheritance rules of faceplate compatibility mode for grouped or nested graphics, see [About graphics visibility during runtime](#).

In the example of using a faceplate to manage fixed-speed and variable speed pumps, the meter should be hidden on the faceplate when an operator is managing a fixed-speed pump.

Graphics include the Boolean **FaceplateMode** property. When a graphic's **FaceplateMode** property is set to True, the graphic is hidden when it contains an invalid reference to a target. In the following example, a CheckBox graphic element toggles the state **FaceplateMode** property of a meter graphic to enable or disable faceplate mode compatibility.



When the operator selects a fixed-speed motor from the faceplate, the meter's PV custom property reference becomes invalid, which will hide the meter when its **FaceplateMode** property is True.

About graphics visibility during runtime

Two conditions determine if a graphic can be shown or hidden during runtime using faceplate compatibility mode. The following table shows the conditions to show or hide a graphic on a faceplate:

FaceplateMode Property	Valid Reference	Invalid Reference
True	Show Graphic	Hide Graphic
False	Show Graphic	Show Graphic

- The **FaceplateMode** Runtime Behavior property must be set to True to show or hide a graphic in faceplate compatibility mode. The **FaceplateMode** default value is False, which shows a graphic under all reference conditions.
- A faceplate graphic is hidden if **FaceplateMode** is True and an invalid reference is made from a graphic to a target attribute from the following:

- **Graphic element**

When **FaceplateMode** is True and a graphic element has an invalid reference through animation or script, only that graphic element is hidden within a graphic. If the graphic element is part of a graphic embedded into another graphic, the embedded graphic still appears although the graphic element with the invalid reference is hidden.

- **Embedded graphic**

When **FaceplateMode** is True and an embedded graphic has an invalid reference, then the entire graphic will be hidden. To show or hide an embedded graphic based on the value of **FaceplateMode**, a reference must be made to the entire graphic that is embedded, not just to a graphic element that belongs to the graphic.

- **Expressions**

Animation expressions, script expressions, or custom property expressions are not evaluated if there is at least one invalid expression.

- **Animations**

If an animation has an invalid reference, then the graphic associated with the animation will be hidden in faceplate mode. If an animation has a valid reference then its animation will execute. Animations with valid references are evaluated even if the graphic with which the animation is associated is hidden by faceplate mode. This is because a graphic element may be associated with several animations, some which may be valid while others are invalid.

- **Scripts**

Named scripts, action scripts, data change scripts value scripts, and On show/On hide scripts, do not run if a script has at least one invalid reference.

- **Quality and status icon**

Global status and quality ignores invalid references and does not show any status and quality indications related to them.

- **Error and warning messages**

When a graphic is set to faceplate compatibility, no error or warning messages are logged for invalid references. The data status graphic element is hidden if it contains an invalid reference.

- **Undeployed GR node**

In faceplate compatibility mode, objects show a Configuration Error status when the GR node is undeployed. Any graphic that shows a Configuration Error message is hidden. The graphic becomes visible again when the GR node is redeployed.

- **Stopped App engine**

In faceplate compatibility mode, objects show a Communication Error status when an App engine is undeployed. Any graphic that shows a Communication Error message remains visible and is not hidden.

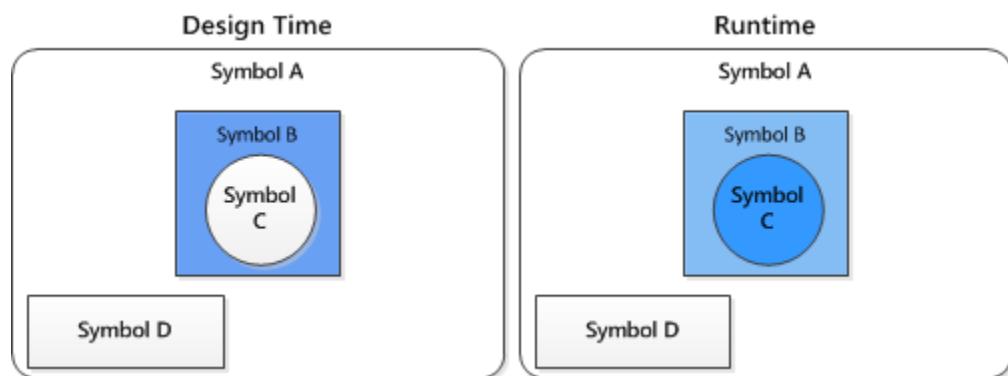
Runtime Behavior of Faceplate Mode

Graphics configured for faceplate compatibility mode follow defined inheritance rules during runtime.

Embedded Graphics

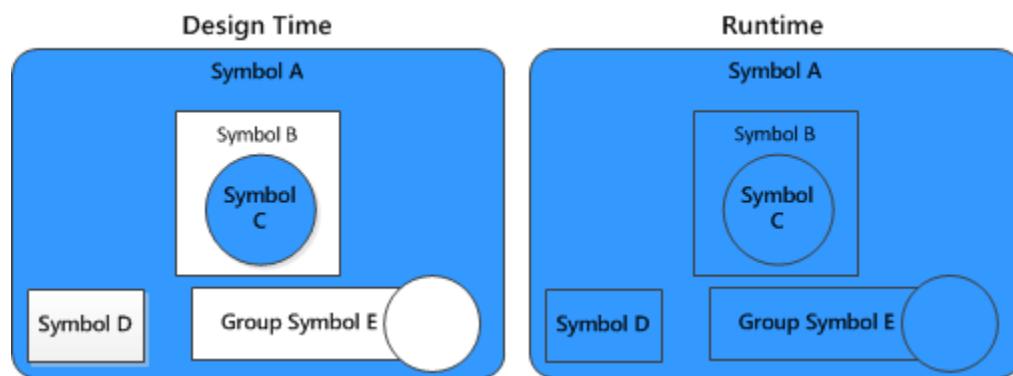
Embedded graphics inherit faceplate compatibility from the parent to child graphic during runtime. But, faceplate compatibility mode is not inherited from the child to the parent embedded graphic.

In the following figure, Symbol B is set to faceplate compatibility during design time. During runtime, Symbol C embedded within Symbol B inherits faceplate compatibility.



Graphic Groups

If a parent graphic is set to faceplate compatibility mode, then all its child embedded graphics and graphic groups inherit faceplate compatibility mode at runtime.



About settings attributes

Settings attributes typically enable or disable visibility options of Situational Awareness Library graphics and some Viewapp global settings. The Settings attributes are assigned default values and load when a ViewApp starts.

Settings attributes can be set from faceplate graphics. You can also embed the **WW_App_Function** widget in a graphic and use it to control settings attributes. The **WW_App_Function** widget is included in Galaxies created from the Default.cab, as part of the **ViewApp UI Library** in the Visualization folder.

The sample **\$InTouchOMI_ViewApp**, which is associated with the **WW_L/Desktop** layout, uses the **WW_App_Function** widget. A User Input animation has been configured to set the value of the

MyViewApp.Settings.TrendDuration attribute. You can configure other attributes similarly. Refer to the sample ViewApp for more information.

Important: Settings attributes are predefined and available only for AVEVA OMI ViewApps. Settings attributes cannot be included in an InTouch application.

All Settings attributes operate in the MyViewApp.Settings namespace. The names of Settings attributes are specified with the MyViewApp.Settings prefix in the form MyViewApp.Settings.*attribute_name*.

Attribute Name	Data Type	Attribute Type	Initial Value	Description
TrendDuration	Integer	Read/Write	15	The length of a trend duration in minutes.
GridLines	Boolean	Read/Write	True	Shows or hides grid lines on a pen trend
PenTrendVisible	Boolean	Read/Write	True	Shows or hides all pen trend lines of a trend graphic
PVVisible	Boolean	Read/Write	True	Shows or hides the process value of a Situational Awareness Library graphic.
PVOptimalRangeVisible	Boolean	Read/Write	True	Shows or hides the optimal range indicator on a Situational Awareness Library graphic.
PVOperatingAutoScale	Boolean	Read/Write	False	Enables or disables meter auto scaling on a Situational Awareness Library graphic.
SPVisible	Boolean	Read/Write	True	Shows or hides a Setpoint on a Situational Awareness Library graphic.
TrackerVisible	Boolean	Read/Write	True	Shows or hides a tracker line on the meter of a Situational Awareness Library

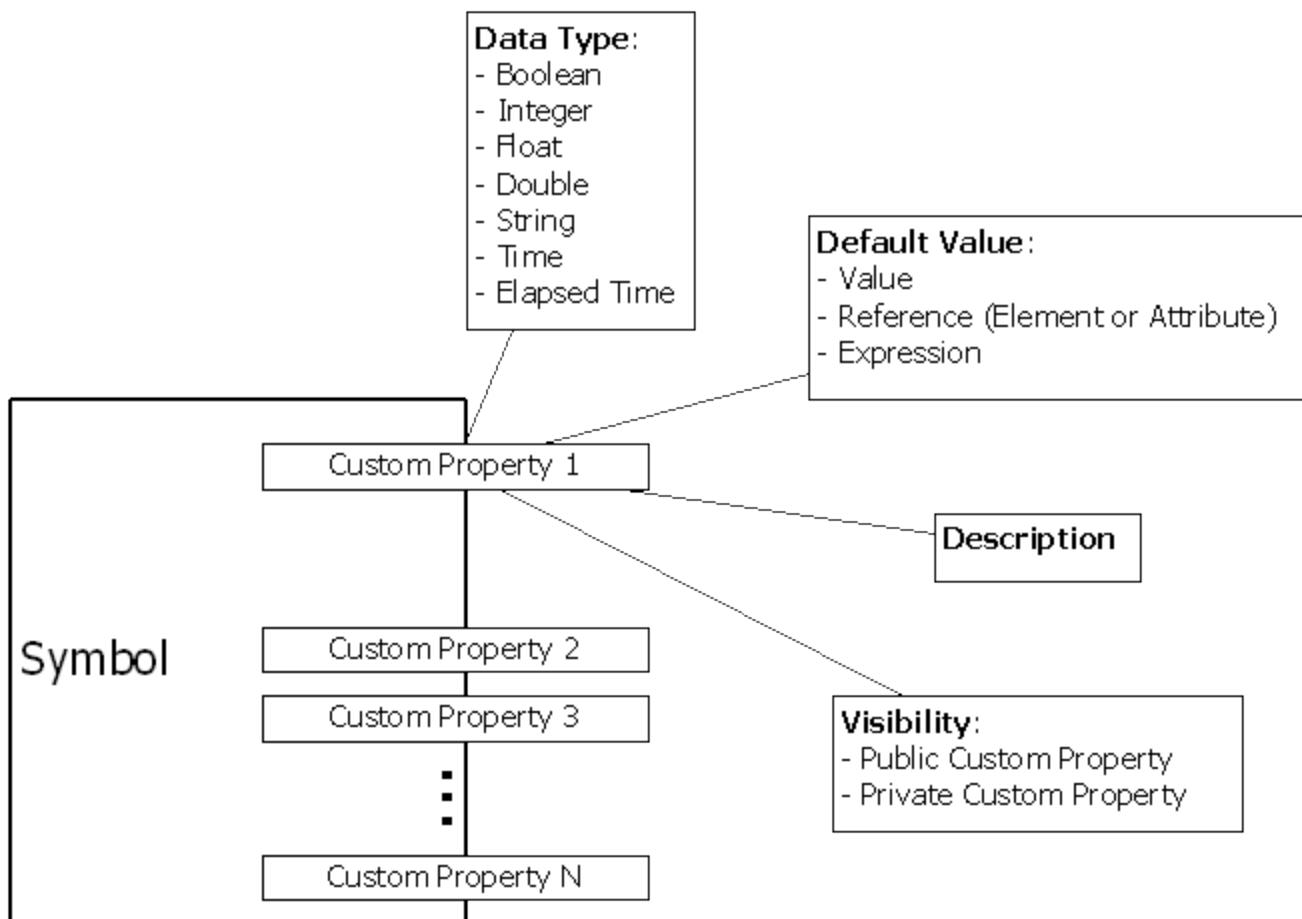
				graphic.
LabelVisible	Boolean	Read/Write	True	Shows or hides the label on a Situational Awareness Library graphic.
EngUnits Visible	Boolean	Read/Write	True	Shows or hides the engineering units of a Situational Awareness Library graphic.
ControllerOP Visible	Boolean	Read/Write	True	Shows or hides the Controller Output for a Situational Awareness Library graphic.
Controller OPCmd Visible	Boolean	Read/Write	True	Shows or hides the Output Command for a Situational Awareness Library graphics.
AlarmIndicator Visible	Boolean	Read/Write	True	Shows or hides an alarm indicator on a Situational Awareness Library graphic.
ChartRange Increment	Integer	Read/Write	5	Sets the precision of the incremental range of chart values shown as measurement lines of Situational Awareness Library Dashboard graphics.
SafetyVisible	Boolean	Read/Write	False	Shows or hides a safety indicator on a Situational Awareness Library graphic.
EnableDisable Sound	Boolean	Read/Write	True	Enables or disables alarm sounds via a Situational

				Awareness Library graphic to alert the operator of an alarm condition. To function properly, this attribute must be connected to a PLC with an integrated audio system.
ShowControl Links	Boolean	Read/Write	True	Shows or hides the control links of a Situational Awareness Library graphic.
ShowToolTips	Boolean	Read/Write	True	Shows or hides the tooltips of a graphic.
ToggleSPTag	Boolean	Read/Write	False	Toggle between a Setpoint or tagname value on a Situational Awareness Library graphic.
DisableHorn	Boolean	Read/Write	False	Enable or disable an external alarm horn via a Situational Awareness Library graphic. To function properly, this attribute must be connected to a PLC with an integrated audio system. This feature is typically used in a control room to allow the operator to troubleshoot the alarm condition without the distraction of the horn blaring.
AckAll	Boolean	Read/Write	False	Enable or disable the

				acknowledgement of all alarms.
OptimizeFor Touch	Boolean	Read/Write	False	Enable or disable optimization of touch gestures at runtime.
ReadOnly	Boolean	Read Only	False	Enable or disable read-only ViewApps at runtime. If false, users have the ability to change attribute values in the ViewApp at runtime. If true, the ViewApp and its attributes are read-only. The ReadOnly property value is assigned by editing a ViewEngine object with the Object Editor. To change the value, you must first undeploy the ViewEngine.

About custom properties for graphics

You can configure and use custom properties to extend the functionality of graphics and use them in combination with InTouch tags. You can use binding with custom properties to dynamically change the reference of a custom property.



You can associate custom properties with functionality you want exposed and that you want to be reusable. You can also use custom properties to connect an embedded Industrial graphic to InTouch tags.

For example, possible uses for using custom properties are:

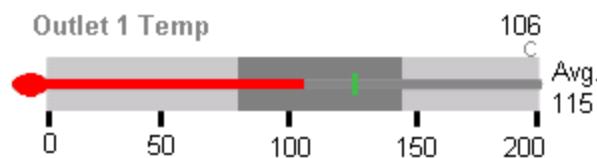
- A "TankLevel" custom property of type Writable Attribute can be given a value of "me.pv".
- A "MaxFillLevel" Custom Property of type Expression can be given a value of "Me.MaxCapacity - 200".

For general information about creating and managing custom properties, see [Custom properties](#) in the Industrial Graphic Editor help. For some examples of using custom properties to show historical data, see the topics below.

Use custom properties to show historical summary data

You can add a custom property to reference historical data collected over a specified period during runtime. The Historian can transform this data to create a set of analog or state statistics that can be shown by Industrial graphic animation during runtime.

For example, consider an example of a temperature meter graphic that shows an optimal temperature range and you would like to know what the average temperature has been over the last 15 minutes. You can add a Value Display animation that shows the average temperature derived from analog temperature data saved in the Historian and referenced by a custom property.



Analog statistical summary data

A custom property analog reference can subscribe to statistics from analog data collected over a defined summary period and saved to the Historian. The following table lists the analog historical statistical data that can be specified for a custom property.

Analog Statistical Data	Description
Average	A time-weighted average calculated from values within a summary period. The average is calculated at the end of the summary period.
Count	A value count calculated from values within a summary period. The count is calculated at the end of the summary period.
First	The first value that occurs within a summary period based on the actual timestamp within the summary period.
Integral	An integral value calculated from values within a summary period. The integral is calculated at the end of the summary period.
Maximum	The first maximum value that occurs within a summary period.
Minimum	The first minimum value that occurs within a summary period.
PercentGood	The ratio of labeled "good" quality data to all data within the summary period. The ratio is expressed as a percentage in the range 0 to 100. PercentGood is calculated at the end of the summary period.
StdDev	Time weighted standard deviation calculated from values within a summary period. The value is calculated using time weighted sums (Integrals) and time weighted sums of squares (IntegralOfSquares) values.

Analog Statistical Data	Description
Last	The last value that occurred in the summary period based on the actual timestamp within the summary period.

State statistical summary data

The Historian stores and retrieves values, where every value gets stored with some timestamp and associated quality. This triplet of value, timestamp, and quality is called VTQ and constitutes the smallest addressable piece of data in the Historian data model.

State summary statistics summarize the states of a value. Four different state value data types are possible: analog (integer), Boolean, string, and Null.

Note: Possible Boolean state values are 0 or 1. True or False strings are not supported.

A custom property state reference can subscribe to state statistics from the Historian as static text, an expression or reference, an aggregate function name, minutes, and state value.

The Historian returns the VTQ for one cycle of a specified state. The quality returned is always OpcQuality. The time returned is always the summary period start time. Value and Time differ based on the aggregate function.

The following table lists state historical statistical data that can be specified for a custom property.

State Statistical Data	Description
Average	Average time a state occurred and completed within a summary period. A partial state within a summary period is ignored for an average calculation. (StateTimeAvgContained)
Minimum	Minimum time a state occurred and completed within a summary period. A partial state is ignored. (StateTimeMinContained)
Maximum	Maximum time a state occurred within a summary period. (StateTimeMax)
Count	Number of times a state occurred and completed within a summary period. A partial state is not counted. (StateCountContained).
Percent	Percentage of the summary period that a state occurred. (StateTimePercent)
Total	Total time a state occurred within a summary period. (StateTimeTotal)

Historical summary period

An Industrial graphic custom property can show historical statistics from Historian data over a defined summary period. During design time, you must specify the summary period to collect Historian summary data by entering values for the **Duration** and **StartTime** options shown on the **Edit Custom Properties** dialog box.

These assigned values are passed as input parameters of the new custom property. A value in minutes must be assigned to the **Duration** option.

The **StartTime** option can be left blank. Auto refresh is applied if a **StartTime** value is not specified.

- If a start time is not specified, then the start and end times of the summary period are calculated as:

Start Time = Current Time - Duration

End Time = Current Time

- If a start time is specified, then the start and end times of the summary period are calculated as:

Start Time = StartTime option value

End Time = StartTime + Duration

The **Duration** option can accept a negative number when a start time is specified. When **Duration** is assigned a negative number, the start time input parameter value becomes the end time of the summary period. The start time is calculated using the formula shown below:

End Time = StartTime assigned option value

Start Time = End Time + Duration (in this case it is negative value)

Duration can accept values from 1 minute to 10080 minutes, which is one week. **StartTime** must be within datetime Min and Max Value. During runtime, history summary data is auto refreshed at an interval that is 25 percent of its duration length when a **StartTime** value is not specified.

Show statistical summary data

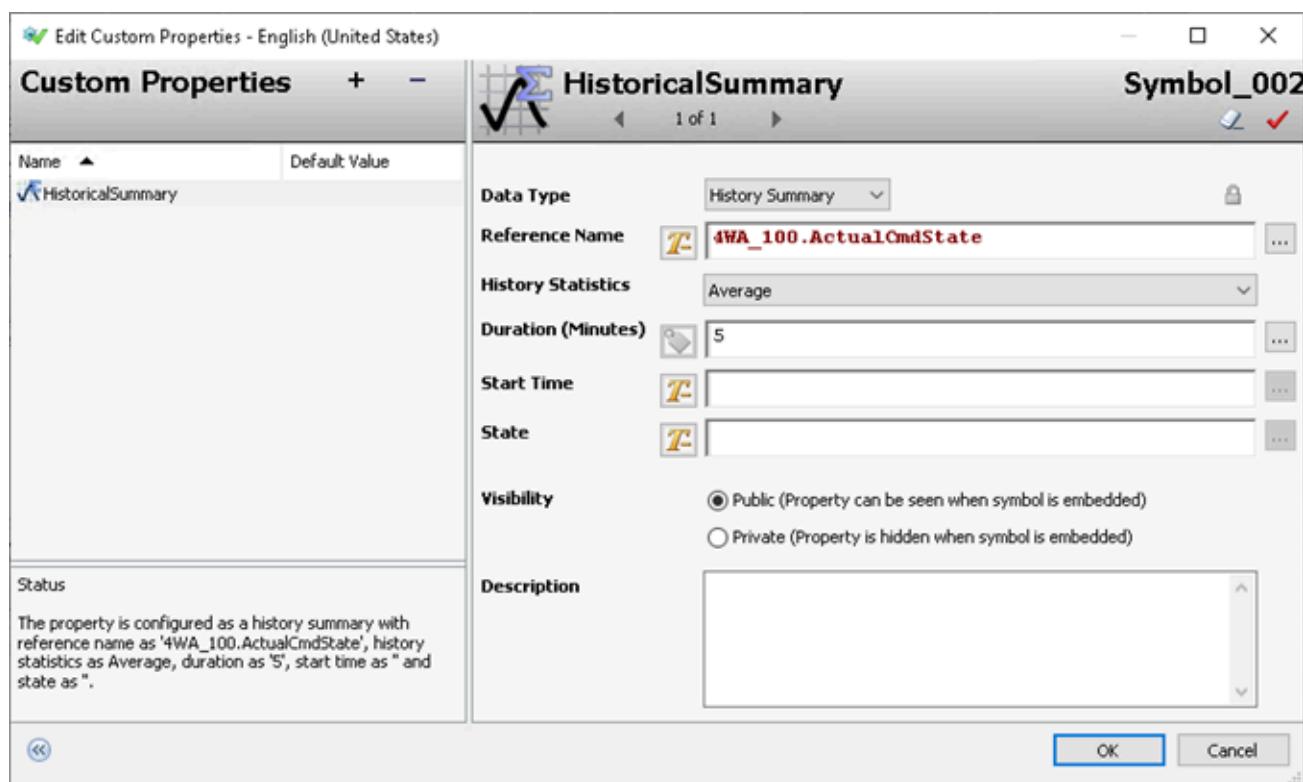
The following procedure explains how to specify custom property options to subscribe to historical statistical data. Before completing this procedure, you must have object or other data that is saved to the Historian and enabled for statistics.

To show historical summary data using custom properties

1. Open the Industrial Graphic Editor.
2. Add a custom property to an Industrial graphic.
 - a. Click the canvas to cancel any selected elements.
 - b. On the **Special** menu, click **Custom Properties**. The **Edit Custom Properties** dialog box appears.
 - c. Click the **Add** icon. A new line is added in the custom properties list.
 - d. Type a name for the new historical summary custom property and press **Enter**.
 - e. You can see the names of the graphic and custom property in the header of the dialog box.
3. Select **HistorySummary** from the **Data Type** field.

Important: The History Summary data type only works with Application Server object attributes intended for AVEVA OMI ViewApps or InTouch HMI managed applications.

The **Edit Custom Properties** dialog box updates to show fields specific to the **HistorySummary** data type.



- Enter a reference to data saved in the Historian in the **Reference Name** field.

The icon to the left of the **Reference Name** field toggles input to the field as **Static Text** or **Expression** or **Reference** mode.

Auto-Detect

The Historian server is auto-detected from the AppEngine on which the reference attribute is running. For example, if the **Reference Name** field is set to UDO.UDA1, the reference is set to the Historian server name configured for the AppEngine on which UDO is running.

Expression

When an expression or reference is typed in the **Reference Name** field, a connection is made to the specified Historian Server. The reference can be an external reference like an object attribute or a custom property.

Note: A reference cannot be made to historical data from an InTouch tag.

- Select the type of historical statistics by selecting an option from the drop-down list of **History Statistics**.

Average is the default type of historical statistic. The following table shows the historical statistics options for analog and state summary data.

Historical Statistics	Analog Historical Data	State Historical Data
Average		
First		
Minimum		

Historical Statistics	Analog Historical Data	State Historical Data
Maximum		
Count		
StdDev		
Integral		
PercentGood		
Percent		
Total		
Last		

6. Set the length of the summary historical period in minutes by entering a value in the **Duration** field.
Acceptable **Duration** values are from 1 to 10080 and the default is 5. Duration can be specified as an integer, an expression, or a reference. For more information about possible **Duration** values, see [Historical summary period](#).
7. Set the start time of the of the summary period in the **StartTime** field.
A start time can be specified as static text, an expression, or a reference. The default start time is the current time.
A time for **StartTime** is optional and can be left blank. Auto refresh is applied if a **StartTime** value is not specified.
For more information about setting a start time, see [Historical summary period](#).
8. Set the type of Historian summary data in the **State** field.
A **State** value can be expressed as an integer constant, static text, an expression, or a reference.
If a string value is provided, then string state summary data is queried from the Historian. If an integer value is entered, the Historian query is for analog state summary data. If a Boolean state summary value is provided, the **State** value must be 0 or 1.
State can be left empty. If empty, the default query is for analog summary data.
To get summary historical data for a Null state, enter "NULL" in the **State** field. The query checks for OpcQuality equal to opcnnull and StringValue "NULL" in the result.

Use binding in custom properties

Application Server object scripting supports a type called "Indirect". It enables you to bind a variable to a reference and read and write to it. This is done using the **BindTo()** method.

Note: The **BindTo()** method binds a variable to a reference as long as the graphic is shown.

For example, the local script variable `ptr` is defined and bound to the reference `ud1_001.Int1`.

```
dim ptr as indirect;
ptr.BindTo("ud1_001.Int1");
```

Within the same script you can use the indirect variable pointer to read from and write to the attribute `ud1_001.Int1`.

Industrial Graphics also use scripting in the same way as the scripting of Application Server.

However, as an Industrial graphic can be embedded into an InTouch window and run anonymously, the time it takes to connect to the reference can be longer than one scan cycle.

For that reason, you cannot use the indirect variable immediately after it is bound to a reference to read from and write to it.

```
dim ptr as indirect;
ptr.BindTo("ud1_001.Int1");
ptr = 42;
```

In the example, the value 42 cannot be written to the reference `ud1_001.Int1` as the binding takes longer.

To avoid this problem, you can modify your Industrial graphic script, using the Industrial Graphic Script Editor to write the value after it is ensured that the binding is complete. The completion of the binding is indicated by the quality of the indirect variable.

- Using artificial delays in the script is not recommended as this may result in script timeouts and would in turn affect UI refresh. For example, do not use a while loop that increments a variable from 1 to <60000. This could cause the runtime application (OMI or InTouch) to go blank or become unresponsive while the script is executing.
- Instead of artificial delays, allow the whole script to continue execution across multiple cycles, without any artificial delays in the script, and set the script's expression to false once the quality of all the attributes is good.

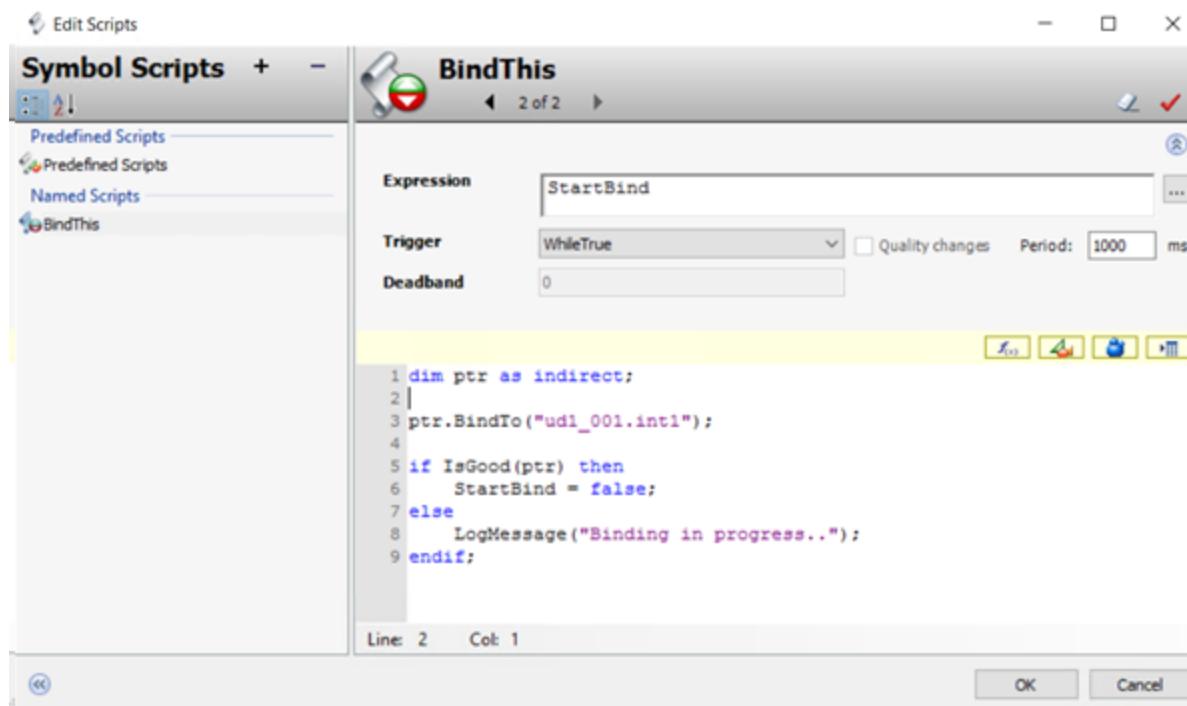
Create a named script, with a trigger set to 'While True' and query for the quality and use the indirect variable to read from and write to the reference when its quality is good. This script will try again every second until the indirect attribute returns with good quality. If the quality is good, then the script exits from the while loop. Once the quality is confirmed, you can use the indirect variable reference in other scripts or button animations.

Note: Make sure to include an exit condition in your script, so that the script does not hang if the binding cannot be made.

The following example script shows you how to do this:

- Expression: Create a Boolean custom property `StartBind` with an initial value of `True`.
- Trigger: `WhileTrue`
- Period: `1000 ms`

```
dim ptr as indirect;
ptr.BindTo("ud1_001.Int1");
if IsGood(ptr)then {if quality not good}
StartBind = false;
else
LogMessage("Binding in progress..");
endif;
```



Note: Similar behavior can occur when you try to bind to a reference of an object that is hosted on a different AppEngine.

Change the expression or reference of a custom property at runtime

You can change the expression or reference of a custom property at runtime by calling the `SetCustomPropertyValue()` method on the graphic using a client script:

```
SetCustomPropertyValue(System.String name, System.String value, System.Boolean isConstant);
```

You can select this method using the Element Browser from within the Industrial Graphic Editor.

This method has three parameters:

- *name* - Name of the custom property to be modified on the graphic. This parameter is of type string, and it can be a reference or a constant.
- *value* - The new value to be set. This parameter is of type string, and it can be an expression, reference, or constant. If the value is given in quotes (""), then the value is considered a constant. If the value is given without quotes, then the value of the expression is considered a reference.
- *isConstant* - A flag that indicates whether the new value will be evaluated as a constant or a reference. This parameter is of type Boolean. If it is set to True (1), then the new value will be treated as a constant. If it is set to False (0), then the new value will be treated as a reference. This parameter only applies when the *value* parameter is a reference or constant and the custom property specified in the *name* parameter is a string or time type. This parameter has no meaning if the custom property is an integer, float, Boolean, or double type.

Note: The *isConstant* parameter does not override the type of input for the *value* parameter. The *value* parameter itself can be either a constant or a reference depending on whether it is enclosed in quotes. The *isConstant* parameter is only determining how the actual value (coming from the *value* parameter) is evaluated.

The whole expression or reference of the custom property is replaced with the new value, regardless if it is overridden or not. No partial replacement is supported.

Only public custom properties on the graphic can be changed.

When the method executes, it overrides any modifications done by previous IOSetRemoteReference() calls from a native InTouch script.

For an example of configuring the custom property as a reference, say you have a Motor_001 object with the string attribute name "State" that stores the current state of the motor ("Running" or "Stopped"). You also have an Industrial graphic that has the string data type custom property "MotorState." The following script code will set the MotorState custom property to Motor_001.State in runtime:

```
GraphicA.SetCustomPropertyValue("MotorState", "Motor_001.State", False);
```

As a result of the call, the function will set the string custom property GraphicA.MotorState to "Motor_001.State" as a reference. The string custom property GraphicA.MotorStatus will resolve that reference and update its value with the reference value ("Running" or "Stopped").

For an example of configuring the custom property as a constant, say you have a Motor_001 object with the Boolean attribute name "State" that reflects the current state of the motor (True or False). You also have an Industrial graphic that has the string data type custom property "MotorState." The following script code causes the MotorState custom property to hold the state of equipment—"Running" or "Stopped"—as text based on the value returned for Motor_001:

```
IF Motor_001.State THEN
GraphicA.SetCustomPropertyValue("MotorState", "Running", True);
ELSE
GraphicA.SetCustomPropertyValue("MotorState", "Stopped", True);
ENDIF;
```

As a result of the call, the function will set the string custom property GraphicA.MotorState to "Running" or "Stopped," depending on the value of Motor_001.State.

About using an External Content item

External content refers to media content outside of a Galaxy, such as an html document, that can be linked to a ViewApp and displayed during runtime.

External content is associated with a ViewApp by placing an External Content item on a layout pane during design time. An External Content item is created and then configured with the External Content editor to identify the location of documents and their content types. An External Content item is managed within the Visualization folder hierarchy of content items, which is visible in the System Platform IDE and in the **Toolbox** or **Assets** areas of the Layout and ViewApp editors.

Create an External Content item

You create an External Content item from the **Visualization folder** of the System Platform IDE. Initially, the External Content item defaults to a set of values that can be changed using the External Content editor.

To create an External Content item

1. Open the IDE and select the **Visualization folder** tab.
2. Select a folder within the **Visualization folder** if you want to create a new External Content item at a specific location.

3. On the **Home** ribbon, in the **Create** area, select **External Content**.

You can also create an External Content item by other methods:

- **Keyboard Shortcut**

Press Ctrl + Shift + C

- **Accelerator Keys**

Press Alt + G + N + C

- **Shortcut Menu**

Right-click a folder of the **Visualization folder**. Select **New**, then **External Content**.

A new **External Content** item is created in the **Visualization folder**.

 ExternalContent_001 * * *

The name of the new item is ExternalContent followed by a three-digit number.

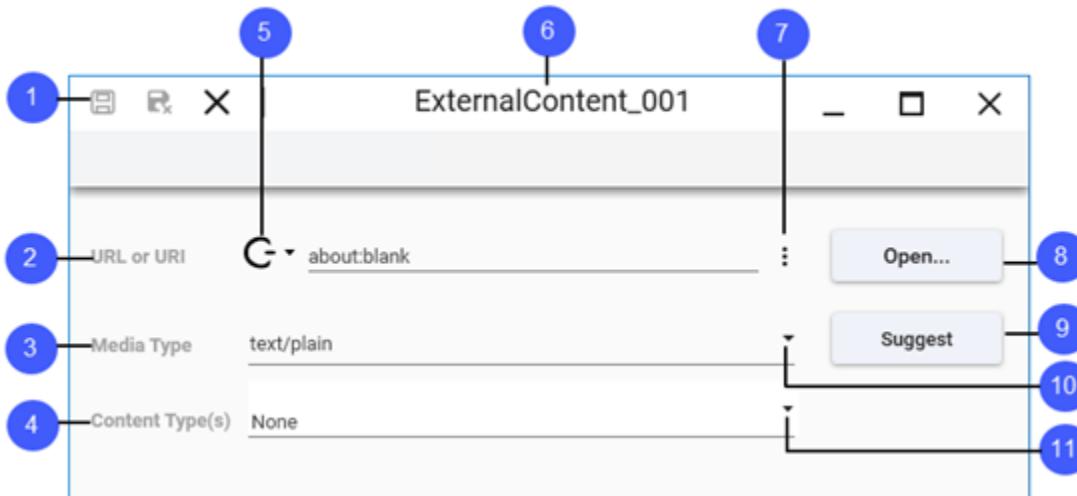
4. Rename the External Content item.

After you create an External Content item, it must be configured to associate it to specific media content. See [About the External Content editor](#).

About the External Content editor

You use the External Content editor to configure an External Content item. Before you configure an External Content item, you must know:

- Location of the external media
- Type of media
- Content type(s) of the layout pane that can host the External Content item



1	Commands to save, save and close, and close the External Content editor.
2	URL or URI that specifies the location of external media using a standard format. A warning message appears if the external content is located on the local computer hosting System Platform.
3	A media type is a two-part identifier that specifies the type of application required to process or view remote content. A media type can be entered in the field or selected from a drop-down list. A warning message appears if a media type is specified that does not have an associated viewer application.
4	Content type assigned to the external media that enables ViewApp algorithms to place content in specific panes during runtime.
5	Icon with drop-down menu to select if the URL/URI field will be a static string or a reference/expression. The icon changes, depending on this selection.  Constant  Reference If you select Reset from the drop-down menu, any value entered in the field is discarded, and the type reverts to Constant with a default value of "about:blank."
6	Name of the External Content item.
7	Drop-down list with options to browse for an external content file or specify a URL using HTTP or HTTPS. When URI/URL is set to Reference or Expression , the drop-down list changes to the Galaxy Browser icon (...), and you can use it to select references.
8	Open validates the location of external media specified in the URL or URI field, if a Constant value has been selected for the field. An attempt is made to display the external media in an application assigned as the default by the operating system, not the app specified for the external content media type. The Open button is disabled if a Reference or

	Expression has been selected. A warning message appears if the content cannot be found at the location specified in the URL or URI field.
9	Suggest is optional. When selected, it automatically selects a media type based on the entry in the URL or URI field, if a Constant value has been selected for the field. The entered value can be changed if the suggestion does not match the expected media type. The Suggest button is disabled if a Reference or Expression has been selected. Note: Ensure the suggested media type is correct. The external media does not appear if the media type is incorrect.
10	Drop-down list of commonly used media types. Media types that have apps in the Galaxy capable of servicing them are shown in bold text.
11	Drop-down list of content types that can be used at preview\runtime to assign the external content to panes of a matching content type. Note: None is not a content type. Instead, it simply means a content type has not been assigned.

Configuration Prerequisites

Before starting the steps to configure an External Content item, ensure the following prerequisites have been met:

- If you are placing media files on a remote computer in your network, ensure connectivity between the computer where the ViewApp will run and the computer where the remote media is stored.

The example screenshot shown above shows a sample configuration with the URI set to a standard Windows mapped drive. For this configuration to work successfully, it is important that the computer to which the ViewApp is launched from has the same drive letter mapped to a location having the referenced content.
- If you intend to show videos within a running ViewApp on a computer running a supported version of Windows Server 2012, ensure the Desktop Experience feature is enabled. Desktop Experience includes required media features that are not enabled by default on the different versions of Windows Server 2012 supported by System Platform.

Configure an External Content item

Each External Content item has three properties that must be assigned values:

- **URL or URI**
- **Media Type**
- **Content Type(s)**

An External Content item is modified from the External Content editor. The editor can be opened by double-clicking an item from the Visualization folder. Also, an External Content item that has been placed onto a layout pane can be edited from the Layout and ViewApp editors by the selecting the item from the Actions list.

To configure an External Content Item

1. Open the IDE and select the **Visualization** tab to show the External Content items available in the Galaxy.
2. Double click an External Content item to open it in the External Content editor.
3. If you are using a static string for this **URL or URI** value, enter the value. The value you enter is the location of the external content specified by a URI-formatted string. A media location must be specified.
 - You can browse for the media file by selecting the vertical dots icon to the right of the field. A drop-down list includes a **Browse for file** option that enables you to browse your network and select a media file. The **URL or URI** field updates and shows a formatted URI path to the file.
 - You can also enter a URL to a web site by selecting the **http://** or **https://** options from the drop-down list. The **http://** and **https://** options assist the user when entering URLs by prefixing the URLs with the selection. See [More information about specifying a URI](#) for information about the format of a URI string.
4. Select the **Constant/Reference** icon if you will enter a reference in the URL or URI field, and select **Reference or Expression**.
 - The vertical dots to the right of the data entry field will change to the Galaxy Browser (...) icon, and you can browse for the reference you want to use.
 - If you choose to use a **Reference or Expression**, the **Open** and **Suggest** buttons are disabled.
5. If you are using a static string (Constant), select the **Open** button to validate the value you entered.

The **Open** button attempts to launch the appropriate media application and display the external content specified in the **URL or URI** field using the default application assigned from the operating system. For example, an external video file will play in Windows Media Player.

The **Open** button is disabled if you are using a **Reference or Expression**, and the External Content object subscribes to the reference at runtime.

6. Select the type of media from the drop-down list of the **Media Type** field.

The value you enter is the MIME type of the content. To be viewable, the media type must be supported by an app within the galaxy. A media type must be specified.

A media type is a two-part name that identifies the type of app required to display the external media within a running ViewApp. The bold media type names listed in the **Media Type** field's drop-down list have apps that can play the external media. Users may also type in a media type that is not available from the drop-down list. However, for the external content to be displayed at runtime, an app must be imported that can service the exact media type.

If you want, selecting the **Suggest** button updates the **Media Type** field with a suggested media type based on the file extension of the external media entered in the **URL or URI** field. Also, it is based on file extension if it's a file in files system (file:\\) or based on the header from server response if it's a http or https.

The **Suggest** button is disabled if you are using a **Reference or Expression**.

See [Specify media types for External Content](#) for information about supported media types.

7. Select the type of content from the drop-down list of the **Content Type(s)** field.

The value you enter is used by content-placement algorithms within AVEVA OMI during runtime to determine the placement of content when AutoFill or ShowContent calls are invoked.

See [Set pane properties: content types](#) for the types of content that can be assigned to layout panes.

8. Save your changes and exit from the External Content item.

Specify the URL of External Content

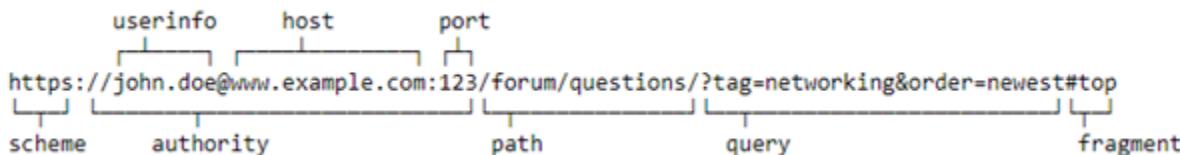
In the context of External Content, a Uniform Resource Identifier (URI) is a string that identifies the pathway to specific content that can appear in a running ViewApp.

All URIs adhere to predefined syntax rules.

- Generic URI syntax consists of a hierarchical sequence of components

```
URI = scheme:[//authority]path[?query][#fragment]
```

Example



- Each URI begins with a scheme name followed by a colon
Examples of common scheme names include http:, https:, and file:
- An optional authority component preceded by two slashes (//)
 - Use information component consisting of a user name and optional password preceded by a colon followed by an at symbol (@)
//username:password@
 - Host subcomponent consisting of a registered hostname or IP address
 - A path component consisting of a sequence of path segments separated by a slash (/). A path is always defined for a URI, though the defined path may be empty
 - An optional query component preceded by a question mark (?), containing a query string of non-hierarchical data.

For more information about the format of a URI, see <https://tools.ietf.org/html/rfc3986>

Example URLs

The following examples show common URIs that identify External Content within a Galaxy

- ftp://ftp.is.co.za/rfc/rfc1808.txt -- ftp scheme for File Transfer Protocol services
- file:///Z:/Media/Runtime%20Language%20Switching.mp4
- http://www.exampleserver.com/documents/index.html

Specify media types for External Content

A media type is a standard two-part string analogous to a MIME type that identifies external content file types and their format. The media type is used to identify external content and the type of app required to display media during runtime.

A media type consists of a type and a subtype, which is further structured into a tree.

```
type "/" [tree "."] subtype ["+" suffix] *[ ";" parameter]
```

For more information about the format of media types, see <https://www.iana.org/assignments/media-types/media-types.xhtml>

System Platform provides a default text/html media type that includes an associated app to display web content.

The drop-down list of the **Media Type** field shows media types in bold text to indicate a viewer app is available in the Galaxy that can service external content with these media types. Other listed media types that appear in plain text require an app to be created and imported into a Galaxy to play the specified media.

Link an External Content item to an object

After you have created and configured an External Content item, you can link it to an object template.

External Content items can be linked only to objects as opposed to graphics, which can be linked to or owned by an object. All External Content items reside in the Visualization folder. A single External Content item can be linked to multiple objects.

To link an External Content item to an object template

1. Open an object template in the **Object Editor**.
2. Select the **Attributes** tab.
3. In the **Content** pane, select the **Link Content**  button.
The Galaxy Browser opens.
4. Navigate to the folder that contains the **External Content** item to be linked.
5. Select the item, then click **OK**.

The item is added to the object and appears in the **Content** tab.

Note: After linking an External Content item, you can modify it by selecting it and then selecting **Edit**. However, any changes you make will apply to all objects that link to the item.

6. Save your editing changes to the object template and exit from the Object editor.
7. Create an instance of the linked object template.
8. Create a layout which contains pane with a Content Type that matches the content type of the External Content item.
9. Add a navigation tree to the layout
10. Create a ViewApp that incorporates the layout.
11. Deploy the ViewApp.
12. In runtime, navigate to the object or any instance created from template.

The linked External Content item will auto-fill the matching pane to show the external content.

Associate External Content to a ViewApp

After you configure an External Content item, you must place it onto a pane of a layout incorporated into a ViewApp. You can place an External Content item into a pane from either the Layout or ViewApp editors.

External Content items are listed in the **Toolbox** or **Assets** tabs of either editor.

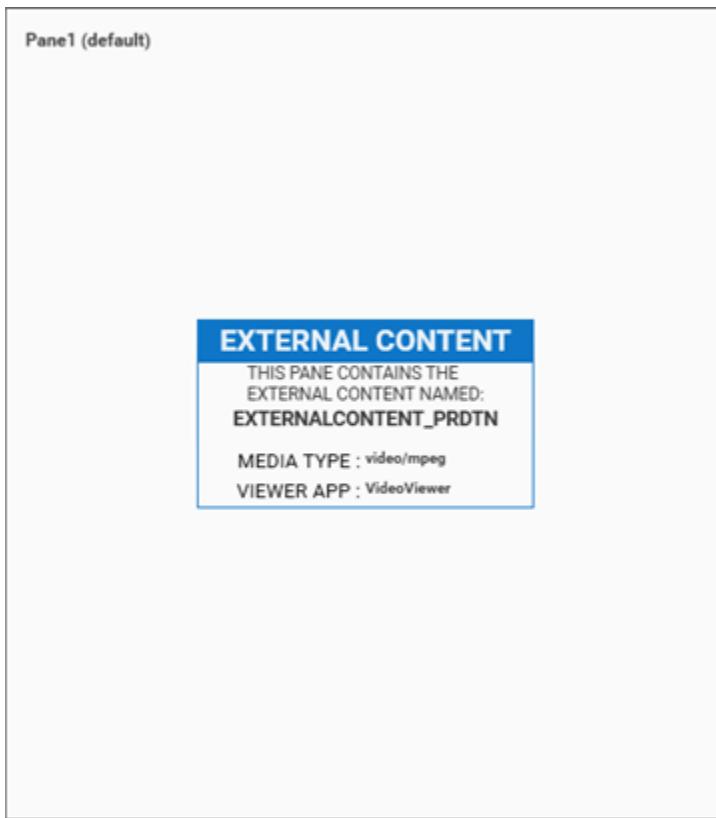
Note: You cannot select an External Content item from the Industrial Graphic Editor.

The External Content item must have been assigned a valid location for a document and a viewer application must be associated with the assigned document media type.

If you attempt to place an External Content item onto a pane that does not have an assigned viewer application to display the media type, a warning message appears after placing the item onto a pane. The ViewApp will not show the document, although the ViewApp still continues to run during runtime.

To associate external content to a ViewApp

1. Open the System Platform IDE and select the **Toolbox** tab to see the list of layouts.
2. Select a layout that you want to insert external content in and open it in the Layout editor.
3. Select the **Toolbox** tab of the Layout editor to show the list of content accessible from the Galaxy.
4. Locate the External Content item to place in the layout.
5. External Content thumbnails appear beneath of the list of content shown in the **Toolbox** list.
6. Drag an External Content thumbnail from the **Toolbox** area onto the desired layout pane.



The External Content thumbnail appears in the pane containing the name of the External Content item, the media type of the external content, and the viewer app that will display the media during runtime. Also, the layout **Actions** area shows the name of the External Content item and the pane that it was placed in.

Editing an External Content Item After Being Placed onto a Pane

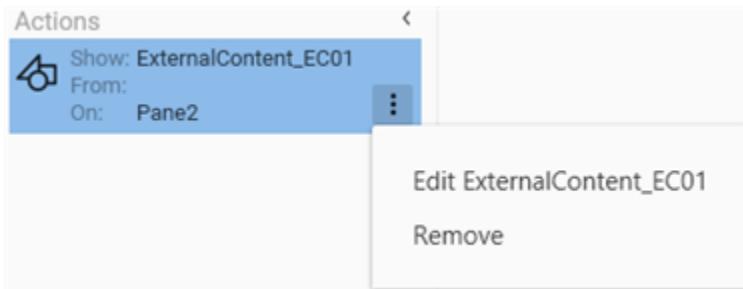
An External Content item can be modified after being placed in a layout or a ViewApp, which automatically propagates the changes

To edit an External Content item in a layout

1. If necessary, open the layout containing an External Content item.

2. Locate the item in the layout **Actions** list.
3. Select the **Options** icon shown at the right of the item in the **Actions** list.

A drop-down list shows an Edit External Content command.



4. Select the edit command.

The External Content editor opens the item for editing.

5. Update the values assigned to the item and save your changes.

The External Content thumbnail in the pane updates to reflect the changes made to the item.

Manage external content

An External Content item can be managed from the IDE using standard options from the right-click shortcut menu or from the **Edit** options of the IDE menu bar.

A ViewApp containing apps designed to display external content can be exported from one Galaxy and imported to a ViewApp in another Galaxy. The External Content editor is capable of recognizing a new media type from the imported app.

An External Content item can be:

- Exported to an aaPKG file
- Imported from an aaPKG file
- Included in a Galaxy backup
- Restored from a Galaxy backup
- Included in an export of layouts or ViewApps that have External Content items associated with them

Create an app for external content

Apps that are designed to load external content when a ViewApp starts must be able to recognize a URI. App developers also need to include a list of the Media Types that can be serviced by the app. An app's supported media types must be specified in an AppManifest.xml file. For more information about specifying media types in the AppManifest.xml file, see the *AVEVA OMI Developer Guide*.

Note: When there is more than one application in a Galaxy capable of servicing a specific media type, the more recent imported application will be used by the ViewApp. Users cannot select an application for a media type.

After importing the app, these Media Types will be added to an internally maintained Media Types registry. Panes will refer to this registry when trying to open External Content. The app that services the Media Type associated to the External Content will first be launched and then the external content loaded.

The overall steps in the workflow to import an app are described in the following list.

1. Create an AVEVA OMI app that recognizes a URI and supports one or more defined media types.
2. Enclose the app in an ArchestrA aaPKG file and export it to a location accessible to your development computer.
3. Open the System Platform IDE.
4. Import an app in an aaPKG file that is designed to load and display a specific type of external media.
5. Create an External Content item.
6. Edit the External Content item.

The **Media Type** field of the External Content item lists the media types supported by the imported app. The External Content editor must be closed and reopened if an app containing an unrecognized media type is imported while the External Content editor is open. Reopening the External Content editor lists the new media type.

Script Editor

AVEVA OMI supports two types of scripts:

- Simple scripts can perform assignments, comparisons, simple math functions, and similar actions.
- Complex scripts can perform logical operations using conditional branching with IF-THEN-ELSE type control structures.

For detailed information about scripting and available script functions, see [Scripting Reference](#) located in the *Application Server Help*.

Add a script to a graphic

You can associate scripts with your graphics. Scripts can add animation to a graphic or its elements that can be executed in runtime.

Caution: If you configure scripts that affect more than element and graphic animation, the script processing may affect performance.

You can:

- Configure the predefined scripts of a graphic.
- Add named scripts to a graphic.
- Edit existing named or predefined scripts in a graphic.
- Rename named scripts in a graphic.
- Remove named scripts from a graphic.
- Substitute references in named or predefined scripts.
- Use element methods in named or predefined scripts.

The autocomplete features available in the Application Server QuickScript editor are also available in the Industrial Graphic Editor script editor. For more information about Application Server scripting, see *the*

Application Server Scripting Guide.

The topics in this section describe using the SignedWrite() function for secured and verified writes. For information about all other aspects of associating scripts with graphics, see [Scripting Reference](#).

Working with the SignedWrite() function for secured and verified writes

This section provides information about the SignedWrite() function, SignedWrite() runtime behavior, scripting tips, and in-depth script examples.

For SignedWrite() scripting information including script syntax, parameters, and basic script examples, see the *Application Server Scripting Guide*.

You can write to an Automation Object attribute that is configured for Secured Write or Verified Write security classification by means of the Industrial Graphics SignedWrite() script function.

The SignedWrite() function can be used only in client scripts, not in Application Object scripts, and only on Attributes that have been configured for Secured Write or Verified Write. Attempting to use the function on an Attribute not so configured will result in an error message at runtime.

SignedWrite() runtime behavior

At runtime, the SignedWrite() function does the following:

1. Checks the target Attribute for Signed Write or Verified Write configuration.
If not so configured, the following error message appears: **Operation Failed. Attribute does not have the correct security classification.**
2. Checks Galaxy security.
If the Galaxy is not secured, the following error message appears: **Operation Failed. Galaxy is not secured.**
3. There are several ways to write to an Attribute configured with Secured Write or Verified Write security classification, it is possible to have multiple SignedWrite() and other Secured/Verified writes pending from the same script, or even from multiple scripts running side-by-side.
4. Determines which dialog is required—Secured Write or Verified Write—and pops up the appropriate dialog box.
If Smart Cards are enabled, the function displays different versions of the Secured Write and Verified write dialog boxes.
5. Lists the predefined comments, if any, from the configured Predefined Comments list. Up to 20 comments are supported.
6. Enables comment editing if the Comment_Is_Editable parameter is configured and comment enforcement is other than PredefinedOnly.
7. Acquires the user credentials and authenticates them.
If the user credentials are invalid, an error message appears. The function will attempt the write only if the credentials are valid.
8. Checks if comment enforcement is mandatory, and displays an error message if the comment is empty.
9. Performs the write if user credentials are valid and the comment entry satisfies the comment enforcement parameter.
10. Provides a return status.

11. Following a Secured or Verified Write a security Event is written to the event log, including the signee name, verifier name, if any, Type of write: "Secured Write" or "Verified Write", Comment, if any entered by user, Reason Description, if any provided, Field Attribute description, if any, or the Short Description of the Application Object, if no Field Attribute description exists.

Each call to SignedWrite() is distinct from any other. The success or failure of any individual write does not affect other attempted writes.

Entering user credentials for SignedWrite() is distinct from logging on to the client application. The user can modify attributes configured with Secured or Verified Write even if another user is logged on, without affecting the session of the logged-on user.

SignedWrite() script execution sequence at runtime

The SignedWrite() function goes into a queue and the script continues executing. The function is queued for operator entry. The script may complete prior to the operator completing the Secured or Verified Write operation.

By contrast, the SignedAlarmAck() script function executes completely synchronously, and waits for user input before proceeding to the next line in the script.

SignedWrite() scripting tips

Bound references in SignedWrite()

If the Attribute parameter string evaluates to the name of a Custom Property and that Custom Property is a bound reference to an Attribute, the SignedWrite() function will write to that indicated Attribute. The Attribute must have the security classification of Secured Write or Verified Write.

The SignedWrite() function supports Custom Properties that are nested bound references. That is, if the string evaluates to the name of a Custom Property and that Custom Property is a bound reference to another Custom Property which itself is a bound reference, the SignedWrite() function will follow through the chain of bound references until it finds an item that is a value. If that item is an Attribute that has the security classification of Secured Write or Verified Write, the SignedWrite() function will write to that item.

SignedWrite() in WhileTrue, WhileFalse, or periodic type scripts

Using the SignedWrite() function with WhileTrue, WhileFalse, or Periodic type scripts can repeatedly execute the script, causing another secured write dialog box to pop up with each trigger. We do not recommend using the SignedWrite() function with WhileTrue, WhileFalse, or Periodic types.

SignedWrite() with OnShow and OnHide scripts

We do not recommend using the SignedWrite() function with OnShow and OnHide scripts. This can cause issues with window functionality, including the window title bar, windows losing correct focus, and windows opening on top of one another.

Examples of using the attribute parameter in the SignedWrite() function

Working from the overall syntax of the SignedWrite() function, the script examples in the following table illustrate a number of approaches to using the Attribute parameter in the SignedWrite() function.

The following String, Boolean, and Integer user defined attribute conditions apply to the script examples:

- User Defined object UD1_001
- String attribute UD1_001.udString1 with the value "WW". Security classification is set to Secured Write.
- Boolean attribute UD1_001.secBool1 with the value false. Security classification is set to Secured Write.
- Integer attribute UD1_001.secInt1 with the value 24. Security classification is set to Secured Write.
- User Defined object UD2_002

String attribute UD2_002.udString2 with a value "UD1_001.udString1" The following custom property conditions apply to the script examples:

- String custom property CP1 with a reference to UD1_001.udString1
- String custom property CP2 with a value "UD1_001.udString1"
- String custom property CP3 with a value "UD1_001"
- Boolean custom property CP4 with a reference to UD1_001.secBool1
- Integer custom property CP5 with a reference to UD1_001.secInt1
- String custom property CP6 with a reference to an attribute on Owning Object UD1_001 as me.udString1

Script example	Function and result
SignedWrite("CP1", "AVEVA", "using redirect", true, 0, null);	Uses the CP1 reference UD1_001.udString1 and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
SignedWrite(CP2, "AVEVA", "using string value", true, 0, null);	Resolves CP2 string value "UD1_001.udString1" to a reference and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
SignedWrite(CP3+".udString1", "AVEVA", "using string expression", true, 0, null);	Resolves the string "UD1_001.udString1" to a reference and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
SignedWrite("UD1_001.udString1", "AVEVA", "using constant string", true, 0, null);	Resolves the string "UD1_001.udString1" to a reference and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".
SignedWrite(UD2_002.udString2, "AVEVA", "using attribute containing string", true, 0, null);	Resolves the UD2_002.udString2 string value "UD1_001.udString1" to a reference and pokes to it the value "AVEVA". Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".

Script example	Function and result
SignedWrite("CP" + "1", "AVEVA", "using redirect from string expression", true, 0, null);	<p>Resolves the expression to "CP1" to use the CP1 reference UD1_001.udString1 and pokes to it the value "AVEVA".</p> <p>Result: The value in UD1_001.udString1 will change from "WW" to "AVEVA".</p>
SignedWrite("CP4", true, "using redirect", true, 0, null);	<p>Uses the CP4 reference UD1_001.secBool1 and pokes to it the value true.</p> <p>Result: The value in UD1_001.secBool1 will change from false to true</p>
SignedWrite("CP5", 37, "using redirect", true, 0, null);	<p>Uses the CP5 reference UD1_001.secInt1 and pokes to it the value 37.</p> <p>Result: The value in UD1_001.secInt1 will change from 24 to 37</p>
SignedWrite("CP6", "AVEVA", "using redirect using relative reference", true, 0, null);	<p>Uses the CP6 reference me.udString1 and resolves it to UD1_001.udString1 and pokes to it the value "AVEVA".</p> <p>Result: That the value in UD1_001.udString1 will change from "WW" to "AVEVA"</p>

Configure the SignedWrite() script function

You can create a dashboard application to automate routine use of Secured and Verified Write by means of the SignedWrite() function.

To configure the SignedWrite() script function

1. Open the System Platform IDE.
2. Create a graphic and associate it with an attribute configured with Secured Write or Verified Write. For more information on associating attributes with graphic, see the Application Server User Guide, "Creating and Working with UDAs" topic.
3. Add the SignedWrite script function to the graphic. The following editor detail shows the buttons configured with scripts in the applied example:
4. Configure the scripted functionality you require. Scripts for the buttons shown in the example are as follows:
 - a. Hard-coded DataUDO.SecUDA: The following example sets the value of 23 to DataUDO.SecUDA. The user optionally can enter a comment, but no pre-defined comment list is available.
`DataUDO.RetStatus=SignedWrite("DataUDO.SecUDA", 23, "Set the Value", True, 0, null);`
 - b. Attribute Pointer has DataUDO.SecUDA: The source to be written to is passed as a parameter to the function. Attribute_Pointer is a custom property whose value is set to DataUDO.SecUDA.
 The following example sets the value of 23 to DataUDO.SecUDA. The user optionally can enter a comment, but no pre-defined comment list is available.
`DataUDO.RetStatus=SignedWrite(Attribute_Pointer, 23, "Set the Value", True, 0, null);`

- c. Attribute Pointer and Pre-Defined List: The pre-defined comment list is an array. This example extends the functionality of example b to force the user to enter a comment (Comment_Enforcement parameter set to 1) and also presents a pre-defined set of comments linked to the DataUDO.PreDefComments[] array.

The following example will set the value of 23 to DataUDO.SecUDA. The user must enter a comment and may use one from the pre-defined comment list.

```
DataUDO.RetStatus=SignedWrite(Attribute_Pointer, 23, "Set the Value", True, 1,  
DataUDO.PreDefComments[]);
```

- d. Variable Array: The predefined list is a pointer to an array. This example extends the functionality of example c to force the user to enter a comment (Comment_Enforcement parameter set to 1) and also presents a predefined set of comments linked to DataUDO.PreDefComments[] array.

The value of custom property CP1 is "DataUDO.PreDefComments[]".

The following example will set the value of 23 to DataUDO.SecUDA. The user must enter a comment and may use one from the pre-defined comment list.

```
dim xInd as Indirect;  
xInd.BindTo(CP1);  
DataUDO.RetStatus=SignedWrite(Attribute_Pointer, 23, "Set the Value", True, 1,  
xInd);
```

- e. All Parameters Variable: The predefined list array is built into the script. All parameters are passed as variables.

The following example will set the value of 23 to DataUDO.SecUDA . The user must enter a comment and may use one from the pre-defined comments list.

```
dim myList[5] as string;  
myList[1] = "Batch Accepted";  
myList[2] = "Batch Rejected";  
myList[3] = "Batch on Hold";  
myList[4] = "Batch Resumed";  
myList[5] = DataUDO.PreDefComments[4];  
DataUDO.RetStatus=SignedWrite(Attribute_Pointer, SignedWrite_Value_Ptr,  
SignedWrite_Reason, Enable_Edit_Comment, Comment_Options, myList[]);
```

How to write to an attribute configured for Secured Write or Verified Write security classification

There are several ways to write to an Attribute configured for Secured Write or Verified Write security classification.

- Any assignment in a script that sets the value of the Attribute, such as

```
A=B;
```

where A references an Attribute that is configured for Secured Write or Verified write security classification.
- Any action on an animation graphic that alters the value of an Attribute that has Signed Write or Verified Write security configuration, such as a user input, a slider, an up/down button on a counter, and other such actions.
- A script that uses the SignedWrite() function.

For information specific to the SignedWrite() function, see [Working with the SignedWrite\(\) function for secured](#)

and verified writes.

Example: how to change element properties using scripts

You can change some properties of elements using scripting. This lets you configure additional runtime behavior to your elements in addition to design-time animation of those elements.

When you write scripts for the graphic or for one of its elements, you can use your HMI's attribute/tag browser to show and select a:

- Property of an element
- Custom property of the graphic

If a reference is not unique, the following order applies:

1. Dimensioned variable references
2. Graphic properties references
3. Custom property references
4. Object attribute references

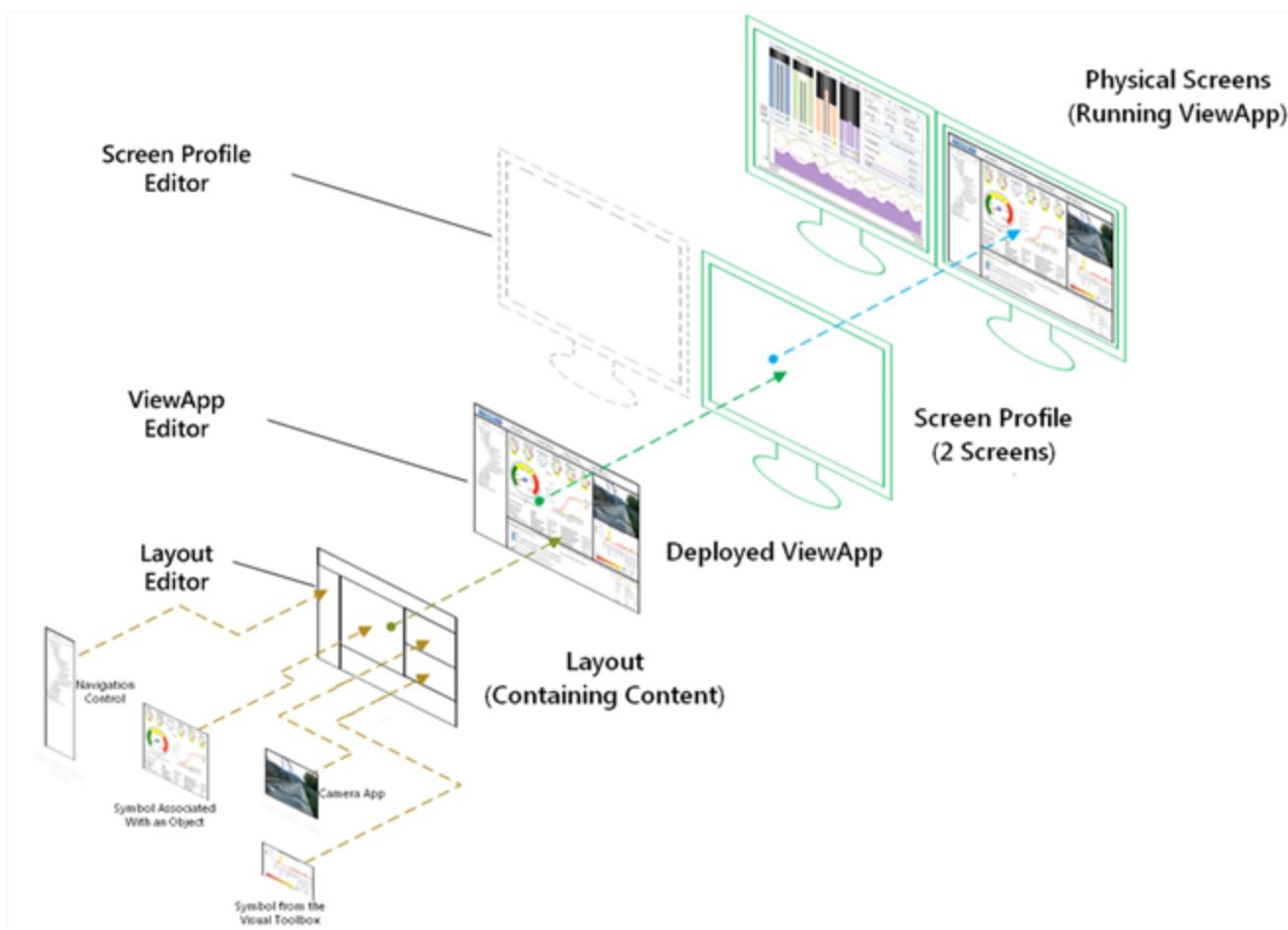
To select an element property or a graphic custom property

1. Open your HMI's attribute/tag browser. It shows the element names and the properties of the selected element.
2. Select an element or graphic from the list. The right pane shows the accessible properties of the selected element or graphic.
3. Select a property from the right pane and click **OK**. The reference appears in the script window.

ViewApp editor

Typically, you build standard objects, graphics, and controls during the design phase of your project. These standards are reusable components that can be inserted into a ViewApp in an almost assembly line process. The following figure shows the major steps of assembling a ViewApp with objects, screen profiles, layouts, and content with the ViewApp editor.

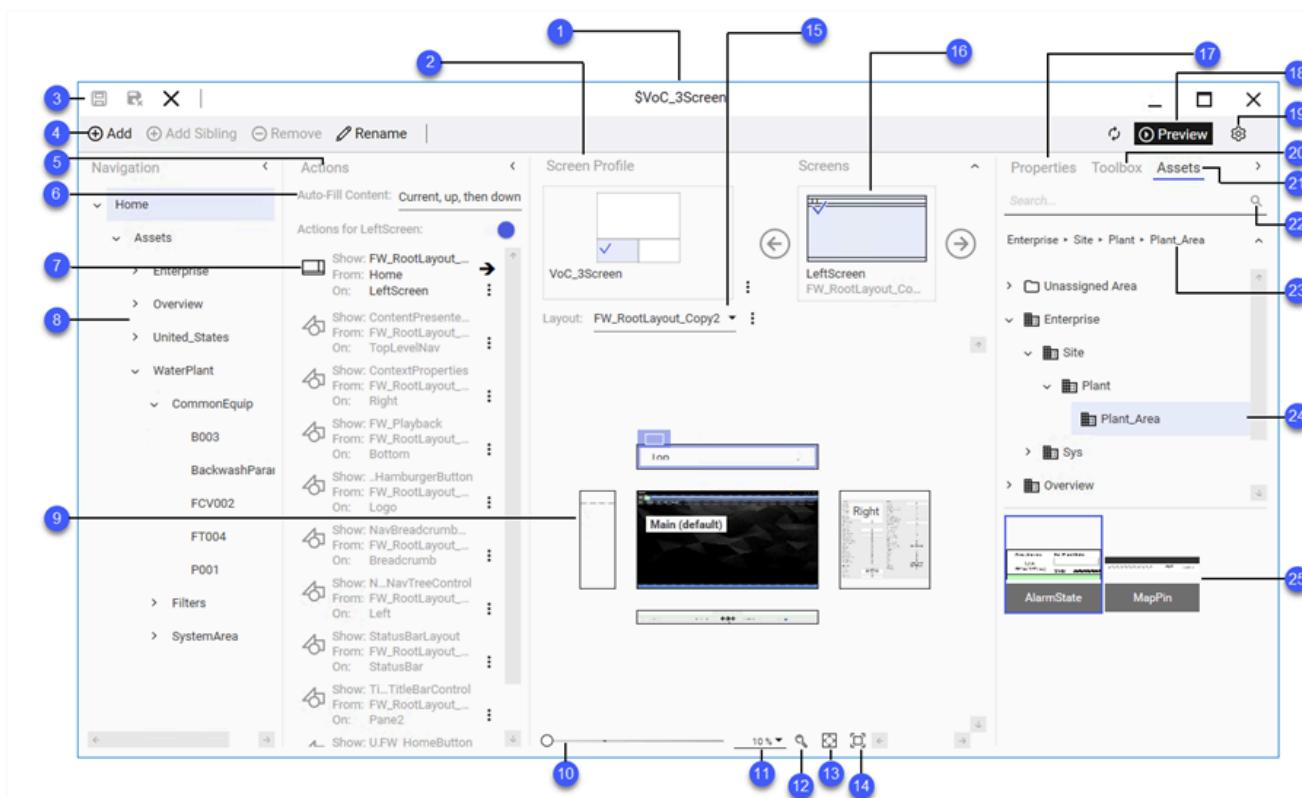
See ViewApp for information about using the ViewApp object.



About the ViewApp Editor

You use the ViewApp Editor to integrate all of the component parts to create a ViewApp. The ViewApp Editor incorporates the functionality to add, replace, and edit screen profiles, layouts, and content.

The figure below shows the various components of the ViewApp Editor



1	Selected Screen Profile	Name and thumbnail representation of the current screen profile associated with the ViewApp being edited. The screen thumbnail with a green background is the active screen in the profile. When a screen profile contains multiple screens, individual screens can be selected. When an individual screen is selected, the screen name appears in the Actions list and the layout assigned to the screen is shown in the editing area of the ViewApp Editor.
2	Custom Navigation Commands	Commands to manage custom navigation items placed in the navigation tree. These commands enable you to add, remove, and rename navigation items included in the ViewApp's navigation model.
3	Title Bar	The Title Bar shows the name of the ViewApp being edited. An asterisk (*) immediately after the

		ViewApp name indicates the edited ViewApp contains changes that have not been saved. The phrase Read Only appears when the ViewApp is open in the ViewApp Editor in read only mode.
4	Command Bar	Commands to save, save and close, and close the ViewApp currently edited with the ViewApp Editor.
5	Actions List	List of navigation actions associated with navigation items and the panes of a layout.
6	Auto-Fill Content	Drop-down list of Auto-Fill options that determine the different types of navigation modes to search for valid content that can be shown in a pane.
7	Action Items	Each action item lists the name of content or container placed in a pane. The action item also shows the source and the pane within the layout.
8	Navigation Tree	Hierarchical tree view of navigation items associated with objects listed in the Model view of the Template folder. Custom navigation can be used to add or remove items from the navigation tree.
9	Pane Content	Thumbnail representation of the content shown in a pane based on the navigation item selected in the navigation tree.
10	Zoom Percent Slider	Moving the Zoom Percent Slider left or right adjusts the zoom percentage of the layout shown in the ViewApp Editor..
11	Zoom Percent Picker	Drop-down list of zoom percentages to set the size of the layout shown in the ViewApp Editor..

12	Zoom Box	The Zoom Box enables a selected area to be zoomed to the full width or height of the current editing area of the ViewApp Editor. The cursor shows a magnifying glass when the Zoom Box is selected. Drawing a rectangle by mouse or by touch within the editing area shows the content within the rectangle at the full width or height of the current editing area.
13	Zoom to Fit	Adjusts the size of the layout to fit entirely within the current area of the ViewApp Editor
14	Zoom to 100%	Sets the size of content shown in a ViewApp Editor to its actual size
15	Layout Selector	Displays the name of the layout assigned to the selected screen. Clicking the arrow at the right of the Layout name lists all available Layouts in the Galaxy. Selecting another layout within the list assigns it to the selected screen.
16	Screens List	Thumbnail list of all screens in the screen profile with an overlay of the layout assigned to the screen.
17	Properties grid	Shows content properties of the graphics or controls currently showing on the selected pane. If a pane is empty, no properties are listed. If the selected pane contains a graphic, the name of the graphic and its pane location appear as content properties. If the selected pane contains a control, the control's properties are listed.
18	Preview	Opens a preview window so you can see what the ViewApp will look like at runtime without having to deploy it. For more information , see About previewing a ViewApp .
19	Additional settings	Opens the Settings dialog box, where you can set additional values

		that affect the behavior of this ViewApp. For more information, see Set additional properties for a ViewApp .
20	Toolbox Tab	Shows the hierarchical folder structure of the Visualization folder. Selecting a folder shows thumbnails of the content within the folder.
21	Assets Tab	Shows a hierarchical list of the objects shown in the Model view of the IDE. Selecting an object from the list shows the thumbnail of any associated content.
22	Search Field	Field to enter a search string to search for properties, graphics, or objects by name.
23	Breadcrumb	When the Toolbox tab is selected, a breadcrumb shows the list of graphic folder names to the content selected from the list. If the Assets tab is selected, a breadcrumb shows the serial list of Areas or other objects to the selected item from the list.
24	Selected Object	Object selected from the list of assets.
25	Associated Content	Content associated with the selected object from the assets list.

Configure the ViewApp Editor

You can use a set of common controls to adjust the editing areas of the ViewApp Editor while you are working on a ViewApp.

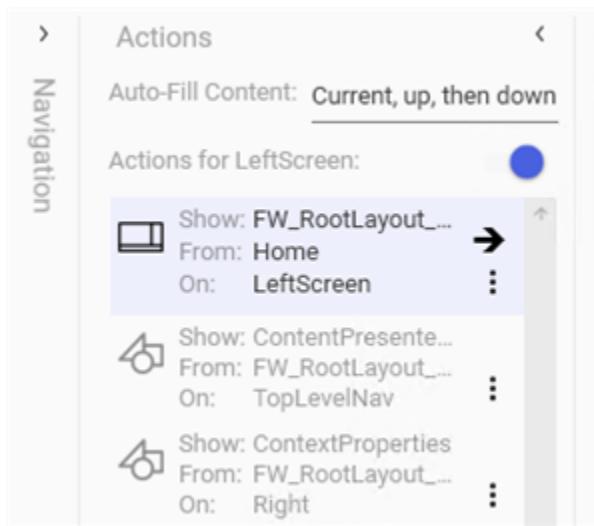
Adjust the Size of ViewApp Editor Areas

The ViewApp Editor consists of a set of columns that show a navigation hierarchy, an Action List, a graphic of the layout being edited, and a tabbed area at the right to show properties, assets, and content within the Visualization and Template folders.

Dividers are vertical or horizontal lines that separate the different areas of the ViewApp Editor. You can resize a ViewApp Editor area by moving its divider with a drag and drop mouse action or by a touch gesture.

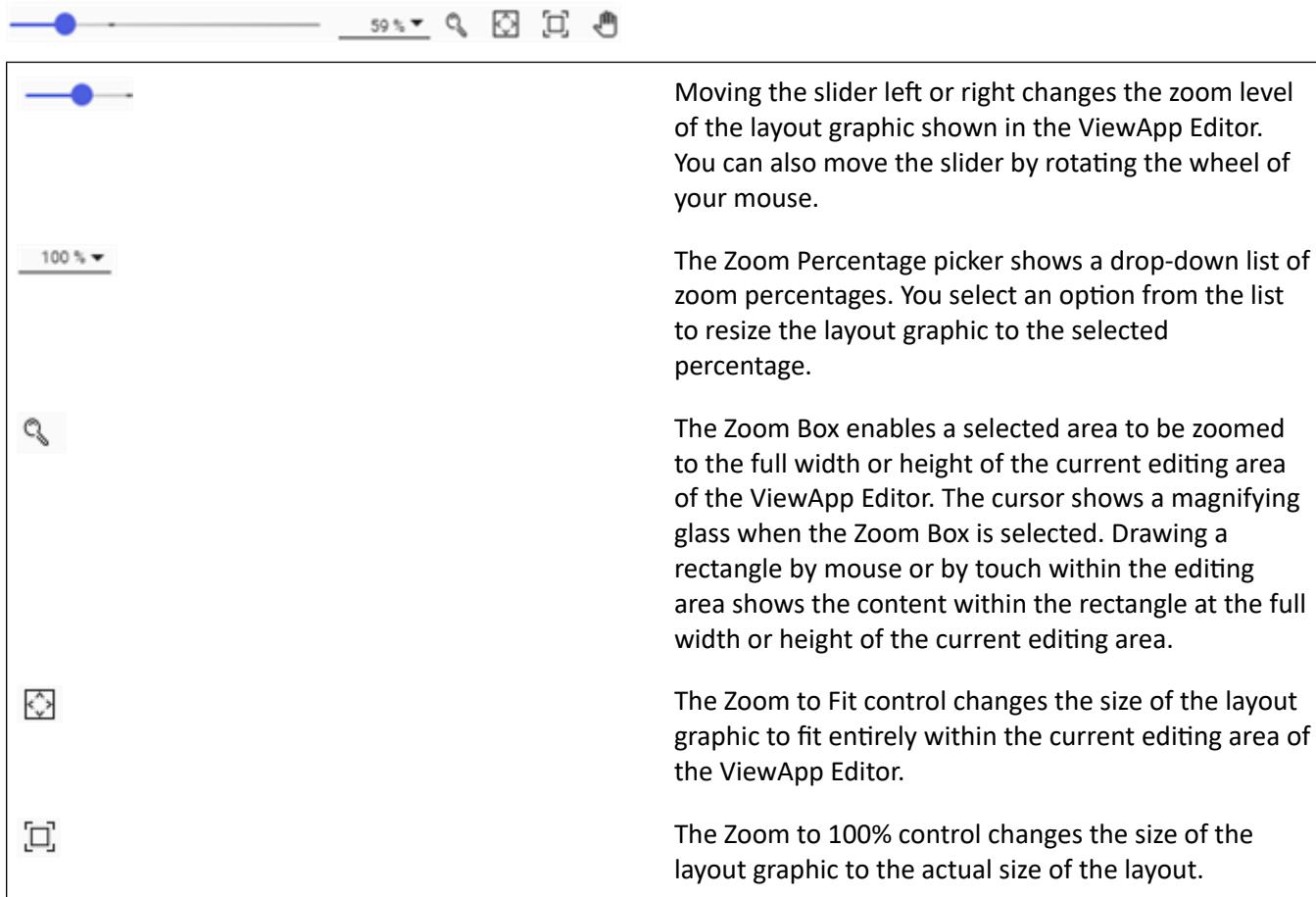
Show or Hide ViewApp Editor Fields

A chevron icon placed near the name of a ViewApp Editor field indicates the field can be expanded or hidden by selecting the chevron.



Change the Viewing Size of a ViewApp in the ViewApp Editor

The middle area of the ViewApp Editor show a graphic representation of the ViewApp currently being edited. A set of pan and zoom controls appear at the bottom of the field to change the size of the Layout graphic. Panning or zooming a layout is useful when the layout is designed for a much larger or smaller screen than the editing area of the ViewApp Editor.





You can pan to different parts of the editing area of the ViewApp Editor with the Pan tool. The cursor changes to a hand when pan mode is selected.

About ViewApp and derived templates

You begin working with a ViewApp by creating a derived template from the \$ViewApp base template in the Template folder of the System Platform IDE. After creating a derived template from the \$ViewApp template, you can begin editing your ViewApp with the ViewApp Editor.

Unlike other objects that are created from templates, you cannot create and edit an instance of a ViewApp. All of your editing is done in the derived template. An instance is created automatically when you assign the derived ViewApp template to a ViewEngine or WebViewEngine.

Before creating a ViewApp, you should have created the screen profiles and layouts that will be integrated into your ViewApp.

- Screen Profile Editor, see [About screen profiles](#)
- Layout Editor, see [AppFramework layouts](#)

Create a ViewApp

You create a ViewApp from a derived template of the \$ViewApp base template. When you start the ViewApp Editor for the first time after creating a Viewapp, the **ViewApp Initialization** wizard appears to select a screen profile and layouts that will be associated with the ViewApp.

Note: Before you create a ViewApp you should have created the screen profiles and layouts that will contain the content of your ViewApp.

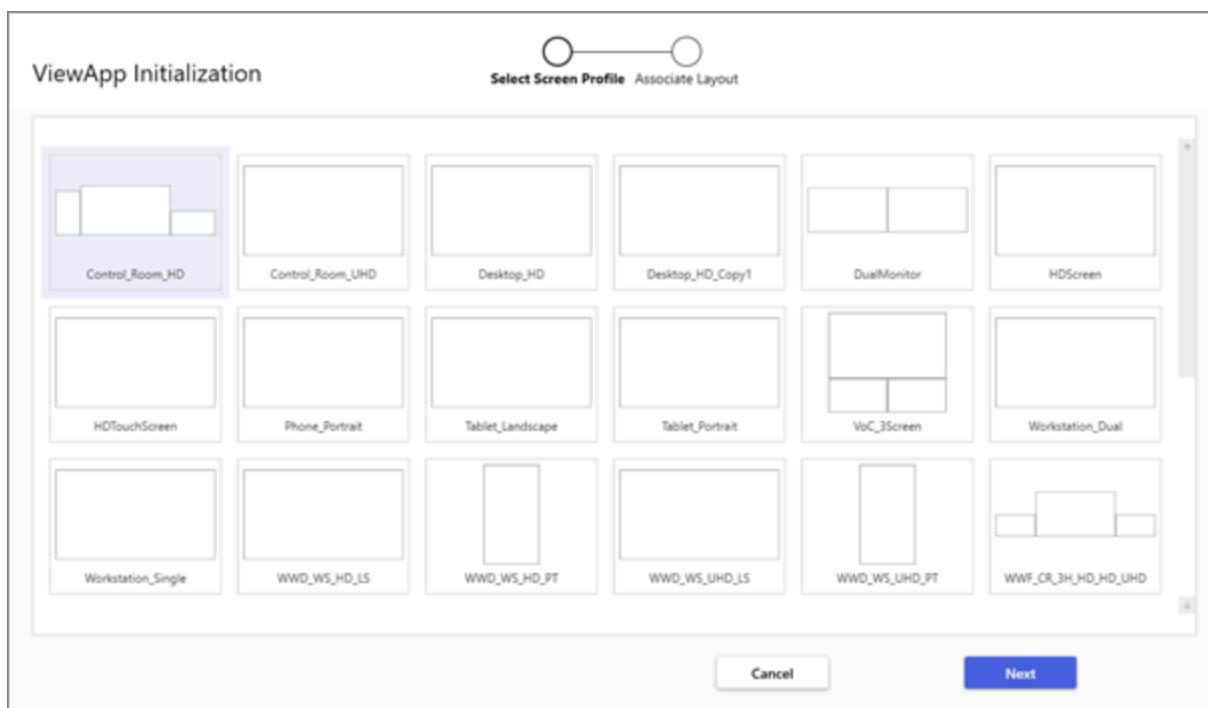
To create a ViewApp

1. Open the IDE.
2. Select the **Template folder** tab and expand the **System** folder to show the \$ViewApp object.
3. Create a derived template of the \$ViewApp object.
4. Assign a ViewApp name to the derived template you created.

Note: Immediately after creating a ViewApp derived template, a red icon appears indicating no screen profiles and layouts have been assigned to the ViewApp.

5. Double-click on the ViewApp icon to start the **ViewApp Initialization** wizard.

The wizard opens and shows the thumbnails of the screen profiles present in the Galaxy. Each thumbnail shows the configuration of screens that belong to the screen profile.



6. Select a screen profile to associate with your ViewApp and select **Next**.

The wizard shows a thumbnail of the screen profile you selected and a list of layouts present in the Galaxy. Each layout appears as a thumbnail graphic showing its pane organization.

7. Associate a layout to one or more screens in the Screen Profile.

a. Select a screen from the screen profile.

The screen border turns green and the name of the screen appears beneath the icon when it is selected.

b. Select a layout to associate with the selected screen.

Note: At least one screen in the screen profile must be associated with a layout before continuing to the ViewApp Editor.

8. Select **Finish** to close the Wizard and open the ViewApp editor.

For more information about the different components of the ViewApp editor, see [About the ViewApp Editor](#).

Manage a ViewApp

You can perform routine tasks to manage a ViewApp from the Template folder. Management tasks include:

- [Delete a ViewApp](#)
- [Rename a ViewApp](#)
- [Export a ViewApp](#)
- [Import a ViewApp](#)

Delete a ViewApp

You can delete a ViewApp from the Templates pane of the IDE.

To delete a ViewApp

1. Open the IDE and display the **Templates** pane.
2. Navigate to the folder containing the ViewApp template to delete.
3. If the ViewApp is open for editing, close it.
4. Select the ViewApp you want to delete.
5. Press the **Delete** key or right-click the ViewApp and select **Delete** from the shortcut menu.
The **Delete** dialog box appears to confirm your delete request.
Click **Yes** to delete the ViewApp.
6. Verify the ViewApp no longer appears in the folder list.

Rename a ViewApp

You can rename a derived ViewApp by selecting it in the Templates pane.

After you enter a new ViewApp name, the name is validated to ensure it meets naming requirements. If it is a duplicate of an existing name or breaks any name rules, a message appears indicating there is a problem with the name.

Pressing the **Esc** key cancels a rename request and the Viewapp reverts to its original name. Committing an empty name cancels the rename request.

To rename a ViewApp

1. Open the IDE and display the **Templates** pane.
2. Navigate to the folder containing the ViewApp template to rename.
3. If the ViewApp is open for editing, close it.
4. Select the ViewApp to rename.
5. Click the ViewApp name or press **F2** to enter edit mode. A blue background appears around the name.
6. Type the new name.
7. Commit the new name by clicking outside of the name.
If the new name is valid, it appears in the Template folder. If the name is invalid, you see a dialog box that describes the problem.
8. If you typed an invalid name, click **OK** and enter another name.

Export a ViewApp

You can export one or more ViewApps from the Templates folder to an aaPKG file. You can then import the ViewApps from the export file to another Galaxy.

To export a ViewApp

1. Open the IDE and display the **Templates** pane.
2. Select one or more ViewApps to export.
3. On the **Home** ribbon, in the **Export** area, select **Selected**, then **As package**.

You can also export a ViewApp from the shortcut menu: right-click the ViewApp name, select **Export**, and then **Selected object(s)**.

The **ViewApp Export Preferences** dialog box appears with an option to export any graphics and other objects included with the ViewApp.

4. Select whether or not to export objects included in the View App and click **OK**.

The **Export Selected objects** dialog box appears with a folder tree to specify where the exported file should be saved and a field to name the export file.

5. Select the folder to save the export file.

6. Assign a name to the export file.

The default export file name is the name of the first selected ViewApp from the Template Toolbox.

7. Click **Save**.

A horizontal bar shows the progress of the ViewApps being loaded into the export file.

8. Verify the export aaPKG file has been placed in the folder you specified.

Import a ViewApp

You can import ViewApp to the Templates folder of the System Platform IDE. Exported ViewApps are saved in an aaPKG file.

During the import process, you have the choice to keep or overwrite any ViewApps with the same name currently in the Template folder with the imported ViewApps.

To import one or more ViewApps

1. Open the IDE and display the **Templates** pane.

2. Select the **Galaxy** ribbon, then select **Import**.

3. Select **Objects**, then **From package**.

The **Import Objects from package** dialog box appears to specify the folder containing a ViewApp export file.

4. Go to the folder where the aaPKG file containing ViewApps is located.

5. Select the aaPKG file and click **Open**.

The **Import Preferences** dialog box opens with options to import ViewApps. You cannot have two ViewApps with the same name or more than one copy of the same version of a profile in the same Galaxy. When you import a ViewApp, you can choose options from the **Import Preferences** dialog box to control how you want naming and version conflicts handled.

6. Click **OK** to import the ViewApps.

A progress bar shows the ViewApps being imported into the Templates folder. After finishing the import, you should see the ViewApps in the list of Templates folder objects.

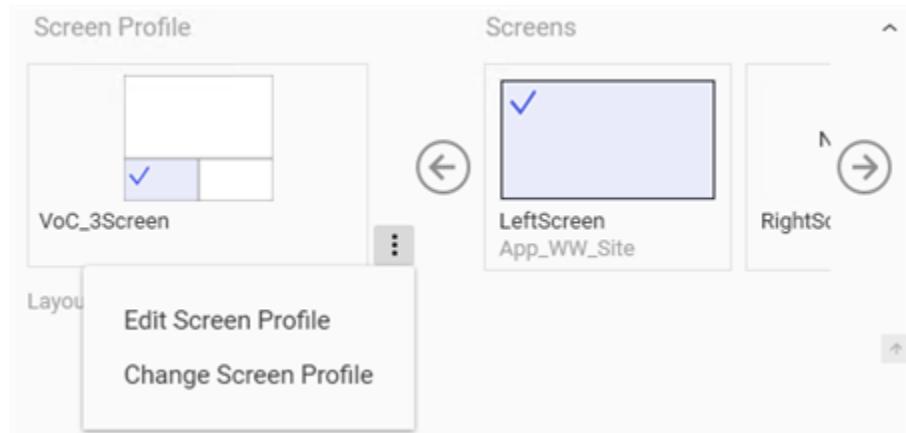
Configure a ViewApp

Typically, you create screen profiles and layouts before integrating them into a ViewApp with the ViewApp Initialization Wizard. The ViewApp Editor provides the capability to replace the current screen profile or edit it by opening the Screen Profile Editor from within the ViewApp Editor. You can also remove a layout associated with a screen or edit it by opening the Layout Editor from within the ViewApp Editor. These topics describe how to

manage the screen profiles and layouts associated with a ViewApp from the ViewApp Editor.

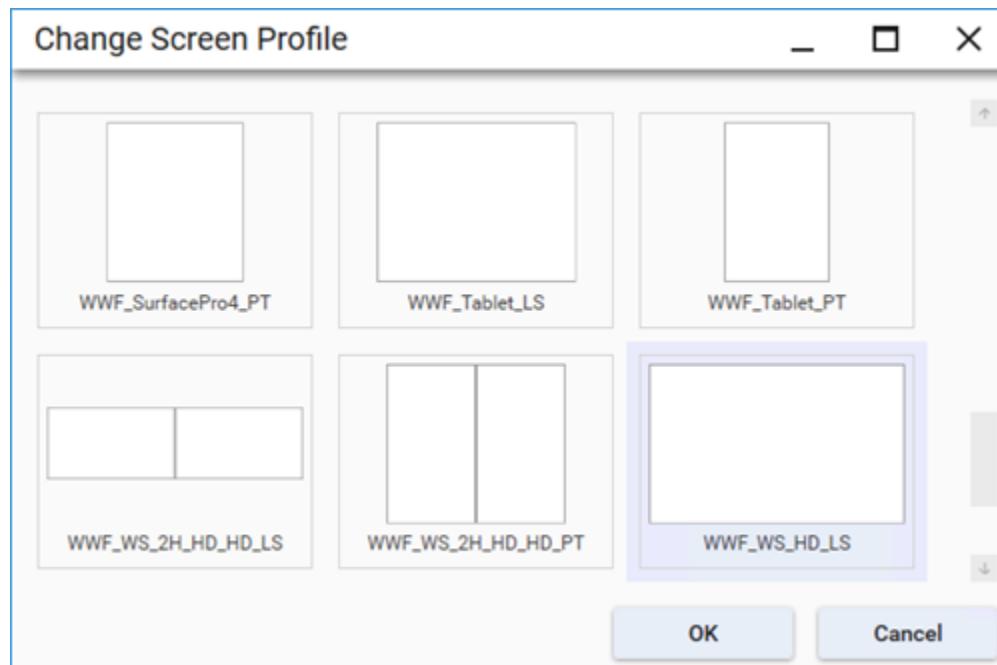
Manage screen profiles in a ViewApp

You select a screen profile with the Initialization Wizard the first time you open the ViewApp Editor. Thereafter, the ViewApp editor shows the screen profile you selected. You can change screen profiles or edit the one you selected within the ViewApp Editor by selecting the **Screen Profile** menu button.



If you select **Edit Screen Profile**, the Screen Profile Editor opens with the current screen profile selected for editing. You can make changes to the screen profile and save it. The Screen Profile Editor closes and returns you to the System Platform IDE.

If you select **Change Screen Profile**, a pop-up window shows thumbnails of other screen profiles defined in the Galaxy.



Select another screen profile from the list and click **OK** to replace the current screen profile with the one you selected.

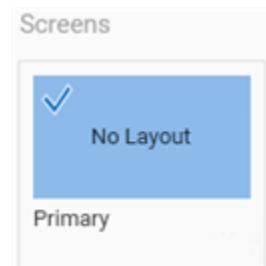
Manage layouts in a ViewApp

You select one or more layouts with the **ViewApp Initialization** wizard the first time you open the ViewApp Editor. Thereafter, the ViewApp editor shows the layouts you selected. You can edit the current selected layout or remove it within the ViewApp Editor by selecting options from the **Layout** menu button.

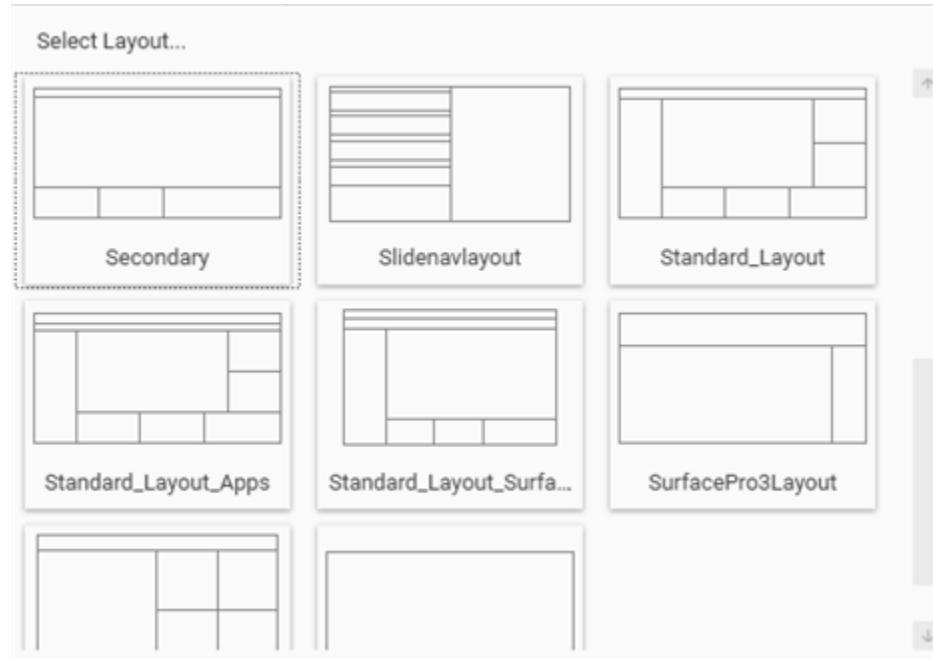


If you select **Edit Layout**, the Layout Editor opens with the current layout shown in the Layout Editor selected for editing. You can make changes to the layout and save it. The Layout Editor closes and returns you to the System Platform IDE.

If you select **Remove Layout**, the layout for the selected screen in the ViewApp Editor is removed. The thumbnail of the layout you removed no longer appears within the selected screen and you see a message that no layout is assigned to the screen. Also, the thumbnail of the layout shown in the ViewApp Editor indicates that it is empty.



You can also change layouts by selecting another layout from the **Layout** field with the ViewApp Editor. Clicking the arrow in the **Layout** field shows a pop-up menu of all layouts defined in the Galaxy.



Select another layout from the list to replace the current or deleted layout with the one you selected. The

selected screen and the layout thumbnail are updated with the layout you selected.

Set additional properties for a ViewApp

The additional settings available for a ViewApp determine the timing of certain actions taken when the ViewApp is open. The available settings are:

- **OnShowTimeoutExtendMs:** This extends the default timeout period for the OnShow script when the ViewApp is opened. The amount you enter is added to the default timeout of 1000 msec (one second). So, the default value of 5000 msec actually sets the OnShow timeout to six seconds. This is the maximum amount of time that the OnShow script will wait this long for the ViewApp's data to be bound. The behavior is:
 - If the data is bound before the timeout period is reached, the OnShow script executes as soon as the data is available.
 - If not all of the data is bound, the OnShow script executes when the timeout period is reached.
- **DataPollingIntervalMs:** How often (in msec) the ViewApp will poll for new data. The default is 1000 msec (one sec).
- **UseLanguageSettingFontOnly:** If this is set to True, when the ViewApp is opened in the OMI web client, it will use only the font that is set for the language the user has selected in the OMI web client. This must be one of the languages defined on the Galaxy configuration **Languages** dialog. This font will override all other font settings for any element of the ViewApp. The default value is False.

To set additional values for a ViewApp

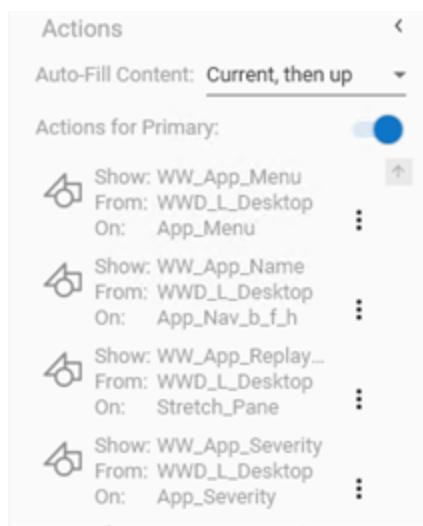
1. On the ViewApp editor, click the **Additional settings** icon at the top right. The settings dialog box opens.
2. On the **Settings** dialog box, enter the desired values in JSON format, like this:

```
{  
  "Web": {  
    "OnShowTimeoutExtendMs" : 5000,  
    "DataPollingIntervalMs" : 1000,  
    "UseLanguageSettingFontOnly" : False  
  }  
}
```

3. When you are done, select **OK** to save your changes and close the **Settings** dialog box.

Add content to a ViewApp

You can use a drag and drop mouse action or touch gesture to add content to a layout within the ViewApp Editor. Assigning content to a pane adds a Show Content action to the ViewApp's navigation Action List, which coordinates the automatic filling of panes during runtime.

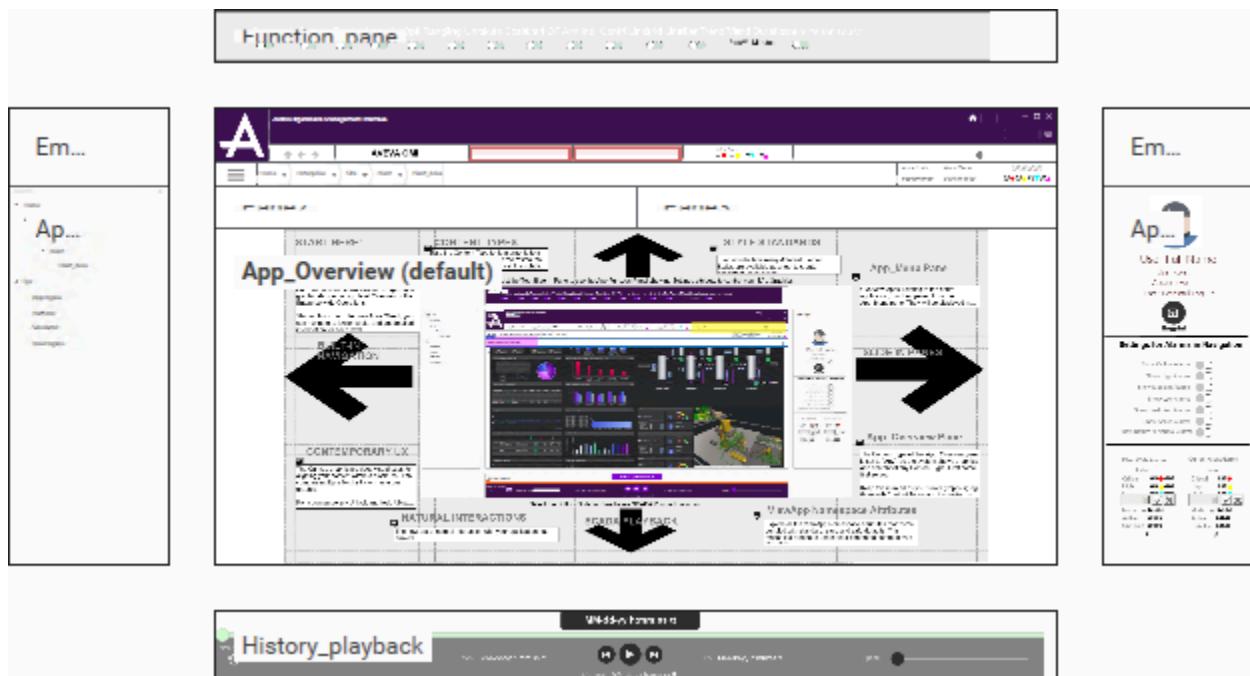


Content from the **Toolbox** or **Assets** tabs at the right of the ViewApp Editor can be dragged on a pane of the layout. Content can include:

- Graphic
- Layout
- Control
- App Instance

While content is dragged over a layout, the background color of the target pane changes to indicate the pane will receive the content when it is dropped. The shape of a mouse cursor also changes to indicate the content can be dropped on the pane. After content has been dropped, a thumbnail appears within the pane representing its content.

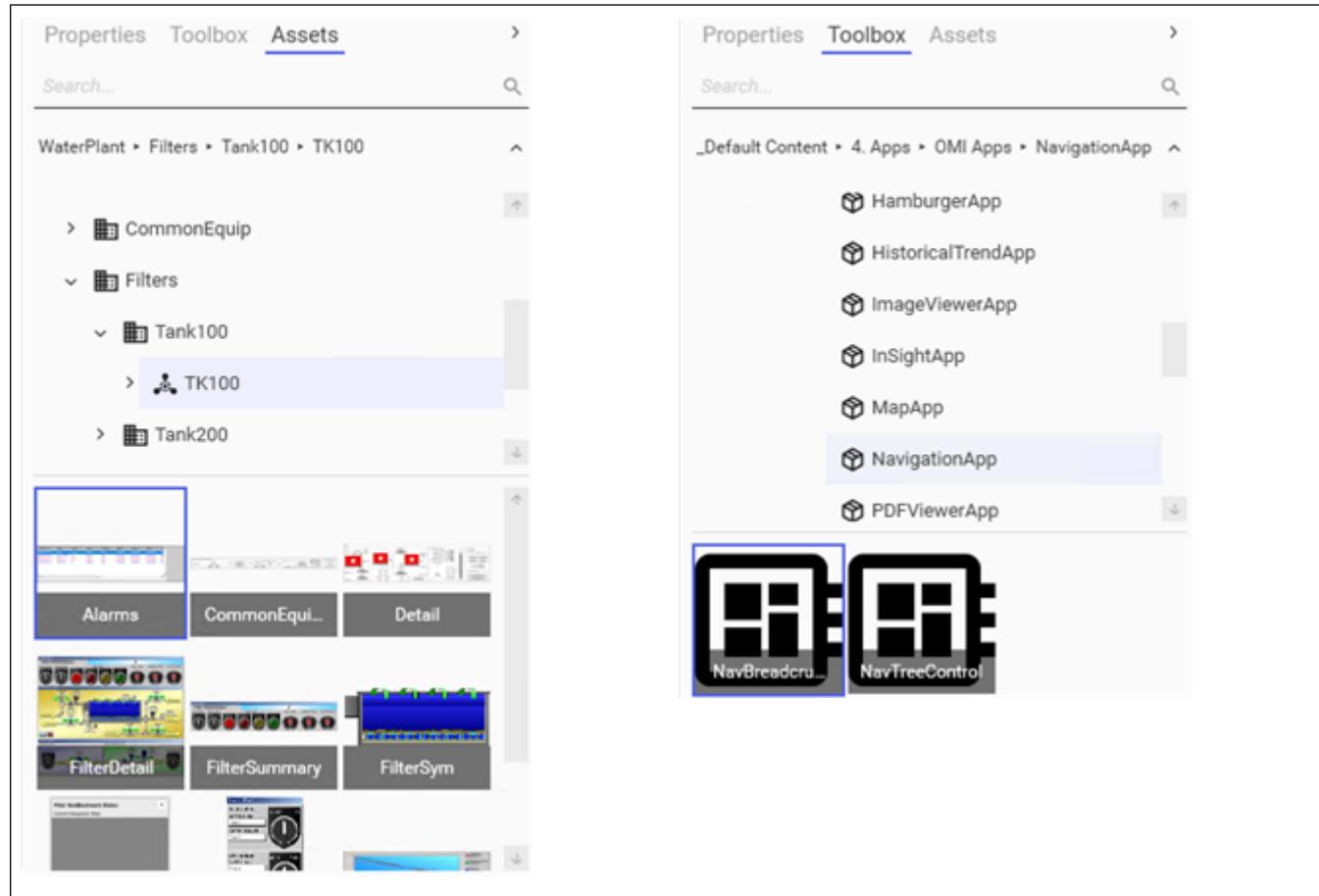
Note: You can also add content to a pane using a touch device by selecting and holding the content to move it over the target pane.



Browse content to add to a ViewApp

You can browse the **Assets** and **Toolbox** tabs to find content that can be added to a layout within the ViewApp Editor. Each item shown in the **Assets** view represents an Area or an Asset.

Each item shown in the **Toolbox** tree represents either a folder or a control library found in the Visualization folder. Selecting an item from a browse tree selects the item and displays any content belonging to that item in the content area beneath the browse list.



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

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

Search for content

You can search within the currently displayed toolbox folders or assets by entering a search term in the **Search** field. Clicking anywhere in the **Search** field hides the hint text and magnifying glass in preparation to enter a search string.

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

conducted.

Asset Searches

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

The screenshot shows a software interface with a top navigation bar containing 'Properties', 'Toolbox', and 'Assets'. The 'Assets' tab is selected and underlined. Below the navigation is a search bar with the text 'Pump|'. Underneath the search bar, a message reads '14 assets matching 'Pump':'. A scrollable list displays 14 items, each consisting of a name and a category. The names are: Pump_001, Pump_002, Pump_003, Pump_004, Pump_005, Pump_006, Pump_007, Pump_008, Pump_009, Pump_010, TestPump_001, TestPump_002, and TestPump_003. The categories are all 'Enterprise'. At the bottom of the list, there are navigation arrows and a small tree diagram icon.

Name	Category
Pump_001	Enterprise
Pump_002	Enterprise
Pump_003	Enterprise
Pump_004	Enterprise
Pump_005	Enterprise
Pump_006	Enterprise
Pump_007	Enterprise
Pump_008	Enterprise
Pump_009	Enterprise
Pump_010	Enterprise
TestPump_001	Enterprise ▶ Site ▶ Plant ▶ Plant_Area
TestPump_002	Enterprise ▶ Site ▶ Plant ▶ Plant_Area
TestPump_003	Enterprise ▶ Site ▶ Plant ▶ Plant_Area

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

Toolbox Searches

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

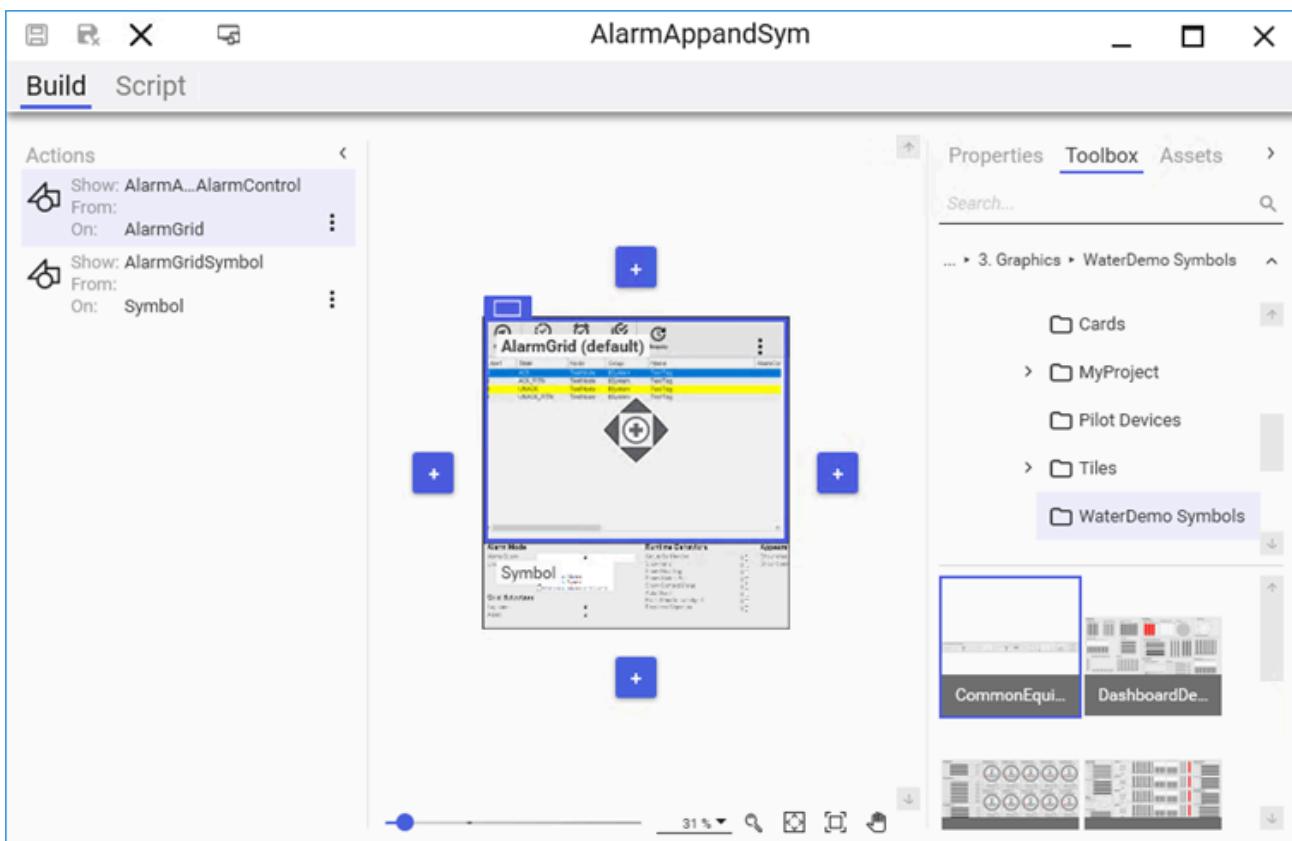


Search results items in the content area can be selected and dropped in a layout pane.

About adding content to an empty pane

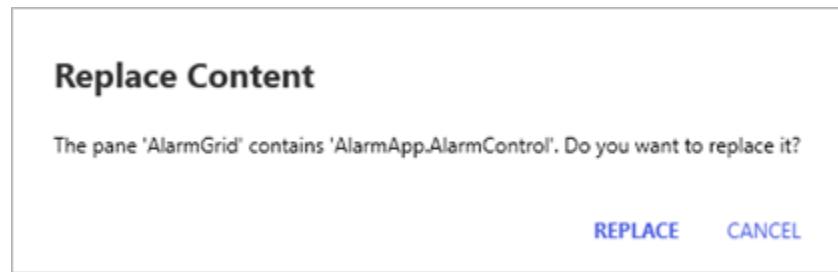
If you add content into an empty pane, then,

- The **Content** property is set to the name of the content added to the pane
- The **Pane** property is set to the name of the pane hosting the content.
- The **Actions** list updates to show an entry containing the name of the content and the name of the pane hosting the content
- A preview of the content appears in the pane



Note: If a layout is added to a pane within the same layout being edited, the cursor changes appearance to indicate the layout cannot be added.

If the target pane being receiving content is a single content pane and already contains content, you see a dialog box asking whether the current content should be replaced or not.



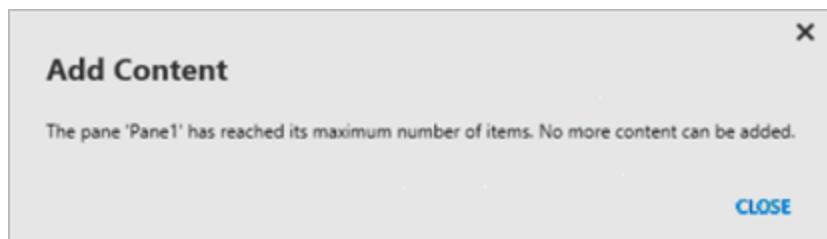
Select **Replace** to update the **Content** property. The thumbnail of the new content appears in the pane.

Select **Cancel** to cancel the add content process. The existing content remains in the pane.

Add content to multi-content panes

If a pane is defined as a multi-content pane, dropping a second content item adds the content and the pane is selected. The **Pane** property updates to show the name of the most recent content added to the pane.

If you attempt to drop a content item that will exceed the maximum limit set for the pane, the add content action will fail. A dialog box appears notifying you that the maximum number of content items has been reached and no more content can be added to the pane.

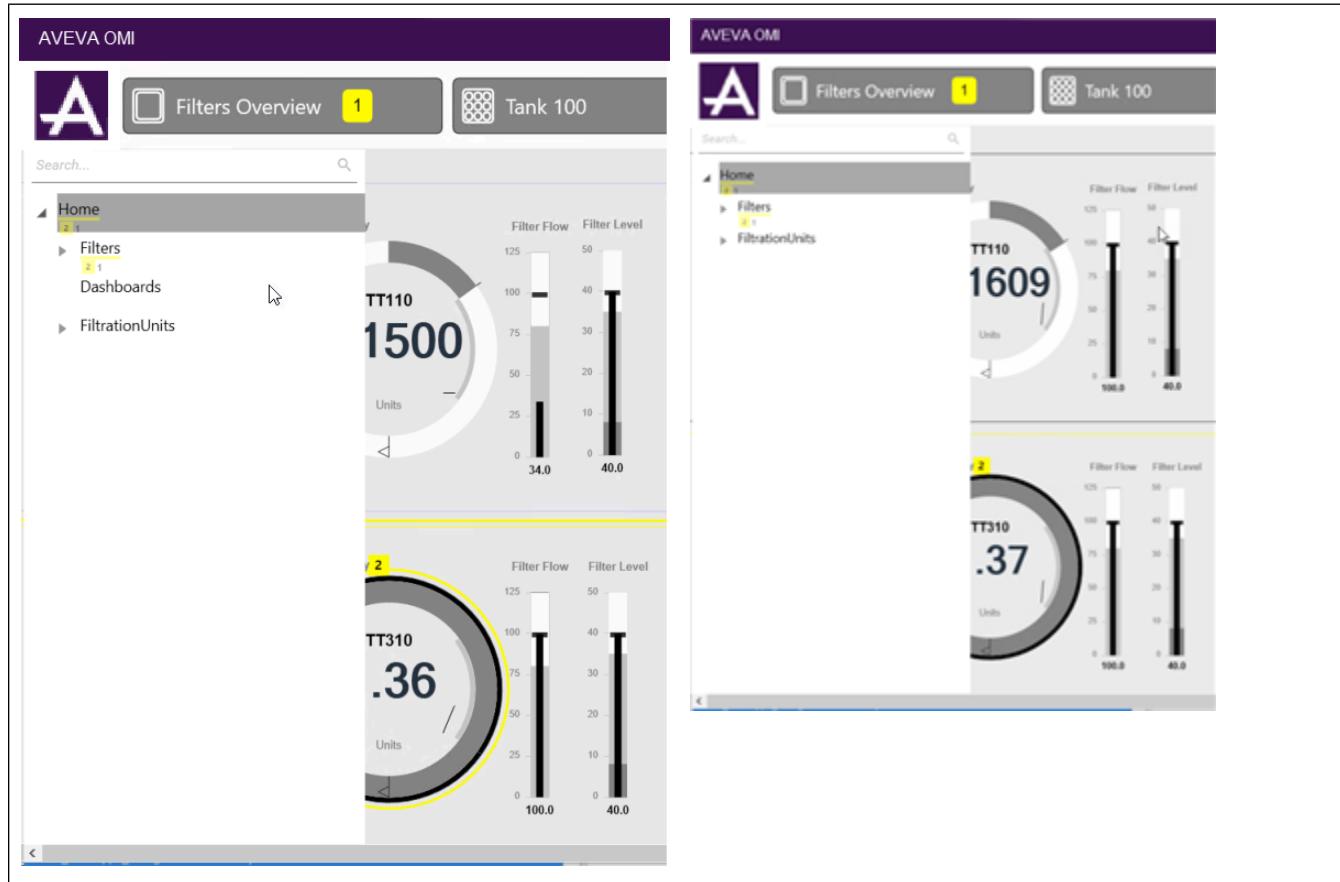


Show or hide ViewApp navigation items

During runtime, ViewApp navigation items can be shown or hidden based on the access level assigned to, and/or user roles selected for, the navigation items. The navigation item will be visible only to logged in users who satisfy each of these conditions, if configured".

Controlling the display of content in a ViewApp by access levels or roles is an effective way of preventing access to a ViewApp's sensitive controls from unauthorized users. The following figures show a ViewApp to monitor tanks and filters in a water treatment plant. The ViewApp includes a **Dashboards** navigation item containing controls that regulate pump flow between tanks, which is the responsibility of supervisors in the plant's control room. Operators only need to see the status of filters and filtration units as part of their work responsibilities.

Navigation item security has been implemented for the **Dashboards** navigation item. The figure on the left shows the ViewApp navigation item shown to supervisors who need access to **Dashboards** controls. The ViewApp on the right hides the **Dashboards** navigation item from operators.



About navigation item security

Security properties can be configured differently to implement various types of navigation item security.

Security	Navigation Item Property Configuration	Navigation Item Visibility Requirements
No Security	Access Level =Not Configured User Roles =Not Configured	Navigation item always visible
Access Level Only Security	Access Level =Integer <N>in the range 0-9999 User Roles =Not Configured	<ul style="list-style-type: none"> • Security must be configured in the Galaxy • User must log in to the ViewApp • Logged-in user must have an access level equal to or higher than the specified value <N> of the Access Level property. • The logged-in user can belong to any role
User Roles Only Security	Access Level =Not Configured User Roles =Role1,Role2,Role3	<ul style="list-style-type: none"> • Security must be configured in the Galaxy • User must log in to the ViewApp • Logged-in user can be assigned any access level • Logged in user must be a member of at least one of the specified user roles.
Access Level and User Roles security	Access Level =Integer <N>in the range 0-9999 User Roles =Role1,Role2,Role3.	<ul style="list-style-type: none"> • Security must be configured in the Galaxy, • User must log in to the ViewApp • Logged-in user must have an access level equal to or higher than the specified value <N> of the Access Level property. • Logged in user must be a member of at least one of the specified user roles.

Configure navigation item security

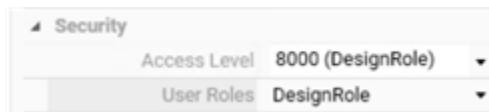
Before you can configure navigation security in a ViewApp, the following prerequisite tasks must be completed:

- Security must be configured for the Galaxy to authenticate users by their user names and passwords as part of the ViewApp login process.
- Security roles must be assigned to those users who will interact with a running ViewApp. Each user must be assigned an access level and user role. For more information about implementing security for users by role and access name, see [About Galaxy security](#).
- Decide how you want to implement security. Navigation items containing secure, restricted content must be assigned an access level, or user role, or both.

The following procedure explains how to assign security to a navigation item associated with content that must be secured

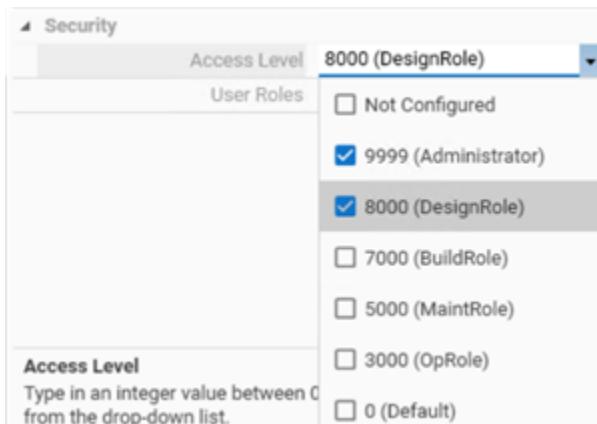
To configure navigation item security

1. Open the ViewApp with the ViewApp Editor.
2. Expand the **Navigation** area of the ViewApp Editor to show the navigation tree listing the items you want to apply security.
3. Select an item from the navigation tree that you want to apply navigation security to.
4. Select the **Properties** tab in the ViewApp Editor to show the **Access Level** and **User Roles** properties.



The following steps of this procedure vary based on how you want to implement navigation item security.

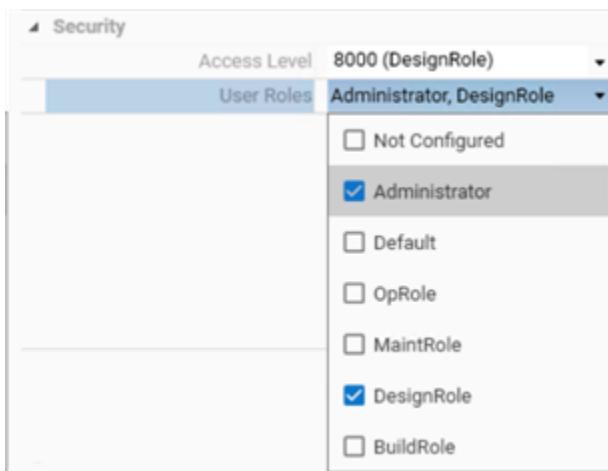
- If you only want to implement **Access Level** security, omit steps 7 and 8.
 - If you only want to implement **User Role** security, omit steps 5 and 6.
 - If you want to implement both **Access Level** and **User Role** security, complete all steps.
5. Select the **Access Level** property field to show a drop-down list of access levels by user roles.



6. Select an access level from the list.

Note: You can also enter an access level less than 9999 directly in the field without selecting an access level from the list. Also, any listed access level greater than the one you selected is also selected in the list.

7. Select the **User Roles** property field to show a drop-down list of user roles.



8. Select one or more user roles from the list.

The selected roles appear in the **User Roles** field in a comma delimited list.

9. Save your changes.

User permissions and navigation item security

The comparison of a navigation item's access level to the logged on user's access level follows inheritance rules. If a parent navigation item is not visible during runtime because its access level is greater than the logged on user's access level, all child navigation items are also hidden, even if the child navigation items have been assigned access levels less than the user's access levels.

If the user cannot access a navigation item during runtime, then the content from the next accessible navigation item is displayed based on the selected Auto-Fill mode. A comparison is made between the user's access level and the next navigation item's access level in the Auto-Fill sequence. If the user is authorized to see the content, it appears in the ViewApp. Otherwise, the Auto-Fill sequence continues until content is shown from an authorized navigation item or the last item in the Auto-Fill sequence is reached.

About previewing a ViewApp

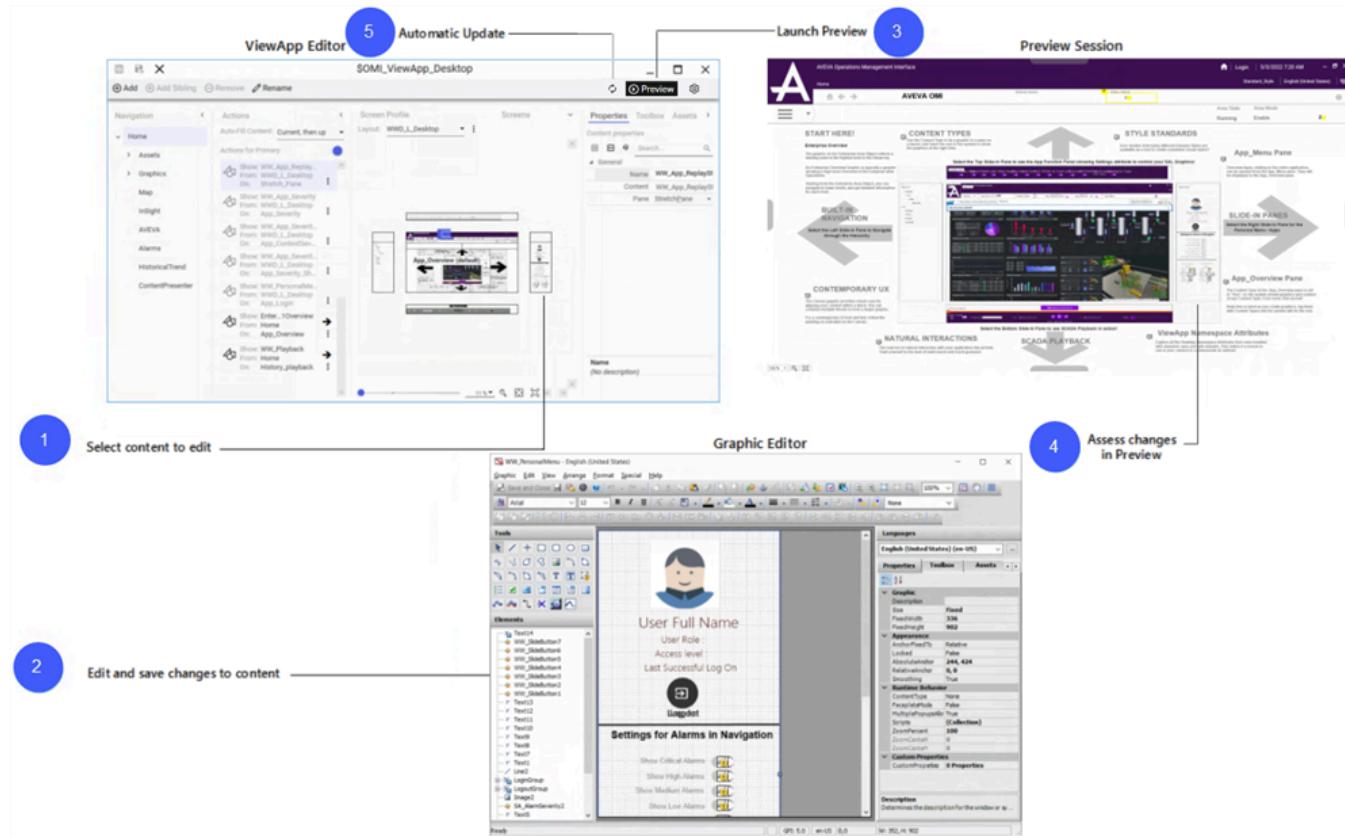
Building an AVEVA OMI ViewApp typically involves significant internal testing before placing a ViewApp into production. Builders must follow an iterative process of modifying a ViewApp and then viewing the running ViewApp to assess the effects of development changes. To run a ViewApp, the Galaxy must be deployed. This repetitive edit, deploy, and assessment process is lengthy and labor intensive.

AVEVA OMI now provides a preview session, which enables builders to quickly assess their changes in a running ViewApp without deploying the ViewApp. A preview launches immediately and runs in a separate window to show the current content being edited.

The following figure shows the major steps in an example workflow to edit a graphic within a ViewApp and then launch a preview session to assess the editing changes in the ViewApp. Select the **Preview** button to preview the ViewApp in a native ViewApp. Preview sessions can remain open while you continue editing the ViewApp.

With some exceptions, when changes are made to the ViewApp, the preview refreshes to show the updates that were made. You can toggle the **Auto Update** icon next to the **Preview** button to activate or prevent updates to a preview session.

Note: You cannot preview what a ViewApp will look like in the OMI web client. While the OMI web client display of a ViewApp is generally very similar to the desktop client display, to see exactly how a ViewApp looks in the web client, you must assign it to a WebViewEngine, deploy it, and open it in the web client. If you make changes to the ViewApp while it is open, refresh the web page to see their effect in the web client.



Preview a ViewApp

The following procedures explain how to launch and close a ViewApp preview. For a complete list of content and ViewApp functions supported in a preview session, see [About ViewApp previews at different stages](#).

Before you request a preview, you should decide if it is acceptable that alarm and historical data are not updated after you make changes to your ViewApp.

A message appears when you initially request a preview that explains preview mode is intended for testing purposes only and not intended for production usage.

PREVIEW Mode is not intended for production usage.

PREVIEW mode facilitates iterative development and shows saved versions of graphical content from the configuration repository which may not match the deployed object.

Preview mode is intended for testing purposes and is not intended for production usage.

Note: You can always turn this prompt back on from the User Information menu in the ArchestrA IDE.

Do not show this message again

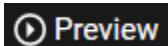
OK

A second message appears if a WinPlatform instance is not deployed for the local computer where the preview request is made. The message provides options to accept a preview that may not show accurate historical and alarm data, or cancel the preview request. If you want to ensure that accurate data appears in a preview, deploy a WinPlatform instance for the local computer before submitting a preview request.

Both messages include a check box to hide the message when the check box is selected. If you want to restore a message that you hid, select the appropriate preview prompt option from the **Configure User Information** dialog. For more information, see [Configure user preferences](#).

To preview a ViewApp

1. Decide how you want to run your preview session.
 - If you want to ensure that all live data in a preview will be updated, deploy the WinPlatform instance for the local computer in which the ViewApp preview will run.
 - If you do not want to deploy the WinPlatform instance on the local computer and are willing to run a preview that does not update live data, select **OK** when you see the warning message.
2. Open a ViewApp in the ViewApp editor.
3. Select **Preview** shown in the upper right corner of the ViewApp editor.



The first message appears with a caution that preview mode is not intended for production usage.

4. Click **OK**.

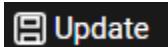
If you did not deploy the ViewApp's Galaxy WinPlatform object, you see a second warning message about the integrity of historical and alarm data shown in the preview.

5. Click **OK** to start the preview session.

The preview session launches from the computer hosting the System Platform IDE. The preview window opens and shows the current focus of the ViewApp as it is edited in the ViewApp editor.

Note: When previewing a ViewApp for the first time after launching the System Platform IDE, it takes several seconds for the **Preview** button to become enabled while the ViewApp editor is preparing the preview data.

If a Preview is active and you have made changes to the ViewApp without saving them, the caption on the **Preview** button changes to **Update**:



6. Make changes to one or more component parts of the ViewApp.

You can select the **Preview** button any time during an editing session. After you select **Preview** again:

- If the ViewApp contains unsaved changes, the ViewApp will be saved and the new configuration is passed to Preview.
- If Auto Update is off and some contents (layout, graphics) have been changed, the changes will be passed to Preview.
- If the navigation node in ViewApp Editor has been changed, the new navigation node will be passed to Preview.
- If the current selected screen has been changed, the current selected screen will be activated in Preview.

7. Select **Save**.

The word **Preview** appears on the button and the preview updates to show the changes you made to the ViewApp.

Note: If you have multiple instances of the same content in a ViewApp, only the edited instance is updated in the preview. Instances of the same content in other panes appear unchanged in the preview.

8. Open the preview window and verify the ViewApp is running.

To close a ViewApp preview

You can also close a ViewApp preview session by doing any of the following:

- Close the System Platform IDE by selecting the **Close** button.
- Close the System Platform IDE by selecting **Exit** from the System Platform IDE **Galaxy** menu bar item.
- Close the ViewApp editor by selecting the **Close** button.
- Close the edited ViewApp by selecting **Save and Close** from the ViewApp editor.
- Close the edited ViewApp by selecting **Close** from the ViewApp editor.
- Select the **Close** button of the preview window.

Other atypical editing actions will also close a preview session:

- Disconnect from the Galaxy Repository
- Restart the computer running the System Platform IDE.
- Change Galaxies from the System Platform IDE.
- Change Galaxy security from the System Platform IDE

About ViewApp previews at different stages

When a builder requests a preview of the ViewApp open in the ViewApp editor, the initial preview shows the current content of the ViewApp's constituent components. The preview shows similar runtime data that would appear in a deployed ViewApp.

Initial Preview

The following list describes the components of a ViewApp that appear in an initial preview request.

- Graphics
 - Graphics from the Visualization folder

- Embedded graphics
 - Graphics currently placed in ViewApp panes
 - Graphics associated with objects
 - Data values associated with graphic custom properties, animations, or attributes.
 - Objects
 - Attributes assigned to an object
 - Graphics associated with an object
 - Hierarchy of objects in the navigation model.
 - Screen Profile associated with the ViewApp
 - Layout associated with the ViewApp
 - Alarms
 - Active alarms from any runtime node.
 - Alarm severity and acknowledgement status
 - Apps
 - Standard apps placed in ViewApp panes
 - Custom apps placed in ViewApp panes
 - Current selected item of the ViewApp's navigation model
 - Current values assigned to the ViewApp namespace
 - Navigation security by assigned Access Levels and content types.
 - Galaxy Styles
 - element styles
 - Format Styles
 - Quality and Status Overrides
 - ViewApp language
 - Historical playback
 - Displays real-time data from attributes whose data is stored in the Historian.
 - Playback controls can be modified by the user from the preview session.
 - Displays historical data from the period set by the playback controls.
- Preview Updates

After editing changes have been made to ViewApp content, the running preview updates to show the changes in the ViewApp after the changes have been saved. Not all changes to a ViewApp necessarily update the preview session.

If a preview session is open, then the following types of editing changes will not appear until the ViewApp editor is closed and opened again.

- Change in language settings
- Change to a Screen-Profile
- Change in Asset hierarchy
- Change to an alias-name

- Change to the object order in the hierarchy shown by the Model view

The remainder of this topic describes the supported updates to a preview session by the various components that can be included as ViewApp content.

Graphics

- Visual or functional changes to embedded graphics.
- Changes to the content type or Access Level assigned to the pane in which the graphic is placed.

Apps

A preview session updates when the following changes are made to apps or controls embedded in a ViewApp.

- Change the properties of apps embedded in a ViewApp.
- Rename an app.
- Delete an app

Note: After renaming or deleting an app, all preview sessions show a popup dialog that indicates the preview session needs to be shutdown or restarted.

- Import an aaPKG file containing one or more new apps and place the imported apps on a layout pane.
- Delete and import an app with updated versions of DLL files with the same name as the original deleted app.
- Change the global configuration values of the InSightApp, Historical TrendApp, and the AlarmApp controls.

ViewApp Navigation

A preview session updates content rendering when the following changes are made to a ViewApp navigation model:

- Change the autofill mode of a navigation item.
- Add or remove custom navigation items.
- Change the position of a custom navigation item in the hierarchy of the navigation model.
- Change the reference that navigation item points to
- Change the property of the navigation item
- Add or remove the content of a navigation item.
- Change the current selected navigation item

ViewApp Layout

A preview session updates when the following changes are made to one or more layouts associated with a ViewApp:

- Delete a layout
- Rename a layout
- Add, delete, merge, or resize a layout pane.

Note: An updated preview shows any panes that have been added or deleted. A message appears when a pane is deleted indicating the content associated with the deleted pane cannot be found.

- Change layout properties.

- Change the properties of a pane within a layout.
- Add or delete a pane from a layout.
- Change the content associated with a layout pane.
- Change properties of content associated with a layout pane.

ViewApp Namespace

A preview session updates when changes are made to a ViewApp Namespace or the attributes associated with it:

- Add, delete, or modify attributes associated with a ViewApp Namespace.
- Rename a ViewApp Namespace.

Historical Playback

A preview updates to show real-time or historical data when the playback controls are adjusted by the user.

Global Styles

Changes to a ViewApp's assigned Galaxy Styles (element styles, Format Styles, and Quality and Status Overrides) will be updated automatically if the WinPlatform is deployed at the IDE node. The ViewApp editor must be closed and reopened again if the WinPlatform is not deployed to show any changes to Global Styles in the preview session.

Alarms

A running preview updates to show the following alarms changes if the computer hosting the preview has its WinPlatform deployed. For more information about the requirements to see historical and alarm data in a preview, see [Preview a ViewApp](#).

- Active alarm monitoring from any runtime node.
- Alarms can be acknowledged or shelved in an active preview session.

Import aaPKG File

A preview session does not refresh during the period when an aaPKG file is being imported. After the import is completed, the preview session refreshes with the changes in the Galaxy.

Switch ViewApp language at runtime

AVEVA OMI language features enable you to develop ViewApps that can be switched to another language at runtime. To enable runtime language switching, you must:

- Configure multiple languages for the ViewApp.
- Export your application text for offline translation.
- Translate one or more exported dictionary files.
- Import one or more translated dictionary files.

This section describes the features and procedures for configuring languages and enabling runtime language switching.

Define and configure galaxy languages

System Platform language features enable you to develop applications that can be switched to another language at run time. To enable runtime language switching, you must:

- Configure multiple languages for the application.
- Export your application text for offline translation.
- Translate one or more exported dictionary files.
- Import one or more translated dictionary files into the Galaxy.

You can set the Galaxy default language and you can define additional languages for purposes of switching graphic and alarm languages at runtime.

The default language is set and languages are added only at the Galaxy level.

Graphics language switching

Runtime language switching applies to all portions of the graphics or animations that you create or configure using the Industrial Graphic Editor or InTouch WindowMaker. The language to display is set when the application is being shown at runtime.

You can only view translations at design time for Industrial Graphics in the Graphic Editor. You cannot switch languages in WindowMaker at design time to see the translations.

About localization workflow

The typical workflow for adding translations to a Galaxy consists of the following:

1. From the **Galaxy** menu, select **Export**, select **Localization**, then choose which option you want to use. If you did not select any objects in the IDE before opening the **Export** menu, the options for choosing **selected** graphics, alarm fields, and internationalized strings will not be active:
 - All internationalized strings
 - Internationalized strings from selected objects (inactive if no objects were selected)
 - Selected Industrial Graphics (inactive if no objects were selected)
 - All Industrial Graphics
 - Selected Alarm Fields (inactive if no objects were selected)
 - All Alarm Fields
2. When exporting Industrial Graphics or internationalized strings, select the language you are exporting for translation and a destination folder.

When exporting alarm fields, select as appropriate:

- Export all alarm messages to a single file
- Create one translation file per area
- You can also limit the export to only alarm messages that have not yet been translated.

Note: For applications that only target sub-areas of the Galaxy, exporting by Area is a more optimal

workflow.

When exporting internationalized strings, you can also limit the export to strings that have not yet been translated.

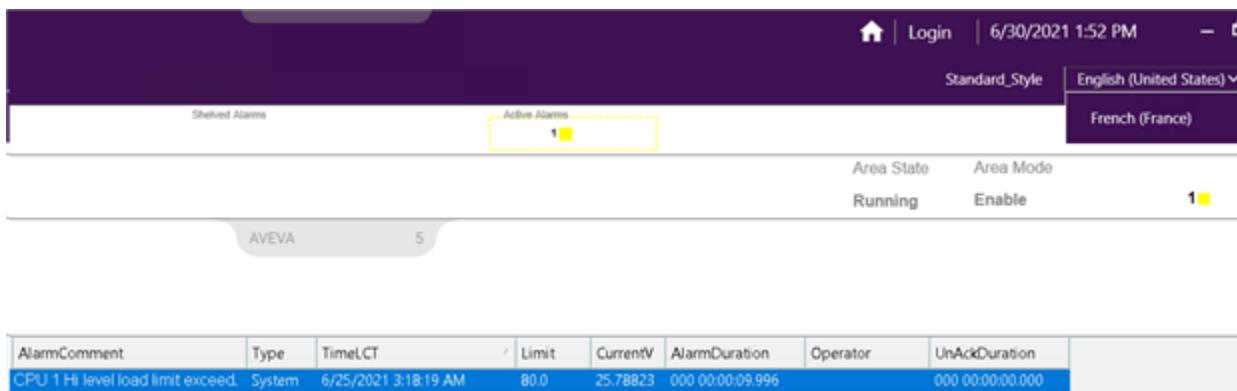
3. After exporting the file, add translations for each field to be localized in the file.
4. Import the translated file(s) back into the Galaxy.
5. For InTouch HMI, import the translated language files into InTouch.
 - In stand-alone InTouch, you can perform a single import operation.
 - In managed InTouch, you must import the language file into each managed InTouch application.
 - For InTouch applications that only target subareas of the Galaxy, exporting by Area is a more optimal workflow.

Since AVEVA OMI applications access the imported files directly from Application Server, there is no additional import required for AVEVA OMI applications.

Alarm app language switching

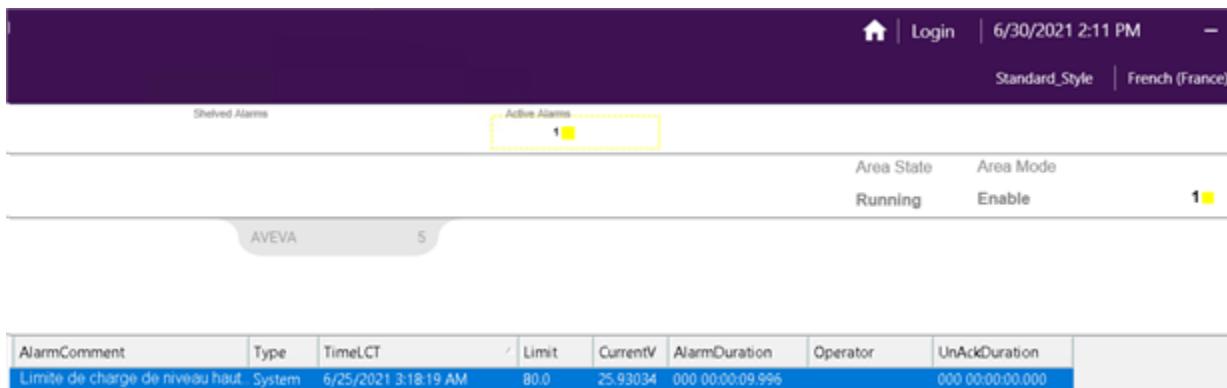
The Alarm app supports alarm comment language switching, which enables you to switch the language of alarm comments that appear in the Alarm app while a View app is running. During runtime, several methods can be used to switch the language of alarm comments that appear in the ViewApp. The following example shows how to switch to another language using the TitleBar app.

Alarm Comment Shown in the Default English Language



AlarmComment	Type	TimeLCT	/ Limit	CurrentV	AlarmDuration	Operator	UnAckDuration
CPU 1 Hi level load limit exceed.	System	6/25/2021 3:18:19 AM	80.0	25.78823	000 00:00:09.996		000 00:00:00.000

Alarm Comment Shown in French by Switching the Language from the TitleBar App



AlarmComment	Type	TimeLCT	/ Limit	CurrentV	AlarmDuration	Operator	UnAckDuration
Limite de charge de niveau haut.	System	6/25/2021 3:18:19 AM	80.0	25.93034	000 00:00:09.996		000 00:00:00.000

The following table shows the major steps to enable alarm comment language switching and references for more

information.

Alarm Comment Translation Workflows	For More Information, See
Configure another language for the Galaxy	Configure languages for a Galaxy
Export alarm comments in a single file to be translated	Exporting Alarm Comments for Offline Translation
Translate alarm comments into another language	Translate exported alarm comment language files
Import the translated alarm comments	Import translated alarm comment language files
Configure the language switching method	Testing Language Switching Functionality at Run Time

If you change an alarm comment after exporting the alarm comments, you must re-export the alarm comments. The translated alarm comments imported into InTouch and displayed at runtime do not change dynamically when the alarm comment is edited.

Configure languages for a Galaxy

The language settings of the Galaxy control which languages are available to graphics and alarm and event messages. You cannot add a language at the graphic or attribute level. Languages are only added at the Galaxy level using the IDE.

When you open a graphic for editing using the Graphic Editor, open a managed InTouch application in WindowMaker, or open an AVEVA OMI ViewApp, the language settings are retrieved from the Galaxy.

For example, you configure English and French languages for the Galaxy. You open graphic G1 in the Graphic Editor. graphic G1 is now configured with English and French languages. Using the IDE, you add German to the list of configured languages. You must close the Graphic Editor and open graphic G1 again to see the German language available for the graphic.

For more information about language switching in Industrial Graphics, see the *Creating and Managing Industrial Graphics User Guide*.

Add a language to a Galaxy

Every Galaxy is associated with a base language. You must configure any additional languages that you want to be supported by the Galaxy.

To add a language to a Galaxy

1. Using the System Platform IDE, open the Galaxy that you want to add a language to.
2. From the ribbon, select **Galaxy**, then select **Configure**.
3. Select **Galaxy**, then **Languages**. The **Configure Languages** dialog box appears.

Configure languages



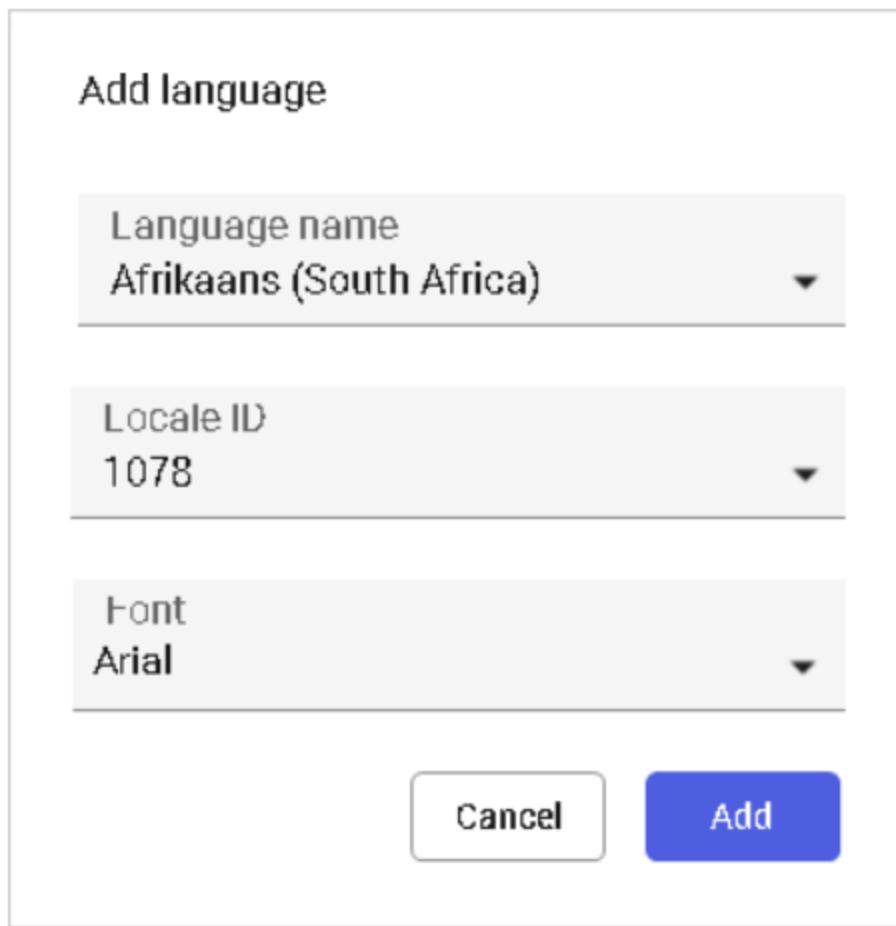
Make default

Language name	Locale ID	Font
English (United States)	1033	Arial
German (Germany)	1031	Arial

Default language English (United States)

The **Configure Languages** dialog box shows the base (default) language of the Galaxy.

4. Click the Add [+] button. The **Add Language** dialog box appears.



5. Specify the language and font for the translated text.
 - In the **Language name** or the **Locale ID** list, select the language to add. If you select the language by name, the corresponding locale ID appears in the **Locale ID** list, and vice versa.
 - In the **Font** list, select a font.
6. Click **Add** to close the **Add Language** dialog box. The language you selected now appears in the **Configure Languages** dialog box.
7. To add more languages, repeat steps 4 through 6.
8. To specify a particular language as the default, select the row in the list and then click **Make Default**.
9. When you are done, click **Save**.

Remove a language from a Galaxy

You cannot remove the default language. At least one language must be configured for a Galaxy.

To remove a language for a Galaxy

1. Using the System Platform IDE, open the Galaxy from which you want to remove a language.
2. From the ribbon, select **Galaxy**, then select **Configure**.
3. Select **Galaxy**, then **Languages**. The **Configure Languages** dialog box appears.
4. Select the language to remove and then click the **Remove[-]** button.

5. When the **Confirm Delete** dialog box appears, click **Delete**.
 - The dialog box provides an additional option to delete the alarm comment translations for the language selected. Leave the check box empty (unselected) if you want to keep the alarm comment translations for the selected language, but just want to remove the language selection option from the galaxy.
6. Click **Delete** to confirm deletion of the selected language (and the alarm comment translations for that language if selected).

Modify the font for a language

The default font for all languages is Arial. You can only set the font for a language at the time you add it. To change the font setting for a previously configured language, you must first delete it and then re-add it as described in [Add a language to a Galaxy](#).

You cannot edit the font for the default language. To change the font for the default language:

1. Set a different language as the default.
2. Delete the language for which you want to change the font.
3. Add back the language and set the font.
4. Set the language you just added as the default.

The configured font for a language is used in design time when a specific translation is supplied for a language. It is also propagated to all translations for secondary languages when the fonts are not specifically overridden by you.

For example, if you create a text element and provide translation for the secondary language without modifying the font, the text is shown using the font from the Galaxy. However, if you manually modify the font, then the text will always use the chosen font. This same behavior also applies to runtime.

For an Industrial graphic to show the updated font from the Galaxy in an InTouch ViewApp, WindowViewer must be restarted. No notification is provided to WindowViewer when the Galaxy font changes.

All managed InTouch applications are updated to use the new font for the selected language.

Change the default language for a Galaxy

When you change the default language for the Galaxy:

- The new default language is shown when the Graphic Editor opens.
- The base language for translation export is changed to the new default language.
- You cannot import translations for that language.
- Graphics opened in design time use the default language when showing text that does not have a specific translation in a secondary language.
- Graphics shown at runtime use the default language if specific translated values are not provided for the currently viewed language.
- Thumbnails for graphics are shown using the default language.

To change the default language of a Galaxy

1. From the ribbon, select **Galaxy**, then select **Configure**.
2. Select **Galaxy**, then **Languages**. The **Configure Languages** dialog box appears.
3. Select the language you want to set as the new default language.
4. Click **Make Default**.

The selected language is now the default language for the Galaxy.

Export files for offline translation

Typically, text strings are sent out for bulk translation. You can export translation files and modify them using a text editor or a spreadsheet program like Microsoft Excel.

Application Server has the following export options for localization:

- All Internationalized strings
- Internationalized strings from selected objects (one or more objects must be selected before this option can be used)
- Selected Industrial Graphics (select the graphics you want to localize before you use this option)
- All Industrial Graphics
- Alarm Fields from selected areas (select the area objects you want to localize before you use this option)
- All Alarm Fields

Export Localization ^

Configure Close

All Internationalized strings
Export language translations for all internationalized string attributes from all objects to a CSV file

Internationalized strings from selected objects
Export language translations for internationalized string attributes from selected objects to a CSV file

Selected Industrial Graphics translations
Export language translations for selected Industrial Graphics to a XML file

All Industrial Graphics translations
Export language translations for all Industrial Graphics to a XML file

Alarm fields from selected areas
Export language translations for alarms from selected objects to a CSV file

Alarm fields
Export language translations for alarms from all objects to a CSV file

If you make changes to your application after you export your dictionary files, you must export the dictionary file again. For more information, see [Export text to an existing dictionary file](#).

You can only export the text strings for one language at a time. You cannot export Industrial Graphics text strings for the default language.

In order to localize a string attribute, it must be of type InternationalizedString. Not all attributes are this type. For a list of InternationalizedString attributes, see [Attributes of Type InternationalizedString](#).

Each graphic in a Galaxy is only exported one time, no matter how many times it is referenced.

When you export the text, you specify a folder for the dictionary files. Creating a new folder to export phrases for each language makes it easy to manage dictionary files. For example, ...\\Galaxy1\\My German Files\\.

Also, alarm message files are exported as tab-delimited text files and have the following convention: Galaxy_<GalaxyName>_<LanguageID>.txt. If you will be exporting language data for different objects at different times, use separate target directories to prevent subsequent exports from overwriting the first export.

When you export the dictionary for an application, you can copy and paste the text file into Microsoft Excel.

Types of language dictionary files

Export operations to support localization can create the following types of dictionary files:

- Dictionary file (type XML) for all Visualization folder graphics, template graphics, and AutomationObject graphics. The file naming convention for graphics is: <GalaxyName>_<LanguageID>.xml. For example, if the

Galaxy name is TestSample and the language being exported is French (Language ID = 1036) then the file name is TestSample_1036.xml.

- Dictionary file (type TXT) for internationalized string attributes. The file naming convention for internationalized strings is: Galaxy_<GalaxyName>_<LanguageID>_Internationalized_strings.txt
- Dictionary file (type TXT) for alarm messages, group names, state messages, and event messages. The file naming convention for alarm messages is: Galaxy_<GalaxyName>_<LanguageID>.Alarm_Comments.txt
- Dictionary file (type XML) for each managed InTouch application.
- Dictionary file for each SmartSymbol (type XML) in managed InTouch applications.

Note: Alarm export files are text files and are therefore named and formatted differently from graphic dictionary files, which are XML files. For further information see [Export alarm comments for offline translation](#).

Export language data for all Industrial Graphics in a Galaxy

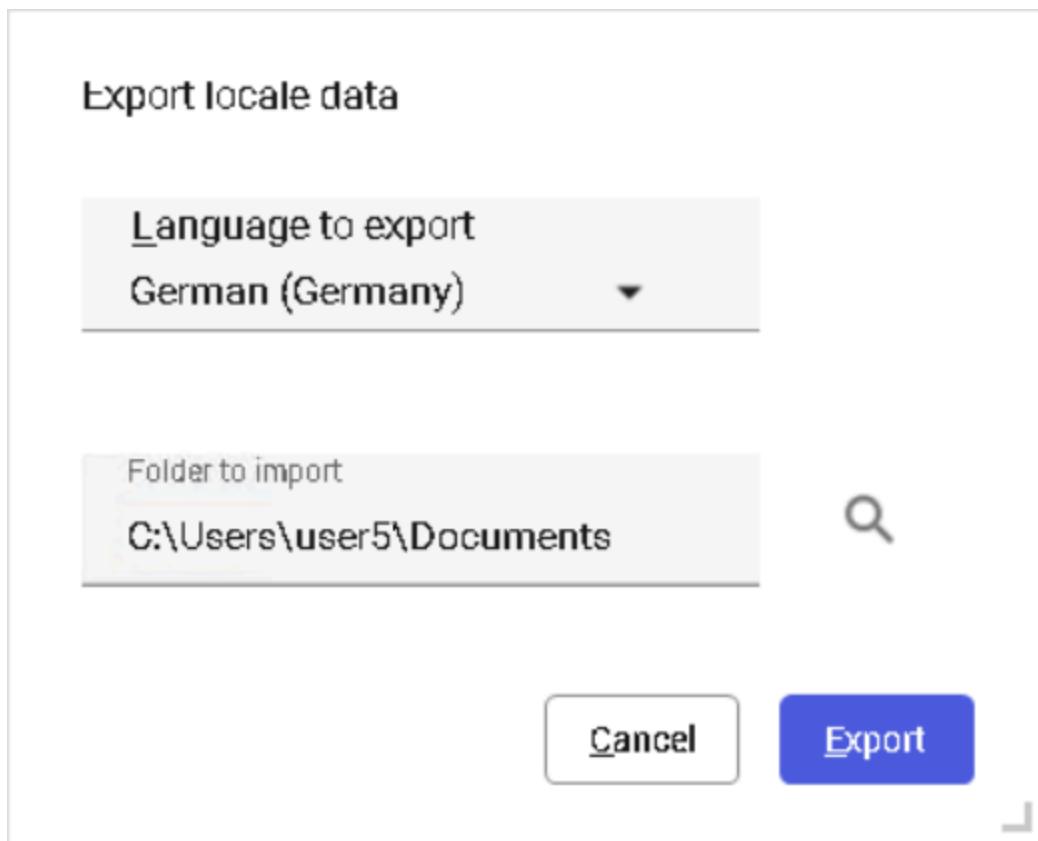
You can export language data for all Industrial Graphics in a Galaxy at one time. The export contains language data for:

- Visualization folder graphics.
- Graphics contained in AutomationObject templates, except for an \$InTouchViewApp template.
- Graphics contained in AutomationObject instances.

The export operation only applies to graphics that are checked in.

To export language data for all Industrial Graphics

1. Using the System Platform IDE, open the Galaxy for which you want to export graphic text.
2. From the ribbon, select **Galaxy**, then select **Export**.
3. Select **Localization**, then **All Industrial Graphics**. The **Export Locale Data** dialog box appears.



4. Configure the export settings.
 - In the **Languages to export** list, select the language dictionary to export. The default language is not listed.
 - In the **Path** box, type the folder to which you want to export the dictionary. Click the browse icon to select an existing folder or create a new folder.
5. Click **Export**. The export progress is shown as the XML file is created.
6. When file creation is complete, click **Close**.

Export language data for specific objects

You can export language data for the following types of Galaxy objects:

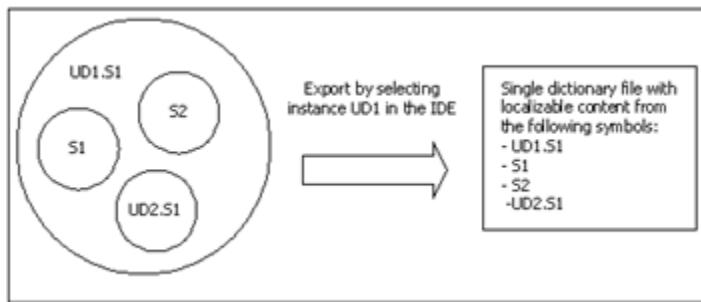
- Visualization folder graphics
- AutomationObject templates and instances
- \$InTouchViewApp templates and instances.

All graphics in an instance or template are exported.

When you export the language data for a graphic, the language data for all graphics referenced by the graphic is also exported. The exported graphics can be referenced through direct embedding or through a show graphic animation.

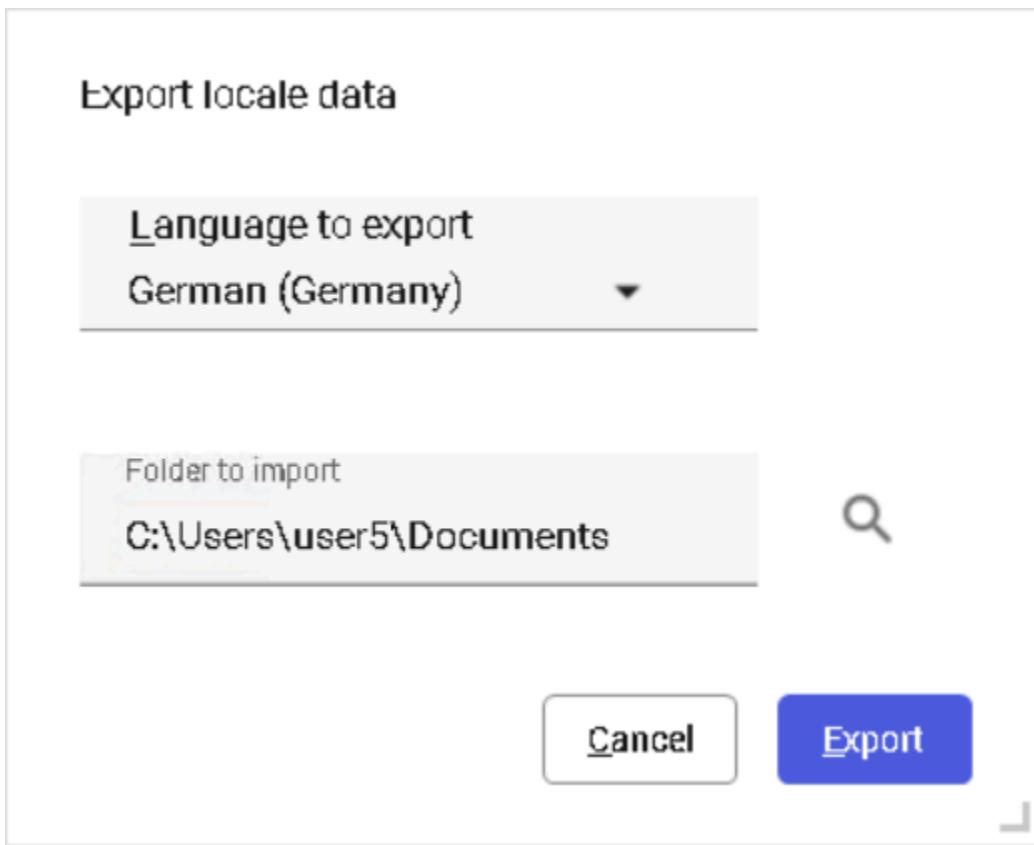
For example, a Galaxy has Symbol1 and Symbol2 defined in the Visualization folder. There are two instances called UserDefined1 and UserDefined2. UserDefined1 includes Symbol1. UserDefined2 includes Symbol1 and

Symbol2. The instance graphic UserDefined1.Symbol1 embeds Symbol1 and Symbol2 from the Visualization folder and one instance graphic UserDefined2.Symbol1. If you select the instance UserDefined1 and export the language data, then the language data would also be exported for the graphics Symbol1, Symbol2, and UserDefined2.Symbol1.



To export language data for specific objects

1. Using the System Platform IDE, open the Galaxy for which you want to export graphic text.
2. Select one or more objects to export.
3. From the ribbon, select **Galaxy**, then select **Export**.
4. Select **Localization**, and then select **Selected Industrial Graphics**. The **Export Locale Data** dialog box appears.



5. Configure the export settings.
 - In the **Language to export** list, select the language dictionary to export. The default language is not listed.
 - In the **Path** box, type the folder to which you want to export the dictionary. Click the browse icon to

select an existing folder or create a new folder.

6. Click **Export**. The export progress is shown as the XML file is created.
7. When file creation is complete, click **Close**.

Export graphic language data for a managed InTouch application

You export language data for a managed InTouch application using the IDE. You cannot export translations from within the InTouch HMI.

The export includes strings for:

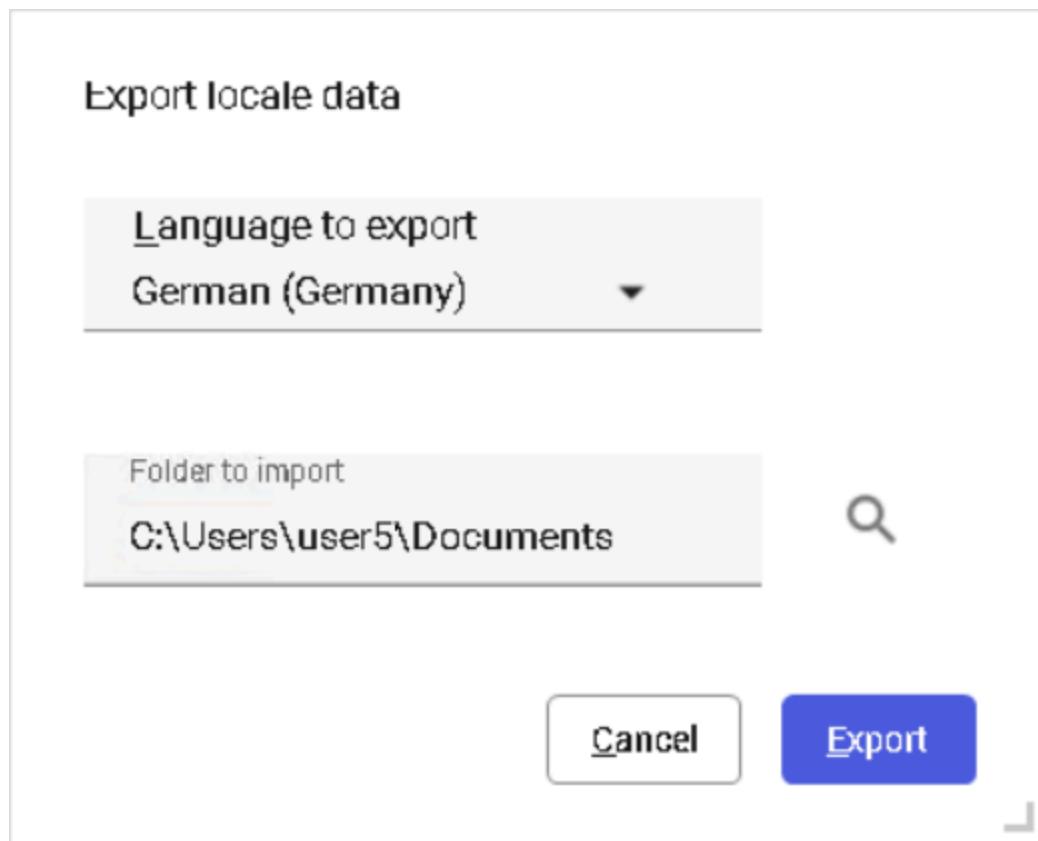
- All InTouch windows
- All SmartSymbols
- Industrial Graphics referenced by the \$InTouchViewApp template

The export causes a cascade export of all referenced Industrial Graphics.

When you export language data for a managed InTouch application, the default language for the Galaxy is ignored. The InTouch default locale is always the installed InTouch locale. If the InTouch installed locale and the Galaxy default language are not the same, the export is skipped for the selected InTouchViewApp and a message is logged.

To export language data for a managed InTouch application

1. Using the System Platform IDE, open the Galaxy for which you want to export graphic text.
2. Select one or more managed InTouch applications.
3. From the ribbon, select **Galaxy**, then select **Export**.
4. Select **Localization**, and then select **Selected Industrial Graphics**. The **Export Locale Data** dialog box appears.



5. Configure the export settings.
 - In the **Language to export** list, select the language dictionary to export. The default language is not listed.
 - In the **Path** box, type the folder to which you want to export the dictionary. Click the browse icon to select an existing folder or create a new folder.
6. Click **Export**. The export progress is shown as the XML file is created.
7. When file creation is complete, click **Close**.

Export graphic language data for a published InTouch application

If you export languages for a published InTouch application, the following are not included:

- Embedded Industrial Graphics
- Custom property overrides
- String overrides.

The export only includes native InTouch translations.

Export text to an existing dictionary file

If you change your application after you translate the text strings, you need to export the text again. For more information, see [Export files for offline translation](#).

If you export a file with the same name to the same directory, you will be asked to confirm that you want to overwrite the existing file.

If you click **Yes**, the existing file is deleted and the current text for graphics in the Galaxy are exported as a new file.

Any existing translations for a graphic are reflected in the new .xml file.

Translate exported graphic language files

The procedures and tools for translating the exported language files are similar for both graphic text and for alarm comments. However, there are important differences. Procedures for each are described in this section.

Translate exported graphic text dictionary files

After you export the dictionary file containing your application text, use Microsoft Excel to edit the text.

To translate an exported dictionary file

1. Open the XML file in Excel. The **Open XML** dialog box appears.
2. Click **As an XML list**, then click **OK**. A message may appear informing you that an XML schema will be created.
3. Click **OK**.

The XML file opens in Excel with columns for the:

- Phrases in your application.
- Translated phrases from the translator.
- Translated font name.
- Translated font properties.
- Translated font size.
- Base font properties.
- Base font size.
- Context, phrase ID, language ID and foreign language ID.

Phrase	Translation	Translated	Translated	Translated	BaseFont	BaseFont	BaseFont	Context
2 Do not edit	Provide transl	Can edit	Can edit	Can edit	Do not edit	Do not edit	0	SP03296
3 #####			0	Arial	B		10	\$Master_Area.AlarmState:Text4;TextElement;Text
4 Mode			0	Arial	B		10	\$Master_Area.AlarmState:Text3;TextElement;Text
5 Area Alarms			0	Arial	B		10	\$Master_Area.AlarmState:Text2;TextElement;Text
6 Only transla	Do not edit		0	Arial	B		12	\$Master_Area.AlarmState:ComboBox1;ElementFont;Font
7 Running			0				0	\$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item0
8 Maintenance			0				0	\$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item1
9 Startup			0				0	\$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item2
10 Shutdown			0				0	\$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item3
11 Testing			0				0	\$Master_Area.AlarmState:ComboBox1;ComboBoxAnimation;Item4
12 Set PlantState			0	Arial	B	10	\$Master_Area.AlarmState:Text1;TextElement;Text	
13 #####			0	Arial	B	10	\$Master_Area.MapPin:Text1;TextElement;Text	
14 ###			0	Arial		10	\$Master_Area.MapPin:Text2;TextElement;Text	
15 Alarms			0	Arial		8	\$Master_Area.MapPin:Inchd;TextElement;Text	

Important: Only modify data in the **Translation**, **TranslatedFontSize**, **TranslatedFontName**, and **TranslatedFontProperty** columns. Do not change any column header. Do not insert or delete rows.

4. Type the language-specific text in the **Translation** column in the row that corresponds with the base language string in the **Phrase** column.

5. If necessary, change the font parameters for the translated strings. If you only provide a translation, the Galaxy-configured font for the language is used to render the text after the translation is imported. If you specify a font, it overrides the Galaxy-configured font.

- In **TranslatedFontName** column, type the font name.
- In the **TranslatedFontProperty** column, type the notation for the font properties:

B = **bold**

I = *italic*

U = underline

For example, if you want the text to be bold, type **B** in the **TranslatedFontProperty** column. If you want the text to be bold and underlined, type **BU** in the **TranslatedFontProperty** column.

6. Save the file using XML Data as the file type.

Important: If you save as another file type, such as XML Spreadsheet, Excel changes the schema and the Galaxy cannot load the file. If you change the name of the XML file, the file will not import properly into the Galaxy.

Import translated language files

You can import alarm comment language files for re-export by area to facilitate the export of new, untranslated alarm comments, and to facilitate re-export of previously translated alarm comments that have been changed.

For graphic text, you must import the translated dictionary files for each language to enable runtime language switching for those languages. All dictionary files for a given language should be placed in the same folder.

You can import files for only one language at a time. When you import, you select the desired language and specify the files to import.

Import translated graphic dictionary files

All affected graphics and relevant objects are checked out before the import begins and are checked in when the import is done. If an affected graphic or object is already checked out, a message appears in the progress dialog box and the import is skipped for the checked-out object.

For a published InTouch application, only the native InTouch translations are imported.

You can configure how you want Galaxy and graphic/object mismatches handled during the import.

The import is skipped if:

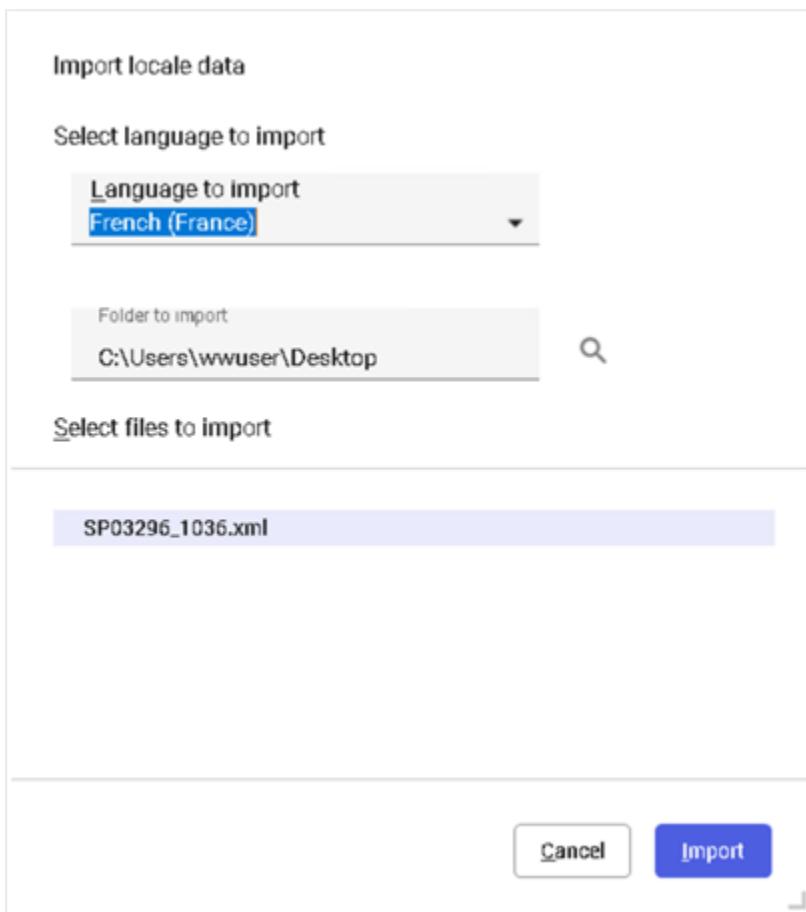
- The default locale specified in the .xml translation file does not match the current default locale of the Galaxy.
- An InTouchViewApp's installed locale does not match the current default locale of the Galaxy.

Important: You cannot cancel the import after it starts.

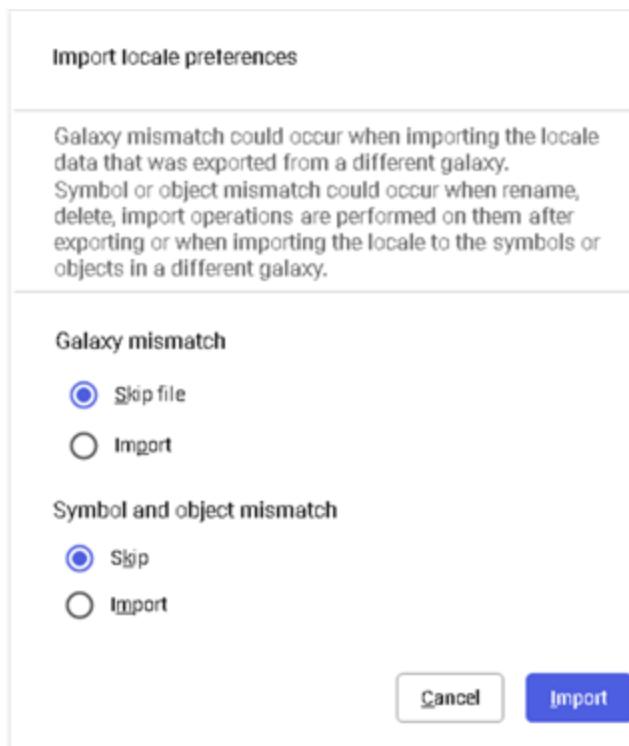
To import a translated dictionary file

1. Using the System Platform IDE, open the Galaxy for which you want to import graphic text.
2. Close all editors and check in all Galaxy objects.
3. From the ribbon, select **Galaxy**, then select **Import**.

4. Select **Localization**, then select **Industrial Graphics translation**. The **Import locale data** dialog box appears.



5. Configure the import settings.
 - In the **Language to import** list, select the language dictionary to import.
 - In the **Folder to import** box, specify the folder that includes the dictionary file to import.
 - In the **Select files to Import** box, select the .xml files to import. Only files that include the current Galaxy name and the locale ID for the selected language are shown.
6. Click **Import**. The **Import Locale Preferences** dialog box appears.



7. In the **Galaxy mismatch** area, configure how you want Galaxy mismatches handled. A Galaxy mismatch occurs when you try to import a translation file that was exported from a different Galaxy. The Galaxy name in the translation .xml file is used to match the current name of the Galaxy.
 - Click **Skip file** to skip all the files that do not contain the current Galaxy name in the file name.
 - Click **Import** to import all the selected files, regardless of what the Galaxy name is in the .xml filename.
 8. In the **Symbol/Object Mismatch** area, configure how you want graphic and object mismatches handled. A graphic or object mismatch occurs when the name of the graphic and the internal ID (GObjectID) of the graphic do not match what is within the .xml file. Objects include AutomationObjects and InTouchViewApp objects.
 - Click **Skip** to skip the graphics and objects that have mismatched names or IDs in the .xml file
 - Click **Import** to import a graphic or object only if it has a matching name or matching ID. If the name resolves to one object, and the ID resolves to another object, then the import is skipped.
- For examples, see [Examples of graphic or object mismatch handling during language imports](#).
9. Click **Import**. The import progress is shown.
 10. Click **Close**. The **Import Language Dictionary Files** dialog box appears.
 11. Click **Check In**. The check in progress is shown.
 12. Click **Close**. A summary of the import is shown.
 13. Click **OK**.

Examples of graphic or object mismatch handling during language imports

The following table shows an example of handling mismatch conditions for a Visualization folder graphic. The bold name/ID is the matching name/ID in the .xml file and current Galaxy during import.

	Graphic Name and ID in the Galaxy while Exporting	Graphic Name and ID in the Galaxy while Importing	Change made after export and before import	Skip Option Selected	Import Option Selected
1	S1, 100	S1, 100	No change	Import to S1	Import to S1
2	S1, 100	S2, 100	Rename S1 to S2	Skip	Import to S2
3	S1, 100	S1, 200	Delete S1, Create/Import S1	Skip	Import to S1
4	S1, 100	S1, 200 S2, 100	Rename S1 to S2 Create/Import S1	Skip	Ambiguous, skip import
5	S1, 100	No S1 and No 100 in the Galaxy	Deleted S1	Skip	Skip import

The following table shows an example of handling mismatch conditions for an AutomationObject graphic. The bold name/ID is the matching name/ID in the .xml file and current Galaxy during import.

	Graphic Name and ID in the Galaxy while Exporting	Graphic Name and ID in the Galaxy while Importing	Object Name and ID in the Galaxy while Importing	Change made after export and before import	Skip Option Selected	Import Option Selected
1	Pump1, 10	S1, 100	Pump1, 10	No change	Import to S1 only if Pump1 has S1 with an ID of 100	Import to S1 only if Pump1 has S1 with an id of 100
2	Pump1, 10	S1, 100	Pump2, 10	Rename Pump1 to Pump2	Skip	Skip import if S1 and 100 pointing to two different graphics in Pump2 (ambiguous) Import to S1 only if Pump2 has S1 or a graphic with id of 100.

	Graphic Name and ID in the Galaxy while Exporting	Graphic Name and ID in the Galaxy while Importing	Object Name and ID in the Galaxy while Importing	Change made after export and before import	Skip Option Selected	Import Option Selected
3	Pump1, 10	S1, 100	Pump1, 20	Delete Pump1 Create and Import Pump1	Skip	Skip import if S1 and 100 pointing to two different graphics in Pump1 (ambiguous) Import to S1 only if Pump1 has S1 or a graphic with id of 100.
4	Pump1, 10	S1, 100	Pump1, 20 Pump2, 10	Rename Pump1 to Pump2 Create and Import Pump1	Skip	Ambiguous, skip import
5	Pump1, 10	S1, 100	No Pump1 and No 10 in the galaxy	Deleted Pump1	Skip	Skip import S1

Language data handling for galaxy operations

If you import or export objects, all language data within the associated graphics is imported or exported.

You can export all objects in the Galaxy, selected instances/templates, or selected graphics from the Visualization folder. All language data is exported as part of the graphics definition, regardless of the languages configured in the Galaxy.

You can import Galaxy objects that contain language data from the same Galaxy or a different Galaxy. All language data in the Industrial Graphics is imported, regardless of what languages are configured for the target Galaxy.

No Industrial Graphic or InTouchViewApp language data is read or modified during the import operation.

If you import an unmanaged InTouch application, the configured locales in the Galaxy are applied to the unmanaged InTouch application. No element translations in the InTouch application are removed during import, even though languages may no longer be visible in the InTouch HMI or available at runtime in WindowViewer.

Export alarm comments for offline translation

You typically send alarm comment text out for bulk translation. You can export alarm comments and modify them using a text editor or a spreadsheet program such as Microsoft Excel. After translation, the files can be re-imported to the Galaxy for runtime alarm comment language switching.

Guidelines and recommendations

The following guidelines and recommendations will help you make best use of the alarm comment language switching feature:

Important: The exported file name is generated automatically and must not be changed.

Organize your export

Follow these recommendations when exporting alarm comments for translation:

- You must export alarm comment text for only one language at a time.
- When you export text, you must specify a folder to place the language files. We recommend that you create a separate folder for each language. For example, ...\\Galaxy01\\ChineseFiles\\.
- For large Galaxies, exporting files for each area, after being translated and re-imported in the Galaxy, will perform faster when the alarm clients access those files.
- For small Galaxies, we recommend that you export the entire Galaxy for translation.

Translation file formatting and editing

Microsoft Excel is recommended for translation file formatting and editing, since it supports large worksheets and provides enhanced formatting capabilities to convert the text file into tabular format. See [Export alarm comments from very large galaxies](#).

Reimport

- We highly recommend re-importing the translated files to the Galaxy after translation. Although re-import is not mandatory, you can use the re-import to optimize the export files.
- If you make changes to your alarm comments after you export comment text, you must export the alarm comments again. For this reason, we further recommend using the import functionality to reimport the language file after each translation. This practice will avoid the need to re-translate alarm comments you have already translated.

For further information and procedures, see the following sections:

[Export alarm comments from very large galaxies](#)

[Export alarm comments by area](#)

[Import translated alarm comment language files](#)

About the alarm comments language file

The alarm comment language switching feature exports alarm comments, messages and other data to a .txt file. That file can be edited directly, but working in a text editor is typically more difficult and prone to errors.

- All files export as Unicode. We recommend that you save the exported files as Unicode.
- We recommend that you edit the .txt file using Microsoft Excel. Be sure to save the edited file as a .txt file.

The language file name is not user-configurable. The file naming convention for alarm comment export is:

Galaxy_<GalaxyName>_<localeID>_Alarm_Comments.txt

or

Galaxy_<GalaxyName>_<AreaName>_<localeID>_Alarm_Comments.txt

where the locale ID is the selected language decimal identification number.

For example, if the Galaxy name is "TestSample" and the exported language is Chinese, the language file name would be Galaxy_TestSample_2052_Alarm_Comments.txt. The locale ID for Chinese is 2052.

For example, if the Galaxy name is "TestSample" and the Area name is "Area001", and the exported language is German, the language file name would be Galaxy_TestSample_Area001_1031_Alarm_Comments.txt.

See [Translate exported alarm comment language files](#) for further information about alarm comment language files.

Export alarm comments from very large galaxies

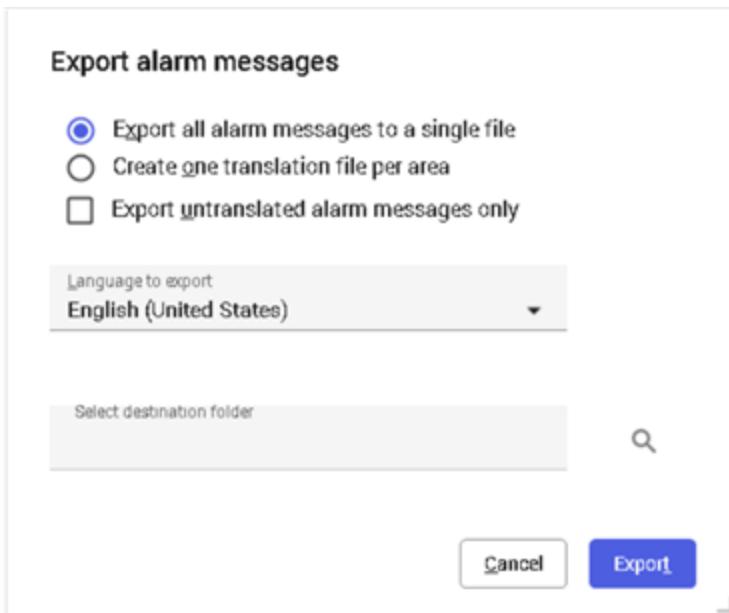
You can export language data in one operation for all alarm comments in a Galaxy. However, very large Galaxies with numerous alarm comments require longer export processing time and can result in a very large .txt language file. For handling large Galaxies and a large number of alarm comments, we recommend that you export alarm comments by Area rather than exporting alarm comments for the entire Galaxy to a single language file. See [Export alarm comments by area](#).

Export all galaxy alarm comments

You can export language data for all alarm comments in a Galaxy at one time. The export operation only applies to objects that are checked in.

To export all Galaxy alarm comments to a single file

1. Using the System Platform IDE, open the Galaxy in which you want to export alarm comments.
2. From the ribbon, select **Galaxy**, then select **Export**.
3. Select **Localization**, then select **Alarm Fields**. The **Export Alarm Messages** dialog box appears.



4. Configure the export settings:
 - a. Select the **Export all alarm messages to a single file** radio button.
 - b. Check the **Export untranslated alarm messages only** check box if you have already exported, translated and re-imported your language file and you want to export only newer, untranslated messages. Otherwise, leave the check box unchecked.
 - c. In the **Language to export** pull-down list, select the language to export. Make sure that you have configured the Galaxy for the language that you want to export, since the export file will contain the locale ID of the language you select. You can export only one language at a time.
 - d. In the **Path** box, type the folder to which you want to export the language file. Click the browse icon to select an existing folder or create a new folder.
5. Click **Export**. The export progress box appears.
6. Click **Close** when the export completes.

Export alarm comments by area

You can export all alarm comments by Area or you can export Alarm comments for a specific Area.

About file names

The *Galaxy_<GalaxyName>_<AreaName>_<localeID>_Alarm_Comments.txt* file name convention applies to specific Area alarm comment exports. Area names will be included the file name. For example, given a Galaxy name "TestSystem", an Area name "Area001", and an export to Chinese (language designation 2052), the language file name would be *Galaxy_TestSystem_Area001_2052_Alarm_Comments.txt*.

Export objects not assigned to an area

System Objects typically are not assigned to an Area. These are:

- Platform Objects
- Device Integration Objects
- Engine Objects

For purposes of language export, these System Objects form their own "Area" or become notification distributors for the purpose of sending alarms. Exporting System Objects follows these important conventions:

- System Objects can only be exported from the **Galaxy** menu. They cannot be selected and exported using a context menu.
- System Objects can only be exported by exporting alarm comments for all areas.

To export alarm comments for all areas

1. Using the System Platform IDE, open the Galaxy for which you want to export alarm comments.
2. From the ribbon, select **Galaxy**, then select **Export**.
3. Select **Localization**, then select **Alarm Fields**. The **Export Alarm Messages** dialog box appears.
4. Configure the export settings:
 - a. Select the **Export all alarm messages to a single file** radio button.
 - b. Check the **Export untranslated alarm messages only** check box if you have already exported, translated and re-imported your language files by Area and you want to export only newer, untranslated messages. Otherwise, leave the check box unchecked.
 - c. In the **Language to export** pull-down list, select the language to export. You can export only one language at a time.
 - d. In the **Destination folder** box, type the folder to which you want to export the language file. Click the browse icon to select an existing folder or create a new folder.
5. Click **Export**. The export progress box appears.
6. Click **Close** when the export completes.

Re-export alarm comments

After exporting and translating alarm comments, you can re-import them into Application Server to help maintain up-to-date alarm comment language files.

After adding new alarm comments that require translation or after modifying existing, already translated alarm comments, you must re-export the language file to update the translations.

Export new untranslated alarm comments

In the **Export Alarm Messages** dialog box, select the **Export untranslated alarm messages only** check box. This creates a file named _untranslated.txt. For example, exporting untranslated alarm comments in Chinese for a Galaxy name "TestSystem" would create a file named Galaxy_TestSystem_2052_Alarm_Comment_Untranslated.txt.

Export modified existing alarm comments

You can make changes to alarm comments that you previously exported, translated and imported back into Application Server. You must re-export the alarm comment language file(s) to update the translations.

In the **Export Alarm Messages** dialog box, select the same language option and folder you selected for the previous export. When asked, confirm that you want to overwrite the existing file.

Translate exported alarm comment language files

Alarm comments are exported to a .txt file, located in the selected directory. You can edit the files with a text editor, but we recommend Microsoft Excel as the editor. Using the format-as-table feature in Excel greatly simplifies editing the specific text to be translated.

To translate an exported alarm comment file

1. Open the .txt file in Excel. The Excel **Text Import Wizard** appears. Follow the wizard instructions to import the text file in tab delimited, general column data format. The file appears unformatted in Excel.

Tip: You can open the file in text editor, and then copy and paste the file contents into a blank Excel worksheet instead of using the import wizard. The results will be the same

	A	B	C	D	E	F	G	H	I	J	K	L
1	Unique Phrases:											
2												
3	PhraseId (Default M Translated Message (EDIT THIS COLUMN)											
4	3 AppEngine											
5	4 Enterprise											
6	5 OMI_ViewApp/Desktop											
7	6 Plant											
8	7 Plant_Area											
9	8 Platform											
10	9 Simulator											
11	10 Site											
12	11 Sys											
13	12 The AppEngine hosts and schedules execution of Application Objects, Areas, and Device Integration Objects.											
14	13 The OPCClient provides connectivity to OPC servers.											
15	15 ViewEngine											
16	16 Device write failed -											
17	17 Device write failed - Array index is out of range -											
18	18 Device write failed - Attribute not writable -											
19	19 Device write failed - Data value out of range -											
20	20 Device write failed - Incorrect data type -											
21	21 Device write failed - Invalid Attribute -											
22	22 Device write success -											
23	23 started											
24	24 started successfully.											
25	25 Stopped											
26	26 stopped successfully.											
27	27 Write access denied -											
28	28 Write access denied - Alarm ACK has already been done -											
29	29 Write access denied - Array index is out of range -											
30	30 Write access denied - Attribute not writable -											
31	31 Write access denied - Data value out of range -											
32	32 Write access denied - Incorrect data type -											
33	33 Write access denied - User does not have Alarm ACK privileges -											
34	34 Write access denied - User does not have permission to modify this type of attribute -											
35	35 Write access denied - User does not have permission to write to this type of attribute											
36	36 Write access denied - Users cannot modify configuration attributes while object is onscan -											
37	37 Write access denied - Verifier does not have permission to verify write for this type of attribute											
38	38 Write failed - Invalid Attribute -											
39	39 Write success -											
40												
41												
42	Alarms:											
43												
44	PhraseId (Alarm (DO NOT EDIT)											
45	3 [_GROUP_NAME_] AppEngine											

2. Format the text as a table in Excel:
- Select all of the rows containing data in Column A through Column C.
 - Select the **Home tab**, then select **Format as Table**.
 - Select the color pattern you want and click **OK** in the **Format as Table** dialog box. Excel will format the region to be edited. Resize the columns as needed.

A	B	C
1 Column1	Column2	Column3
2 Unique Phrases:		
3		
4 Phraseld (DO NOT EDIT)	Default Message (DO NOT EDIT)	Translated Message (EDIT THIS COLUMN)
5 3 AppEngine		<localized text message 1>
6 4 Enterprise		<localized text message 2>
7 5 OMI_ViewApp/Desktop		<localized text message 3>
8 6 Plant		<localized text message 4>
9 7 Plant_Area		<localized text message 5>
10 8 Platform		<localized text message 6>
11 9 Simulator		<localized text message 7>
12 10 Site		<localized text message 8>
13 11 Sys		<localized text message 9>
14	The AppEngine hosts and schedules execution of Application Objects, Areas, and Device Integration Objects.	<localized text message 10>
15	The OPCClient provides connectivity to OPC servers.	<localized text message 11>
16	13 ViewEngine	<localized text message 12>
17	15 Device write failed - Device write failed - Array index is out of range -	<localized text message 13>
18	17 range -	<localized text message 14>
19	18 Device write failed - Attribute not writable	<localized text message 15>
20	Device write failed - Data value out of range -	<localized text message 16>
21	20 Device write failed - Incorrect data type -	<localized text message 17>
22	21 Device write failed - Invalid Attribute -	<localized text message 18>
23	22 Device write success -	<localized text message 19>
24	23 started	<localized text message 20>
25	24 started successfully.	<localized text message 21>
26	25 Stopped	<localized text message 22>
27	26 stopped successfully.	<localized text message 23>
28	27 Write access denied -	<localized text message 24>
29	Write access denied - Alarm ACK has already been done -	<localized text message 25>
30	Write access denied - Array index is out of range -	<localized text message 26>
31	Write access denied - Attribute not writable -	<localized text message 27>
32	Write access denied - Data value out of range -	<localized text message 28>
33	32 Write access denied - Incorrect data type -	<localized text message 29>
34	Write access denied - User does not have Alarm ACK privileges -	<localized text message 30>

The Excel display is divided into four regions:

- Column 1 contains Phrase ID numbers used internally in the language file. **Do not edit** this column.
 - Column 2 contains the exported alarm messages. **Do not edit** this column.
 - Column 3 contains the alarm message translations. Edit this column with your translation text.
 - Below the alarm message text columns are the alarms and phrase IDs, which map to column 1 and are used internally in the language file. Do not edit this area.
3. Translate the alarm message text. Type the alarm message translations into the appropriate rows in column 3.

Important: Only edit the text in column 3, translations. No other text in the file may be edited. You can edit the original alarm comment text only within the System Platform IDE.

4. Save the file as a .txt file in the directory you originally selected for export.

Important: Do not change the file name or the file type, otherwise it will not import correctly into the Galaxy.

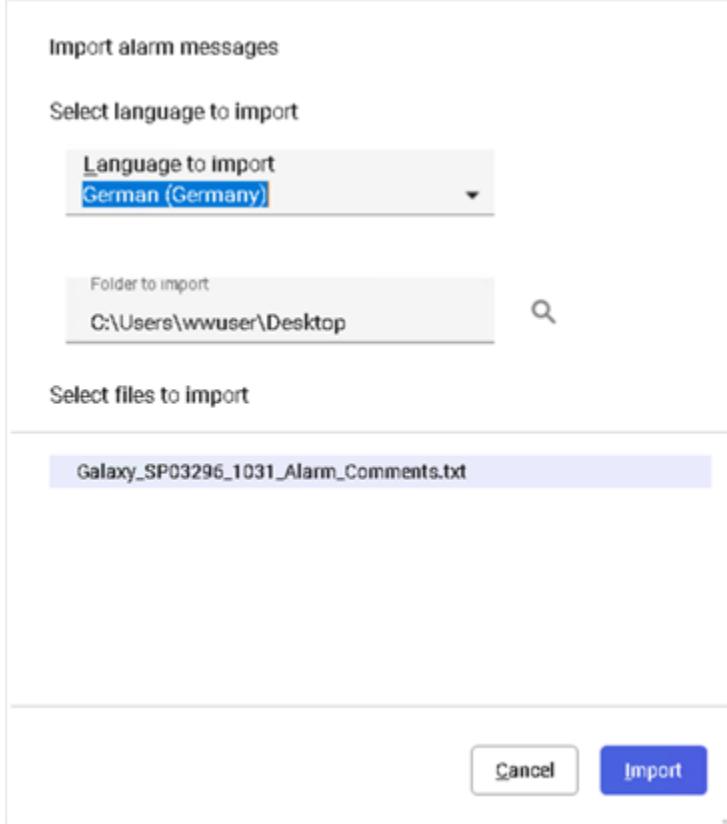
Import translated alarm comment language files

You can import translated alarm comment language files only from the **Galaxy** menu, not from the context menu for specific Areas.

To import alarm comment language files, you must first have exported alarm comments for translation of some or all of the alarm comments.

To import a translated alarm comment language file

1. Using the System Platform IDE, open the Galaxy in which you want to import translated alarm comments.
2. On the **Galaxy** menu, select **Import**, then select **Localization**, and then select the **Alarm Fields** menu option. The **Import Alarm Messages** dialog box appears.



3. Configure the import settings:
 - a. In the **Language to import** pull-down list, select the language to import. You can import only one language at a time.
 - b. In the **Folder to import** box, type or browse to the folder where you previously exported your language file. The available language files appear in the **Select files to import** list.
 - c. Select a file or files to import.
4. Click **Import**. The import progress box appears.

5. Click **Close** when the import completes.

Note: If an alarm name or message do not match, the alarm comment import will display a message for each failed alarm message import.

Test the language switching functionality at run time

We recommend that you test the runtime language switching functionality. Follow the procedures outlined in this chapter to enable runtime language switching in your application. Then import the appropriate language file into InTouch. The import step applies to both stand-alone and managed InTouch applications.

To test the language switching functionality

1. Open the application in WindowViewer.

2. On the **Special** menu, point to **Language**, and then click the name of the language to switch to.

The information from the corresponding translated dictionary file (if one exists and has been imported into InTouch) loads and appears.

3. When you are done, click **Close**.

About predefined language attributes

AVEVA OMI includes a set of predefined Language attributes. These attributes identify the currently-selected language used by the ViewApp by name or by an ID number, and can be used in action scripts and animations to define actions that can change the current language and locale in a running ViewApp. These attributes can also be incorporated programmatically into an WPF-based app that can be added to a ViewApp. When used in an app, users can directly select a new language from a list of configured languages.

A typical runtime language workflow based on language attributes is for the user to select a graphic element shown in a ViewApp that includes an action script or animation containing a Language attribute. The script or animation switches to another pre-configured language. When implemented inside of an app, for example, in the TitleBar app, users can be presented with a drop-down list of all configured languages within the Galaxy, and can select one of the listed languages.

About attribute setting

All language attributes operate in the MyViewApp.Language namespace. Language attribute names are specified with the prefix MyViewApp.Language, in the form MyViewApp.Language.*attribute_name*.

When an attribute is set, it updates the other two attributes with their equivalent values. For example, setting CurrentLanguage will also set CurrentLCID with the LCID number and CurrentCultureInfo with the Culture Information that correspond to the attribute value set for CurrentLanguage.

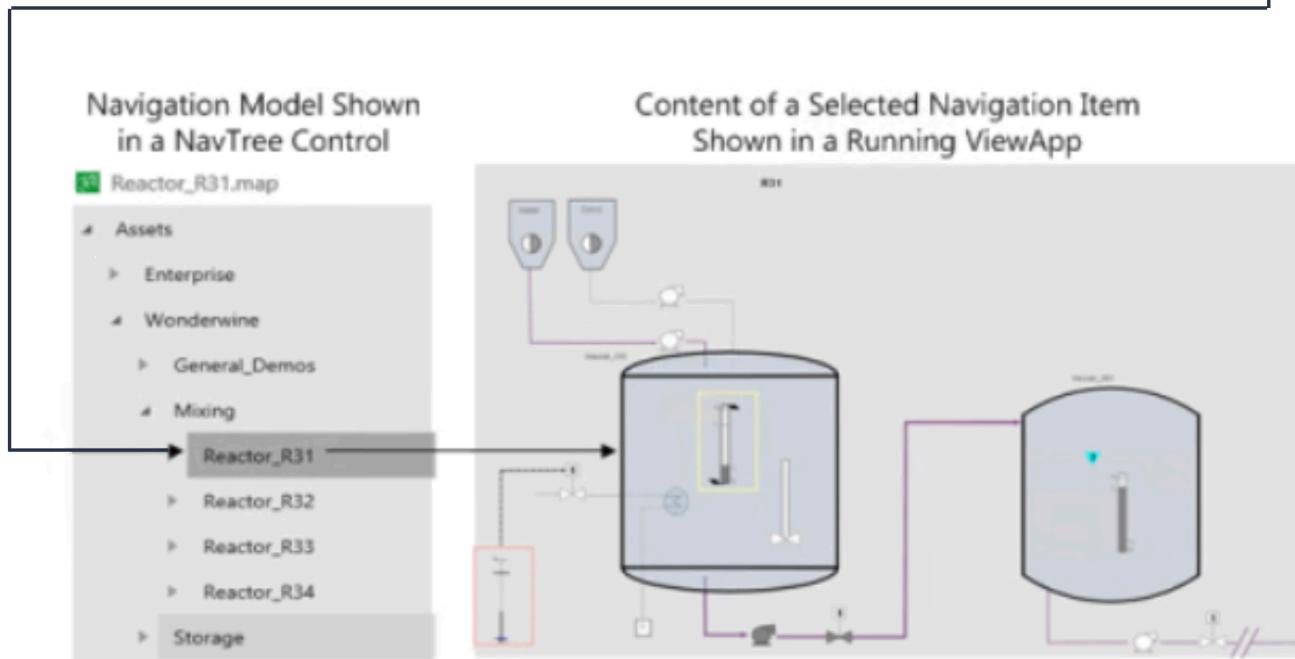
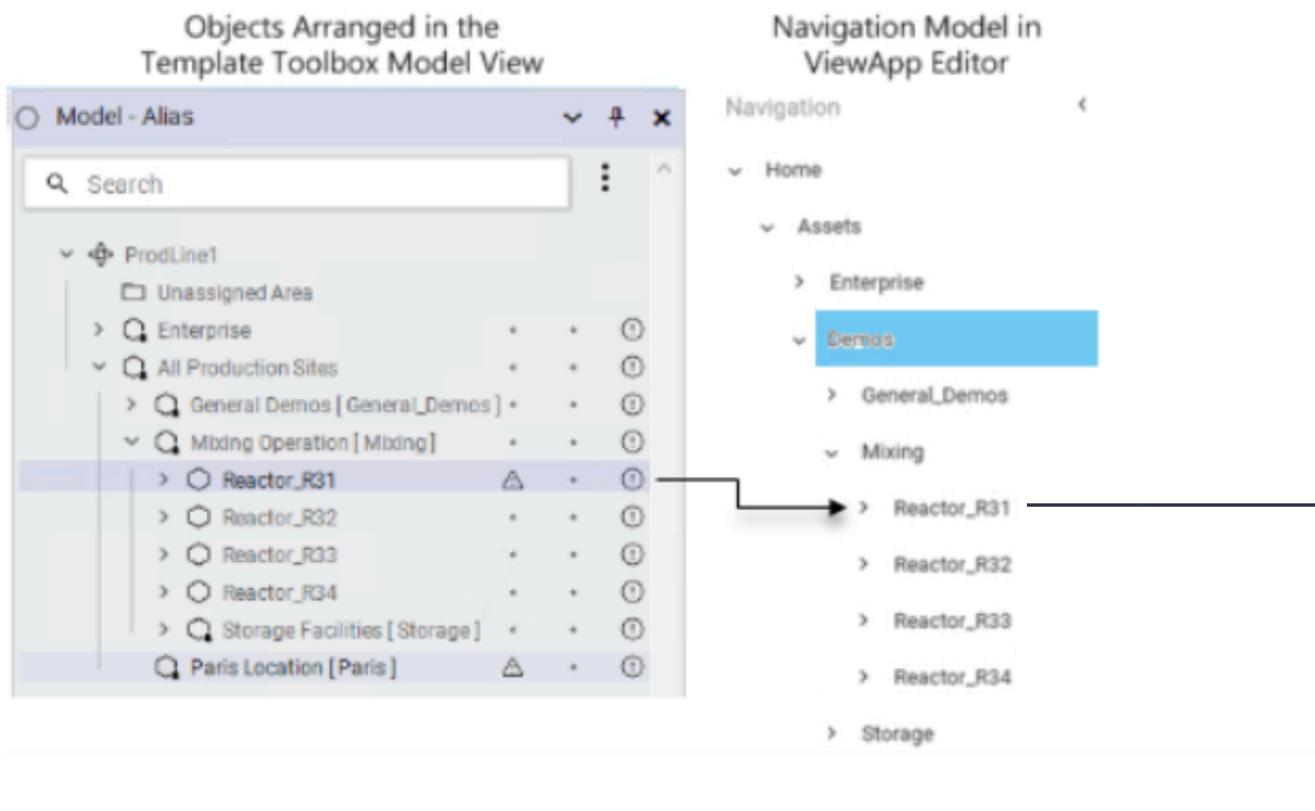
Attribute Name	Data Type	Retentive	Attribute Type	Initial Value	Description
CurrentLanguage	String	True	Read/Write	None	String value of the language and locale, as specified for the

					.NET framework. For example, English-United States.
CurrentLCID	Integer	True	Read/Write	None	Integer value of the language and locale, as specified for the .NET framework. For example, 1033 (for English-United States).
CurrentCultureInfo	String	True	Read/Write	None	String value (abbreviated) of the language and locale, as specified for the .NET framework. For example, EN- US.

When an attribute is set, it updates the other two attributes with their equivalent values. For example, setting CurrentLanguage will also set CurrentLCID with the LCID number and CurrentCultureInfo with the Culture Information that correspond to the attribute value set for CurrentLanguage.

About ViewApp navigation

Navigation enables users to browse for and show information obtained from the constituent elements of a running ViewApp. The simplest type of navigation uses a visible component like a tree control that shows the hierarchical organization of objects or graphics used in the ViewApp. Selecting an item from the tree launches an action that shows graphics or other types of content that appear in one or more panes of the ViewApp's layout.



ViewApp navigation consists of three essential components:

- Navigation model

A navigation model is the view composition engine that drives the presentation of content from a layout or ViewApp to a screen viewed by a user. A navigation model defines a hierarchy of navigation items, references to visual content used by navigation aware controls, and the actions to execute when a navigation item is selected.

- Navigation item

A navigation item is a unique element of the navigation model that contains the necessary information to show a visual representation of an item in various forms and triggers one or more navigation actions after being selected during runtime.

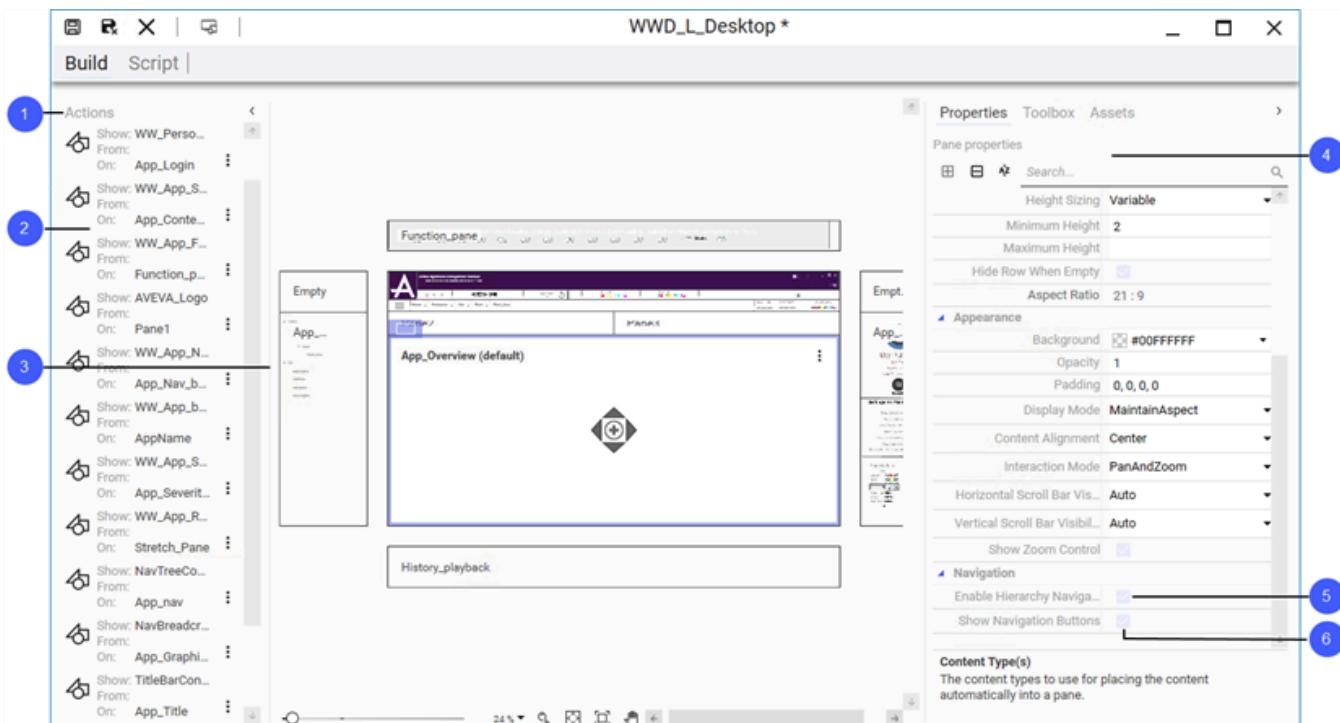
- Navigation actions

Navigation behavior during runtime by navigation aware controls to show the information present in the navigation model.

The Layout and ViewApp editors provide integrated components and properties to configure navigation in a ViewApp. For more information about configuring navigation in each editor, see [About the Layout Editor and navigation](#), and [About the Layout Editor and navigation components](#).

About the Layout Editor and navigation

The following figure shows the major components and pane properties of the Layout Editor to configure navigation for a ViewApp.

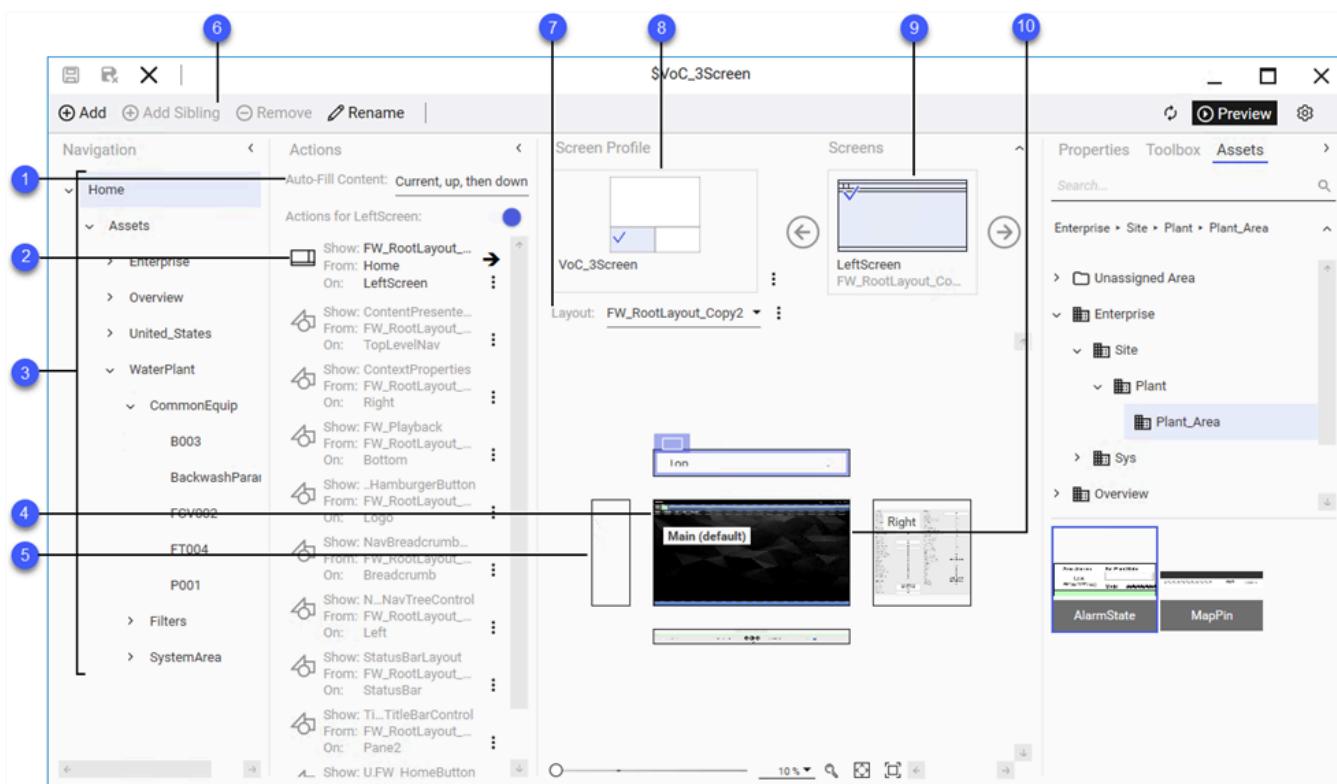


1	Actions List	Show navigation actions list associated with content added to the panes of a layout.
2	Actions	Navigation action of a piece of content added to a pane.
3	Navigation Control	Navigation tree control that shows the ViewApp's navigation model.

		During runtime, users can select a navigation item from the navigation control to show content.
4	Use For Auto-Fill	Although not shown in this graphic of the Layout Editor, the Use For Auto-Fill property determines if the pane can be filled with content automatically by Auto-Fill mode or not. Auto-Fill modes are selected from the ViewApp Editor.
5	Enable Hierarchy Navigation	Enables hierarchical navigation from the current context to show other content within a multi-content pane or neighboring navigation items by means of a vertical swipe gesture or by clicking on navigation buttons shown on the pane when ShowNavigationButtons property is set to true.
6	Show Navigation Buttons	When set to true, shows graphic buttons on the pane to make a hierarchical or context navigation action while a ViewApp is running.

About the Layout Editor and navigation components

The following figure shows the major components of the ViewApp Editor to configure navigation for a ViewApp.



1	Auto-Fill Options	The Auto-Fill Content field includes a drop-down list of Auto-Fill options that set the possible navigation behaviors specific to the selected navigation item. For more information about Auto-Fill navigation, see About auto-fill navigation .
2	Actions List	The Actions field shows a list of content assigned to the panes of the layout applied to the selected screen. Each item in the Actions list includes a set of layout or content actions to remove or edit the pane's current content.
3	Navigation Model	The Navigation field shows the hierarchy of navigation items in the ViewApp. The entire set of navigation items is the navigation model, which includes references to visual content and assigned navigation behavior when an item is selected.

4	Configuration View	Thumbnail representation of the content that appears in a pane while configuring ViewApp navigation. The icon dynamically updates as you add content to panes or assign navigation behaviors to items.
5	Navigation Control	A Navigation tree control shows a visual representation of the ViewApp's navigation model with a set of selectable navigation items arranged in a hierarchical tree. During runtime, users can select a navigation item from the control to set the context of the ViewApp's view.
6	Command Bar	Commands to add, remove, or rename navigation items to create a custom navigation model of a ViewApp. For more information about creating custom navigation in a ViewApp, see About ViewApp custom navigation .
7	Layout	The Layout field shows the name of the layout applied to the selected screen in the screen profile. The Layout field contains a drop-down list of all layouts in the Galaxy. You can select another layout from the list to replace the current layout assigned to the screen. For more information about assigning another layout to the current screen from the ViewApp Editor, see Manage layouts in a ViewApp . The Layout field also includes commands to remove the current layout from the selected screen or edit the layout itself.

8	Screen Profile	The Screen Profile field shows an Icon view of the screens of the selected screen profile. A green screen background identifies the selected screen assigned to the ViewApp currently edited in the ViewApp Editor. The Screen Profile field also includes commands to remove the current screen profile from the selected screen or edit the screen profile itself.
9	Layout assignment to a screen	Visual representation of the selected layout applied to the selected screen in a screen profile. The names of the screen and layout appear beneath the icon image of the layout. For more information about assigning the initial layouts to the screens of a screen profile with the ViewApp Initialization wizard, see Create a ViewApp .
10	Content Pane	Content can be selected to appear in a pane by an assigned content type. If the assigned content type of a graphic, control, or other content matches the assigned content type of the pane, the content appears in the pane during runtime. For more information about content navigation, see About content types and navigation .

About ViewApp hierarchical navigation

When a ViewApp's navigation model is organized as a multi-level hierarchy of navigation items, users can make an action by mouse or touch to show content from another item or another piece of content in the navigation model.

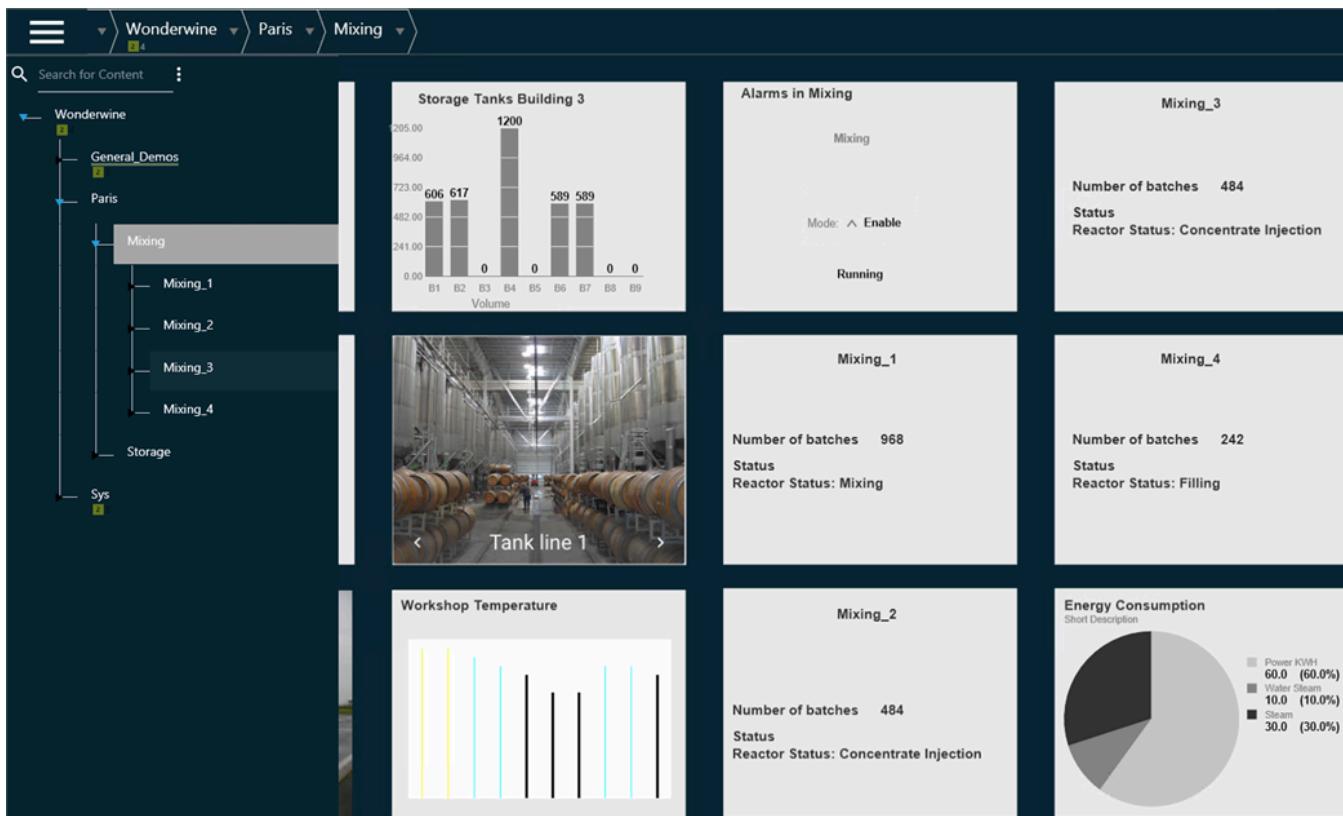
A layout can be configured with panes associated to each level in the hierarchy. Panes can be configured to support two types of hierarchical navigation behaviors:

- The user can make an action that navigates to the next item at the same level of the hierarchy and show content from that item.

- The user can make an action that navigates to the next piece of content within a multi-content pane.

Based on a pane property, tabs appear on the inner border of a pane that indicates hierarchical navigation is supported. When the user clicks or taps on a tab, the hierarchical navigation behavior is executed.

The following screen capture shows the behavior of hierarchical navigation. The current selected item is Mixing, which is a sibling item to Storage in the hierarchy. Users can show the content from these two sibling items by selecting the respective items shown in the navigation model. The Mixing navigation item is the parent of four child items.



The navigation breadcrumb shown above the navigation tree shows the selected Mixing item and its four child items with the vertical hierarchy represented as horizontal segments.



About ViewApp hierarchical navigation

To incorporate hierarchical navigation into a ViewApp, the following general configuration goals should be

considered for your navigation model..

- Design the hierarchy of your navigation model to show similar content from similar pieces of equipment . For example, if you want to show data from a set of similar pumps, place the associated navigation items at the same sibling level in the model hierarchy. This enables the user to navigate quickly from pump to pump using hierarchical navigation.
- If you want to design a pane that shows increasingly detailed information from a single navigation item, create a stack or tabbed multi-content pane configured for hierarchical navigation. Place the content in order of detail to the different tabs or layers of the pane. Users can then drill-down through the content to see more detailed information with a simple hierarchical navigation action.

Each pane of a layout can be configured for hierarchical navigation. The following pane properties must be configured within the Layout Editor to enable hierarchy navigation.

- **Enable Hierarchy Navigation**

This pane property must be set to True to enable hierarchical navigation. The **Show Navigation Buttons** pane property is hidden unless **Enable Hierarchy Navigation** property is set to True.

- **Show Navigation Buttons**

This property must be set to True to show navigation tabs on the pane at runtime. When enabled, users select the tabs by mouse or touch to navigate to the next sibling navigation item in the hierarchy or other content in the same pane.

If you want to use another visual element than the default tabs to trigger a hierarchical navigation action, set **Show Navigation Buttons** to False and create a replacement visual element that supports animation or action scripts. ViewApp Navigation attributes can be used to make hierarchical navigation actions when the user selects the visual element.. See [Use navigation attributes in action scripts](#) for a general procedure to incorporate custom hierarchical navigation in a ViewApp.

Touch swipe gestures can be used to trigger hierarchical navigation actions. Set **Show Navigation Buttons** to False to hide the default hierarchical navigation tabs when a ViewApp is intended to be run on a touch device. On touch devices, swipe indicators appear within the pane that give users visual cues about the current swipe gestures that can be made.

Trigger hierarchical navigation actions

During runtime, users can use a mouse, a keyboard, or touch gestures to trigger hierarchical navigation actions.

Mouse

When the user clicks once on a navigation Item, the item is selected and its hierarchical commands are triggered. Clicking once on a hierarchical navigation tab within the inner border of a pane triggers the pane's navigation action.

Keyboard

Users can press keyboard up, down, left, or right arrows to move to different navigation items in the hierarchy. Pressing Enter triggers the hierarchical action associated with the selected item.

Touch

Users tap a hierarchical navigation item to select it. If a pane is enabled for hierarchy navigation, users can swipe up to move to the previous sibling navigation item in the hierarchy. Swiping down moves to the next sibling item in the hierarchy. The pane shows swipe indicators to indicate the current valid swipe gestures within the pane

and a count of how many pieces of previous sibling or next sibling content exists within the navigation item.

Layout panes can be filled with content using Auto-Fill mode. In a single layout the visible pane content can originate from different navigation items. Using hierarchy navigation in any pane uses the specific navigation item for the swiping behavior that placed that original content in the pane.

When a multi-content pane has hierarchy navigation enabled, the first content placed into the pane determines the navigation item, which may place future pieces of content into the pane. This ensures the multi-content pane only holds content coming from a single navigation item. With this restriction, an up down swiping behavior is not contingent on the content which is visible to the user, reducing confusion about the final navigation item after a swipe gesture to the next or previous sibling item.

When a multi-content pane is populated with multiple pieces of content, users can utilize content navigation to move through the multiple pieces of content in the pane. Swiping left in the pane moves to the previous sibling content. Swiping right moves to the next sibling pane content.

About auto-fill navigation

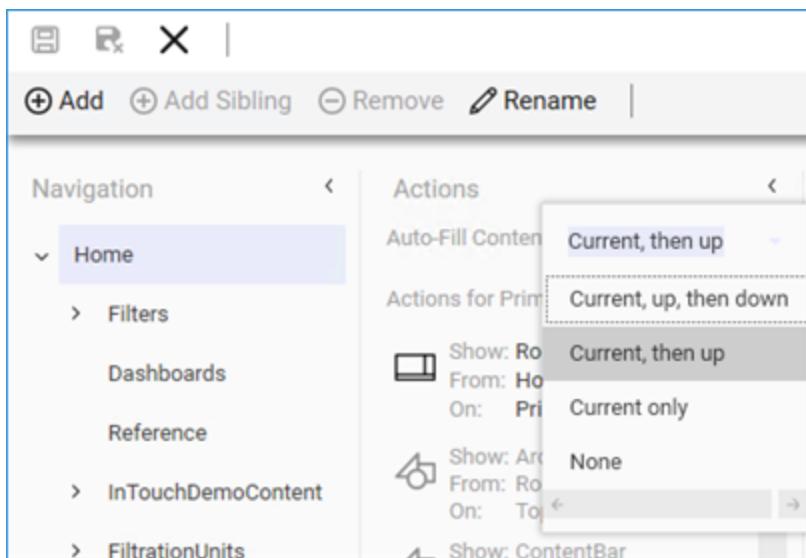
Auto-Fill is a navigation item behavior that controls how content is automatically placed in panes of a layout without manually configuring each view of a running ViewApp.

Before Auto-Fill navigation executes in a running ViewApp, appropriate layouts must be assigned to the available screens of a screen profile in a specific sequence.

- First, layouts are determined by examining the current navigation item to determine if there are any specific layouts that should be applied to screens.
- If any screens remain empty without layouts, the parent navigation item is queried to continue to search for layouts appropriate for screens.
- This process continues until the root node of the navigation hierarchy is reached, or all screens are assigned layouts.

After the appropriate layouts are determined, Auto-Fill navigation executes against the available layouts. If you link a layout to an asset and create a custom navigation item in a ViewApp with the navigation item set to the asset context, the autofill will attempt to fill the pane from the linked asset. You can override that autofill behavior by dragging and dropping content into the layout.

The ViewApp Editor contains a set of selectable Auto-Fill options to control how content is automatically placed in panes.



Note: The ViewApp's assigned layout panes must have the **Use For Auto-Fill** property enabled. For more information about layout pane properties, see [Set pane properties 1](#).

- **None**

In **None** mode, no content is automatically shown unless specific commands have been created to show content. You must assign all layouts for all screens and manually assign content to specific panes using drag and drop operations. Any commands in parent navigation nodes or child navigation nodes are not considered in assigning content to screens or panes. The **None** option is available only for custom navigation items.

- **Current only**

In **Current only** mode, you must assign layouts to screens, but any navigation item content is automatically placed in the available panes using the best available pane algorithm.

- **Current, then up**

In **Current, then up** mode, Auto-Fill searches for layouts that initially fill the screens prior to determining the content to be shown in specific panes. The current navigation item is inspected for any commands which place layouts or content full screen. For any screens that remain empty, Auto-Fill examines the parent navigation item for additional ShowLayout or ShowScreen commands to apply to empty screens.

After content has been assigned to screens, any content residing on the current navigation item is shown to the screen using the best available pane algorithm for each individual piece of content. For any remaining empty panes, Auto-Fill looks up to the parent navigation item for additional commands or content, which apply to empty panes. This process continues until all panes are filled, or the root of the navigation hierarchy is reached. If Auto-Fill intends to replace a pane's content with the same content already in the pane, the command is skipped to avoid disturbing the existing content. In **Current, then up** mode, some panes may remain empty.

- **Current, up, then down**

The **Current, up, then down** mode is identical to **Current, then up** mode with an additional action. If the Auto-Fill search reaches the root of the navigation hierarchy and there are still empty screens or panes, the first child of the current navigation item is inspected to see if it has content that applies to the available screens or panes. This process continues down the navigation hierarchy by inspecting the first child navigation item at each level until arriving at a leaf node, then Auto-Fill stops.

About content types and navigation

Content types enable you to classify content within a ViewApp. Content assigned to a pane can be shown or hidden during runtime based on whether the assigned graphic content type matches the content type of the pane or not. This mutual association between a pane and a graphic content type is restrictive in that it limits the content that can appear in a pane to a specific subset content types. With some exceptions, a pane can only host defined content types from the available graphics that match the content type.

There are exceptions to content type matching:

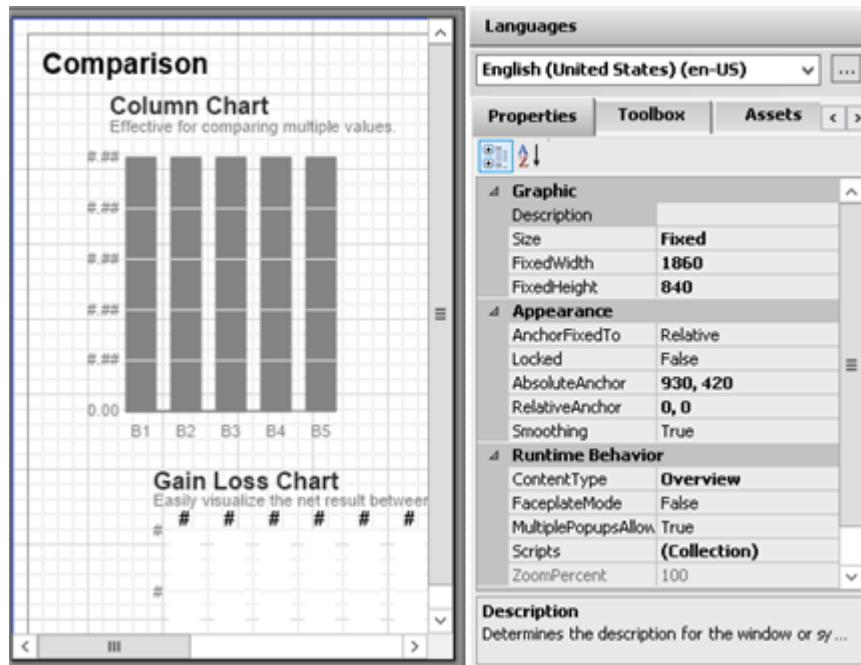
- Panes can be assigned the **Select All** content type, which matches any type of graphic content. The pane can show any type of graphic content.
- Graphics can be assigned the **All** content type, which matches any type of pane content. The graphic can be shown in any type of pane.
- Graphics and panes can be assigned multiple content types.

The selected content types appear in a comma-delimited list in both the Industrial Graphic Editor and Layout Editor.

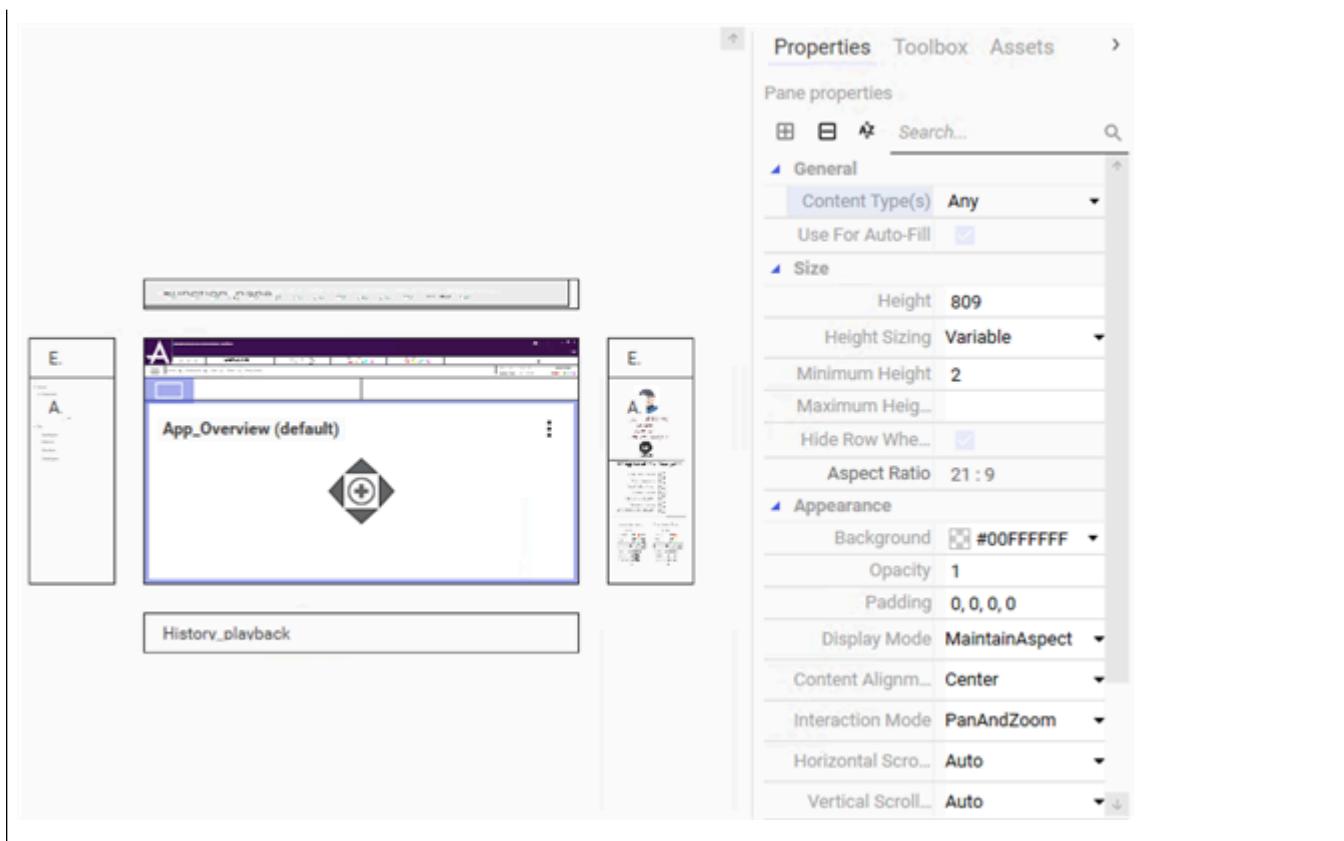
Assign a content type

A content type can be assigned to a graphic by assigning a value to the **ContentType** property in the Industrial Graphic Editor. Similar content types can be assigned as a value of the **Content Types** pane property of the Layout Editor. Content is shown in the appropriate pane if the assigned value of the **ContentType** property matches the content type of the pane or a direct reference to the pane.

Graphics Editor ContentTypes Property



Layout Editor Content Types Property



Action sequencing when content or other types are applied in ViewApp

Content and other types of ViewApp navigation follow a distinct sequence of steps to populate the available panes of a layout shown on a screen during runtime. When the user selects a navigation item and triggers its navigation actions, the following steps occur:

1. All screens within the ViewApp's selected screen profile are considered empty without content.
2. If the current navigation item specifies a layout for use on a screen, layouts are placed on the screens of the profile using inheritance. If any screens remain empty, the current navigation item's parent item is queried to determine if a layout is specified for any empty screen.
3. All panes belonging to the layouts assigned to screens are assumed to be empty before the composition engine assigns any content to the panes.
4. Content is shown in the appropriate pane based on content type matching between content and the pane or a direct reference to the pane.

Layout panes containing pre-existing content are not considered in content type matching. But, a direct reference to a navigation action replaces pre-existing pane content.

5. Other panes without any assigned content are enumerated to retrieve their individual content types.
6. For each type of pane, the current level in the hierarchy is queried to see:
 - a. If any explicit actions exist that specify a pane in the destination screen
 - b. If any implicit actions exist based on the type of content.

Implicit navigation actions are triggered by an available pane that matches the content type of the content. A multi-content pane can be the destination of multiple implicit action calls while executing an

Auto-Fill navigation action.

7. If any actions are found at that level of the navigation hierarchy, they are shown in the panes.
8. For any remaining empty panes, the current navigation item's parent is retrieved and the search starts again at step 3.
9. The search ends when content is found for all of the panes or the root of the navigation hierarchy is reached.

If there are multiple panes with the same content type, the pane chosen to show the content is selected based on the descending alphabetical order of the pane names.

About ViewApp custom navigation

The ViewApp Editor includes a set of custom navigation commands to create a navigation model for a ViewApp based on custom navigation items. Each custom navigation item can be assigned one or more pieces of content that appear in a specific pane during runtime.

Custom navigation gives users the ability to view specific types of content associated with navigation items that are not required to replicate the hierarchy of objects shown in the Model view of the Template folder. During runtime, the ViewApp's layout panes auto fill with the associated content when the user selects the custom navigation item.

Add custom navigation to a ViewApp

This topic describes the general steps to incorporate custom navigation into a ViewApp using a simple example of replacing the entire default navigation model with a set of custom navigation items.

The ViewApp Editor includes custom navigation commands shown on the menu bar.

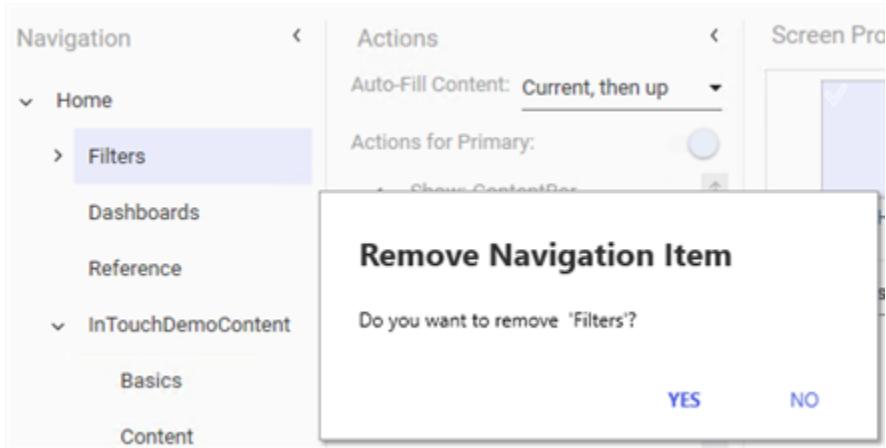
 + Add + Add Sibling - Remove ⚙ Rename

Add	Adds a custom navigation item as a child of the current selected item in the navigation model.
Add Sibling	Adds a custom navigation item as a sibling of the current selected item in the navigation model. Sibling items are at the same level within a navigation hierarchy.
Remove	Removes the selected navigation item and all child items beneath it in the navigation hierarchy.
Rename	Select to rename the current selected item. A blue background behind the current item name indicates the item can be renamed by typing over the current name.

The following procedure describes how to use these commands to create custom navigation items and associate content to them. See [Configure properties for custom navigation items](#) for a description of the steps to configure the properties of a custom navigation item.

To add custom navigation to a ViewApp

1. Create a ViewApp that includes a layout with a NavTree control that shows the hierarchy of default navigation items.
2. Select the immediate navigation item beneath the root item shown in the **Navigation** area of the ViewApp Editor.
3. Select **Remove** from the menu bar of the ViewApp Editor.
4. Select **Yes** to confirm that you want to remove navigation items.



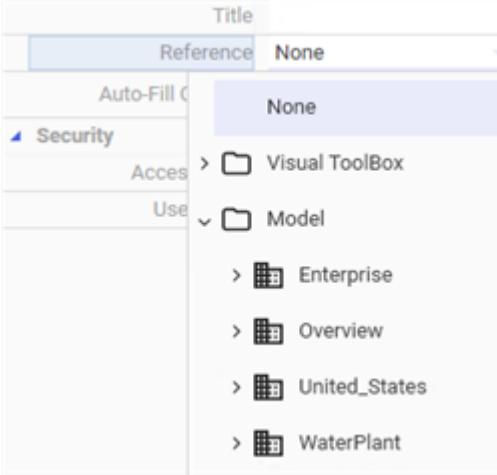
5. Click **Add** from the menu bar to add a custom navigation item.
The navigation item appears beneath the root as a child item. Blue text indicates the navigation item can be renamed.
6. Rename the item you added.
7. Select the custom navigation item you added.
8. Select the **Toolbox** or **Assets** tab to show their respective lists of graphics or objects.
9. Select a graphic or object that you want to associate with the custom navigation item and drag it to the pane you want to show this content when the custom navigation item is selected during runtime.
The **Actions** list updates to show the content you associated with the custom navigation item.
10. Repeat these steps to build a complete navigation model with custom navigation items.

For information about configuring pane properties containing custom navigation content, see [Configure properties for custom navigation items](#).

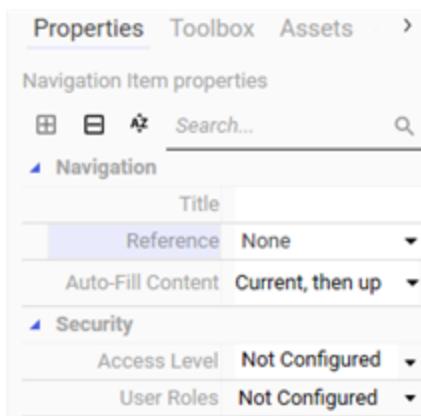
Configure properties for custom navigation items

After associating a piece of content to a custom navigation item, its properties need to be configured.

Custom Navigation Properties	Description
Title	Name assigned to the custom navigation item.
Reference	Reference node in the asset hierarchy that is the anchor point of a custom navigation item. Click at the right of the Reference field to show a drop-down list

	<p>of assets defined for the ViewApp's navigation model.</p>  <p>Select an asset from the list. Any content associated with the asset you selected appears in the Actions list.</p>
Auto-Fill Mode	The Auto-Fill mode assigned to the panes containing content assigned to the custom navigation item. Click at the right of the Auto-Fill Mode field to show a drop-down list of Auto-Fill modes. Select a mode from the list.
Access Level	<p>Security access level assigned to content associated with the custom navigation item. Click at the right of the Access Level field to show a drop-down list of Access Levels for the different security roles defined in the Galaxy.</p> <p>Not Configured) 0 (Default) 9999 (Administrator)</p> <p>To view the content assigned to a custom navigation item, the runtime user role must be assigned an Access Level that meets or exceeds the the Access Level assigned to the content.</p>
User Roles	User roles allowed to access content associated with the custom navigation item. Click at the right of the User Roles field to show a drop-down list of the roles defined in the Galaxy.

The following figure shows the default property values assigned to a custom navigation item.



To edit custom navigation item properties

1. Open the ViewApp with the ViewApp Editor.
2. Select the pane containing custom navigation content.
3. Select the custom navigation item added to the the **Navigation** hierarchy.
4. Select the **Properties** grid of the ViewApp Editor.
You should see the navigation item's properties
5. Set values to the properties.
6. Click **Save**.

ViewApp custom navigation rules

The following list describes some restrictions and behaviors when building a custom navigation model for a ViewApp.

- Add a new navigation item

You cannot add a custom navigation item at the same level of the navigation hierarchy that contains a sibling navigation item with the same name.

- Configure a reference to an **Assets** or **Toolbox** folder

When you configure a reference to an **Assets** or **Toolbox** folder, the name of the navigation item does not change and can be renamed if the new name does not match an existing sibling navigation item.

You cannot add a new custom navigation item beneath a navigation item, which references an **Assets** or **Toolbox** folder. The **Add** menu option should be disabled.

- Asset hierarchy root node behavior

You can select the root node of the asset hierarchy, as viewed from the **Template folder** of the System Platform IDE, for a reference to include asset nodes. The user can also select any child node in the asset hierarchy for the reference.

- Valid content in the **Visualization folder** hierarchy

You can select any folder in the **Visualization folder** hierarchy, including the root of the tree.

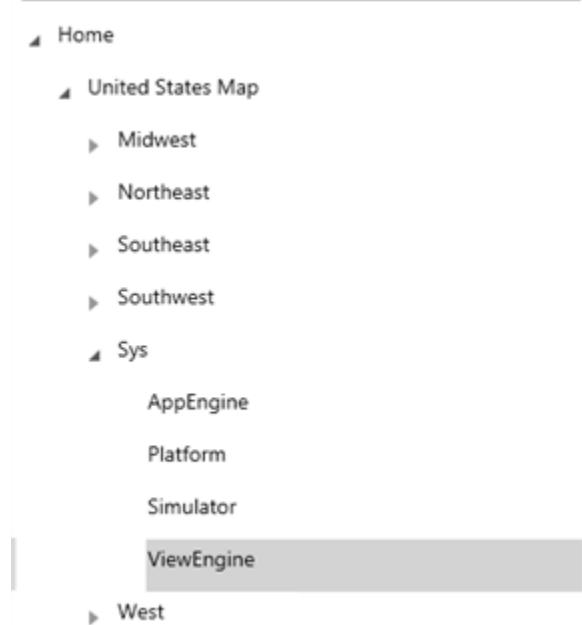
- Removing a Navigation Item

You can remove a navigation item from the hierarchy. All children items are removed if their parent is removed. A confirmation dialog box appears and requests confirmation that you want to remove the

navigation item.

Hide assets in ViewApp navigation during runtime

System Platform includes the ability to hide assets in ViewApp navigation during runtime. The ability to hide assets from a ViewApp's navigation model is beneficial in cases where the assets themselves provide little value to an operator during runtime. An example is the case of child asset navigation items that are purely system assets in the Sys [System] folder, which appear in the navigation model of a ViewApp during runtime.



None of the system assets that appear in the Sys folder provide any meaningful content in a running ViewApp. The system assets within the Sys folder need to be hidden and not shown in a ViewApp's navigation model during runtime.



Hide assets from a ViewApp navigation model

The requirements to hide an asset from a ViewApp navigation model are listed below.

- The asset must be undeployed from the **Model** view to rename its alias name.

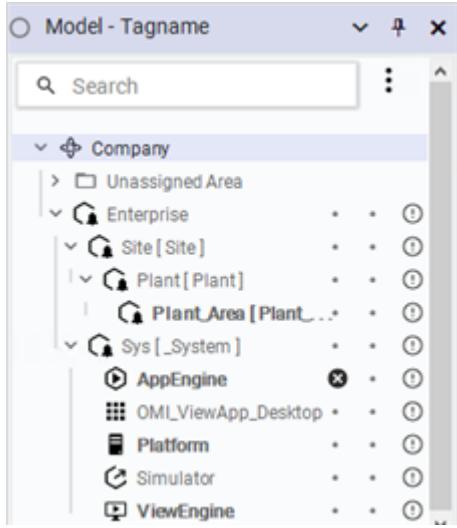
- The asset to be hidden must be a child navigation item in the ViewApp navigation model.
- You must change the alias name of the asset to start with an underscore.

To Hide Assets from a ViewApp Navigation Model

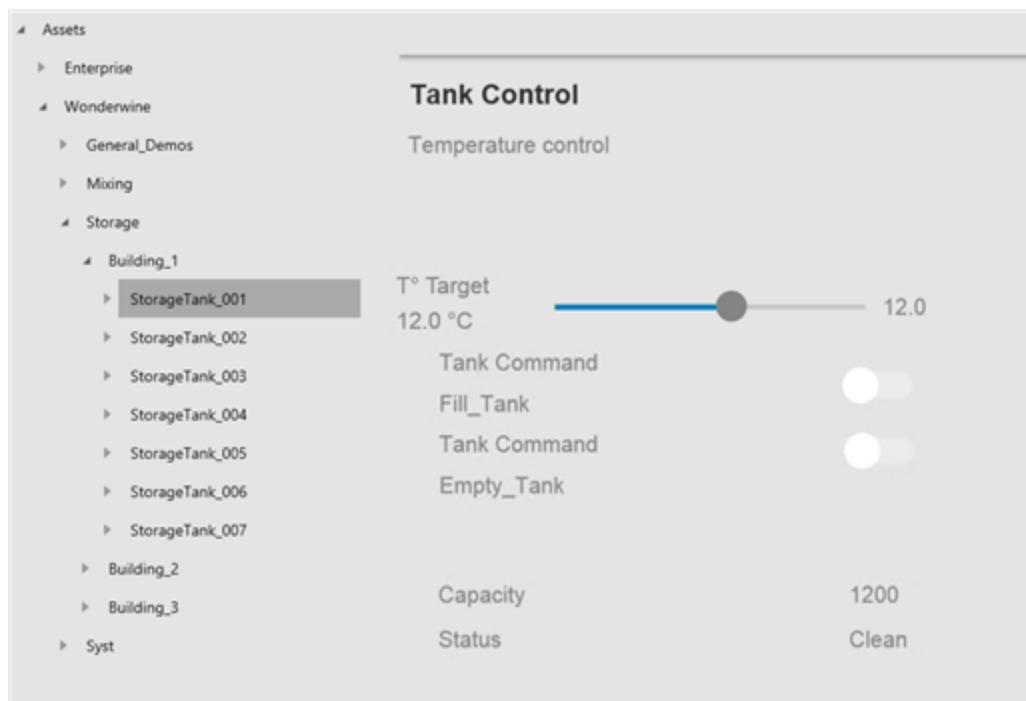
1. Open the System Platform IDE and show the list of assets in the **Model** view at the lower left of the IDE.
 2. If necessary, undeploy the child asset to be hidden.
 3. Right-click the asset to show the shortcut menu.
 4. Select the **Rename Alias** option from the menu.
 5. Type an underscore character (_) as the first character of the asset's alias name and click **OK**.
 6. Deploy the ViewApp.
 7. Run the ViewApp to verify the renamed asset no longer appears in the navigation model.
- Also, the hidden asset does not appear when you run a ViewApp in preview mode.

About ViewApp navigation hierarchical display

AVEVA OMI includes a NavTree app that shows a ViewApp's navigation items arranged in a hierarchical tree. During configuration time in the ViewApp Editor, the NavTree app shows this navigation item hierarchy based on the Model view of object instances. This view represents the navigation model of the ViewApp being edited.

Model View	NavTree App in ViewApp Editor
	<ul style="list-style-type: none"> Assets <ul style="list-style-type: none"> Enterprise Site Plant Plant_Area Sys <ul style="list-style-type: none"> AppEngine Platform Simulator ViewEngine

During runtime, users can select the chevron icon next to a navigation item to expand or collapse the tree of navigation items. When an item is selected from the tree, the item's navigation commands are executed and associated content appears in layout panes that meet the command requirements.



About display of controls in the ViewApp navigation hierarchy

Aside from the standard **Content** and **Pane** properties of all objects and graphics, the NavTree app includes sets of **Behavior**, **Alarms**, **Appearance**, and **Search** properties:

- Enable the **Show Root** property to show the root navigation item of the navigation model during runtime.
- Enable the **Show Alarms** property to show any active local and aggregated alarms on the individual items within the navigation model.
- Enable the **Show Search Bar** property to show a search field above the runtime Navigation tree to enter a search string. For more information about searching a ViewApp's navigation model during runtime, see *Search View App Content During Runtime* in the AVEVA OMI help.
- The **Appearance** properties determine the color, size, and font family of text shown on the navigation tree. Each property contains a drop-down list of binding options. Some properties only provide a **Reset** option to reset a property value to its default.

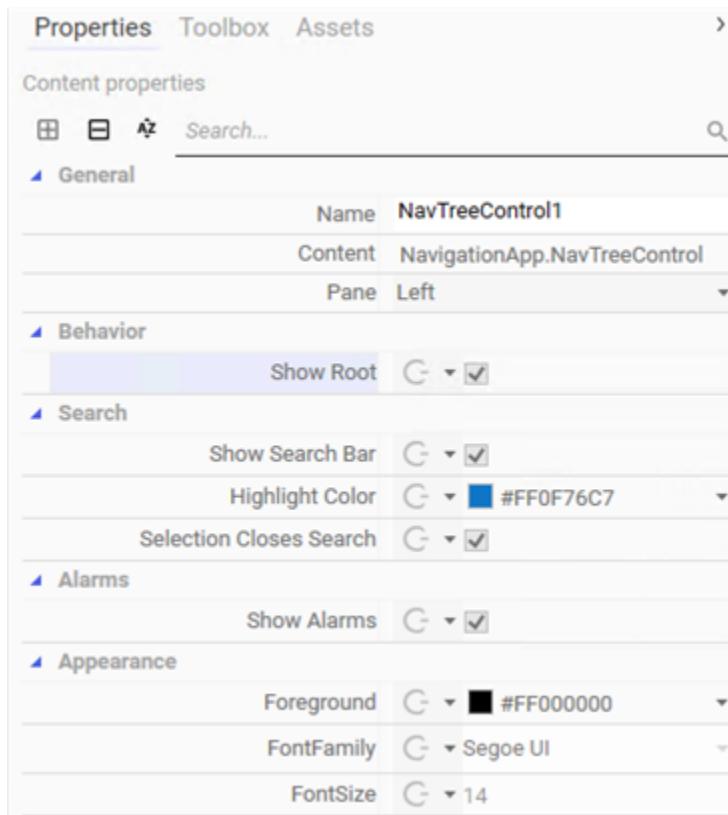
In addition to a static binding, dynamic binding values can also be configured.



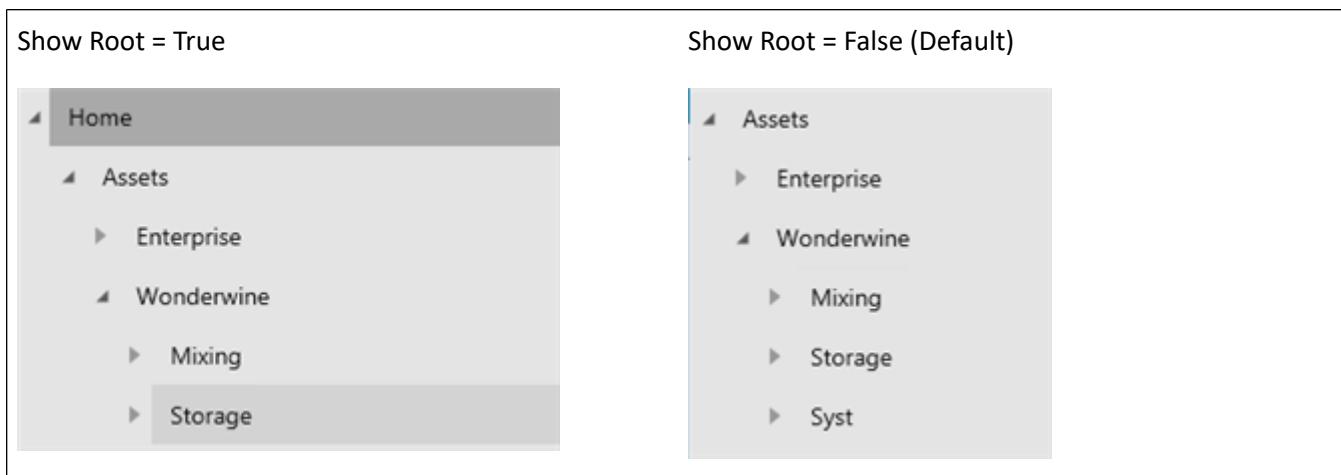
The type of control property /attribute binding can be specified by selecting a value from the drop-down list.

Constant	Static binding to the specified path. Enter a value in
-----------------	--

	the data entry field.
In	Dynamic root path with read only binding by the control to a reference.
Out	Dynamic root path with write only binding by the control to a reference.
InOut	Dynamic root path with read/write binding by the control to a reference.
Reset	Reset the current property to its default value.



When **Show Root** is selected, the root navigation item appears at the top of the NavigationTree's hierarchical list of navigation items. The root navigation item is the default, selected startup item when a ViewApp starts.



The root navigation item is a custom navigation item, which supports the following common behaviors of all navigation items:

- The default name of the root navigation item is Home.
- The root navigation item can be renamed like any navigation item.
- The root navigation item can be a reference to an asset or a graphic item.
- The root navigation item can hold custom actions defining layouts and content in layouts.

The following behaviors are specific to the root navigation item:

- The root navigation item cannot be deleted.
- Sibling items to the root cannot be created.

When **Show Alarms** is selected (it is selected by default), alarm counts from the **AlarmCntsBySeverity** attribute are displayed on any NavigationTree item with an active alarm. If the alarm has been aggregated from a contained object, alarms are shown at 50% opacity (transparency) on the containing object. If alarm is on the object, it is shown as a solid color. The alarm color and graphic reflect the severity of the alarm. Graphics and animations for the alarm counts are taken from the Galaxy style library.

Set values for the color and font properties. Select the color property and then select the small triangle at the right of both property fields to show the color picker.

If you want to create event handler scripts, select the **Available Events** field to show a list of NavTreeControl events.

About predefined navigation attributes

AVEVA OMI includes a set of predefined Navigation attributes. Read-only navigation attributes identify neighboring navigation items of the selected item by name or fully qualified path location. Read-write attributes can be used in action scripts and animations to define actions that can change the current navigation item in a running ViewApp. When the context of the ViewApp shifts from the current navigation item to a destination navigation item, the associated content of the destination item appears in the ViewApp.

A typical runtime navigation workflow based on Navigation attributes is for the user to select a graphic element shown in a ViewApp that includes an action script or animation containing Navigation attributes. The script or animation creates a navigation action based on the attributes to shift the view context to another navigation

item in the ViewApp's navigation model.

Predefined navigation attribute rules

Navigation attributes are available only for use in AVEVA OMI ViewApps. Navigation attributes cannot be used in an InTouch application because the MyViewApp.Navigation prefix is regarded as a configuration error.

All Navigation attributes operate in the MyViewApp.Navigation namespace. Navigation attributes are specified with the MyViewApp.Navigation prefix in the form MyViewApp.Navigation.*attribute_name*.

Notes:

- Attribute names with *Path* return the fully qualified path of a navigation item within the navigation model.
- Attributes with the word *Title* in their names return the user friendly name set as the value of the **Title** property of a navigation item. When a navigation item's **Title** property is not configured, then the name of the navigation item is used instead.

Attribute Name	Data Type	Read/Write	Initial Value	Description
Current	String	Read/Write	None	Returns the name of the current internal name of a navigation item. Should always be assigned a value.
CurrentTitle	String	Read Only	None	CurrentTitle returns the name specified in the Title property of the current navigation item. If the navigation item's Title property is empty, CurrentTitle returns the current value.
CurrentPath	String	Read/Write	None	CurrentPath can be assigned the fully qualified path of a navigation item. When a value is set to CurrentPath , it is validated to see if the navigation item with the fully qualified path exists in the navigation model. If the

Attribute Name	Data Type	Read/Write	Initial Value	Description
				<p>navigation item exists, then the value is accepted. Otherwise, the previous value of CurrentPath is retained.</p> <p>A successful path assignment moves the navigation item to a new value and triggers Auto-Fill behaviors. Other Navigation attributes like Parent, NextSibling, PrevSibling, Current, CurrentAsset are updated.</p>
CurrentAsset	String	Read/Write	None	<p>CurrentAsset can be assigned the name of an asset if the current navigation item is an asset.</p> <p>When a value is set to CurrentAsset, it is validated to see if an asset with the given name exists in the navigation model. If an asset with given name exists, then the value is accepted.</p> <p>Otherwise, the previous value of CurrentAsset is retained.</p> <p>A successful asset assignment moves the navigation item to a new value and triggers Auto-Fill behaviors. Other attributes like</p>

Attribute Name	Data Type	Read/Write	Initial Value	Description
				Parent, NextSibling, PrevSibling, Current, CurrentAsset are updated.
CurrentAsset Title	String	Read Only	None	CurrentAssetTitle returns the user friendly name of the current asset. CurrentAssetTitle returns CurrentAsset if empty.
CurrentAsset Path	String	Read Only	None	CurrentAssetPath returns the fully qualified name of the current asset path. CurrentAssetPath returns empty if the current asset is empty.
CurrentArea	String	Read/Write	None	CurrentArea can be assigned the name of the current asset's area if the current asset is valid. When CurrentArea is set to a value, the incoming value is validated to see if an area with the given name exists in the navigation model. If an area with given name exists, then the CurrentArea value is accepted. Otherwise the previous value of the CurrentArea is retained. A successful area

Attribute Name	Data Type	Read/Write	Initial Value	Description
				assignment moves the navigation item to a new value and triggers Auto-Fill behaviors. Other Navigation attributes like Parent, NextSibling, PrevSibling, Current, CurrentAsset are updated.
CurrentArea Title	String	Read Only	Non	CurrentAreaTitle returns the user friendly name of the current asset's area. CurrentAreaTitle returns "CurrentArea" if empty.
CurrentArea Path	String	Read Only	None	CurrentAreaPath returns the fully qualified path of the current area. CurrentAreaPath returns empty if the current asset is empty.
CurrentRoot Area	String	Read Only	None	Returns the name of the current asset's Root area if the CurrentAsset is a valid asset. This is the area at the root of the Model Hierarchy (not the navigation hierarchy) which contains the CurrentAsset. This value is populated even if the Root Area node is not contained in the Navigation model.

Attribute Name	Data Type	Read/Write	Initial Value	Description
CurrentRootAreaTitle	String	Read Only	None	Returns the user friendly name of the current asset's Root area. If the root area is not part of the navigation model, this value will be empty.
CurrentRootAreaPath	String	Read Only	None	Returns the fully qualified name of the current root area item using the ".Title" strings to form the path. Returns empty if the root area is not part of the Navigation model.
Root	String	Read Only	None	Root returns the internal name of the root item in the current navigation hierarchy.
RootTitle	String	Read Only	None	RootTitle returns the user friendly name of the root item in the current navigation hierarchy. If the root item's Title property is empty, RootTitle returns Root as its assigned value.
RootPath	String	Read Only	None	RootPath returns the fully qualified path of the root item using the Title strings to form the path.
Parent	String	Read Only	None	Parent returns the internal name of the parent navigation

Attribute Name	Data Type	Read/Write	Initial Value	Description
				item in the current navigation hierarchy. If there is no parent item, Parent returns empty.
ParentTitle	String	Read Only	None	ParentTitle returns the user friendly name of the parent item in the current navigation hierarchy. If the parent item's Title property is empty, ParentTitle returns the Root value.
ParentPath	String	Read Only	None	ParentPath returns the fully qualified path of the parent item in the current navigation hierarchy. If there is no parent, ParentPath returns empty.
PrevSibling	String	Read Only	None	PrevSibling returns the internal name of the previous sibling item in the current navigation hierarchy. If there is no previous sibling, no value is associated with PrevSibling .
PrevSibling Title	String	Read Only	None	PrevSiblingTitle returns the friendly name of the previous sibling item in the current navigation hierarchy. If the previous sibling's Title property is empty, PrevSiblingTitle retains the current

Attribute Name	Data Type	Read/Write	Initial Value	Description
				value.
PrevSibling Path	String	Read Only	None	PrevSiblingPath returns the fully qualified path of the previous sibling navigation item. No value is set to PrevSiblingPath if there is no previous sibling.
PrevSibling Count	Integer	Read Only	None	Returns the number of previous sibling items remaining in the current navigation hierarchy.
NextSibling	String	Read Only	None	NextSibling returns the internal name of the next sibling item in the current navigation hierarchy. No value is set to NextSibling if there is no next sibling item.
NextSibling Title	String	Read Only	None	NextSiblingTitle returns the friendly name of the next sibling item in the current navigation hierarchy. If the next sibling's Title property is empty, NextSiblingTitle returns NextSibling as its assigned value.
NextSibling Path	String	Read Only	None	NextSiblingPath returns the fully qualified path of the next sibling navigation item. If there is no next

Attribute Name	Data Type	Read/Write	Initial Value	Description
				sibling in the path, NextSiblingPath returns empty.
NextSibling Count	Integer	Read Only	None	Returns the number of next sibling items remaining in the current navigation hierarchy.
FirstChild	String	Read Only	None	Returns the internal name of the first child item in the current navigation hierarchy. If there is no first child, returns empty.
ChildCount	Integer	Read Only	None	Returns the number of child items in the current navigation hierarchy.
FirstChildPath	String	Read Only	None	Returns the fully qualified path of the first child, using the "Title" strings to form the path, for example, "\<rootname>\<node name>". If there is no first child, returns empty.
FirstChildTitle	String	Read Only	None	Returns the friendly name of the first child of the current navigation item. If there is no first child the string is empty.

Use navigation attributes in action scripts

This example explains how to use Navigation attributes in action scripts associated with a graphic to enable users to shift the context of a ViewApp to sibling navigation items, or return to the root level item in the navigation tree.

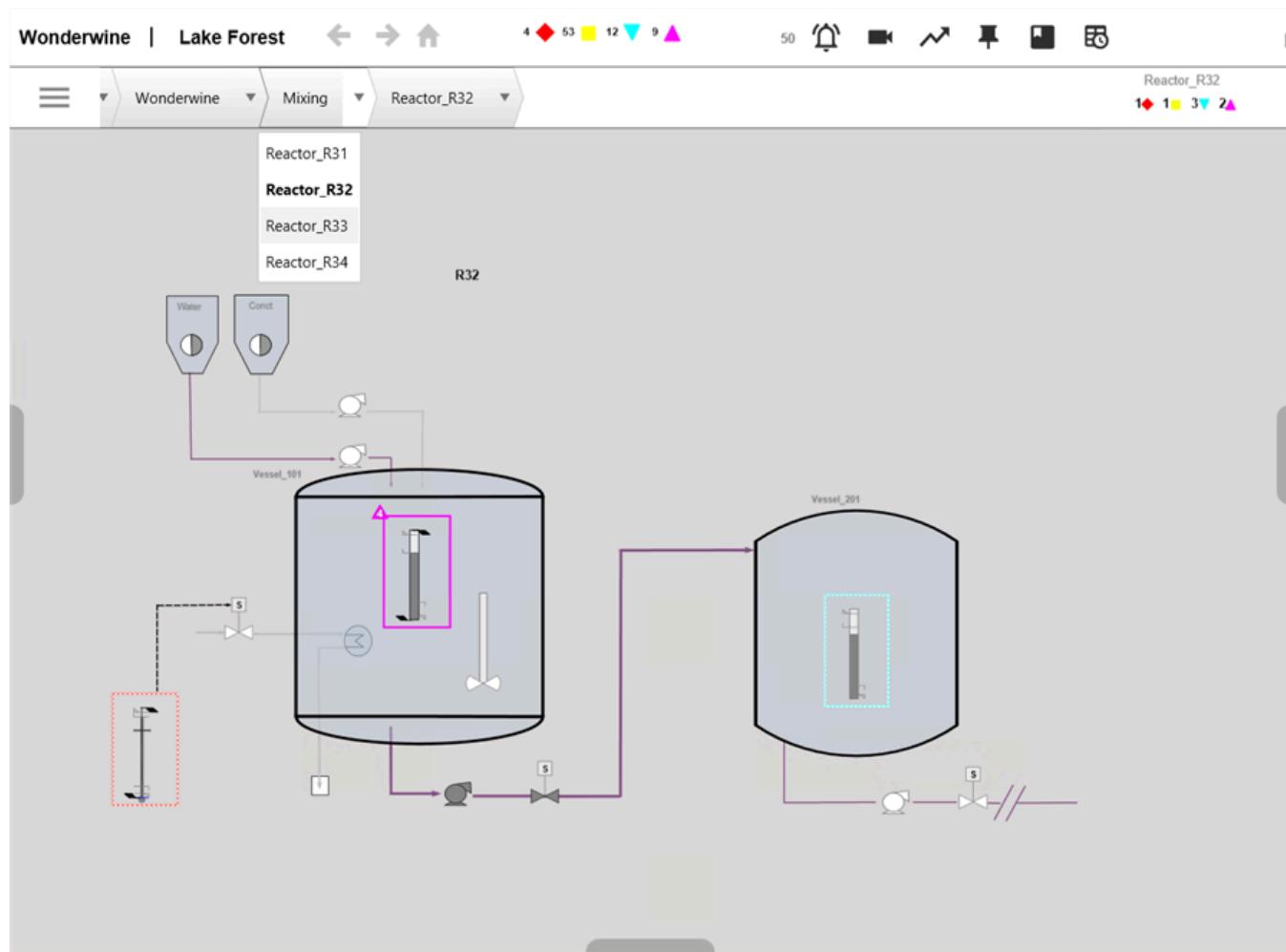
An example ViewApp is designed to show the mixing and storage areas of a plant, which contain reactors and tanks. Users need the capability to quickly drill down within the ViewApp to see the details of individual reactors and tanks.

In the Model view of the IDE, the Mixing and Storage areas are sibling items in the navigation hierarchy. The Mixing area contains four reactors, which are siblings of each other. The Storage area contains three sibling buildings. Each building contains a set of storage tanks, which are siblings of each other. During runtime, the ViewApp's NavTree control shows the hierarchical relationships of the different items in the navigation model.

- ▲ Wonderwine
 - ▲ Mixing
 - ▶ Reactor_R31
 - ▶ Reactor_R32
 - ▶ Reactor_R33
 - ▶ Reactor_R34
 - ▲ Storage
 - ▲ Building_1
 - ▶ StorageTank_001
 - ▶ StorageTank_002
 - ▶ StorageTank_003
 - ▶ StorageTank_004
 - ▶ StorageTank_005
 - ▶ StorageTank_006
 - ▶ StorageTank_007
 - ▶ Building_2
 - ▶ Building_3

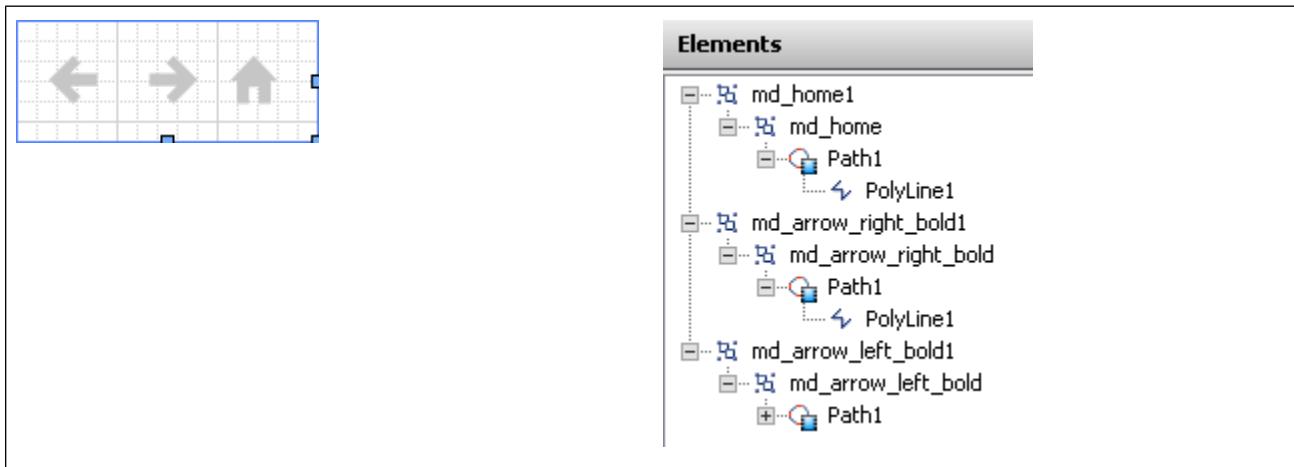
Users need the ability to shift quickly between sibling views. Also, users need to be able to immediately navigate to the root item in the navigation tree that shows an overview of the entire plant operation.

The graphic near the top of the ViewApp containing left/right arrows and a home icon is the visual element the user interacts with to navigate between sibling views and show the plant overview. The left arrow shows the previous item in a set of sibling items. The right arrow shows the next item. The home icon shows the plant overview.

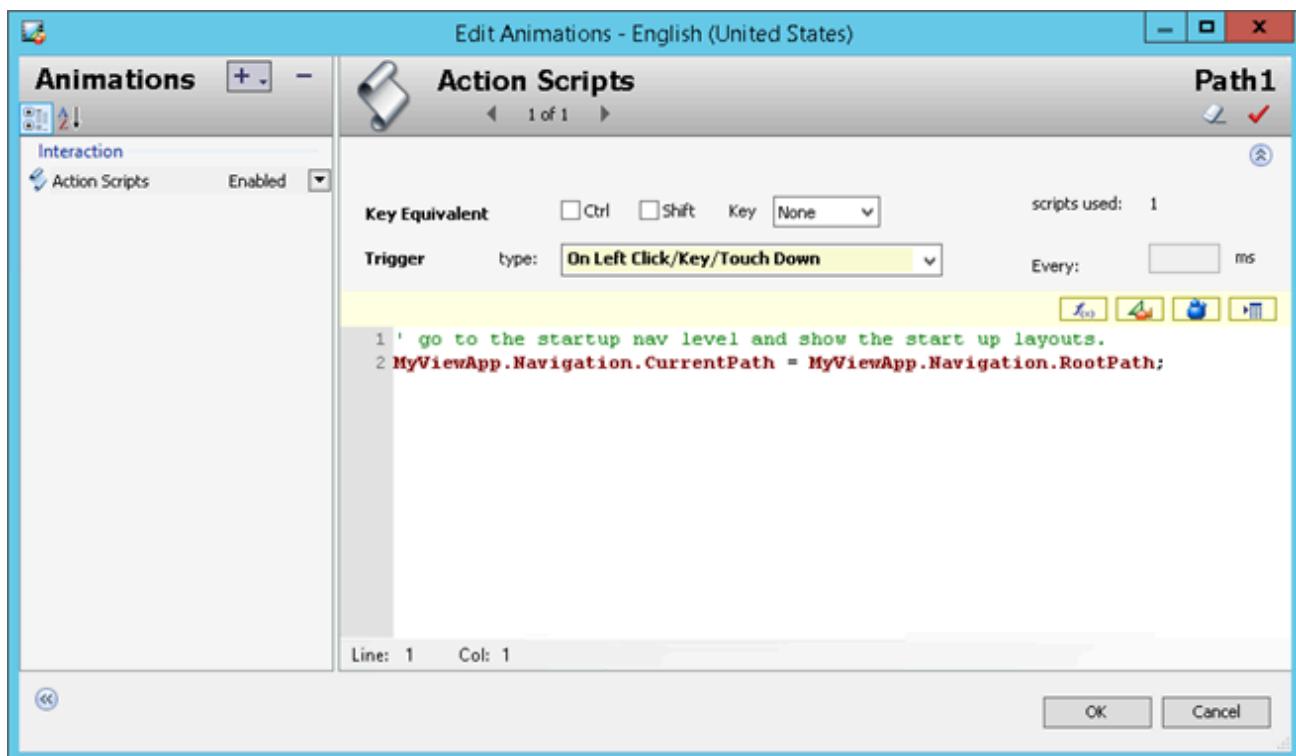


To set navigation pathways with Navigation attributes

1. Create a graphic containing left/right arrows and a home icon graphic element.



2. Select the Path element within the home graphic element to show the **Edit Animations** dialog box.
3. Select **Action Scripts** as the type of animation.
4. Enter a Navigation attribute expression to set the current navigation path to the root path.



- Repeat steps 2-4 to set actions scripts for the left/right arrow graphic elements.

The image contains two side-by-side screenshots of the 'Action Scripts' dialog. The left one is for a 'Right arrow' graphic element, and the right one is for a 'Left arrow' graphic element. Both dialogs show a single script with a trigger 'On Left Click/Key/Touch Down'. The 'Right arrow' script code is:

```

1 ' go to the next sibling nav level
2 MyViewApp.Navigation.CurrentPath = MyViewApp.Navigation.NextSiblingPath;

```

The 'Left arrow' script code is:

```

1 ' go to the previous sibling nav level
2 MyViewApp.Navigation.CurrentPath = MyViewApp.Navigation.PrevSiblingPath;

```

- Save your changes to the graphic.

About client controls embedded into a graphic

You can:

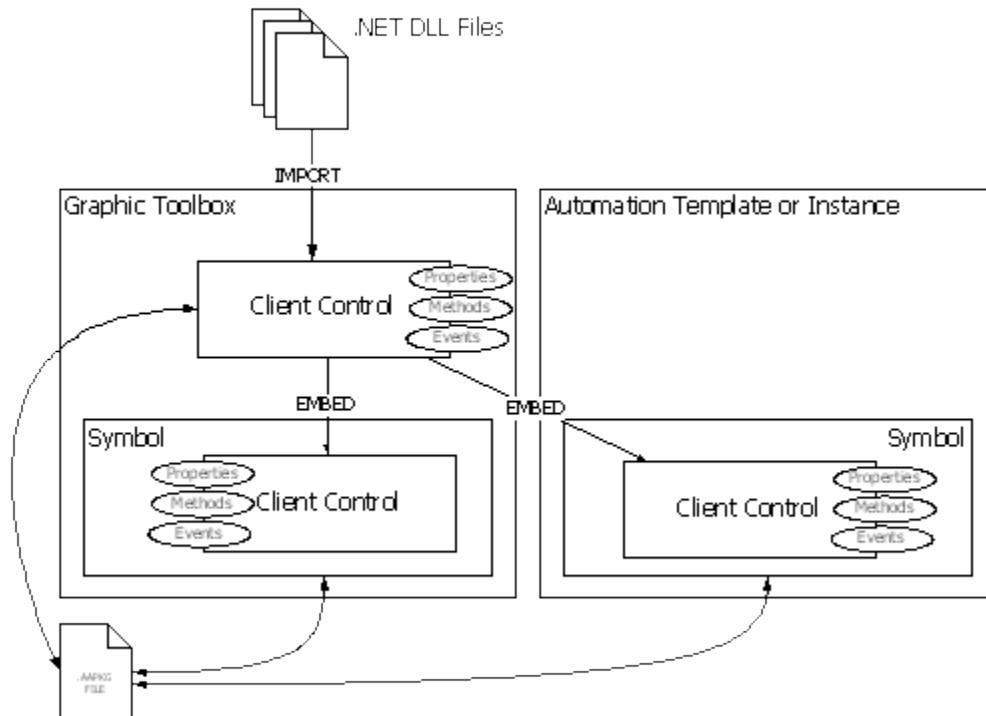
- Import and embed client controls into a graphic.
- View and edit the properties of the client control.
- Bind the properties of the client control with attributes and element references.
- Configure scripts for client control.

- Animate client controls.
- Export a client control.
- Configure a client control with security.
- Ensure that dynamically loaded assemblies are included with the primary client control assembly when an application is deployed
- View additional client control information such as the files the client control uses and what objects and graphics are using the client control.

Client controls give you functionality contained in .NET controls you can use in graphics. To use this functionality, you must:

- Import the .DLL file that contains one or more client controls. The client control is imported into the Visualization folder.
- Browse and embed one or more of the client controls into a new or existing graphic. The client controls appear as elements.
- View and edit the exposed client control properties.
- Bind the client control properties to attributes, graphic custom properties or InTouch tags. Do this using data binding animation.
- Configure scripts for client control. Do this using the animation.

You can then use the graphic containing the embedded client control in a ViewApp.



Import client controls: options

You can import client controls into the Industrial Graphic Editor from .NET Dynamic Link Library (.DLL) files. After importing client controls, you can organize them in the Industrial Graphic Editor as you would with

Industrial Graphics. For more information, see [Organize client controls](#).

You can also import client controls that have previously been exported in an Archestra package (.aaPKG) file.

If you import a newer version of a client control that you are already using in your HMI software as embedded Industrial graphic, you need to restart the HMI software.

Import client controls

You can import one or more client controls from .DLL files. The client controls can be contained in one single .DLL file or span multiple files.

To import a client control, you must have security permissions to import graphic objects.

Note: If you select .NET .DLL files that do not contain client controls, the import process ignores these and continues at the next .DLL file.

To import client controls

1. On the IDE **Home** ribbon, select **Galaxy**, then select Import.
2. Select **Visualization**, then **Client Controls**. The **Import Client Control(s)** dialog box appears.
3. Select one or more .NET .DLL files that contain the client controls you want to import and click **Open**. The **Import Preferences** dialog box appears.
4. Select the appropriate option for the import and click **Import**. The **Import Client Control(s)** dialog box appears.
5. When the client controls are imported, click **Close**. The imported client controls appear in the Visualization folder.

Note: If the import fails, a message in the **Import Client Control(s)** dialog box indicates the error.

Example of installing the TagPicker control

To install the TagPicker client control

1. On the IDE **Home** ribbon, select **Galaxy**, then select Import.
2. Select **Visualization**, then **Client Controls**. The **Import Client Control(s)** dialog box appears.
3. Browse to the C:\Program Files (x86)\Archestra\Framework\Bin\ViewAppFramework\SharedLibs folder, select the aaHistClientTagPicker.dll file and click **Open**. The **Import Preferences** dialog box appears.
4. Select the appropriate option for the import and click **Import**.
5. When the import is complete, click **Close**.
6. Open the **Visualization folder** and expand the Galaxy node. aaTagPicker is listed as a client control.

Import previously exported client controls

You can import one or more previously exported client controls from an Archestra package file (.aaPKG). Previously the client controls may have been:

- Exported without a graphic or an object instance or template.

- Embedded in a graphic and the graphic was exported.
- Embedded in a graphic and contained in an object instance or template and the object was exported.

To import a previously exported ArchestrA package containing one or more client controls

- Import the exported client controls the same way as you would import an AutomationObject (.aaPKG). For more information, see the *Application Server User's Guide*.

Organize client controls

You can organize the client controls within the Visualization folder the same way as you would with Industrial Graphics. You can:

- Rename client controls.
- Move client controls in and out of folders.
- Delete client controls.

For more information, see [Organize graphics](#).

View and change the properties of client controls

When you embed a client control into a graphic, the native properties of the client control are imported into the Properties Editor in the **Misc** group.

Also the element container of the client control has properties such as:

- Name
- X, Y, Width, Height, AbsoluteOrigin, RelativeOrigin, and Locked
- FillColor
- TextColor and Font
- Enabled, TabOrder, TabStop, and Visible

The element container properties override the native properties of the client control.

You can view and change the properties of the control in the Properties Editor.

To view or change the properties of a client control

1. Select the embedded client control on the canvas.
2. In the Properties Editor, locate a:
 - Container property in the property categories **Graphic**, **Appearance**, **Fill Style**, **Text Style** or **Runtime Behavior**.
 - Native property in the **Misc** property category.
3. View or change the located property.

To reset a client control back to its original size

- On the **Edit** menu, click **Control - Original Size**. The AutoSize property is set to False.

Example of changing a property of the TagPicker control

Install and embed the TagPicker Control into a graphic as described in:

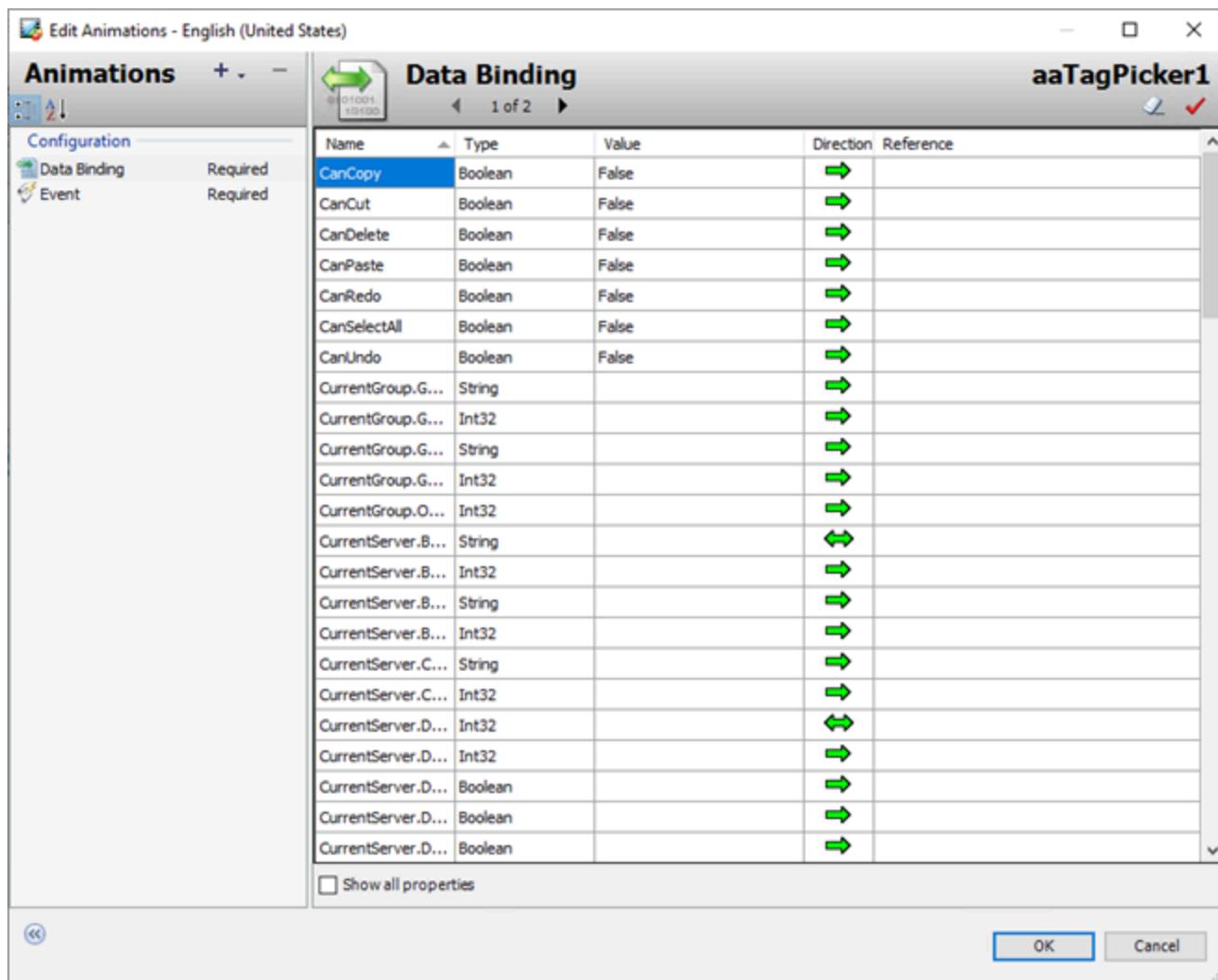
- [Example of installing the TagPicker control](#).
- [Embed the TagPicker client control](#).

In this example the "Tag Picker" caption of the TagPicker control is hidden.

1. Select the embedded TagPicker client control. The Properties Editor shows all properties of the client control.
2. In the **Misc** property category, locate the property **HideCaption**.
3. Assign the value True to it and click **Enter**. The caption "Tag Picker" of the TagPicker client control is hidden.

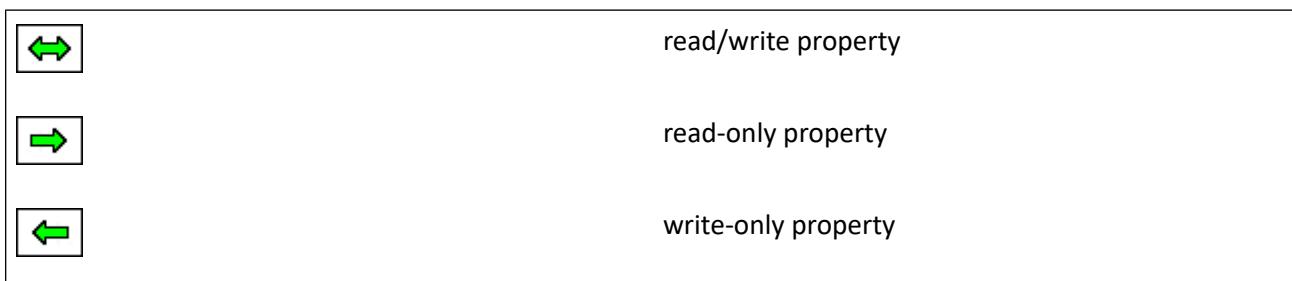
Bind client control properties to attributes or element references

You can bind the properties of an embedded client control to attributes or element references. This lets you use attributes and element references as source and consumer of data for the client control properties. You do this with the Data Binding animation.



The Data Binding table contains the following information:

- **Name** - name of the client control property
- **Type** - the .NET data type of the property
- **Value** - the default value of the client control property
- **Direction** - indicates if the property is read/write or just read-only



- **Reference** - the attribute or element reference the property is bound to

You cannot remove the **Data Binding** animation.

To bind a client control property with an attribute or element reference

1. Double-click the embedded client control on the canvas. The **Edit Animations** dialog box appears and the Data Binding animation is selected by default.
2. Locate the client control property that you want to bind with an attribute or element reference.
3. Double-click the **Reference** box.
4. Do one of the following:
 - Type an attribute or element reference.
 - Browse for an attribute or element reference by clicking the **Browse** button.
5. Repeat above for any other properties you want to bind.
6. Click **OK**.

Example of data binding in the TagPicker control

Install and embed the TagPicker control into a graphic as described in:

- [Example of installing the TagPicker control](#).
- [Embed the TagPicker client control](#).

In this example, the Boolean graphic custom property **HCV** controls the visibility of the ActiveFactory TagPicker caption.

Do the following:

1. Create a Boolean custom property and rename it **HCV**.
2. In the Industrial Graphic Editor, double-click the embedded ActiveFactory TagPicker control.
3. From the list of properties in the **Data Binding** configuration area, locate the **HideCaption** property.
4. Double-click the **Reference** box of the **HideCaption** property.
5. In your HMI's attribute/tag browser, select the **HCV** custom property and click **OK**.
The HideCaption property is now assigned to the element reference HCV.
6. Click **OK**.
7. Place a button on the canvas and configure it with a Boolean pushbutton animation that toggles the custom property HCV.
8. Save and close the graphic.
9. Embed the graphic in a managed InTouch application and test the data binding by clicking on the button in WindowViewer. When you do so, the visibility of the caption of the TagPicker control is toggled.

Configure client control event scripts

You can configure a script that is executed when a client control event occurs. You do this using the **Event** animation.

To configure a script for a client control event

1. Double-click the embedded client control on the canvas. The **Edit Animations** dialog box appears.
2. In the animation list, click **Event**. The right panel shows the configuration.
3. In the **Event** list, select the event for which you want to execute a script. The **Parameters** list shows for the selected event:
 - **Type**: the data type of each parameter.
 - **Name**: the name of each parameter.
4. In the script area, write the event script.
5. If you want to insert an event parameter in your script, click the **Select Event Parameter** icon. Select the parameter. The parameter name is inserted into the script at the cursor position.
6. If you want to configure scripts for other, select the event from the **Event** list. The script area is cleared and you can write the script for the newly selected event.
7. When you are done, save and close.

Example of configuring an event script for the TagPicker control

Install and embed the TagPicker control into a graphic as described in:

- [Example of installing the TagPicker control](#).
- [Embed the TagPicker client control](#).

In this example, when one of the tags is picked by double-clicking on it, a message is logged in the Log Viewer. First however, you need to:

- Import the script function library from the file aaHistClientDatabase.dll.
- Configure a connection to a valid and running Historian Server.

To import the script functions from aaHistClientDatabase.dll

1. On the Home ribbon, select **Galaxy**, then select **Import**.
2. Select **Visualization**, then **Client Controls**. The **Import Client Controls** dialog box appears.
3. Browse to the aaHistClientDatabase.dll and select it. By default, this file is in the C:\Program Files (x86)\ArchestrA\Framework\Bin\ViewAppFramework\SharedLibs folder.
4. Click **Open**. The import starts and finishes with a message.
5. Click **OK**.

To connect the TagPicker control to the Historian Server

1. On the canvas, place a button next to the TagPicker control.
2. Double-click the button. The **Edit Animations** dialog box appears.
3. Add an **Action Script** animation to the animation list.
4. In the script area, type the following script:

```
Dim NewServer as ArchestrA.HistClient.Database.aaServer;
```

```
Dim statusMessage as String;
NewServer = aaTagPicker1.Servers.Add("MyHistorian");
NewServer.LoginID = "MyUserName";
NewServer.Password = "MyPassword";
NewServer.LogOn( statusMessage );
LogMessage ("Connection" + statusMessage);
```

In the script, replace the strings MyHistorian, MyUserName and MyPassword with the Historian server name, a valid user name, and a password to connect to the server.

5. Close the **Edit Animations** dialog box.

You can now configure the client control event to log a message every time the user picks one or more tags by double-clicking on them:

1. In the Industrial Graphic Editor, double-click the embedded ActiveFactory TagPicker control.
2. In the animation list, click **Event**.
3. In the **Event** list, click the **OnTagsPicked** event.
4. In the script area, type the following:

```
LogMessage("User picked one or more tags.");
```
5. Save and close the **Edit Animations** dialog box.
6. Save and close the Industrial Graphic Editor.
7. Embed the graphic in a managed InTouch application.
8. Switch to runtime and connect to a valid IndustrialSQL Server source.
9. Double-click on one of the tags in the TagPicker control.
10. Check the SMC Log Viewer. The message "User picked one or more tags" appears.

Animate client controls

Every client control has these animation types:

- Data binding animations determine which attributes and element references can read and write to the client control.
- Event animations assign scripts to individual client control.

You can add the following animations that correspond to the supported client control container properties.

- Visibility
- Fill Style
- Text Style
- Location Horizontal
- Location Vertical
- Width
- Height
- Tooltip

- Disable

If you configure these animations, the resulting behavior and appearance overrides the behavior and appearance given by the native properties of the client control.

To add animation to embedded client controls

1. Double-click the embedded client control on the canvas. The **Edit Animations** dialog box appears.
2. Add animations as you would with any other element.

Export client controls

You can export client controls as ArchestrA package files (.aaPKG). You can export them:

- Directly from the Industrial Graphic Editor.
- Indirectly when you export graphics that reference them.

You can import the client controls again from exported .aaPKG files.

To export client controls directly as ArchestrA package files

1. In the **Visualization folder**, select one or more client controls that you want to export.
2. On the **Home** ribbon, in the **Export** area, select **Selected**, then select **As package**.
3. Follow the general procedure for exporting AutomationObjects. For more information, see the *Application Server User's Guide*.

Secure client controls

The client controls use the same security setting as the graphics. You can set the security for client controls and graphics in the **Security** panel of the IDE. For more information, see [Configure security for graphics](#).

Include dynamically loaded assemblies with the client control

When the primary client control assembly is imported into the HMI/SCADA project or application during an application's deployment, the system identifies all statically-linked dependent assemblies and imports them into the HMI/SCADA application as well. However, if the client control contains dynamically loaded assemblies, these assemblies are not automatically loaded in the HMI/SCADA application or project.

There are two methods for ensuring that the client control's dynamically loaded assemblies are included in the galaxy when the primary assembly is imported:

- By including the list of dynamically loaded assemblies in an XML manifest resource that is embedded in the primary assembly. The advantage of this method is that the required configuration information is packaged with the assembly, so no any other packing mechanism is required.
- By including the list of dynamically loaded assemblies in an external XML configuration file that is stored in the same directory as the primary assembly.

Both methods can be used simultaneously to provide redundancy, in the event that one of the dynamically loaded assembly lists is missing a required assembly.

Requirements for both inclusion methods

- Store all dynamically loaded assemblies, as well as all non-system static dependencies of these dynamically loaded assemblies, in the same directory as the primary assembly.
- If a dynamically loaded assembly is loading another assembly dynamically, then include the other assembly as a dynamically loaded assembly of the primary assembly. This is a requirement because the system will not search recursively for static or dynamic dependencies.

Sample XML for a dynamically loaded assembly list

A sample list of dynamically loaded assemblies in XML format is shown below. The XML list format is the same for an embedded manifest resource or an external configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
<DependentFiles>
<DependentFile>
<FileName>DynDepAsm1.dll</FileName>
</DependentFile>
<DependentFile>
<FileName>DynDepAsm2.dll</FileName>
</DependentFile>
<DependentFile>
<FileName>DynDepAsm3.dll</FileName>
</DependentFile>
</DependentFiles>
</Root>
```

XML schema for the dynamically loaded assembly list

The following XML schema is applicable for the dynamically loaded assembly XML list whether the list is provided as an embedded manifest resource or an external configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<xss:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
 xmlns:xss="http://www.w3.org/2001/XMLSchema">
<xss:element name="Root">
<xss:complexType>
<xss:sequence>
<xss:element minOccurs="1" maxOccurs="1" name="DependentFiles">
<xss:complexType>
<xss:sequence>
<xss:element minOccurs="1" maxOccurs="unbounded" name="DependentFile">
<xss:complexType>
<xss:sequence>
<xss:element minOccurs="1" maxOccurs="1" name="FileName"
 type="xs:string"/>
</xss:sequence>
</xss:complexType>
</xss:element>
</xss:sequence>
</xss:complexType>
</xss:element>
</xss:schema>
```

```
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

Embed the XML manifest resource in the primary assembly

To embed the XML manifest resource in the primary assembly

1. In Visual Studio, add the XML list file to the client control assembly project. Any file name can be used, with the extension .aaCFG.
2. Change the **Build Action** property value to Embedded Resource.

After compilation, the XML will be available in the assembly as an embedded manifest resource file.

During the client control import operation, the system will read any embedded XML manifest resources with the extension .aaCFG. The system will then import any listed assemblies that are stored in the same location as the primary assembly.

Include the XML manifest resource in an external configuration file

To include the XML manifest resource in an external configuration file

1. Create the XML list file using the same root name as the primary assembly but with the extension .aaCFG. For example, if the primary assembly name is MyControl.dll, then the configuration file name would be MyControl.aaCFG.
2. Store the file in the same directory as the primary assembly.

During the client control import operation, the system will look for a file that has the same root name as the primary assembly but with the extension .aaCFG and in the directory in which the primary assembly is stored. If this file is found and an embedded XML manifest resource exists, the system will consolidate the two lists to eliminate duplicate entries. The system will then import any listed assemblies that are stored in the same location as the primary assembly.

Prevent dynamically loaded assembly import issues

Refer to the following guidelines to prevent issues with importing the dynamically loaded assemblies.

- Verify that the XML is valid. Invalid XML in the embedded manifest resource or the configuration file will result in the entire client control import operation for the selected primary assembly to be canceled.
- Include all assemblies on which the dynamically loaded assemblies are directly or indirectly dependent, in the same directory as the primary assembly and included in the XML list. If the system is unable to locate and load any of the direct or indirect dependencies, the entire client control import operation for the selected primary assembly will fail.

- If a dynamically loaded assembly is going to load another assembly dynamically, check that the other assembly is included in the XML list. If any such assemblies are not included in the primary assembly's manifest resource or configuration file, the import operation will succeed. However, these indirectly loaded assemblies will not be imported, which can result in the client control not behaving correctly during execution.

View additional client control information

You can view:

- Which .DLL files, or assemblies, are used for the client control.
- The class name, vendor, and version.
- Which Industrial Graphics use the client control.

This information is contained in the Client Control Properties panels.

The client control properties are different than the properties of the embedded client control. The client control properties can be viewed in the IDE directly. The properties of the embedded client control can be viewed in the Properties Editor of the Industrial Graphic Editor.

View client control assemblies

You can view which Client Control .DLL files, or assemblies, are used for the client control.

To view the client control assemblies

1. In the **IDE Visualization** folder, select the client control.
2. On the **Home** ribbon, in the **Edit** area, select **Properties**. The **Properties** dialog box appears.

The screenshot shows the 'aaTrendControl properties' dialog box. The 'General' tab is selected. The 'Codebase' section lists 'ArchestrA.ClientControl.1'. The 'Derived from' section lists '\$ClientControl'. The 'Errors' and 'Warnings' sections both show 0. The 'Class name' section lists 'ArchestrA.HistClient.UI.aaTrendControl'. The 'Vendor' section lists 'AVEVA_Software_LLC_aaHistClientTrendControl.dll'. The 'Version' section lists '6.0.0.0'. The 'Primary assembly' section lists 'aaHistClientTrendControl.dll'. Below these sections is a dropdown menu labeled 'Supported languages' with 'Selected Option' selected. A scrollable list titled 'Additional assemblies' contains several DLL files: aaHistClientTrendControl.dll, aaHistClientTrendObject.dll, aaAFCommonTypes.dll, SMDiagnostics.dll, aaHistClientUI.dll, aaHistClientUtil.dll, ArchestrA.CommonUI.ThemeColor.dll, aahSHelper.dll, and Historian_DBAPI.dll. In the bottom right corner of the dialog box is a blue 'Close' button.

3. On the **General** tab, you can view:
 - The **Primary Assembly**, which is the main .DLL file.
 - **Additional Assemblies**, which are linked to the main .DLL file and automatically loaded.

View class name, vendor, and version of a client control

You can view the class name, vendor, and version of a client control in its **Properties** panel.

To view the class name, vendor, and version of a client control

1. In the IDE **Visualization** folder, select the client control.
2. On the **Home** ribbon, in the **Edit** area, select **Properties**. The **Properties** dialog box appears.

3. Click the **General** tab.

Class Name:	ArchestrA.HistClient.UI.aaTagPicker
Vendor:	Invensys_Systems__Inc__aaHistClientUI.dll
Version:	1.0.0.0

View objects and graphics referencing client controls

You can view which Industrial Graphics are using a given client control. This can be viewed in the **Properties** dialog box of the client control.

To view objects and graphics referencing a client control

1. In the **IDE Visualization** folder, select the client control.
2. On the **Home** ribbon, in the **Edit** area, select **Properties**. The **Properties** dialog box appears.
3. Click **Referenced By**. The list of objects and graphics using the client control is shown.

Ensure dynamically loaded assemblies for client controls are imported

When the primary client control assembly is imported into the galaxy during an application's deployment, all statically-linked dependent assemblies are identified and imported into the galaxy as well. However, if the client control contains dynamically loaded assemblies, these assemblies are not automatically loaded in the galaxy.

There are two methods for ensuring that the client control's dynamically loaded assemblies are included in the galaxy when the primary assembly is imported:

- By including the list of dynamically loaded assemblies in an XML manifest resource that is embedded in the primary assembly. The advantage of this method is that the required configuration information is packaged with the assembly, so no any other packing mechanism is required.
- By including the list of dynamically loaded assemblies in an external XML configuration file that is stored in the same directory as the primary assembly.

Note: Both methods can be used simultaneously to provide redundancy, in the event that one of the dynamically loaded assembly lists is missing a required assembly.

Dynamically loaded assemblies rules

- All dynamically loaded assemblies, as well as all non-system static dependencies of these dynamically loaded assemblies, must be stored in the same directory as the primary assembly.
- If a dynamically loaded assembly is loading another assembly dynamically, then the other assembly must be included as a dynamically loaded assembly of the primary assembly. This is a requirement because the system will not search recursively for static or dynamic dependencies.

XML sample for listing the file location of dynamically loaded libraries

A sample list of dynamically loaded assemblies in XML format is shown below. The XML list format is the same for an embedded manifest resource or an external configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<Root>
  <DependentFiles>
    <DependentFile>
      <FileName>DynDepAsm1.dll</FileName>
    </DependentFile>
    <DependentFile>
      <FileName>DynDepAsm2.dll</FileName>
    </DependentFile>
    <DependentFile>
      <FileName>DynDepAsm3.dll</FileName>
    </DependentFile>
  </DependentFiles>
</Root>
```

XML schema for the dynamically loaded assembly list

The following XML schema is applicable for the dynamically loaded assembly XML list whether the list is provided as an embedded manifest resource or an external configuration file.

```
<?xml version="1.0" encoding="utf-8"?>
<xss:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
  xmlns:xss="http://www.w3.org/2001/XMLSchema">
  <xss:element name="Root">
    <xss:complexType>
      <xss:sequence>
        <xss:element minOccurs="1" maxOccurs="1" name="DependentFiles">
          <xss:complexType>
            <xss:sequence>
              <xss:element minOccurs="1" maxOccurs="unbounded"
                name="DependentFile">
                <xss:complexType>
                  <xss:sequence>
                    <xss:element minOccurs="1" maxOccurs="1"
                      name="FileName" type="xs:string"/>
                  </xss:sequence>
                </xss:complexType>
              </xss:element>
            </xss:sequence>
          </xss:complexType>
        </xss:element>
      </xss:sequence>
    </xss:complexType>
  </xss:element>
</xss:schema>
```

Embed the XML manifest resource in the primary assembly

To embed the XML manifest resource in the primary assembly

1. In Visual Studio, add the XML list file to the client control assembly project. Any file name can be used, but the name must have the extension .aaCFG.
2. Change the Build Action property value to Embedded Resource.

After compilation, the XML will be available in the assembly as an embedded manifest resource file.

During the client control import operation, the system will read any embedded XML manifest resources with the extension .aaCFG. The system will then import any listed assemblies that are stored in the same location as the primary assembly.

Include the XML manifest resource in an external configuration file

To include the XML manifest resource in an external configuration file

1. Create the XML list file using the same root name as the primary assembly but with the extension .aaCFG. For example, if the primary assembly name is MyControl.dll, then the configuration file name would be MyControl.aaCFG.
2. Store the file in the same directory as the primary assembly. During the client control import operation, the system will look for a file that has the same root name as the primary assembly but with the extension .aaCFG and in the directory in which the primary assembly is stored. If this file is found and an embedded XML manifest resource exists, the system will consolidate the two lists to eliminate duplicate entries. The system will then import any listed assemblies that are stored in the same location as the primary assembly.

Tips on dynamically loaded assemblies

Refer to the following guidelines to prevent issues with importing the dynamically loaded assemblies.

- Make sure that the XML is valid. Invalid XML in the embedded manifest resource or the configuration file will result in the entire client control import operation for the selected primary assembly to be cancelled.
- All assemblies on which the dynamically loaded assemblies are directly or indirectly dependent must be included in the same directory as the primary assembly and included in the XML list. If the system is unable to locate and load any of the direct or indirect dependencies, the entire client control import operation for the selected primary assembly will fail.
- If a dynamically loaded assembly is going to load another assembly dynamically, make sure that the other assembly is included in the XML list. If any such assemblies are not included in the primary assembly's manifest resource or configuration file, the import operation will succeed. However, these indirectly loaded assemblies will not be imported, which can result in the client control not behaving correctly during execution.

Embed client controls

You can embed an installed client control into a graphic as you would embed a graphic within another graphic.

We recommend that you not overlap client controls with other elements on the canvas. Otherwise, the client controls may not work correctly.

To embed a client control into an Industrial graphic

1. On the **Edit** menu, click **Embed Industrial Graphic**. Your HMI's attribute/tag browser appears.
2. Browse to the location that contains the client control.
3. Select a client control from the right panel and click **OK**. The pointer changes to paste mode.
4. Click the canvas where you want to embed the client control. The client control is placed onto the canvas.

Embed the TagPicker client control

To embed the TagPicker client control, follow the steps of the [Example of installing the TagPicker control](#). Then do the following:

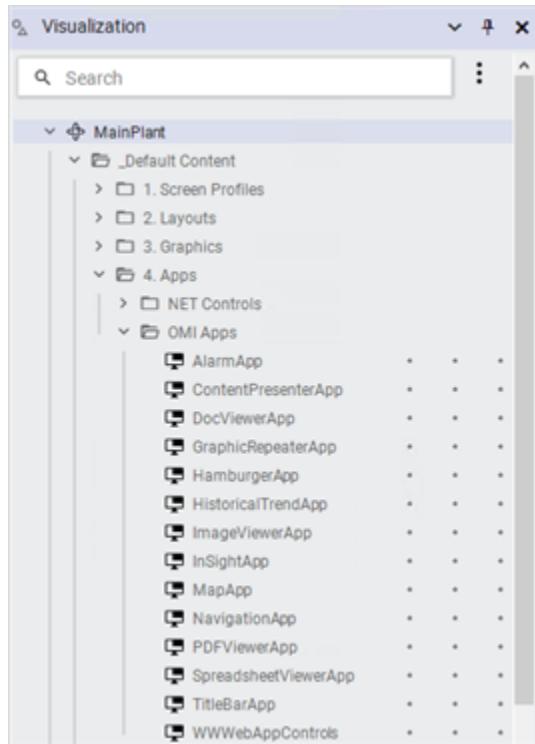
1. Create a new graphic in the Visualization folder.
2. Open the graphic in the Industrial Graphic Editor.
3. On the **Edit** menu, click **Embed Industrial Graphic**.
4. Select **aaTagPicker** and click **OK**.
5. Click on the canvas near the top left corner. The TagPicker control is placed on the canvas.

About OMI Apps

AVEVA OMI Apps are a collection of one or more controls primarily developed with Windows Presentation Foundation (WPF). Other technologies such as Windows Forms (WinForms) and HTML5 can also be used. You can also create your own AVEVA OMI Apps, which can be imported via WPF interoperability. See [Import OMI Apps \(WebHelp\)](#) for more information.

The **Visualization** folder includes a set of default AVEVA OMI Apps, which you can place in a ViewApp by dragging and dropping them onto a pane in the Layout or ViewApp editors. The default OMI Apps are listed below. For complete information about the default OMI Apps, click any of the links to the left to go directly to information about that OMI App.

The following figure shows the **Visualization** folder path to AVEVA OMI apps in the System Platform IDE.



AlarmApp

The AlarmApp can be placed in a ViewApp to show current and historical alarms and events. Alarms or events appear in tabular form with each row representing a single alarm or event. Background color of a row indicates the severity and current state of an alarm.

User1	State	Node	Group	Name	AlarmComment	Type	TimeLCT	/	Limit	CurrentValue	AlarmDuration
4	UNACK	VM-TST01	Asset_Demos	Xl_201.PV.Lo	Current scaled <Miscellaneous Meter>...	Lo	8/22/2017 7:42:43 PM	25.0		000 00:00:00.000	
2	UNACK_RTN	VM-TST01	Mixing	R34.ReactLevel.LoLo	This is the Reactors Level	LoLo	8/22/2017 7:20:41 PM	100.0		000 00:02:20.007	
3	UNACK_RTN	VM-TST01	Mixing	R34.ReactTemp.Lo	This is the Reactors Temp	Lo	8/22/2017 7:33:22 PM	25.0	104	000 00:08:40.020	
1	UNACK_RTN	VM-TST01	Mixing	R32.ReactTemp.HiHi	This is the Reactors Temp	HiHi	8/22/2017 7:40:21 PM	190.0	141	000 00:04:40.009	
3	UNACK_RTN	VM-TST01	Mixing	R32.ReactTemp.Hi	This is the Reactors Temp	Hi	8/22/2017 7:41:01 PM	180.0	141	000 00:01:40.009	
3	UNACK	VM-TST01	Mixing	StorageTank_R32.ProdLevel.Hi	Level In Storage tank	Hi	8/22/2017 7:43:51 PM	8000.0	8903	000 00:00:00.000	
1	UNACK_RTN	VM-TST01	Mixing	R34.ReactTemp.HiHi	This is the Reactors Temp	HiHi	8/22/2017 7:45:42 PM	190.0	104	000 00:01:20.014	
3	UNACK_RTN	VM-TST01	Mixing	R34.ReactTemp.Hi	This is the Reactors Temp	Hi	8/22/2017 7:47:02 PM	180.0	104	000 00:03:19.997	
4	UNACK_RTN	VM-TST01	Mixing	R34.ReactLevel.Hi	This is the Reactors Level	Hi	8/22/2017 7:47:22 PM	1500.0		000 00:16:20.043	
2	UNACK_RTN	VM-TST01	Mixing	R32.ReactLevel.LoLo	This is the Reactors Level	LoLo	8/22/2017 7:48:01 PM	100.0	1800	000 00:01:10.000	
4	UNACK_RTN	VM-TST01	Mixing	R32.ReactLevel.Lo	This is the Reactors Level	Lo	8/22/2017 7:49:01 PM	400.0	1800	000 00:03:20.018	
2	UNACK_RTN	VM-TST01	Mixing	R31.ReactLevel.LoLo	This is the Reactors Level	LoLo	8/22/2017 7:49:28 PM	100.0	463.5	000 00:00:34.995	
4	UNACK_RTN	VM-TST01	Mixing	R31.ReactLevel.Lo	This is the Reactors Level	Lo	8/22/2017 7:49:50 PM	400.0	463.5	000 00:01:40.010	
3	UNACK_RTN	VM-TST01	Mixing	R31.ReactTemp.Lo	This is the Reactors Temp	Lo	8/22/2017 7:52:38 PM	25.0	120.1	000 00:02:10.175	
3	UNACK	VM-TST01	Mixing	StorageTank_R34.ProdLevel.Hi	Level In Storage tank	Hi	8/22/2017 7:52:42 PM	8000.0	8759	000 00:00:00.000	
4	UNACK	VM-TST01	Mixing	R32.ReactLevel.Hi	This is the Reactors Level	Hi	8/22/2017 7:53:11 PM	1500.0	1800	000 00:00:00.000	
3	UNACK_RTN	VM-TST01	Mixing	R32.ReactTemp.Lo	This is the Reactors Temp	Lo	8/22/2017 7:54:21 PM	25.0	141	000 00:04:20.318	
1	UNACK_RTN	VM-TST01	Mixing	R31.ReactTemp.HiHi	This is the Reactors Temp	HiHi	8/22/2017 7:55:43 PM	190.0	120.1	000 00:00:19.998	
3	UNACK_RTN	VM-TST01	Mixing	StorageTank_R33.ProdLevel.Hi	Level In Storage tank	Hi	8/22/2017 7:55:52 PM	8000.0	6064	000 00:02:01.182	
3	UNACK_RTN	VM-TST01	Mixing	R31.ReactTemp.Hi	This is the Reactors Temp	Hi	8/22/2017 7:56:09 PM	180.0	120.1	000 00:00:50.025	
4	UNACK_RTN	VM-TST01	Mixing	R31.ReactLevel.Hi	This is the Reactors Level	Hi	8/22/2017 7:56:08 PM	1500.0	463.5	000 00:04:05.379	
4	UNACK	VM-TST01	Mixing	R34.ReactLevel.Lo	This is the Reactors Level	Lo	8/22/2017 7:56:23 PM	400.0		000 00:00:00.000	
2	UNACK_RTN	VM-TST01	Mixing	R33.ReactLevel.LoLo	This is the Reactors Level	LoLo	8/22/2017 7:56:40 PM	100.0	828	000 00:00:30.013	
4	UNACK_RTN	VM-TST01	Mixing	R33.ReactLevel.Lo	This is the Reactors Level	Lo	8/22/2017 7:56:46 PM	400.0	828	000 00:00:43.011	
3	UNACK_RTN	VM-TST01	Mixing	R33.ReactTemp.Lo	This is the Reactors Temp	Lo	8/22/2017 7:57:18 PM	25.0	140.8	000 00:00:26.007	
3	UNACK	VM-TST01	Mixing	StorageTank_R31.ProdLevel.Hi	Level In Storage tank	Hi	8/22/2017 7:57:23 PM	8000.0	8460	000 00:00:00.000	
1	UNACK_RTN	VM-TST01	Mixing	R33.ReactTemp.HiHi	This is the Reactors Temp	HiHi	8/22/2017 7:57:55 PM	190.0	140.8	000 00:00:04.018	
3	UNACK_RTN	VM-TST01	Mixing	R33.ReactTemp.Hi	This is the Reactors Temp	Hi	8/22/2017 7:57:59 PM	180.0	140.8	000 00:00:10.006	
4	UNACK_RTN	VM-TST01	Mixing	R33.ReactLevel.Hi	This is the Reactors Level	Hi	8/22/2017 7:58:00 PM	1500.0	828	000 00:00:49.009	
4	UNACK_RTN	VM-TST01	Asset_Demos	Xl_201.PV.Roc.DecreasingHi	Current scaled <Miscellaneous Meter>...	ROClo	8/22/2017 7:58:06 PM	0.0		000 00:00:06.009	
4	UNACK	VM-TST01	Asset_Demos	Xl_201.PV.Roc.IncreasingHi	Current scaled <Miscellaneous Meter>...	ROCHi	8/22/2017 7:58:06 PM	0.0		000 00:00:00.000	

Get started with the AlarmApp

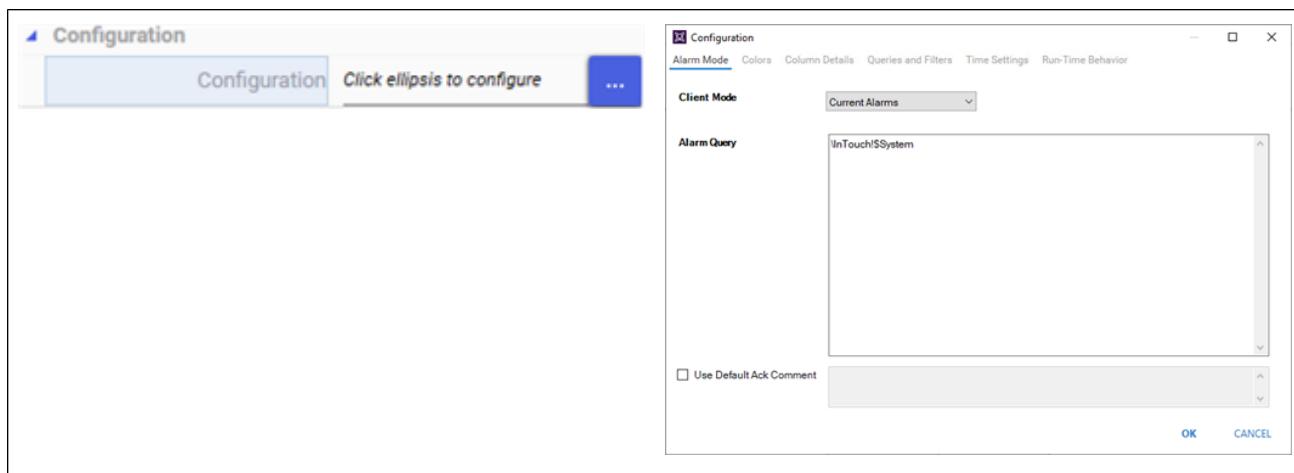
The AlarmApp includes two sets of properties that can be assigned values during design time.

- **Runtime and context properties shown from the Properties page of the Layout and ViewApp editors**

Two sets of properties are shown on the **Properties** area of the Layout and ViewApp editors after placing an AlarmApp on a layout pane. These properties are a subset of AlarmApp's **Run-Time Behavior** properties. These run-time properties have been assigned default values that minimize the number of configuration tasks required to get the AlarmApp running. For more information, see [Configure runtime properties of the AlarmApp](#).

- **All AlarmApp properties shown from the Configuration dialog box**

The **Properties** page includes an option to show the **Configuration** dialog. The **Configuration** dialog includes all AlarmApp properties that can be assigned values. For more information about configuring an AlarmApp from the **Configuration** dialog box, see [Set AlarmApp properties](#).



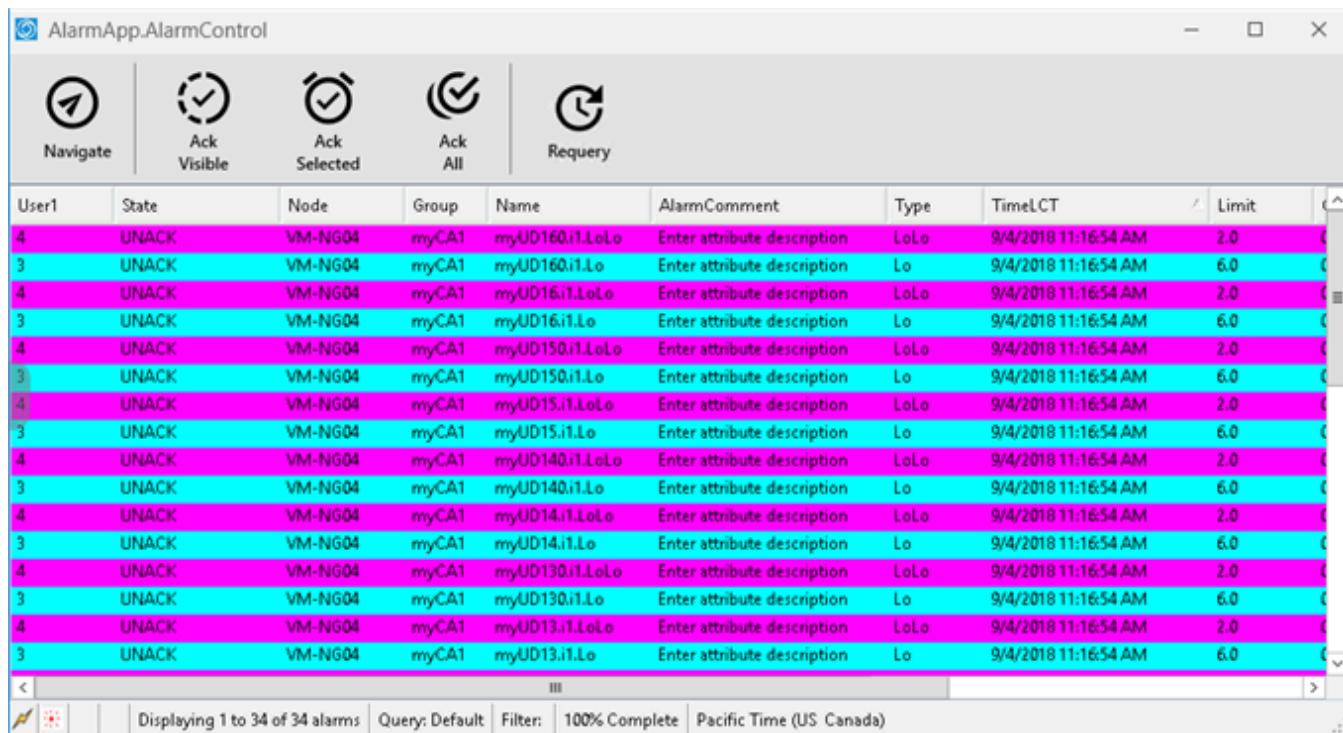
Related AlarmApp properties are grouped in a set of tabs shown on the **Configuration** dialog:

- **Alarm Mode:** Type of data shown by the AlarmApp and the default alarm database query
- **Colors:** Colors shown on the alarm grid, window, heading, and alarm records.
- **Column Details:** Column order, width, and order of alarm records.
- **Queries and Filters:** Database query for alarm records and filters saved as favorites for re-use.
- **Time Settings:** Date and time format of alarm record time stamps.
- **Run-Time Behavior:** Options that determine the physical appearance of the AlarmApp and options to manage alarms shown while the ViewApp is running.

Configure runtime properties of the AlarmApp

The AlarmApp includes a subset of run-time properties whose values determine the visual aspects or functional behavior of the AlarmApp during run time. All properties are assigned default values that enable the AlarmApp to run immediately in a ViewApp with minimal configuration. For more information about configuring all AlarmApp properties, see [Set AlarmApp properties](#).

The following screen shot shows the AlarmApp running in a ViewApp after accepting all default values assigned to the run-time properties listed on the **Properties** page of the Layout or ViewApp editors.



In the tabbed **Properties** area of the Layout or ViewApp editors, many AlarmApp properties include a user control to assign a static or dynamic value based on the type of binding between the property and its associated reference.

Constant	Static binding to the specified property value.
In	Dynamic property value with read only binding by the control to a reference.
Out	Dynamic property value with write only binding by the control to a reference.
InOut	Dynamic property value with read/write binding by the control to a reference.
Reset	Reset the current property to its default.

A check box to the right of some properties indicates the initial default value of a Boolean property. When the check box is selected, the property is set to true.

The AlarmApp shows two different types of properties in the **Properties** area of the Layout editor. The **Run-Time Behavior** properties are the native properties of the AlarmApp. This topic provides descriptions of these properties. The remaining properties are from the underlying .NET alarm control that are exposed and can be edited from the Layout script editor. .NET property groups are identified by the **AlarmClientControl** prefix in their titles that appear on the **Properties** page. For example, the **AlarmClientControl.Accessibility** title appears immediately beneath the **Area Hierarchy** group of properties.

Descriptions of .NET properties, methods, and events are easily accessible from the Microsoft Web site and are not described in System Platform documentation. For more information about the .NET properties that appear in the AlarmApp, see the .NET property descriptions at the Microsoft web site. At the time of writing, they could be found at: <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms>.

To configure run-time properties of the AlarmApp

1. Open the Layout or ViewApp editor and show the items listed in the **Toolbox** tab.
2. Select **AlarmApp** from the **Toolbox** list to show its preview thumbnail.

The **Toolbox** pathway to the AlarmApp is:

_Default Content > 4. Apps > AVEVA OMI Apps > AlarmApp

3. Drag and drop the preview thumbnail on a layout pane.
4. Select the preview thumbnail on the pane and select the **Properties** grid.

The **Properties** grid shows a list of AlarmApp run-time properties. The two screen captures shown below show the complete set of AlarmApp run-time properties and their assigned default values.

The **Content** and **Pane** property fields, listed under the **General** in the **Properties** grid, show the assigned content type and the name of the pane in which the AlarmApp is placed.

5. Set values for the AlarmApp properties. The following tables describe AlarmApp run-time properties.

Run-Time Behavior Properties

Follow Current Context	<p>Boolean value that determines if the AlarmApp shows alarms from the current area selection. Follow Current Context is enabled by default. Behavior is determined by the setting of the Only Show Alarms on Current Asset property.</p> <p>When Follow Current Context is disabled, no alarms related to the current area or asset are displayed within the AlarmApp grid at run time. The Only Show Alarms on Current Asset property does not have any effect if Follow Current Context is disabled.</p> <p>For more information about how the Follow Current Context property works with the Only Show Alarms on Current Asset property, see Follow Current Context property.</p>
Only Show Alarms on Current Asset	<p>Boolean value that determines if the AlarmApp shows aggregated alarms for all assets contained in the currently-selected area, or if it shows only alarms for the currently-selected asset. When this property is enabled, only alarms for the currently-selected asset are shown, even if the asset is an area. Note that this property only has an effect when Follow Current Context is enabled.</p> <ul style="list-style-type: none">• Default: When this property is disabled and Follow Current Context is enabled, alarms related to the current area are displayed within the AlarmApp grid at run time. If the current asset is not an area, the AlarmApp grid traverses up the hierarchy to aggregate all alarms from the area, including other contained assets, that

	<p>contains the selected asset.</p> <ul style="list-style-type: none"> When both this property and Follow Current Context are enabled: only alarms related to the current <i>asset</i> are displayed within the AlarmApp grid at run time. Alarms for contained assets are not shown. This is the case, even if the current asset is an area. When Follow Current Context is disabled, this property has no effect.
Navigate on Double Click	<p>Boolean value that determines if the user must double-click on an alarm to navigate to the asset that is in alarm. The default is true.</p> <p>If set to false, navigation by selecting an alarm is disabled. See also Show Navigate, below, under Command Bar.</p>
Use Description as Comment	<p>Boolean value that determines if an attribute description is used as the default comment for an acknowledged alarm. False is the default value to not use the description as the default comment.</p> <p>When this property is enabled, the comment popup window states that if left blank, the attribute description is used as the default comment:</p> <p>This property is only applicable when an Ack Alarm command button is used. It has no effect when the shortcut (right-click) menu is invoked.</p> <p>If the "Use Default Ack Comment" checkbox in the Configuration dialog is checked, no comment popup window is shown. Instead, when alarms are acknowledged, the text entered in the Configuration dialog is sent as the comment.</p>
Group by Header	Boolean value that allows alarms to be grouped by column in the alarm grid when the AlarmApp is in historical mode.
Show Grid	Boolean value that determines if the alarm grid shows row and column lines during run time. True is the default to show grid lines. The value set to the Show Grid property overrides the equivalent value set from the Configuration dialog.
Show Heading	Boolean value that determines if each column of the alarm grid shows a heading during run time. True is the default value to show alarm grid headers. The value assigned to the Show Headers property overrides the equivalent property set from the

	Configuration dialog.
Show Status Bar	Boolean value that determines if the status bar appears beneath the grid area of the AlarmApp during run time. True is the default to show the status bar.
Show Context Menu	Boolean value that determines if a shortcut menu containing command options appears when users right-click in the alarm grid. True is the default to show a shortcut menu. For more information, see Show Context Menu property .
Auto Scroll to New Alarms	Boolean value that determines if the alarm grid will automatically scroll to the bottom to view new alarms. When set to True , the alarm grid scrolls to the newest alarm.
Flash UnAck Alarms	Boolean value that determines if unacknowledged alarms are shown in flashing text and background colors on the alarm grid during run time. True is the default to show unacknowledged alarms in flashing colors. For an example that shows a flashing unacknowledged alarm, see Flash UnAck Alarms property . Note: Flash UnAck Alarms only works at run time if the Client Mode property is set to Current Alarms (1) or Recent Alarms (2) . The Configuration dialog will validate the setting and disallow enabling Flash UnAck Alarms if Client Mode is set to one of the historical modes (3, 4, or 5). The Property tab does not perform this validation.
Requires Ack Signature	Boolean value that determines if a signature is required to acknowledge an alarm. The default is false (no signature required to acknowledge an alarm). When set to true , the Requires Ack Signature property overrides the corresponding value set in the Configuration dialog, and the user must provide a signature to acknowledge alarms.
Ack Max Priority	Numerical value (1-999) that specifies the maximum priority of an alarm that requires an acknowledgement signature. Note: Ack Max Priority only works at run time if the

	<p>Requires Ack Signature property is enabled. The Configuration dialog will validate that the property is enabled before it lets you set Ack Max Priority. The Property tab does not perform this validation.</p>
Ack Min Priority	<p>Numerical value (1-999) that specifies the minimum priority of an alarm that requires an acknowledgement signature.</p> <p>Note: Ack Min Priority only works at run time if the Requires Ack Signature property is enabled. The Configuration dialog will validate that the property is enabled before it lets you set Ack Min Priority. The Property tab does not perform this validation.</p>

Configuration Properties

Configuration	<p>Selecting the ellipsis displays the Configuration dialog box. All AlarmApp properties can be configured in the Configuration dialog box. However, properties that are shown in the Properties grid of the Layout or ViewApp editors should be configured in the tab.</p> <p>The settings in the Configuration dialog box will be updated to match the settings in Properties grid. With some minor exceptions, changing the settings with the Properties grid will also update the settings in the Configuration dialog box.</p> <p>For more information about assigning property values from the Configuration dialog box, see Set AlarmApp properties.</p>
----------------------	---

Run-Time Values Properties

Selected Row Tagname	<p>This is a read-only property that can be used to bind a View application namespace attribute or an Application Server attribute reference to the Selected Row Tagname. This is only applicable when a user selects the row (manual selection), not when the system selects the row. The selected row tagname appears in the alarm grid at run time.</p> <p>Although you can configure the reference as two-way (read-write), it functions as an output only (read-only) property at run time.</p>
Selected Row Asset	<p>This is a read-only property that can be used to bind a View application namespace attribute or an Application Server attribute reference to the Selected Row Asset. The selected row asset appears in the alarm grid at run time.</p>

	Although you can configure the reference as two-way (read-write), it functions as an output only (read-only) property at run time.
--	--

Appearance Properties

Foreground	Foreground color of text and command buttons that appear in the Command bar. #FF000000 is the default foreground ground, which is black. Another foreground color can be selected by selecting the inverted triangle at the right of the Foreground data entry field to show a color picker. If the AlarmApp has been configured to show an area hierarchy tree view, the names of areas appear in the foreground text color.
Background	Background color of the Command bar that appears above the alarm grid. The default value is No color , which appears white. Another background color can be selected by selecting the inverted triangle at the right of the Background data entry field to show a color picker. The default background is no color. If the AlarmApp has been configured to show an area hierarchy tree view, the background color extends to the full area of the complete tree view.

Command Bar Properties

Show Command Bar	Boolean value that determines if a Command bar appears above the alarm grid area during run time. True is the default to show a Command bar. The Command bar displays large command buttons that are suitable for touch screen devices. By setting the Show... properties that follow to True, the Command Bar can display the following command buttons: For more information about the Command bar, see Show Command Bar property .
Show Navigate	Boolean value that determines if a navigate command button is shown in the Command bar. The default is true . When true, a user select an alarm from the alarm

	grid and press the navigate command to move to the asset that is in alarm. See also Navigation on Double Click , above, under Run-Time Behavior .
Show Ack Selected	Boolean value that determines if the Show Ack Selected command button appears on the Command bar. When users select one or more alarms from the grid and select the Show Ack Selected command, the selected alarms are acknowledged.
Show Ack All	Boolean value that determines if the Show Ack All command button appears on the Command bar. True is the default to show the Show Ack All command. After users select the Show Ack All command during run time, all alarms are acknowledged.
Show Ack Visible	Boolean value that determines if the Show Ack Visible command button appears on the Command bar. True is the default to show the Show Ack Visible command. After users select the Show Ack Visible command during run time, only those alarms that are visible in the Alarm grid are acknowledged.
Show Requery	Boolean value that determines if the Requery command button appears on the Command bar during run time. True is the default to show the Requery command. After the Requery command is selected, the AlarmApp requeries the historical database and shows the refreshed alarm data.
Show Connect	Boolean value that determines if the Show Connect command button appears on the Command bar during run time. False is the default to hide the Show Connect command. When set to True and users select the Show Connect icon, a request is submitted to connect the AlarmApp to the historical alarm database.
Show Disconnect	Boolean value that determines if the Show Disconnect command button appears on the Command bar during run time. False is the default to hide the Show Disconnect command. When set to True and users select the Show Disconnect command, a request is submitted to

	<p>disconnect the AlarmApp from the historical alarm database.</p> <p>The Show Disconnect command requires a pre-configured Historian connection.</p>
--	--

Alarm Mode Properties

Alarm Query	<p>Alarm Query is a read-write property to get or set an alarm query dynamically at runtime. It can be used to bind a reference to a ViewApp namespace attribute or to an application object attribute. This query overrides the query configured in the Configuration dialog.</p> <p>If you select the ViewApp context via a Navigation App (NanOsecOnd or NavBreadcrumbControl), or if the context is set by using the MyViewApp.Navigation namespace, the focus of the Alarm Query grid will not change unless the Follow Current Context property is enabled.</p> <p>Conversely, changing the Alarm Query will not change the navigation context or change the context of the ViewApp. The only way to change the focus of the Alarm Query at run time is through direct user interaction. You cannot, for example, add a script that will link the actions. For more information about query syntax, see Alarm queries.</p> <ul style="list-style-type: none"> • You can bind an external reference of string type to the Alarm Query. • Users can modify the Alarm Query property at runtime. • The Alarm Query is overridden at run time if Follow Current Context is enabled in the alarm property grid. • If the Alarm Query run time property is disabled, the Alarm Query Configuration dialog value is used as the default.
Client Mode	<p>Read/Write integer value (1-5) that determines the type of alarm data shown in the AlarmApp.</p> <p>The Client Mode property can dynamically bind to a namespace reference or to an attribute.</p> <p>When assigned a value, the Client Mode property overrides the client mode value specified in the Configuration dialog box.</p> <p>The following Client Mode integer values determine the type of alarm data shown in the AlarmApp.</p>

	<p>1 = Current Alarms, which is the default</p> <p>2 = Recent Alarms and Events</p> <p>3 = Historical Alarms</p> <p>4 = Historical Events</p> <p>5 = Historical Alarms and Events</p> <p>For more information about specifying the type of alarm data to be shown in the AlarmApp, see Client Mode property.</p>
--	--

Area Hierarchy Properties

Show Area Hierarchy	<p>The area hierarchy tree view is used to filter alarms by area (within the navigation hierarchy of the ViewApp), without changing the context of the ViewApp.</p> <p>Show Area Hierarchy is a Boolean value that determines if the area hierarchy tree view appears within the AlarmApp. False (default value) hides the area hierarchy tree view.</p> <p>If you select the ViewApp context via a Navigation App (NavTreeControl or NavBreadcrumbControl), or if the context is set by using the MyViewApp.Navigation namespace, the focus of the Alarm Query grid will not change unless the Follow Current Context property is enabled.</p> <p>Conversely, changing the Alarm Query will not change the navigation context or change the context of the ViewApp. The only way to change the focus of the Alarm Query at run time is through direct user interaction. You cannot, for example, add a script that will link the actions. For more information about query syntax, see Alarm queries.</p>
Show Alarm Indicator	Boolean value that determines if an alarm indicator appears beneath an area name shown on the area hierarchy of the AlarmApp. True is the default to show alarm indicators.
Hover Color	Background color that appears when the user hovers directly over an area hierarchy node with a mouse. The default Hover Color value is #FFD3D3D3, which is light gray.
Selection Color	Selection color that appears when the user selects an area shown in the area hierarchy node. The default Selection Color value is #FFA9A9A9, which is dark gray.

Font Size	Specifies the font size of area names that appear in the area hierarchy within the AlarmApp. The default font size is 14 points.
Font Family	Specifies the font family of area names that appear in the area hierarchy within the AlarmApp. Segoe UI is the default font.

Event Handlers Property

Available Events	Lists all available events of the AlarmApp from a list that appears by double-clicking on the data entry field. Select one or more events from the list to create event handlers that can be used in scripting. For more information about AlarmApp event handler scripts, see Write an event handler script .
-------------------------	--

Follow Current Context property

The value assigned to the **Follow Current Context** property determines if the AlarmApp responds to context changes and shows alarms for the currently-selected context of the ViewApp. When set to True, the displayed alarms are from the selected area or asset items of a ViewApp navigation model based on the value assigned to the **Only Show Alarms on Current Asset** property. The table that follows shows the displayed alarms based on the different combinations of property values, and references the following navigation hierarchy.

```

Plant [Area]
|_ Plant_Area [Area]
|_ Tank [Asset]
|_ Inlet_Valve [Asset]
|_ Outlet_Valve [Asset]

```

Follow Current Context	Only Show Alarms on Current Asset	Displayed Alarms
True	False	<p>The alarm query shows alarms from the current area of the navigation model.</p> <p>If you select an asset under an area, the AlarmApp traverses up the navigation hierarchy until it finds the area that contains the selected asset.</p> <p>Alarms are aggregated for the entire area, including all assets contained in the area, and are displayed in the AlarmApp grid at run time.</p>

		<p>Example: Selecting Inlet Valve from the navigation hierarchy shows the same alarms in the alarm grid as the following alarm query: \Galaxy!Plant_Area</p>
True	True	<p>The alarm query shows alarms from the current selected asset of the navigation model.</p> <p>If you select an asset under an area, the AlarmApp displays alarms only for selected asset. If the selected asset is an area, only alarms for the area are displayed. If assets contained by the area have alarms, these alarms are not shown.</p> <p>Example 1: Selecting Inlet_Valve from the navigation hierarchy shows the same alarms in the alarm grid as the following alarm query: \Galaxy!Plant_Area!Inlet_Valve.*</p> <p>Example 2: Selecting Plant_Area from the navigation hierarchy shows the same alarms in the alarms grid as the following alarm query: \Galaxy!Plant_Area!Plant_Area.*</p>
False	False	<p>If Follow Current Context is set to False, the value assigned to the Show Alarms on Selected Asset property does not have any effect.</p> <p>No alarms related to the current area or asset are displayed within the AlarmApp grid at run time. Instead, the AlarmApp shows alarms from the specified query assigned from the Configuration dialog or, if configured, from the Alarm Query property.</p>
False	True	

Alarm queries

The AlarmApp supports standard Galaxy alarm query formats, such as:

```
\galaxy!Area_001
```

Alarm query syntax is the same for both Current Alarms Mode and Recent Alarms and Events Mode. Queries in Historical Alarms, Historical Events, and Historical Alarms and Events modes are actually Alarm Database queries, which follow rules and syntax for SQL Server database queries.

The AlarmApp also supports relative references for Galaxy alarms in alarm queries. For all alarm modes, relative references are resolved at run time at the point of query to the Alarm Manager or Alarm Database.

You must put the reference part of the alarm query between less-than (<) and greater-than (>) characters.

The following table shows examples of alarm queries.

Alarm Query	Description
\provider!group	Shows all alarms from the given provider and group. Example \intouch!Group_A
\provider!group!tagname	Shows all alarms from the given provider, group and tag. Example \galaxy!Mixing_Area!RotorCtrl
\node\provider!group	Shows all alarms from the given provider and group from a given node. Example \remote\intouch!Group_B
\node\provider!group!tagname	Shows all alarms from the given provider, group and tag from a given node. Example \grnode\galaxy!Packaging_Area!Wrapper1
\galaxy!<Area>!<tagname>.*	Shows all alarms from the Automation Object. Alarms from other Automation Objects in the same area are ignored.
\galaxy!<myArea.tagname>	Shows all alarms from the Area object hosting the Automation Object
\galaxy!<myPlatform.tagname>	Shows all alarms from the Winplatform object hosting the Automation Object.
\galaxy!<myEngine.tagname>	Shows all alarms from the AppEngine object hosting the Automation Object. At run-time the AlarmApp resolves the myEngine attribute to detect the host.

Alarm query syntax

For AVEVA OMI, we recommend that you configure Alarm Queries with the Alarm Query property, rather than through the Configuration dialog. The following information is intended for InTouch HMI users:

The run-time alarm comment language switching feature requires slightly different alarm query syntax. In the WinPlatform object, when you enable InTouch alarm provider, you can enable **Register using Galaxy_<GalaxyName>** instead of Galaxy.

This option will register the platform to the alarm subsystem using the Galaxy name preferred by "Galaxy_" instead of just the word "Galaxy". This allows an InTouch application to monitor alarms from multiple Galaxies and avoid name conflicts.

Syntax changes slightly when Galaxy_GalaxyName is enabled:

- Use \\ for computer name
- Use \ for Galaxy or Galaxy_<GalaxyName>
- Use ! for Area

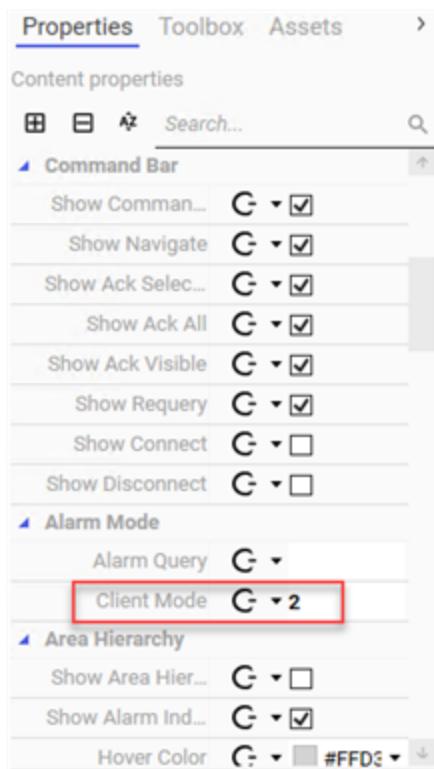
For example: \\Galaxy\MyGalaxy!Area001

If Galaxy_GalaxyName is not enabled in WinPlatform, then the default behavior described in Alarm Queries applies.

You can determine if Galaxy_<GalaxyName> has been enabled by monitoring the run-time attribute of the platform: ITAlarmProvider.ProviderNameAsGalaxyNameEnabled.

Client Mode property

You can set the client mode in the AlarmApp property grid, under the Alarm Mode category, by entering value from 1 to 5. If you prefer, you can bind an external reference of an integer type to the client mode, instead of setting a static value.



- Client Mode = 1: display current alarms
- Client Mode = 2: display recent alarms & events
- Client Mode = 3: display historical alarms
- Client Mode = 4: display historical events
- Client Mode = 5: display both historical alarms and events

Note: The default client mode tab within the Configuration dialog box is "Current Alarms."

Show Grid property

You can bind a Boolean external reference to the Show Grid property.



- At run time, a user can modify the Show Grid property to show or hide grid lines in the AlarmApp.
- If the Show Grid property is disabled at run time, the AlarmApp reverts to the configuration value.

Show Headers property

You can bind a Boolean external reference to the Show Headers property.

- At run time, a user can modify the Show Headers property to show or hide the Header column on the AlarmApp.
- If the user disables the Show Headers property at run time, the AlarmApp reverts to the configuration value.

Show Status property

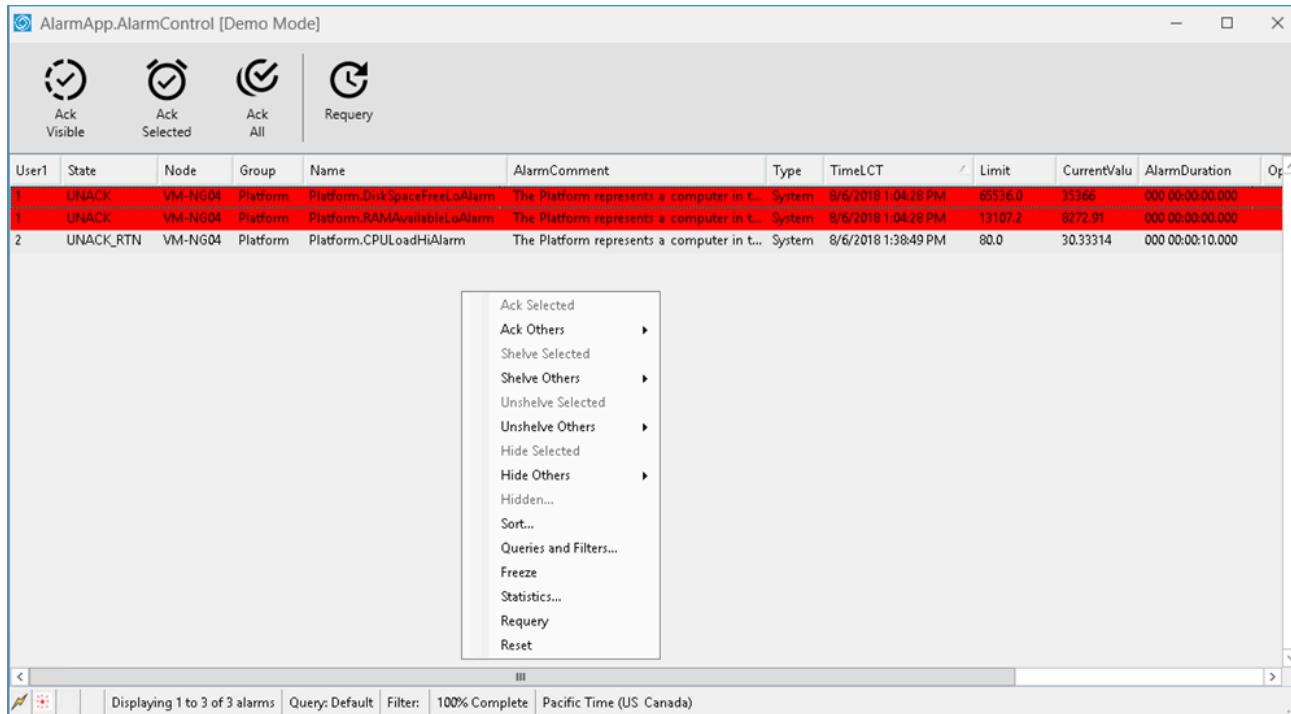
You can bind a Boolean external reference to the Show Status property .

- At run time, a user can User modify the Show Status property to show or hide the status bar in the AlarmApp.
- If the Show Status property is disabled at run time, the AlarmApp reverts to the configuration value.

Show Context Menu property

You can bind a Boolean external reference to the Show Context Menu property.

- At run time, a user can enable or disable the Show Context Menu property.
- If the Show Context Menu property is disabled at run time, the AlarmApp reverts to the configuration value.



Note: The default **Show Context Menu** property provides a subset of possible commands that are added to the shortcut menu. For a complete set of context menu commands, see [Configure the runtime shortcut menu](#).

Auto Scroll property

- When the Auto Scroll property is true at run time, the vertical scroll bar on the alarm grid scrolls to the bottom of the alarm grid to show the most recent alarm record.
- When the Auto Scroll property is false at run time, the alarm grid remains at the current view and does not scroll.

Flash UnAck Alarms property

- When the Flash UnAck Alarms property is true at run time, unacknowledged alarms will flash in the AlarmApp to provide an enhanced indication to the user that there is an unacknowledged alarm.
- When the Flash UnAck Alarms property is false at run time, unacknowledged alarms do not flash in the AlarmApp.

Ack Signature property

Use the Ack Signature property to bind external references of Boolean type to the Required ACK Signature in the AlarmApp property grid in alarm mode category.

- Users can modify the Required ACK Signature property at runtime.
- If the Required ACK Signature property is disabled at run time, the AlarmApp reverts to the configuration value.

Ack Min Priority property

- This property sets the value for Minimum Priority from 1 to 999. Minimum Priority sets the threshold that an alarm must reach to require an acknowledgement signature.
- If the Required Ack Signature property has been changed at run time, the Min Priority property is ignored and all alarms below the Max Priority setting will require an acknowledgement signature.

Ack Max Priority property

- This property sets the value for Maximum Priority from 1 to 999. Max Priority sets the threshold that an alarm must not exceed to require an acknowledgement signature.
- If the Required Ack Signature property has been changed at run time, the Min Priority property is ignored and all alarms below the Max Priority setting will require an acknowledgement signature.

Show Command Bar property

A command bar appears above the alarm grid area of the AlarmApp. The command bar shows a set of large icons that represent AlarmApp commands, which are recommended for ViewApps that will be displayed on portable touch devices with relatively small screens. Users can simply tap an icon shown on the screen to run the command.

The following figure shows a command bar with the default set of command icons that are selected from the **Properties** page of the ViewApp or Layout editors.

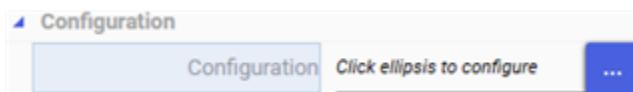
	AlarmComment	Type	TimeLCT	Limit
inopqrstuvwxyz6#\$\$____.i1.Lo	Enter attribute description	Lo	9/4/2018 11:16:54 AM	6.0
inopqrstuvwxyz6#\$\$____.i1.LoLo	Enter attribute description	LoLo	9/4/2018 11:16:54 AM	2.0
opqrstuvwxyzuvwxyz012345.i1.Lo	Enter attribute description	Lo	9/4/2018 11:16:54 AM	6.0
opqrstuvwxyzuvwxyz012345.i1.LoLo	Enter attribute description	LoLo	9/4/2018 11:16:54 AM	2.0

Set AlarmApp properties

AlarmApp properties can be configured from the **Configuration** dialog, which provides access to all properties expected to be modified by users. The **Configuration** dialog sets the initial property values of the AlarmApp.

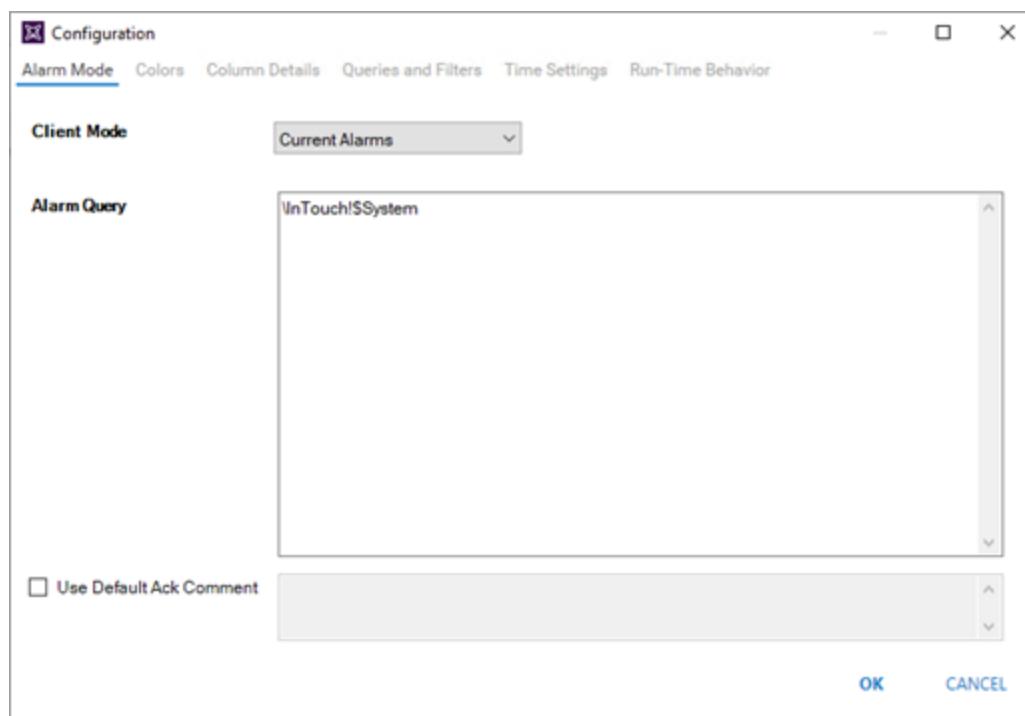
To configure AlarmApp properties

1. Place the AlarmApp on a layout pane.
2. Select the AlarmApp.
3. Select the **Properties** tab to show AlarmApp properties.
4. Locate the **Configuration** property and select the ellipsis button shown at the right of the **Configuration** field.



The **Configuration** dialog box appears with tabs to configure all related properties of the AlarmApp.

Important: Only use the **Configuration** dialog to configure properties that are not exposed by the **Properties** tab. For example, if you want to configure alarm acknowledgment colors or Time Settings, use the **Configuration** dialog, as these settings cannot be set through the **Properties** tab.

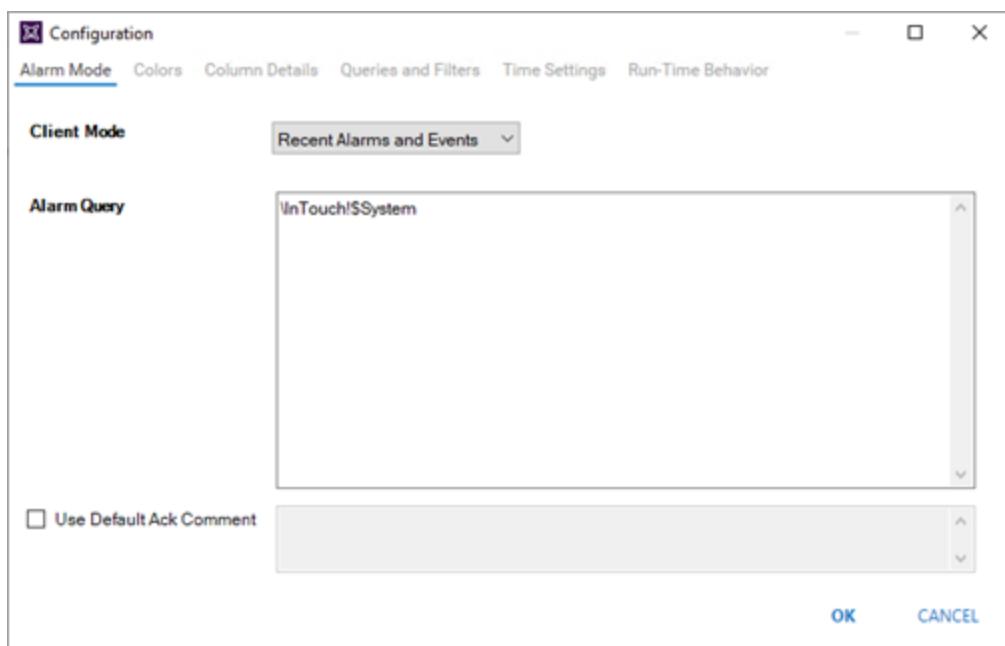


Show current alarms and events

You use the Client Mode property integer property in scripting to switch the AlarmApp to show either current alarms or recent alarms and events at run time.

To show current alarms

1. Select the AlarmApp placed on a pane.
2. If necessary, click **Properties** to show the **Properties** pane. In the properties list, click the ellipsis button (...) next to the **Configuration** field to open the **Configuration** dialog box.
3. If necessary, click the **Alarm Mode** tab. The **Alarm Mode** page appears.



4. In the **Client Mode** list, click **Current Alarms**.
5. In the **Alarm Query** box, type the alarm query. To create a new line in the **Alarm Query** box, press **Ctrl + Enter**. For more information on the valid syntax, see [Alarm queries](#).
6. If you want to use a default acknowledgement comment, select the **Use Default Ack Comment** check box and type a comment in the text box.
7. Click **OK**.

To show recent alarms and events

1. Double-click the AlarmApp on the canvas. The **Edit Animations** dialog box appears.
2. Click **Alarm Mode**. The **Alarm Mode** page appears.
3. In the **Client Mode** list, click **Recent Alarms and Events**.
4. In the **Alarm Query** box, type the alarm query. To create a new line in the **Alarm Query** box, press **Ctrl + Enter**.

The alarm query must follow one of the following syntax:

- \\node\\provider!group
- \\provider!group
- HotBackupName (InTouch HMI only; not applicable to AVEVA OMI)

For example:

```
\intouch!$system \galaxy!Area_001
```

When the AlarmApp is hosted by Automation Object templates or instances, you can specify one of the following alarm queries:

- \\galaxy!<myArea.Tagname> to retrieve alarms and events from the Area object hosting the Automation Object template or instance.
- \\galaxy!<Area>!<Tagname>.* to retrieve alarms and events from the Automation Object template or instance.

For more information on alarm queries, see [Alarm queries](#).

5. If you want to use a default acknowledgement comment, select the **Use Default Ack Comment** check box and type a comment in the text box.
6. Click **OK**.

Show historical alarms and events

You can set the AlarmApp to show one of the following:

- Historical alarms from the Alarm Database
- Historical events from the Alarm Database
- Historical alarms and events from the Alarm Database
- Historical events from History Blocks
- Historical alarms from History Blocks
- Historical alarms and events from History Blocks

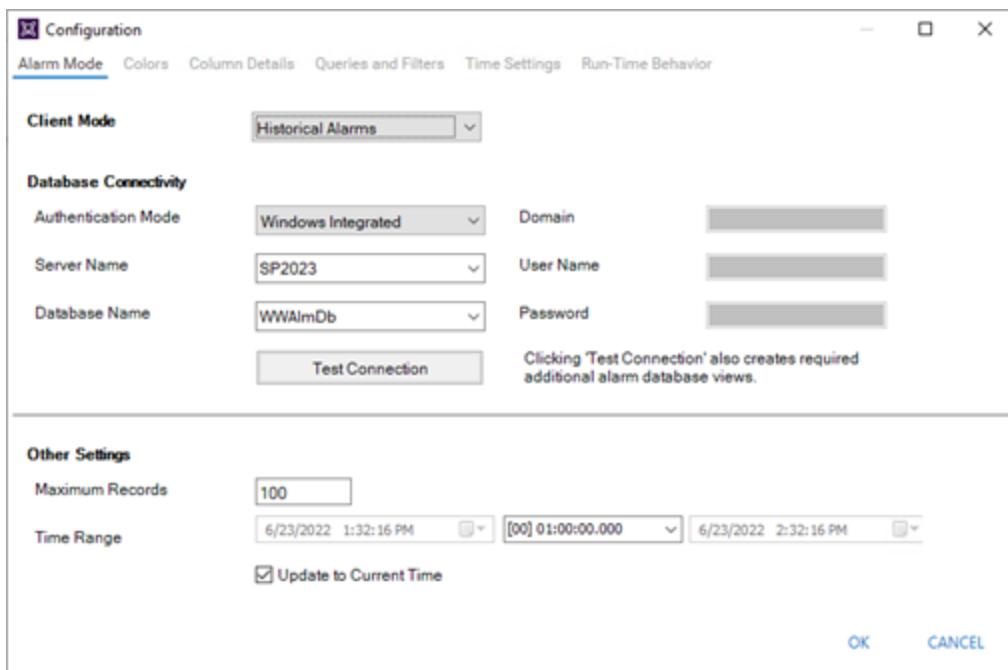
When you configure the AlarmApp to show historical alarms and/or events, you also configure the following:

- Server name hosting the Alarm Database
- Authentication information to connect to the Alarm Database
- Maximum number of records to retrieve from the Alarm Database
- Time range or duration to show in the AlarmApp
- If the AlarmApp should update to the current client time

The AlarmApp works with both the ArchestrA Database (A2ALMDB) and with History Blocks.

To show historical alarms and/or events

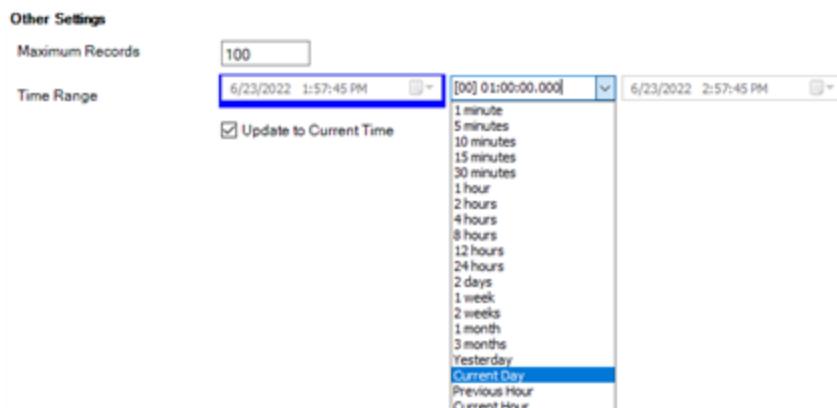
1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. If necessary, click the **Alarm Mode** tab. The **Alarm Mode** page appears.
5. In the **Client Mode** list, click:
 - **Historical Alarms** to only show alarms from the Alarm Database. No events are shown
 - **Historical Events** to only show events from the Alarm Database. No alarms are shown
 - **Historical Alarms and Events** to show both alarms and events from the Alarm Database



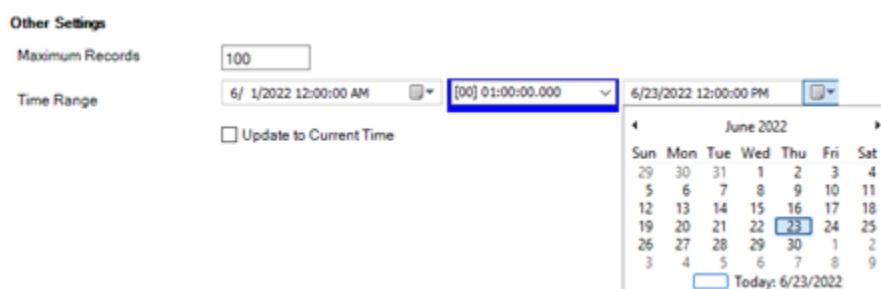
6. In the **Authentication Mode** list, select one of the following:
 - Windows Integrated to use the authentication of the currently logged-on Windows user
 - Windows Account to use a given Windows user authentication
 - SQL Server to use SQL Server authentication mode
7. In the **Server Name** list, either select or type the name of the server hosting the Alarm Database.
8. In the **Database Name** box, type the name of the Alarm Database. For the ArchestrA Database, enter A2ALMDB, and for Historian block storage enter History Blocks.
9. If you are using Windows Account authentication mode, type the domain, user name, and password in the **Domain**, **User Name** and **Password** boxes.
10. If you are using SQL Server authentication mode, type user name and password in the **User Name** and **Password** boxes.
11. Click **Test Connection**. The connection to the Alarm Database is tested and a result message appears. If necessary, check your authentication information.
12. Click **OK**.

To set maximum records and time range

1. Show the AlarmApp's **Configuration** dialog box.
2. If necessary, click the **Alarm Mode** tab. The **Alarm Mode** page appears.
3. Make sure the **Client Mode** field is set to Historical Alarms, Historical Events, or Historical Alarms and Events.
4. In the **Maximum Records** box, type the number of records to view from the control at one instance. The valid range of maximum records is from 1 to 32766.
5. To set a fixed length interval, select an interval from the middle list of the **Time Range** pickers.



6. To use a specific start time and end time, clear **Update to Current Time**, and select the start time from the list at the left and the end time from the list at the right of the Time Range pickers.



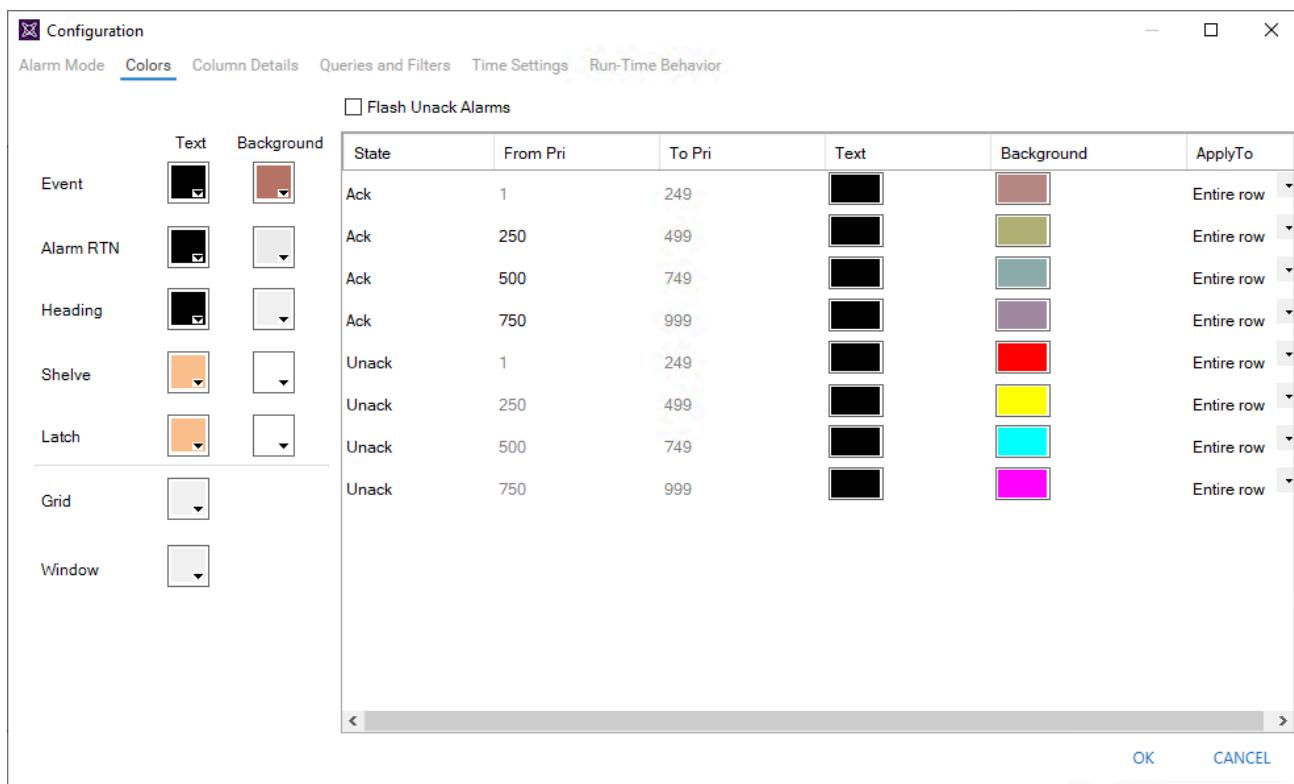
7. Click **OK**.

Set alarm colors

The **Colors** tab of the **Configuration** dialog lets you set a number of color options for your run-time display. Through this dialog, you can:

- Differentiate types of alarm records by assigning different colors let users more easily identify alarm types.
- Configure colors to apply to specific priority ranges to differentiate alarm records by priority or severity (see note).
- Apply the color settings to an entire row or just to the column that contains the priority alarm.
- Set the control background color, the grid color, and the heading colors.

Note: When the **Client Mode** property is set to show current alarms, only the Priority column setting is available; Severity is not available for current alarms. When Client Mode is set to show historical alarms, both Priority and Severity columns are supported. For more information, see [Show current alarms and events](#) and [Show historical alarms and events](#).



An alarm display configured for historical alarms is shown below. The configuration settings are on the left, the run-time display is on the right.

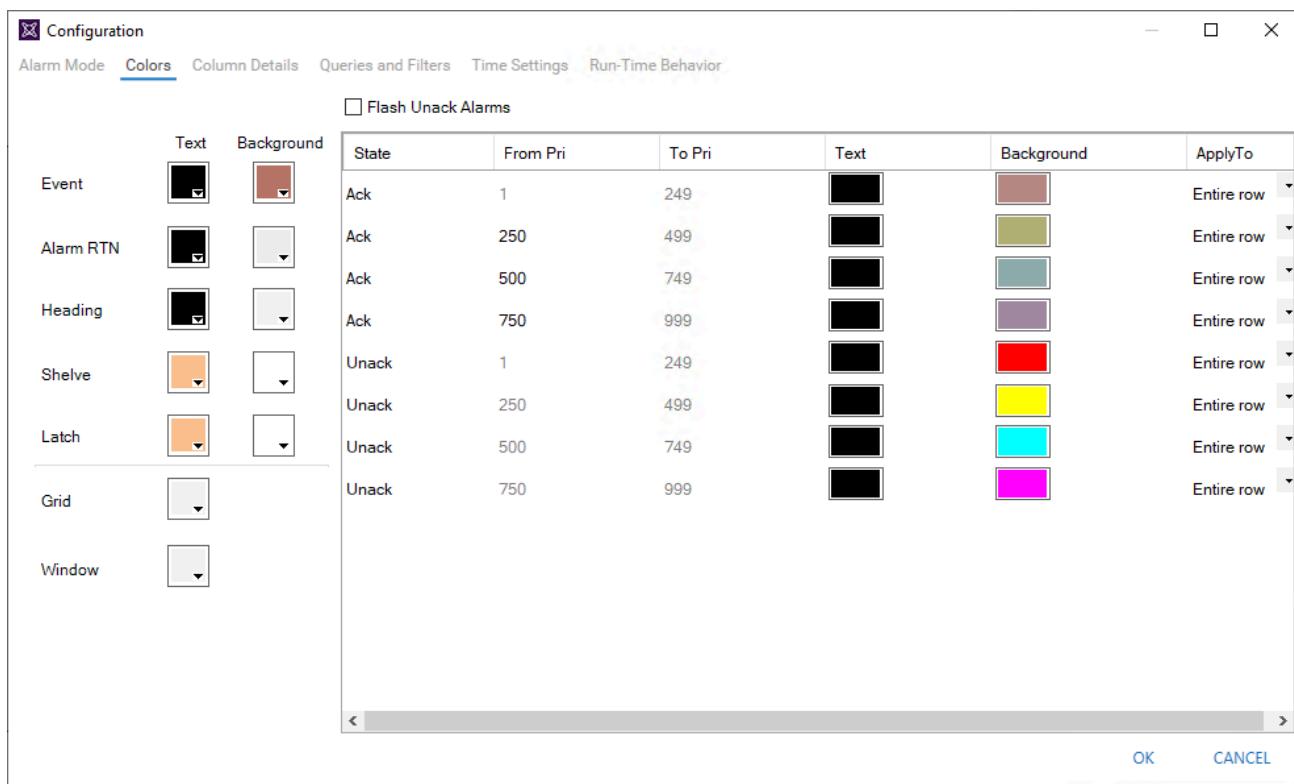
User1	Priority	Severity	State	Group	Name	AlarmComment
3	600	3	ACK_ALARM	Plant_Area	UDO2_alarmapp.Empty	Std ACK - bharat4
3	600	3	ACK_RTN	Plant_Area	UDO2_alarmapp.Empty	Std ACK - bharat4
3	600	3	UNACK_RTN	Plant_Area	UDO2_alarmapp.Empty	Going out of alarm because
3	600	3	UNACK_ALARM	Plant_Area	UDO2_alarmapp.Empty	Going out of alarm because
3	600	3	UNACK_ALARM	Plant_Area	UDO2_alarmapp.Empty	
3	600	3	UNACK_ALARM	Plant_Area	UDO2_alarmapp.Empty	
2	500	2	ACK_ALARM	Plant_Area	UDO1_alarmapp.test1	Std ACK - bharat3
2	500	2	ACK_ALARM	Plant_Area	UDO1_alarmapp.test1	Std ACK - bharat2

Set event record colors

You can set text color and background color for event records. Use the `EventColor.ForeGround` Property and `EventColor.BackGround` Property properties in scripting to set the event alarm record text color and background color at run time.

To set text and background colors for event records

1. Select the Alarm app placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.



4. Configure the event record text color. Do the following:
 - a. Click the color field next to **Event** and under **Text**. The color picker appears.
 - b. Select a color and click **OK**.
5. Configure the event record background color. Do the following:
 - a. Click the color field next to **Event** and under **Background**. The color picker appears.
 - b. Select a color and click **OK**.
6. Click **OK**.

Set return to normal alarm record colors

You can set text color and background color for "return to normal" alarm records.

To set text and background colors for "return to normal" records

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field.
4. Click the **Colors** tab. The **Colors** page appears.
5. Configure the "return to normal" record text color. Do the following:
 - a. Click the color field next to **Alarm RTN** and under **Text**. The color picker appears.
 - b. Select a color and click **OK**.
6. Configure the "return to normal" record background color. Do the following:
 - a. Click the color field next to **Alarm RTN** and under **Background**. The color picker appears.

- b. Select a color and click **OK**.
7. Click **OK**.

Set heading, grid, and window color

You can set text color and background color for the heading, the grid color, and the AlarmApp window color. Use the corresponding HeadingColor.ForeGround Property, HeadingColor.BackGround Property, GridColor Property, and WindowColor Property properties in scripting to set the colors for heading, grid, and window.

To set heading, grid, and window color for the AlarmApp

1. Select the AlarmApp placed on a pane.
2. Select the Properties grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field.
4. Click the **Colors** tab. The **Colors** page appears.
5. Do one of the following:
 - a. Configure the heading text color by clicking the color box next to **Heading** and under **Text**. If the color box does not open, you need to select the **Show Heading** option on the **Run-Time Behavior** page first.
 - b. Configure the heading background color by clicking the color box next to **Heading** and under **Background**. If the color box does not open, you need to select the **Show Heading** option on the **Run-Time Behavior** page first.
 - c. Configure the grid color by clicking the color box next to **Grid**. If the color box does not open, you need to select the **Show Grid** option on the **Run-Time Behavior** page first.
 - d. Configure the window color by clicking the color box next to **Window**.

Set alarm shelving colors

You can set text color and background color for alarms that are temporarily shelved.

To set alarm shelved colors

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. Click the **Colors** tab. The **Colors** page appears.
5. Configure the shelf record text color. Do the following:
 - a. Click the color field next to **Shelve** and under **Text**. The color picker appears.
 - b. Select a color and click **OK**.
6. Configure the shelf record background color. Do the following:
 - a. Click the color field next to **Shelve** and under **Background**. The color picker appears.
 - b. Select a color and click **OK**.
7. Click **OK**.

Set priority ranges and column options for alarm records

You can use alarm priority ranges to filter alarms. The AlarmApp can show alarms within a given range with a different text and background color.

The AlarmApp supports four alarm ranges defined by three breakpoints:

$1 < \text{breakpoint 1} < \text{breakpoint 2} < \text{breakpoint 3} < 999$

You can also set the run-time behavior of the alarm display

To set priority ranges for alarm records

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. Click the **Colors** tab. The **Colors** page appears.
5. In the **From Pri** column in the list at the right, locate the break point you want to change. These are values except 1 or 999.
6. Select the value and enter a new value. The new value must fall in the range between the previous breakpoint and the next breakpoint.
7. In the **ApplyTo** column, select whether the information you entered should apply to the entire row or just the affected column. To apply the settings to only the column for current alarms, select **Priority**. If the AlarmApp is configured for historical alarms, you can choose between **Priority** and **Severity** if you want the settings to apply only to the affected column.

State	From Pri	To Pri	Text	Background	ApplyTo
Ack	1	249			Entire row
Ack	250	499			Entire row
Ack	500	749			Entire row
Ack	750	999			Entire row
Unack	1	249			Entire row
Unack	250	499			Entire row
Unack	500	749			Entire row
Unack	750	999			Entire row

8. Press **Enter**. All values and settings in the list are updated.
9. Click **OK**.

Set colors for acknowledged alarms

You can set the text and background colors for records of acknowledged alarms. For each of the alarm priority ranges, you can set a text color and a background color.

To set colors for acknowledged alarm records

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. Click the **Colors** tab. The **Colors** page appears.
5. In the list at the right, locate the Ack record and priority range for which you want to change the text or background color.
6. Click the color box in the **Text or Background** column of the line. The color picker appears.
7. Select a color.
8. Click **OK**.

Set colors for unacknowledged alarms

You can set the text and background colors of unacknowledged alarm records. For each of alarm priority range, you can set a text color and a background color.

To set colors for unacknowledged alarm records

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. Click the **Colors** tab. The **Colors** page appears.
5. In the list at the right, locate the **Unack** record and priority range for which you want to change the text or background color.
6. Click the color box in the **Text or Background** column of the line. The color picker appears.
7. Select a color and click **OK**.

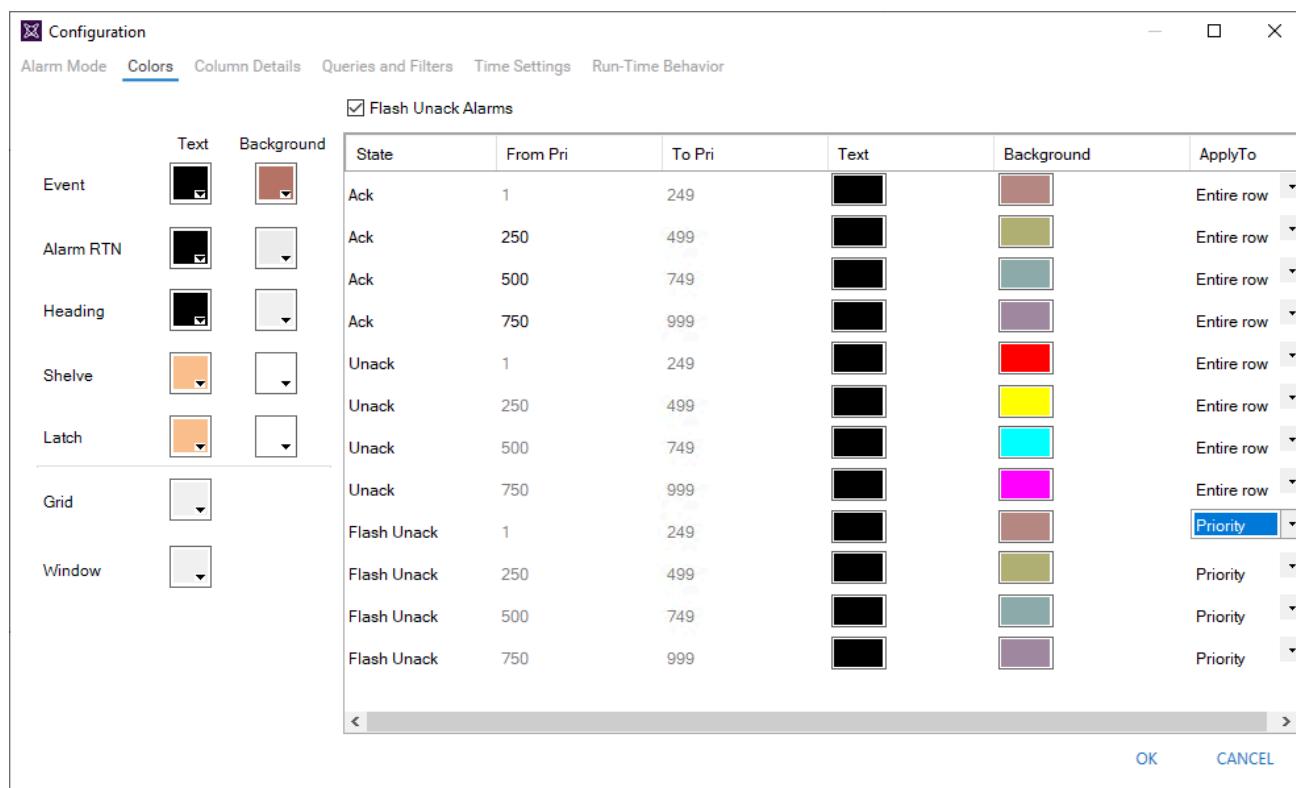
Set unacknowledged alarms to flash

Instead of showing unacknowledged alarm records in predefined constant text and background color, you can configure the AlarmApp to flash unacknowledged alarms repetitively in other text and background colors to identify unacknowledged alarms more quickly.

You can also set the entire row to flash, or just the column containing the unacknowledged alarms.

To set flashing and colors for unacknowledged alarm records

1. From the Layout editor, select the pane containing the AlarmApp.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. Click the **Colors** tab. The **Colors** page appears.
5. Select the **Flash Unack Alarms** check box.
6. For each priority range of the Unacked Alarms, select from the **ApplyTo** dropdown whether you want the entire row to flash, or just the column with the Priority alarm. The default is to flash the entire row.



Note: You cannot select the Flash UnAck Alarms check box if the Client Mode property is set to one of the historical data modes.

7. In the list on the right, locate the Unack record and priority range for which you want to change the text or background color. Do the following:
 - a. Click the color box in the **Text** or **Background** column of the line. The color picker appears.
 - b. Select a color and click **OK**.
8. Locate the **Flash Unack record** and priority range for which you want to change the text or background color. Do the following:
 - a. Click the color box in the **Text** or **Background** column of the line. The color picker appears.
 - b. Select a color and click **OK**.
9. Click **OK**.

Rename, resize, and reorder columns

You can select which columns are displayed in the AlarmApp. You can also rename, resize, and change the order of columns. You can also select which columns are used for sorting the alarm list and select the sorting order of those columns are displayed, and the sort order the alarm list.

The screenshot shows the 'Column Details' tab of the Configuration dialog box. On the left is a table of columns with checkboxes for 'Display Name', 'Width', and 'Original Name'. To the right are 'Sorting' controls for four levels and 'Column ordering buttons' (up and down arrows). Below is a 'Grid preview' showing a subset of the columns. Arrows point from the text labels to their respective elements.

	Display Name	Width	Original Name
<input checked="" type="checkbox"/>	User1	75	User1
<input checked="" type="checkbox"/>	State	120	State
<input checked="" type="checkbox"/>	Node	90	Node
<input checked="" type="checkbox"/>	Group	100	Group
<input checked="" type="checkbox"/>	Name	250	Name
<input checked="" type="checkbox"/>	AlarmComment	235	AlarmComment
<input checked="" type="checkbox"/>	Type	70	Type
<input checked="" type="checkbox"/>	TimeLCT	200	TimeLCT
<input checked="" type="checkbox"/>	Limit	75	Limit
<input checked="" type="checkbox"/>	CurrentValue	75	CurrentValue
<input checked="" type="checkbox"/>	AlarmDuration	140	AlarmDuration
<input checked="" type="checkbox"/>	Operator	125	Operator
<input checked="" type="checkbox"/>	UnAckDuration	140	UnAckDuration
<input type="checkbox"/>	Comment	100	Comment

Sorting

First Sort Criteria: TimeLCT, Ascending

Second Sort Criteria: None, Ascending

Third Sort Criteria: None, Ascending

Fourth Sort Criteria: None, Ascending

Column ordering buttons

Grid preview

OK CANCEL

All changes you make in the **Column Details** list are shown in the grid preview. You can also use the grid preview to resize columns or change their order by dragging and dropping the column headers.

Column headers can be localized along with other graphic text when you export, translate, and re-import language files. The translated language files must be imported to support run-time language switching.

System Platform 2023 R2 SP1 includes two new columns that you can enable for display in the AlarmApp:

- AlarmMessage
- AckMessage

<input type="checkbox"/>	UTCTime	100	UTCTime
<input type="checkbox"/>	Priority	100	Priority
<input type="checkbox"/>	AlarmMessage	235	AlarmMessage
<input type="checkbox"/>	AckMessage	235	AckMessage

These columns are not enabled by default. When enabled, the AckMessage and AlarmMessage columns will appear in the AlarmApp during run-time.

Name	AlarmComment	Type	Limit	AckMessage	AlarmMessage	Priority
UDO01.StringCode.LoLo	ACK 1	LoLo	10.0	ACK 1	QR Codes	500
UDO01.contact.LoLo	contact	LoLo	10.0		contact	500
UDO01.contact.Lo	contact	Lo	25.0		contact	500
UDO01.QRCColor.LoLo	QR	LoLo	10.0		QR	500
UDO01.QRCColor.Lo	QR	Lo	25.0		QR	500

Important: If you rename or reorder column headers, you must repeat the graphic text translation procedures. If you do not, your changes will not be available for run-time language switching.

Rename columns

You can rename the column headers in the AlarmApp.

To rename column headers

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field.
4. Click the **Column Details** tab. The **Column Details** page appears.
5. In the **Column Details** list, locate the column header you want to rename and click on it.
6. Type a new name and press **Enter**. The **Column Details** list and the grid preview are updated with the new name.

Display Name	Width	Original Name
User1	75	User1
State	120	State
Node	90	Node
Group	100	Group
NewName	150	Name
AlarmComment	235	AlarmComment
Type	70	Type
TimeLCT	200	TimeLCT
Limit	75	Limit
CurrentValue	75	CurrentValue
AlarmDuration	140	AlarmDuration
Operator	125	Operator
UnAdvDuration	140	UnAdvDuration
Quality	100	Quality
TimeLCTOAT	120	TimeLCTOAT
Value	120	Value

7. Click **OK**.

Resize columns

You can resize columns in the AlarmApp by:

- Entering a numeric size value

- Dragging the column header boundary width with the pointer in the grid preview

To resize the column numerically

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. Click the **Column Details** tab. The **Column Details** page appears.
5. In the **Column Details** list, click the **Width** value of the column to resize.
6. Type a new width in pixels and press **Enter**. The **Column Details** list and the grid preview are updated.

Column Details

Display Name	Width	Original Name
<input checked="" type="checkbox"/> User1	75	User1
<input checked="" type="checkbox"/> State	120	State
<input checked="" type="checkbox"/> Node	90	Node
<input checked="" type="checkbox"/> Group	100	Group
<input checked="" type="checkbox"/> AlarmComment	235	AlarmComment
<input checked="" type="checkbox"/> Name	250	Name
<input checked="" type="checkbox"/> Type	70	Type
<input checked="" type="checkbox"/> TimeLCT	200	TimeLCT
<input type="checkbox"/>	75	

7. Click **OK**.

Note: To set all column properties (width, names, and sort order) back to their original values, click **Reset**.

Change column order

You can change the order of the columns in the AlarmApp by:

- Moving column names up and down in the Column Details list using buttons
- Dragging the column header with the pointer in the grid preview

You also can reset the column widths and order to their default values. Resetting the column widths and order also resets the names to their default values.

To change the column order

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. Click the **Column Details** tab. The **Column Details** page appears.
5. Do one of the following:
 - Select the column in the **Column Details** list and click the up or down arrow to move it right or left in the display.

- In the grid preview, drag and drop the name of the column you want to move.
The grid preview and the **Column Details** list shows the new column order.
6. Click **OK**.

Sort alarms

You can configure how the AlarmApp sorts alarm records at run time. By default, the AlarmApp lists alarm records by time in ascending order.

You can sort alarm records in ascending or descending order based on a primary column, an optional secondary sort column, and an optional tertiary sort column.

Sorting

First Sort Criteria	
TimeLCT	Ascending
Second Sort Criteria	
None	Ascending
Third Sort Criteria	
None	Ascending
Fourth Sort Criteria	
None	Ascending

You can configure the sorting columns and directions either in lists or with the grid preview.

To set sorting columns and directions with lists

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.
4. Click the **Column Details** tab. The **Column Details** page appears.
5. In the Sorting area, do the following:
 - a. Select the primary sort column in the **First Sort Column** list and a sorting direction in the list to its right.
 - b. Optionally, select the secondary sort column in the **Second Sort Column** list and a sorting direction in the list to its right.
 - c. If you set the **Second Sort Column**, optionally select the tertiary sort column in the **Third Sort Column** list and a sorting direction in the list to its right.

The grid preview is updated and shows arrows for the sorted columns and their sort directions.

Time (LCT)	/	State	▼	Type	/	0

To set sorting columns and directions with the grid preview

1. Select the AlarmApp placed on a pane.
 2. Select the **Properties** grid to show AlarmApp properties.
 3. Select the ellipsis button shown at the right of the **Configuration** field.
 4. Click the **Column Details** tab. The **Column Details** page appears.
 5. In the grid preview, click on a column to select it for sorting. An arrow appears on the column header and the change is also shown in the Sorting area lists.
 6. To change the sorting direction, click on the column header again. The arrow changes on the column header and the change is also shown in the Sorting area lists.
-
- Note:** If you click on a column header after releasing the Shift key, all sorting information is lost and the selected column is the new primary sorting criteria.
7. To set secondary and tertiary sorting, hold the Shift key and repeat from step 5.
 8. Release the Shift key.
 9. Click **OK**.

Filter alarms

You can filter current and historical alarms by using query filters. A query filter is a collection of filter criteria in a logical construct.

For example, you can filter alarms by defining a query filter that only shows alarms with priorities larger than 500 and smaller than 750.

You can re-use the filter queries you define for historical alarms for current alarms and vice versa. You can also re-use filter queries you define at design-time at run time and vice versa.

Important: Query filters for current alarms and recent alarms and events require at least **Provider** and **Group** as filter criteria. These must use the equal sign.

When you use TimeLCT, TimeOAT, or TimeLCTOAT as filter criteria for historical alarm modes, you must ensure that the TimeSelector.StartDate and TimeSelector.EndDate properties do not limit the query. Otherwise, the AlarmApp can possibly not return all alarm and event records.

Set the TimeSelector.StartDate property earlier than any time filtering requirement, and the TimeSelector.EndDate later than any time filtering requirement.

Wildcards in queries

In current alarm queries, you can use wildcards only in the Tagname part of the query and not in the Provider, Group, or Node part of the query. A valid example is:

\galaxy!Mixing!RotorBlade*

In query filters that are used for historical queries, you must convert the operator and wildcard to SQL syntax according to the following table:

	Current Query	Historical Query
Operator	=	Like

Wildcard	*	%
----------	---	---

For example:

Provider = 'galaxy' AND Group = 'Mixing' AND Name Like 'RotorBlade%'

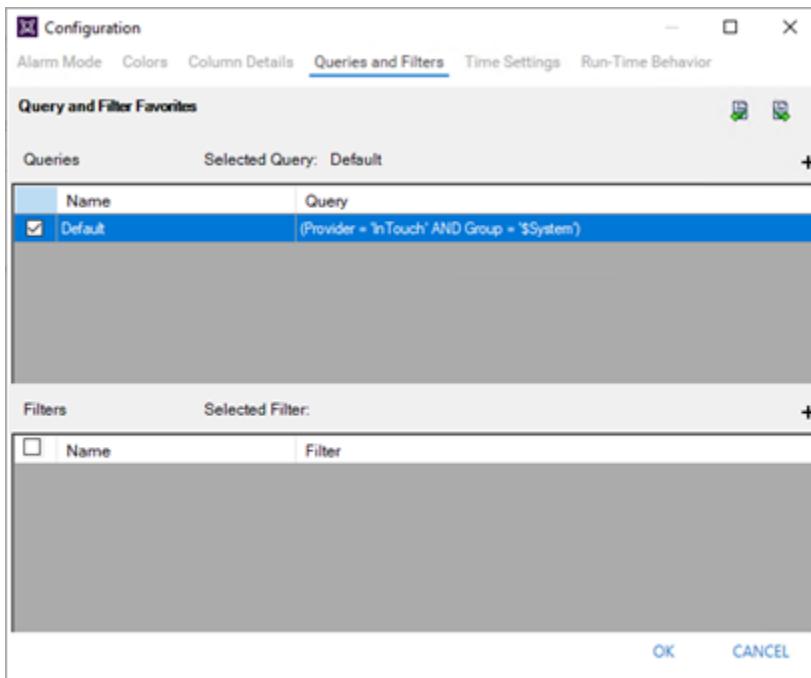
If you want to use a query filter containing a wildcard for a current query and a historical query, create two separate query filters.

Use an existing query

You can use an existing query filter to filter the alarms shown in the AlarmApp.

To use an existing query filter

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Queries and Filters** tab. The **Query and Filter Favorites** page appears.



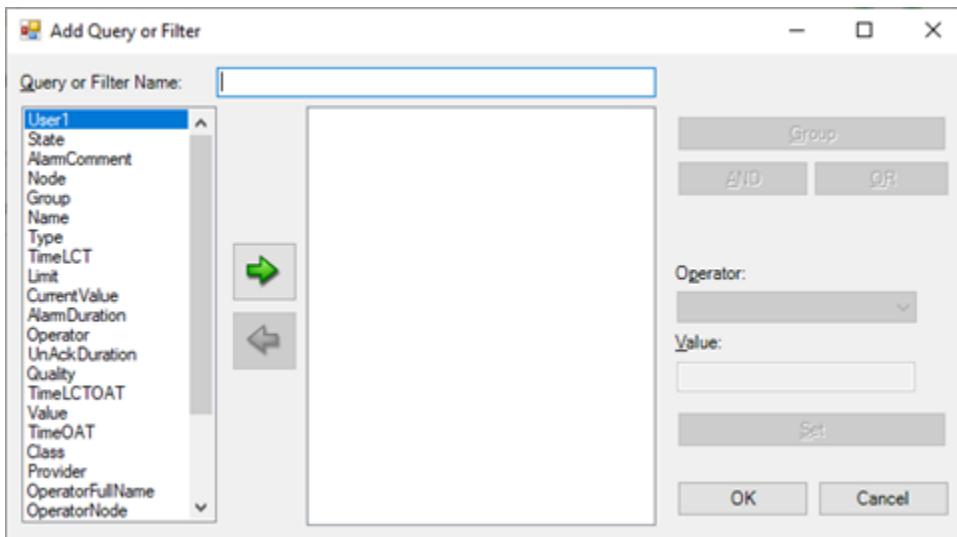
5. In the **Queries** list, select a query filter.
6. Click **OK**.

Add a new query filter

You can define a new query filter to filter the alarms shown in the AlarmApp. The new query filter is saved as a favorite in the **Query Filter Favorites** list.

To add a new query filter

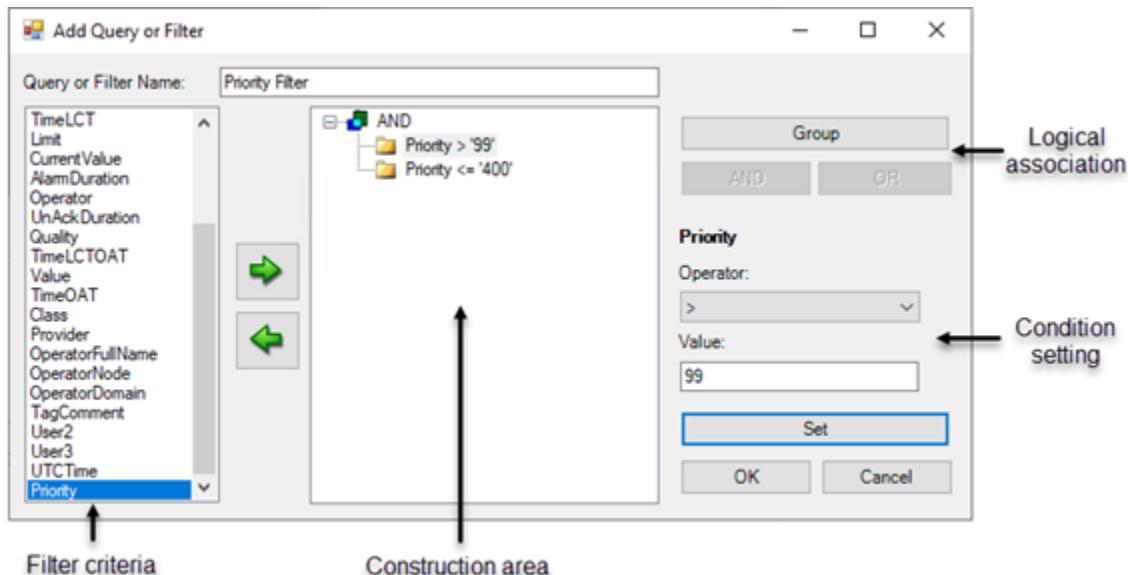
1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field.
4. Click the **Queries and Filters** tab. The **Query and Filter Favorites** page appears.
5. Select the **Add** button at the right of the Queries area. The **Add Query or Filter** dialog box appears.



6. Construct queries or filters as needed.

Construct filters

You use the **Add Query or Filter** dialog box to create or edit a filter graphically.



To construct a filter

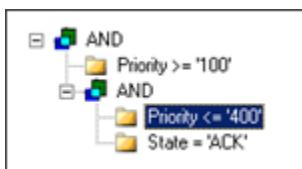
1. If you want to change the filter name, type a new unique name in the **Filter Name** box.

2. Add filter criteria to the construction area by selecting a column name on the left and clicking the right arrow button. When you add filter criteria to the construction area, they are automatically logically connected by AND.



3. If necessary, remove filter criteria by selecting them in the filter construction area and clicking the left arrow button.
4. To change the logical operator, select it in the filter construction area, and then either:
 - Click **AND** or **OR**
 - Right-click and select **AND** or **OR** from the shortcut menu
5. To group filter criteria logically, either:
 - Drag a filter criteria in the construction area over another filter criteria
 - Select one filter criteria, click **Group**, and then click the other filter criteria

By default, the filter criteria are logically grouped with **AND**. If necessary, you can select the **AND** item in the tree and click **OR** to change it to an OR grouping.



6. Assign values to filter criteria.

Note: If you are using the **Value** column as a filter criteria, you may get unexpected results at run time. The items in the **Value** column are sorted alphabetically, not numerically. This is because the **Value** column can contain strings.

Do the following:

- a. Select a filter criteria in the construction area.
- b. Select an operator from the **Operator** list.
- c. Type or select a value in the **Value** box.
- d. Click **Set**. The filter criteria is updated in the construction area.
7. To cut, copy, or paste individual filter criteria or filter criteria branches, right-click on the filter criteria and select the appropriate option from the shortcut menu.
8. When you are done, click **OK**.

Modify a query filter

You can modify an existing query filter using the **Modify Filter** dialog box.

To modify an existing query filter

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field.

4. Click the **Queries and Filters** tab. The **Query and Filter Favorites** page appears.
5. Select an existing query filter in the **Filters** list.
6. Click the ellipsis button under the **Details** column at the far right of the list. The **Modify Query or Filter** dialog box appears. For more information, see [Construct filters](#).
7. Click **OK**.

Delete a query filter favorite

You can delete any non-default query filter favorites.

To delete a query filter favorite

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field.
4. Select the Queries and Filters tab. The **Query and Filter Favorites** page appears.
5. Select an existing query filter in the **Query Filter Favorites** list.
6. Click the **Delete** button.
7. When a message appears, click **Yes**.

Export query filter favorites

You can export the query filter favorites list to an XML file. The XML file containing the query filter favorites can be imported to another AlarmApp in design time or run time. Do not edit this file directly. The default query filter favorite is not exported to the XML file.

To export the query filter favorites list

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field.
4. Click the **Queries and Filters** tab. The **Query and Filter Favorites** page appears.
5. Click the **Export** button. The **Export Query Filter Favorites** dialog box appears.
6. Select a location and a name for the XML file and click **Save**.

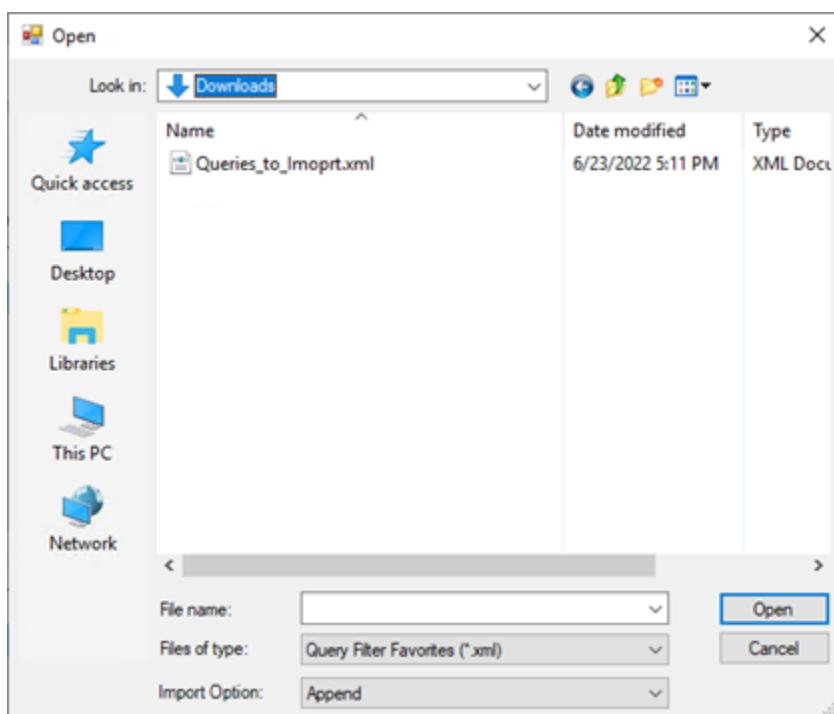
Import query filter favorites

You can import the query filter favorites list from an XML file.

To import the query filter favorites list

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field.
4. Click **Query Filters**. The **Query Filters** page appears.

5. Click **Import**. The Import Query Filter Favorites dialog box appears.



6. In the **Import Option** list, select either:

- **Append** to append the imported query filters to the existing query filters. If query filter names in the imported XML conflict with existing query filters, you are prompted to confirm the import for each filter
- **Overwrite** to replace all existing query filters with the imported query filters

7. Browse to the XML file and click **Open**.

Set time zone and format

You can set the time zone in which the AlarmApp shows the alarm and event records. By default, the time zone is set to the client computer's current time zone at design time.

You can set the time format of the alarm and event records. You can select between two different time format sets:

- Default Time Format
- .NET Time Format

Default Time Format	.NET Time Format
Format Options	.NET Time Format
%m - Two-digit month	MM - Two-digit month
%b - Three-letter month	MMM - Three-letter month
%B - Full month name	MMMM - Full month name
%d - Two digit day	dd - Two digit day
%Y - Four-digit year	yyyy - Four-digit year
%y - Two-digit year	yy - Two-digit year
%#x - Full day and date	
%H - Hours (24 hour format)	HH - Hours (24 hour format)
%I - Hours (12 hour format)	hh - Hours (12 hour format)
%M - Minutes	mm - Minutes
%S - Seconds	ss - Seconds
%s - Milliseconds	fff - Milliseconds
%p - AM or PM display	tt - AM or PM display

Set the time zone

You can set the time zone in which the AlarmApp shows the alarm and event records.

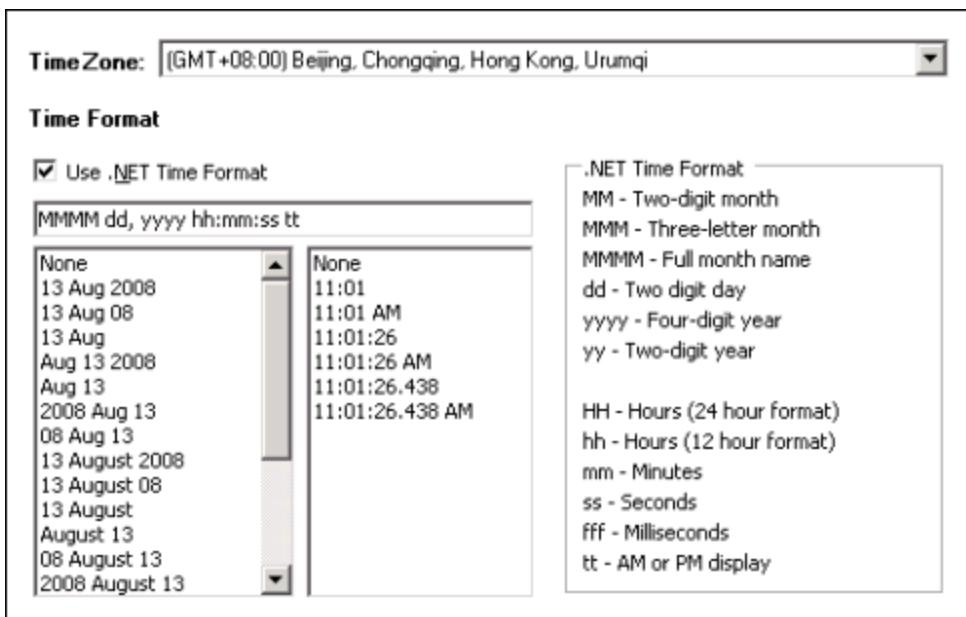
You can either set the time display to a predefined time zone, or to the client time zone. The client time zone is the time zone of the computer on which the AlarmApp is running.

The **Client Time Zone** setting is useful if you are deploying an application using the AlarmApp to a different time zone.

For example, if you develop your application in the Pacific Time zone and deploy it to two computers in the time zones Central Time and Eastern Time, you can ensure the AlarmApp shows the local time for each deployment by setting the time zone to Client Time Zone.

To set the time zone

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Time Settings** tab. The **Time Settings** page appears.
5. In the **Time Zone** field, select a time zone.



6. Click **OK**.

Set a time format

You can set the time format in which the AlarmApp shows the alarm and event records. You can either use a predefined datetime format, or compose one.

To set the time format

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field to show the AlarmApp's **Configuration** dialog box.
4. Select the **Time Settings** tab.
5. In the Time Format area, do the following:
 - a. Make sure Use .NET Time Format is cleared.
 - b. Select date format codes from the list at the right. The equivalent date format code appears in the box above.
 - c. Select time format codes from the list at the right. The equivalent time format code is appended to the date format string.

If you want to customize the datetime format, modify the codes in the box as follows:

Code	Purpose	Example
%m	Two-digit month	03
%b	Three-letter month	Mar
%B	Full month name	March

%d	Two-digit day	17
%Y	Four-digit year	2018
%y	Two-digit year	18
%#x	Full day and date	Monday, March 26, 2018
%H	Hours in 24 hour format	14
%I	Hours in 12 hour format	2
%M	Minutes	54
%S	Seconds	34
%s	Milliseconds	333
%p	AM or PM	PM

6. Click **OK**.

Set a .NET time format

You can set the .NET datetime format in which the AlarmApp shows the alarm and event records. You can either use a predefined datetime format, or compose one. The predefined date format is based on the short date format setting of the operating system and may vary from computer to computer.

To set the .NET datetime format

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the **Configuration** field to show the AlarmApp's **Configuration** dialog box.
4. Click **Time Settings**. The **Time Settings** page appears.
5. In the **Time Format** area, do the following:
 - a. Select the **Use .NET Time Format** check box.
 - b. Select date format codes from the list at the right. The equivalent date format code appears in the box above.
 - c. Select time format codes from the list at the right. The equivalent time format code is appended to the time format string.
6. If you want to customize the datetime format, modify the codes in the box as in the table below. For more information, see the Microsoft Knowledge database on .NET datetime formats.

Code	Purpose	Example
M	Single-digit month	9

MM	Two-digit month	04
MMM	Three-letter month	Mar
MMMM	Full month name	March
d	Single-digit day	3
dd	Two-digit day	03
ddd	Abbreviated day of the week	Mon
ddd	Day of the week	Monday
yyyy	Four-digit year	2018
yy	Two-digit year	18
HH	Hours in 24 hour format	13
hh	Hours in 12 hour format	10
mm	Minutes	33
ss	Seconds	34
fff	Milliseconds	234
tt	AM or PM	PM

7. Click **OK**.

Configure runtime behavior

You can configure the behavior and appearance of the AlarmApp at run time, for example:

- Showing and Hiding parts of the AlarmApp
- Specifying if the AlarmApp queries the alarm database when it starts up
- Scrolling to new alarms
- Hiding warnings, errors, and messages
- Restricting operator access to parts of the AlarmApp
- Specifying AlarmApp freeze behavior
- Customizing the "no records" message
- Configuring alarms to require an ACK signature
- Configuring alarms to require a SHELV signature
- Customizing the run-time shortcut menu

Show heading, grid, or status bar

You can show and hide parts of the AlarmApp at run time, such as the heading, grid, or status bar.

User1	State	Node	Group	Name
3	UNACK_RTN	RM-NGL...	Area_001	UserDefined_001.UDA4
2	UNACK_RTN	RM-NGL...	Area_001	UserDefined_001.UDA3
1	UNACK_RTN	RM-NGL...	Area_001	UserDefined_001.UDA2
1	UNACK_RTN	RM-NGL...	Area_001	UserDefined_001.UDA1
4	UNACK_RTN	RM-NGL...	Area_001	UserDefined_001.UDA6
4	UNACK_RTN	RM-NGL...	Area_001	UserDefined_001.UDA5
2	UNACK	RM-NGL...	Area_002	UserDefined_002.UDA13
2	UNACK	RM-NGL...	Area_002	UserDefined_002.UDA12
1	UNACK	RM-NGL...	Area_002	UserDefined_002.UDA11
1	UNACK	RM-NGL...	Area_002	UserDefined_002.UDA10
4	UNACK	RM-NGL...	Area_002	UserDefined_002.UDA18
4	UNACK	RM-NGL...	Area_002	UserDefined_002.UDA17
3	UNACK	RM-NGL...	Area_002	UserDefined_002.UDA16

To show the heading, grid, or status bar at run time

1. Select the AlarmApp placed on a pane.
 2. Select the **Properties** grid to show AlarmApp properties.
 3. Select the ellipsis button shown at the right of the **Configuration** field to show the AlarmApp's **Configuration** dialog box.
 4. Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.
 5. To show or hide parts of the AlarmApp during run time, do any of the following:
 - Select the **Show Heading** check box to show the heading at run time, or clear it to hide the heading at run time.
 - Select the **Show Grid** check box to show the grid at run time, or clear it to hide the grid at run time.
 - Select the **Show Status Bar** check box to show the status bar at run time, or clear it to hide the status bar at run time.
- Important:** If you hide the status bar, you will not be able to see important indicators, such as the New Alarms, Hidden Alarms, and Frozen Grid indicators.
6. Click **OK**.

Automatic query for alarms on startup

You can configure the AlarmApp to automatically query the Alarm Database when the control starts up at run time.

By default, current alarms and recent alarms and events are automatically queried when the AlarmApp starts at run time. You can disable the automatic query if the AlarmApp is:

- Configured to mainly use query filters
- Controlled mainly by scripts

To query the Alarm Database automatically on start up

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.
5. Select the **Query on Startup** check box.
6. Click **OK**.

Scroll automatically to new alarms

If an operator views multiple pages of alarms, new alarms may go unnoticed. You can configure the AlarmApp to scroll automatically to show new alarms. Use the AutoScroll Boolean property in scripts to scroll automatically to new alarms.

However, if the AlarmApp scrolls automatically to new alarms, it may be hard for the operator to view and analyze older alarms if new alarms occur. If the AlarmApp is frozen, it will not scroll automatically to new alarms.

To scroll automatically to new alarms

1. Select the AlarmApp placed on a pane.
2. Select the Properties grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.
5. Select the **Auto Scroll to New Alarms** check box.
6. Click **OK**.

Hide errors, warnings, and status messages

You can prevent a message dialog box from opening when errors, warnings, or status messages occur in the AlarmApp. Even if you hide errors, warnings and status messages are sent to the Operations Control Logger.

To hide error and warning messages

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.
5. Select the **Hide Errors and Warnings** check box.
6. Click **OK**.

Restrict user access to rows and columns

You can prevent the operator from:

- Resizing columns
- Selecting rows
- Selecting multiple rows

Use this feature for interfaces where it is easy to accidentally resize columns or select rows.

To prevent the operator from resizing columns

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.
5. Clear the **Allow Column Resizing** check box.
6. Click **OK**.

To prevent the operator from selecting rows

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.
5. In the **Row Selection** list, click:
 - **No** to prevent operator from selecting rows
 - **Single** to allow operator to only select one row
 - **Multiple** to allow operator select multiple rows
6. Click **OK**.

Retain hiding when changing the alarm query filter

You can configure the AlarmApp to hide alarms even if the alarm query filter changes.

To retain hiding when change the alarm query filter

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.
5. Select the **Retain Hidden** check box.

6. Click **OK**.

Override a frozen grid

You can configure the AlarmApp to unfreeze the grid after a given time in seconds. Use this option to make sure that new alarms appear on the grid after a specified time.

The AlarmApp also unfreezes if you change one of the following:

- Alarm Mode
- Alarm Query
- Query Filter

To override the frozen grid

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.
5. Select the **Auto Resume** after check box and type the number of seconds after which the grid unfreezes.
6. Click **OK**.

Customize the no records message

You can customize the message that appears when there are no records to show in the grid.

To customize the "no records" message

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.
5. Select the **Show Custom 'No Records' Message** check box and type a message you want to show in the AlarmApp when there are no alarm records.
6. Click **OK**.

Configure the AlarmApp to require an ACK signature

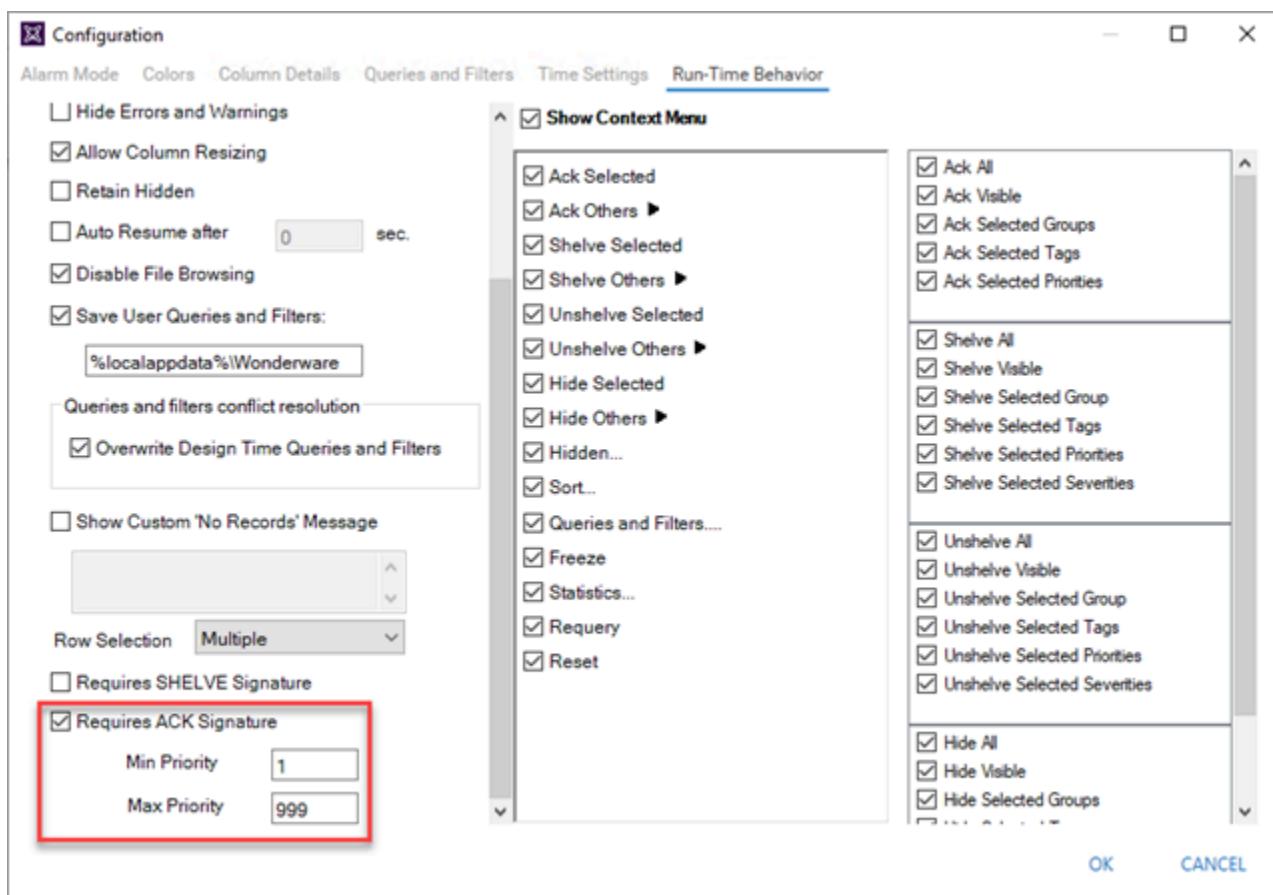
Even if you are logged into the AVEVA OMI ViewApp, acknowledging the alarms that fall within a specific priority range may need to be authenticated by you if required by your company or industry. You can configure the Alarm Control to provide such functionality by selecting the **Requires ACK Signature** check box in the **Object Properties** dialog of the Alarm Control.

After the Alarm Control has been configured to require a signature for acknowledgement, you need to set its minimum and maximum priority values. The valid range of the minimum and maximum priority values is 1 to

999.

To configure the Alarm Control to require ACK signature

1. Select the Alarm Control placed on a pane.
2. Select the **Properties** grid to show Alarm Control properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the Alarm Control's **Configuration** dialog box.
4. Click the **Run-Time Behavior** tab. The **Run-Time Behavior** page appears.



5. Select the **Requires ACK Signature** check box. (You may need to scroll the left column to make it visible.) The **Min Priority** and **Max Priority** boxes are enabled.
6. Enter the minimum and maximum priority values for the alarm range window that will require authentication.
7. Click **OK**.

You can configure the alarm signature requirement as well as the minimum and maximum values in run time using the Object Editor.

Configure the AlarmApp to require a SHELVE signature

Operators can temporarily shelve selected alarms from the list of an AlarmApp's active alarms. A shelved alarm is suppressed and removed from the list of active alarms. Typically, operators shelve lower severity nuisance alarms

because they provide little diagnostic value and interfere with the operator's ability to manage a plant process. An alarm is shelved for a specified period. After the period ends, alarms are automatically unshelved and appear again in the list of active alarms. Operators can also manually unshelve an alarm before the end of the specified shelved period.

By default, Medium and Low severity alarms are enabled for shelving. Critical and High severity alarms are not because of the potential risk of shelving and ignoring alarms that represent serious operating states.

The AlarmApp provides a configuration option to restrict alarm shelving only to operators authorized by application security. When an operator selects one or more alarms and attempts to shelve them, the AlarmApp verifies if a shelve signature is required.

- If no shelve signature is required

The AlarmApp shows a simple **Shelve Comment** dialog box with a **Duration** field to select the number of hours (1, 2, 4, 8, 12) to shelve the alarm and a **Reason** field to enter a mandatory comment.

- if a shelve signature is required and application security is active

The AlarmApp shows a Shelve Comment dialog box with fields for the operator to enter user credentials (name, password, domain) in addition to selecting a shelve duration and entering a mandatory comment.

By default, the logged-in user appears in the Username field. If the application security type is ArchestrA, then ArchestrA appears in the Domain field and cannot be edited. If the credentials are valid, the AlarmApp attempts to shelve the selected alarms.

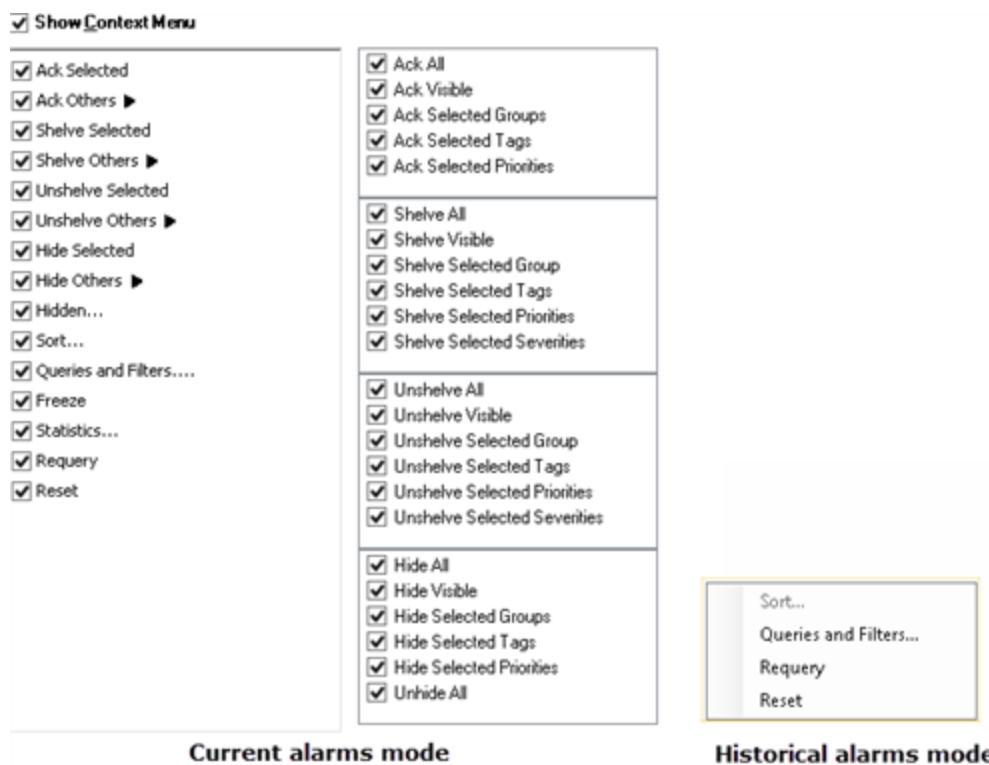
The AlarmApp shows an error message if the operator enters invalid credentials. When the operator clicks OK on the error message, the Shelve Comment dialog appears again with the entered user name, comment, and duration. The Password (or PIN) is blank. The operator can attempt to authenticate again or cancel.

To configure the AlarmApp to require a SHELVE signature

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Select or clear **Requires SHELVE Signature** based on whether operators need to enter their credentials to shelve alarms or not.
5. Click **OK**.

Configure the runtime shortcut menu

You can configure the AlarmApp's shortcut menu to show only selected options at runtime. The shortcut menus showing historical alarms (or events) and current alarms (or recent alarms and events) are different.



For the current alarms Context menu, you can also show or hide entire shortcut submenus.

To hide the Context menu

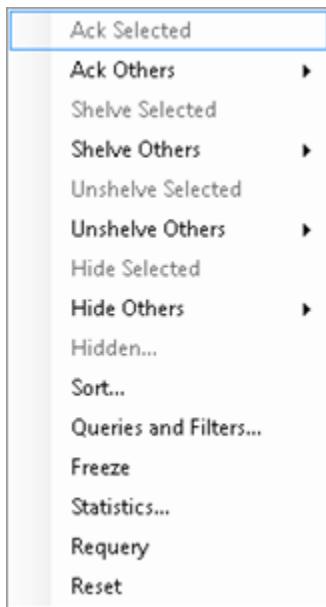
1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.
5. Clear the **Show Context Menu** check box.
6. Click **OK**.

To show or hide Context menu options

1. Select the AlarmApp placed on a pane.
2. Select the **Properties** grid to show AlarmApp properties.
3. Select the ellipsis button shown at the right of the Configuration field to show the AlarmApp's **Configuration** dialog box.
4. Click **Run-Time Behavior**. The **Run-Time Behavior** page appears.
 - a. Select the options you want to appear on the run-time Context menu (if applicable for the selected client mode).
 - b. Clear the options you want to hide from the operator on the run-time Context menu.
5. Click **OK**.

Use the AlarmApp at runtime

During run time, users can right-click on the AlarmApp to show a shortcut menu containing commands to monitor and manage alarms. Also, other shortcut options enable users to sort and filter alarms shown by the AlarmApp.



Note: Many of the commands shown in the shortcut menu can be shown on the command bar as large icons that can be easily selected on touch screen devices. For more information about integrating a command bar in an AlarmApp, see [Show Command Bar property](#).

Refresh the AlarmApp grid

You can refresh the AlarmApp grid at run time. The AlarmApp retrieves alarm record data based on the Alarm Database time range settings.

To refresh the AlarmApp

1. Right-click the AlarmApp grid at run time. The shortcut menu appears.
2. Click **Requery**.

View status bar information

The status bar shows you information about the current AlarmApp grid. Depending on the current AlarmApp mode (real time, historical), the status bar information shows different information.

View status bar information of current modes

If the AlarmApp is showing current alarms or recent alarms and events, the status bar shows the following:



Element	Icon(s)	Description
Client Mode		Indicates the AlarmApp is showing the current alarms (or recent alarms and events).
New Alarms		Appears if new alarms have occurred. If you move the pointer over the indicator, the tooltip shows how many alarms are unacknowledged.
Hidden Alarms		Appears if any alarms are currently hidden. If you move the pointer over the indicator, the tooltip shows how many alarms are hidden.
Frozen Grid		Appears if the AlarmApp is currently frozen
Alarm Records		Shows the total number of alarm records and which alarms are currently shown.
Query Filter		Shows the name of the current query filter favorite.
Retrieval		Shows the percentage of alarms retrieved from all alarm providers. If this percentage is less than 100%, not all alarm providers are providing alarm data. Use the Alarm Statistics dialog box to detect which alarm providers are not providing alarm data.
Time Zone		Shows the current time zone of the AlarmApp. Move the pointer over the time zone to show the full information in a tool tip.

View status bar information of historical modes

If the AlarmApp shows historical alarms or events, the status bar shows the following fields:

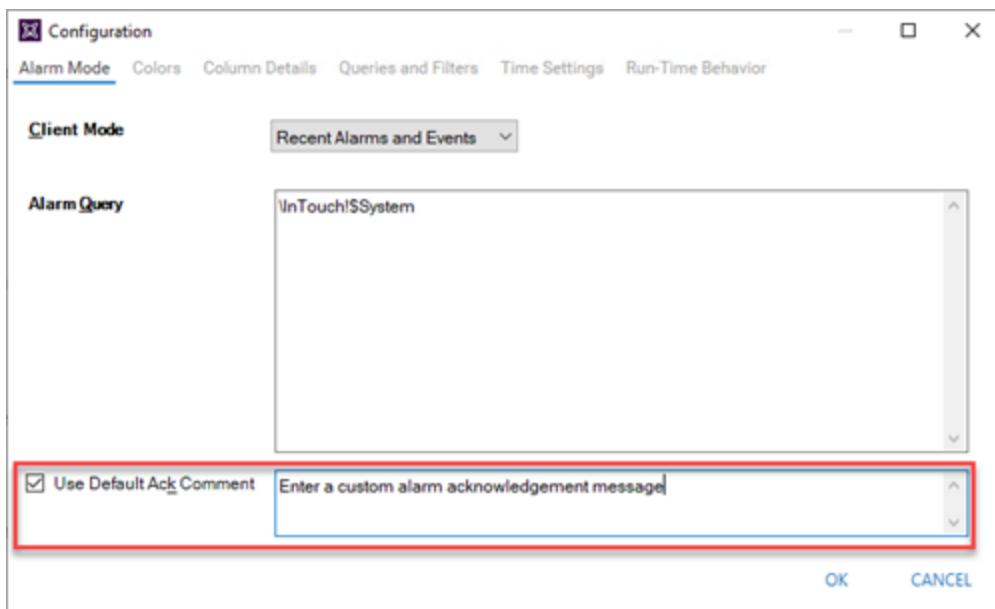
Element	Description
Client Mode	

Element	Description
	Indicates the AlarmApp is showing historical alarms and/or events.
Alarm Records	Displaying 1 to 13 of 28 alarms Shows the total number of alarm records and the number of alarms shown by the AlarmApp.
Alarm Database	localhost - WWAlmDb Shows the name of the server hosting the Alarm Database and the Alarm Database name.
Connection Status	Connected Shows the connection status to the Alarm Database.
Time Zone	Beijing, Hong Kong, Uramqi Shows the current time zone of the AlarmApp. Move the pointer over the time zone to show complete time zone information in a tool tip.
Requery	 Click this button to retrieve latest alarm records from the Alarm Database.

Acknowledge alarms

You can configure the AlarmApp to require an alarm to be acknowledged even if the condition causing the alarm has passed. This ensures that an operator is aware of events that caused a temporary alarm state, but have returned to normal. You acknowledge alarms at run time using a shortcut menu or by script methods.

You can also configure a comment in the **Configuration** dialog box to use when alarms are acknowledged at run time. Select the **Use Default Ack Comment** check box to enable a default comment, and enter the comment string in the text box to automatically send the acknowledgement comment when a user acknowledges an alarm at run time. The default acknowledgement comment cannot be enabled from the **Properties** tab.



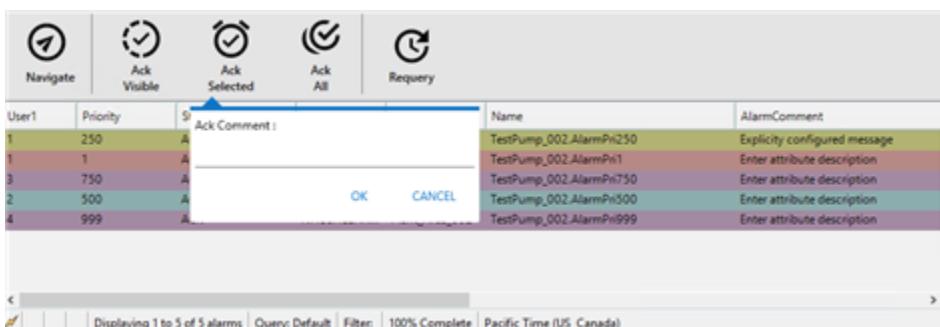
When **Use Default Ack Comment** is disabled, the **Ack Comment** popup is always displayed when a user acknowledges an alarm. You can configure the Ack Comment to automatically fill in the attribute description as the comment by enabling the **Use Description as Comment** Run-Time Behavior property in the **Properties** tab. You can acknowledge alarm records directly from the AlarmApp. You can acknowledge:

- One or more selected alarms
- All alarms, including alarms not visible due to the limited space of the AlarmApp
- All visible alarms
- All alarms with common values, such as provider names, group names, priority ranges, and tag names.

To acknowledge selected alarms using the AlarmApp grid

1. Select one or more alarms in alarm state.
2. Right-click on an alarm shown in the AlarmApp and click **Ack Selected**.

If no default acknowledgement statement is configured for the AlarmApp, the **Ack Comment** popup appears.



Note: If the the **Use Description as Comment** property is enabled, the **Ack Comment** popup will contain the message, "Leave blank to use Attribute Description as Ack Comment."

3. Type an alarm acknowledgement comment and click **OK**.

To acknowledge other alarms using the AlarmApp grid

1. Select one or more alarms in alarm state.
2. Right-click the AlarmApp grid, point to **Ack Others**, and click one of the following:
 - **Ack All** to acknowledge all alarms in alarm state Ack Visible to acknowledge all visible alarms
 - **Ack Selected Group** to acknowledge alarms with the same provider names and group names of one or more selected alarms in alarm state.
 - **Ack Selected Tag** to acknowledge alarms with the same provider names, group names, and tag names within the priority ranges of one or more selected alarms in alarm state.
 - **Ack Selected Priority** to acknowledge alarms with the same provider names, group names, and within the priority ranges of one or more selected alarms in alarm state.
3. If no default acknowledgement statement is configured for the AlarmApp, the **Ack Comment** popup appears.
4. Type an alarm acknowledgement comment and click **OK**.

Acknowledge alarms with signature

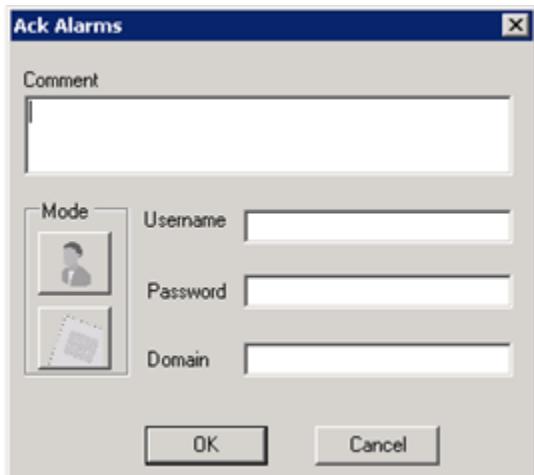
You must provide your signature in the **Ack Alarms** dialog box to acknowledge alarms. A signature is required when the **Require ACK Signature** check box is selected and any of the selected alarms falls within the configured priority range, or if no one is logged on to the ViewApp. You are not required to be logged on to the ViewApp to acknowledge alarms. However, if you are logged on, the **Username** box displays the username.

When you select the **Require ACK Signature** check box, the acknowledgment comment is prefixed in the updated Alarm Record. If one or more of the selected alarms falls within the configured priority range, the comment is prefixed with "Signed ACK -" indicating that it is a signed acknowledgment. Otherwise, it is prefixed with "Std ACK -" indicating that it is a standard acknowledgment.

Note: If the selected alarms do not require a signature, then the **Ack Alarms** dialog box displays the **Comment** box. You can enter a comment before acknowledging the alarm.

To provide a signature to acknowledge alarms with user name

1. Select one or more alarms in the alarm state.
2. Right-click the AlarmApp grid and click **Ack Selected**.
3. If the selected alarms require a signature, or if you are not logged on to InTouch, then the **Ack Alarms** dialog box appears. If the Smart Card authentication system is not configured on your computer, the following dialog appears:



4. In the **Comment** box, enter or modify the alarm comment.
5. In the **Username** box, enter your user name.
6. In the **Password** box, enter your password.
7. In the **Domain** box, enter the domain and click **OK**.

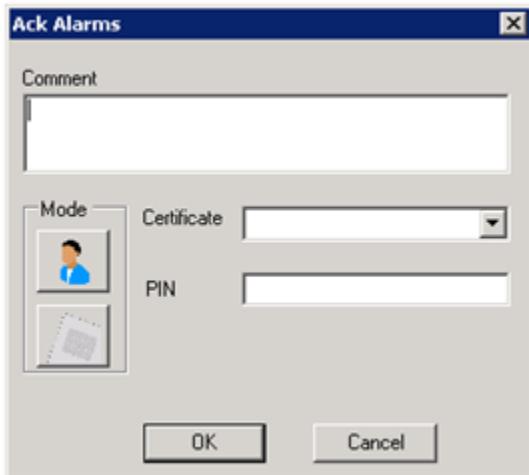
Note: If you enter invalid credentials, the AlarmApp displays an error message. When you click **OK**, the **ACK Alarms** dialog box appears again with the user name and the comment you had provided. You must enter the correct password.

Acknowledge alarms using a Smart Card

You can use a Smart Card to provide the authentication to acknowledge alarms if your computer supports Smart Card authentication. In that case, the AlarmApp displays the Ack Alarms dialog box with the Smart Card authentication dialog. You must have the Smart Card inserted in the Smart Card reader attached to your computer.

To provide a signature to acknowledge alarms using Smart Cards

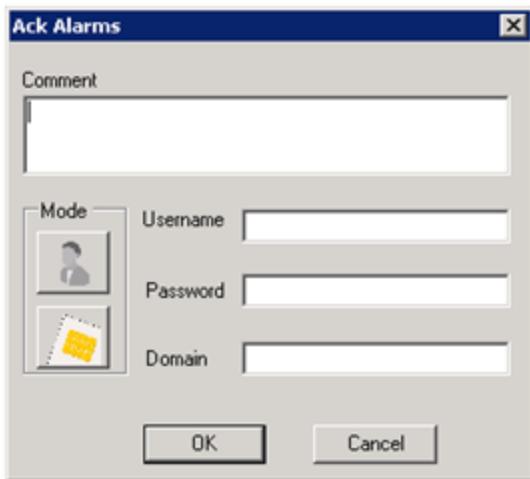
1. Select one or more alarms in an alarm state.
2. Right-click the AlarmApp grid and click **Ack Selected**.
3. If the selected alarms are configured to require a signature and if Smart Card Authentication is selected in InTouch, the **Ack Alarms** dialog box appears where the Smart Card button under **Mode** is disabled.



4. In the **Comment** box, enter or modify the comment.
5. From the Certificate list, select the Smart Card currently inserted into the reader in your system.

Note: If a card is newly-inserted or removed from the reader, you can update the list of cards by selecting the appropriate Smart Card from the Certificate list.

6. In the **PIN** box, enter the personal identification number and click **OK**.
 - a. If you enter an invalid PIN, the system displays an error message. When you click **OK**, the ACK Alarms dialog box appears again with the user credential and the comment you had provided. You must enter the correct PIN.
 - b. If you need to provide your log on credentials instead of the Smart Card details to verify your signature, click the **User Name Authentication** button under **Mode**. A different version of the **Ack Alarms** dialog box appears where the Smart Card button under **Mode** is enabled.



Dismiss alarms at runtime

When an alarm occurs, the run time operator (or system) can acknowledge the alarm to indicate that they are aware of the alarm. An acknowledged alarm that has returned to a normal value will continue to be displayed if the latching feature has been enabled for the Galaxy. With latching enabled, an acknowledged alarm that has returned to normal condition will be placed in the LATCHED alarm state.

LATCHED alarms can be displayed in the current alarms mode to show that the alarms did occur. Alarms go to the LATCHED state if:

- You ACK an alarm from UNACK_RTN state.
Or
- You return an alarm from ACK state.

To view the LATCHED state, you have to enable the LATCHED state under the global settings. For more information, see the "Enable a Latched State" in the AVEVA OMI help.

You can dismiss the LATCHED alarms to remove the LATCHED alarms from the current mode of the Alarm Client Control grid. The dismissed LATCHED alarms would be visible in the recent mode of the Alarm Client Control.

To dismiss selected alarms

1. Select one or more alarms in **LATCHED** state.
2. Right-click anywhere on the grid and select **Dismiss Selected**.

The selected alarm is removed from the grid.

To dismiss other alarms

1. Select one or more alarms in **LATCHED** state.
2. Right-click anywhere on the grid, point to **Dismiss Others**, and select one of the following options:
 - **Dismiss All** to dismiss all alarms in alarm state
 - **Dismiss Visible** to dismiss all visible alarms
 - **Dismiss Selected Groups** Groups to dismiss alarms with the same provider names and group names of one or more selected active alarms

- **Dismiss Selected Tags** to dismiss alarms with the same provider names, group names, and tag names of one or more selected active alarms

The relevant alarms are removed from the grid.

Shelve and unshelve alarms at runtime

Operators can shelve alarms to temporarily suppress them for a fixed period. Shelving an alarm means temporarily removing it from the AlarmApp's main alarm list and placing it on a shelved list. Shelving is normally controlled by an operator to handle irrelevant nuisance alarms that have not been caught by filtering or alarm suppression mechanisms.

When shelving an alarm from the AlarmApp, operators set an associated time period in which the alarm remains shelved and enter a mandatory comment. Operators can select from a list of AlarmApp Context commands during run time to:

- Shelve one or more selected alarms
- Shelve all alarms
- Shelve only those alarms visible in the AlarmApp
- Shelve all alarms within the same alarm group as an alarm selected from the AlarmApp
- Shelve alarms by selected tags or attributes
- Shelve alarms by selected alarm priorities
- Shelve all alarms that have the same severity as an alarm selected from the AlarmApp

When application security is used, alarms can be shelved and unshelved only by operators with proper authorization. For more information about setting shelving authorization, see [Configure the AlarmApp to require a SHELVE signature](#)

Shelved alarms are automatically unshelved at the end of the specified time period. Operators can also manually unshelve alarms and return them to an active state. Alarms are unshelved automatically at the end of the shelving time period. An unshelved alarm reappears in the AlarmApp active list and resumes its state at the time it was shelved. Operators can manually unshelve a shelved alarm before the end of the shelved period and enter an optional comment.

Shelve alarms

You can shelve alarms directly from the AlarmApp by selecting commands from the Controls menu of the AlarmApp.

To shelve selected alarms using the AlarmApp

1. Select one or more alarms in alarm state.
2. Right-click the AlarmApp grid and click **Shelve Selected** from the shortcut menu.

The **Shelve Comment** dialog box appears to set a shelf duration and enter a comment.



3. Select an alarm shelving duration from the **Duration** field.

To shelve other alarms using the AlarmApp

1. Select one or more alarms in alarm state.
2. Right-click the AlarmApp, point to **Shelve Others**, and click one of the following shortcut commands:
 - **Shelve All** to shelve all alarms in alarm state
 - **Shelve Visible** to shelve all visible alarms
 - **Shelve Selected Groups** to shelve alarms with the same provider names and group names of one or more selected active alarms
 - **Shelve Selected Tags** to shelve alarms with the same provider names, group names, and tag names of one or more selected active alarms
 - **Shelve Selected Priorities** to shelve alarms with the same provider names, group names, and within the same priority ranges of one or more selected active alarms
 - **Shelve Selected Severities** to shelve alarms with the same provider names, group names, and within the same severity ranges of one or more selected active alarms

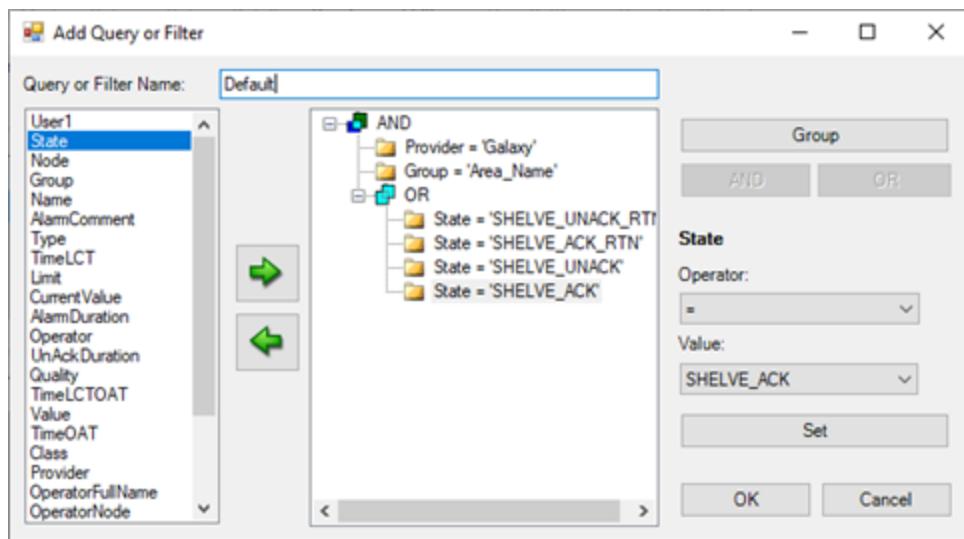
The **Shelve Comment** dialog box appears with fields to set a shelf duration and enter a comment.

3. Select an alarm shelving duration from the **Duration** field.
4. Type a mandatory alarm shelving comment in the **Reason** field and click **OK**.

Show shelved alarms

After alarms are shelved, they no longer appear as active alarms in the AlarmApp. The default query of the AlarmApp does not show shelved alarms. You must configure an AlarmApp query filter similar to the following example to show shelved alarms.

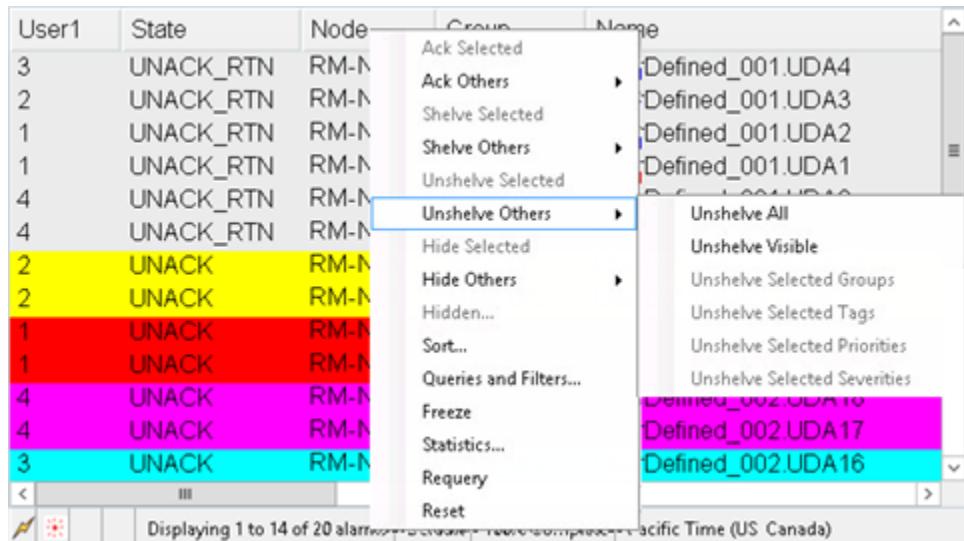
(Provider = 'Galaxy' AND Group = 'Area_Name' AND (State = 'SHELVE ACK RTN' OR State = 'SHELVE_UNACK' OR State = 'SHELVE UNACK RTN' OR State = 'SHELVE ACK'))"



For more information about configuring queries to filter alarms, see [Filter alarms](#).

Unshelve alarms

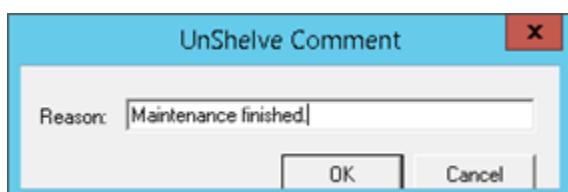
Manually unshelving alarms follows a similar sequence of steps to shelve alarms using Context menu commands. You should have created a query that shows the current shelved alarms.



To unshelve selected alarms using the AlarmApp

1. Select one or more shelved alarms.
2. Right-click the AlarmApp grid and click **Unshelve Selected** from the Context menu.

The **Unshelve Comment** dialog box appears.



3. Type an optional alarm unshelving comment in the **Reason** field and click **OK**.

To unshelve other alarms using the AlarmApp

1. Select one or more alarms in alarm state.
2. Right-click the AlarmApp, point to **Unshelve Others**, and click one of the following shortcut commands:
 - **Unshelve All** to unshelve all shelved alarms
 - **Unshelve Visible** to unshelve all visible shelved alarms
 - **Unshelve Selected Groups** to unshelve alarms with the same provider names and group names of one or more selected shelved alarms
 - **Unshelve Selected Tags** to unshelve alarms with the same provider names, group names, and tag names of one or more selected shelved alarms
 - **Unshelve Selected Priorities** to unshelve alarms with the same provider names, group names, and within the same priority ranges of one or more selected shelved alarms
 - **Unshelve Selected Severities** to unshelve alarms with the same provider names, group names, and within the same severity ranges of one or more selected shelved alarms

The **Unshelve Comment** dialog box appears.

3. Type an optional alarm unshelving comment in the **Reason** field and click **OK**.

Sort alarms at runtime

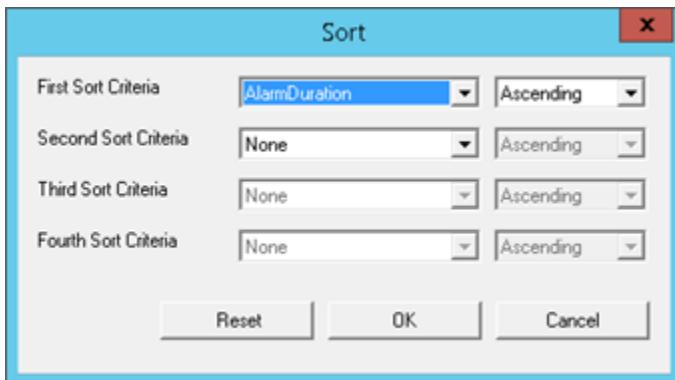
The AlarmApp supports alarm sorting for up to three columns at design time and run time. At run time, operators can configure sorting of even more columns by clicking on the column headers of the Alarm ControlAlarmApp. You can sort alarms in ascending or descending direction for selected columns.

You can sort alarms at run time in similar way as design time. Any changes you make to the sorting at run time are lost when you switch back to design time.

Note: If you are sorting by the Value column, the items in the column are sorted alphabetically, not numerically. This is because the Value column can contain strings.

To set sorting columns and directions with lists at run time

1. Right-click the AlarmApp grid and click Sort. The Sort dialog box appears.



2. Optionally, select the second sort column in the **Second Sort Criteria** list and a sorting direction in the list to its right.
3. If you set the **Second Sort Criteria**, optionally select the third sort column in the **Third Sort Criteria** list and a

sorting direction in the list to its right.

4. Click **OK**.

To set sorting columns and directions in the grid at run time

1. In the AlarmApp grid, click on a column header to set sorting for the column. An arrow appears on the column header.
2. To change the sorting direction, click on the column header again. The arrow changes direction on the header.

Note: If you click on a column header after releasing the Shift key, all sorting information is lost and the selected column is the new primary sorting criteria.

3. To set sorting for second and third columns, repeat step 3 while pressing the Shift key.
4. Release the Shift key.

Filter alarms at runtime

You can filter alarms at run time by using the filters you defined at design time.

If you did not define a filter according to your needs at design time, you can still create new filters at run time, or modify existing filters.

If you saved filters to an XML file, you can load them from a file at run-time.

Filters you define at run-time are not saved for use at design time. To re-use filters you create or modify at run-time, export the filter list to an XML file, and import the XML file at design time.

If you are showing historical alarms or events, you can use the filtering mechanism provided by the grid technology instead of using filter favorites.

Use an existing query

At run time, you can use any filter you defined at design time, regardless if you defined it for the current modes or historical modes.

To use an existing query filter

1. Right-click the AlarmApp grid and click **Query Filters**. The **Query Filters** dialog box appears.
2. Select the filter from the list and click **OK**.

The alarm records are filtered and the current filter name appears in the status bar.



Add a new query filter at runtime

At run time, you can create new queries or filters to limit the number of alarm records.

New queries or filters are saved and can be reused for future sessions. Upon close and restart of the AlarmApp window, added queries or filters will be available. For details see [Save runtime modifications to queries and filters](#)

filters.

To add a new query or filter at run time

1. Right-click the AlarmApp grid and click **Queries and Filters**. The **Queries and Filters** dialog box appears.

The configuration is the same as in design time. For more information, see [Add a new query filter](#).

Modify a query filter

At run time, you can modify a query filter. The modification of query filters is not saved for future use and is only valid for the current session. If you want to save the modifications, you must also export the query filters to an XML file.

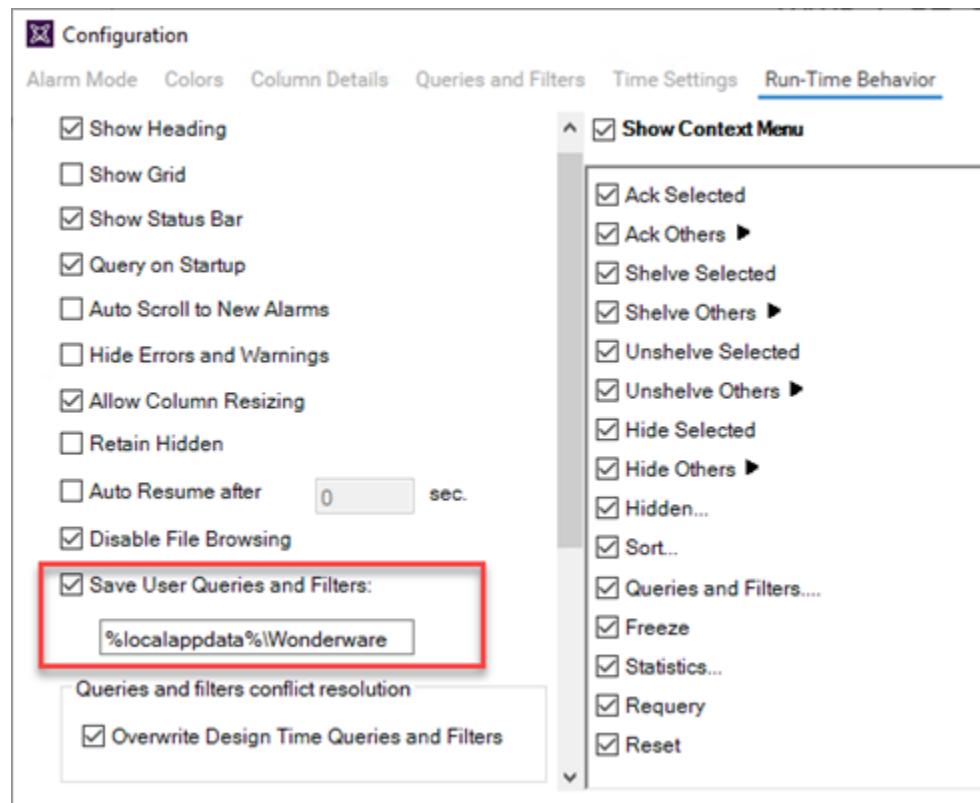
To modify a query filter

1. Right-click the AlarmApp grid and click **Query Filters**. The **Query Filters** dialog box appears.

The configuration is the same as in design time. For more information, see [Modify a query filter](#) in the Filter alarms section.

Save runtime modifications to queries and filters

You can save modifications made to queries and filters at run time. There is an option to save user queries and filters in the **Run-Time Behavior** tab of the **Edit Animations** dialog box. It is selected by default, as shown below:



When this option is selected, modifications made to queries or filters at run-time are saved on a per user and AlarmApp instance to a XML file. For example, a saved query will save as UserQueryFilter_001.xml.

Modifications are saved in the following default directory:

C:\Users\<OS Login Name>\AppData\Local\Wonderware

The map between the AlarmApp instance and the above XML file is saved in a map file, "InstancesMap.dat".

Modified queries and filters will be available upon closing and restarting the AlarmApp window or WindowViewer.

If you uncheck the **Save User Queries and Filters** option, updates made to queries and filters during run-time will not be saved.

Note: When the **Save User Queries and Filter** option is checked, the **Overwrite Design Time Queries and Filters** option is enabled and checked by default. When checked, this option resolves naming conflicts between design-time queries and filters and run-time queries and filters loaded from the XML file. Naming conflicts will be resolved by overwriting the design time queries and filters.

Delete a query filter

At run time, you can delete a query filter.

After you delete a query filter at run time, it is only deleted for the current session. If you want to save the list of query filters without the deleted query filter, you must export the query filters to an XML file. For more information, see [Export query filter favorites](#).

To delete a query filter

1. Right-click the AlarmApp grid and click **Query Filters**. The **Query Filters** dialog box appears.
2. The configuration is the same as in design time. For more information, see [Delete a query filter favorite](#).

Import query filter favorites

At run time, you can import the list of query filters from an XML file.

To import query filter favorites

1. Right-click the AlarmApp grid and click **Query Filters**. The **Query Filters** dialog box appears.
The configuration is the same as in design time. For more information, see [Import query filter favorites](#) in the Filter alarms section.

Export query filter favorites

At run time, you can export the list of query filters to an XML file for future use. After exporting, you can import the query filter from the XML into design time.

Note: The default query filter favorite is not exported to the XML file.

To export query filter favorites

1. Right-click the AlarmApp grid and click **Query Filters**. The **Query Filters** dialog box appears.
The configuration is the same as in design time. For more information, see [Export query filter favorites](#) in the Filter alarms section.

Filter alarms with client-based filtering

The grid technology used in the AlarmApp enables you to filter the grid contents after the data has been retrieved from the data source.

You can filter historical alarms and/or events in the following ways for any selected column:

Filter	Description
(All)	No filtering. All records are shown for the selected column.
(Custom)	Configure a more complex filter for the selected column. For example, a filter that can compare values of different columns.
(Blanks)	Filter by blank grid values only.
(NonBlanks)	Filter by grid values that show content.
Values	Filter by the selected value.

If a filter is applied to any column in the AlarmApp, the filter icon in the column header appears in blue.

Column header here to group by that column			
	State	Type	Class
..	ACK_RTN	LO	VA
..	ACK_RTN	HIHI	VA
..	ACK_RTN	LOLO	VA

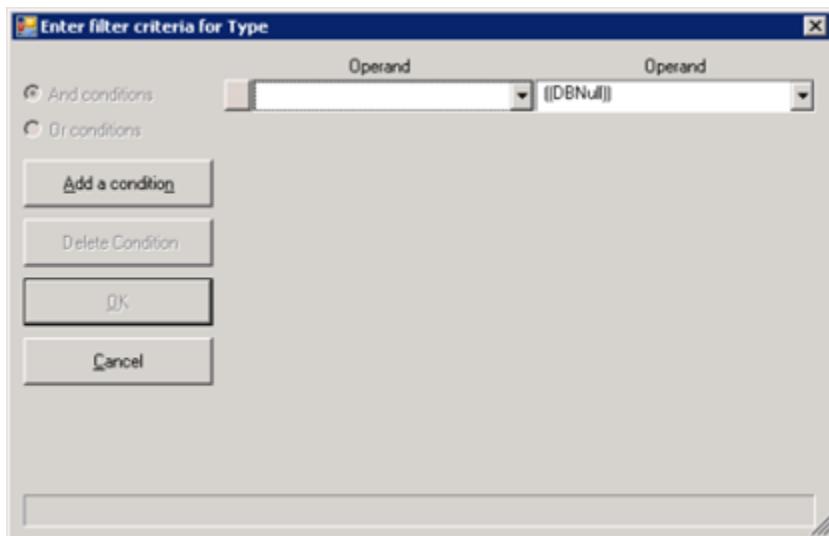
To filter alarms with client-based filtering

1. Click the filter icon on the column you want to filter by. A menu appears.



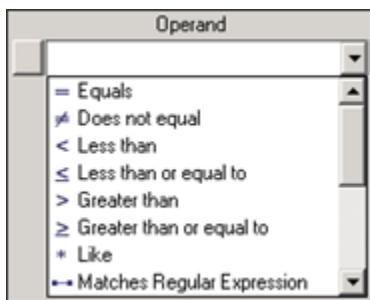
2. Select one of the following:
 - (All) to switch off filtering
 - (Custom) to define a more complex filter
 - (Blanks) to filter by blank values
 - (NonBlanks) to filter by non blank values
 - A value to filter by the value

If you selected (Custom), a dialog box appears.

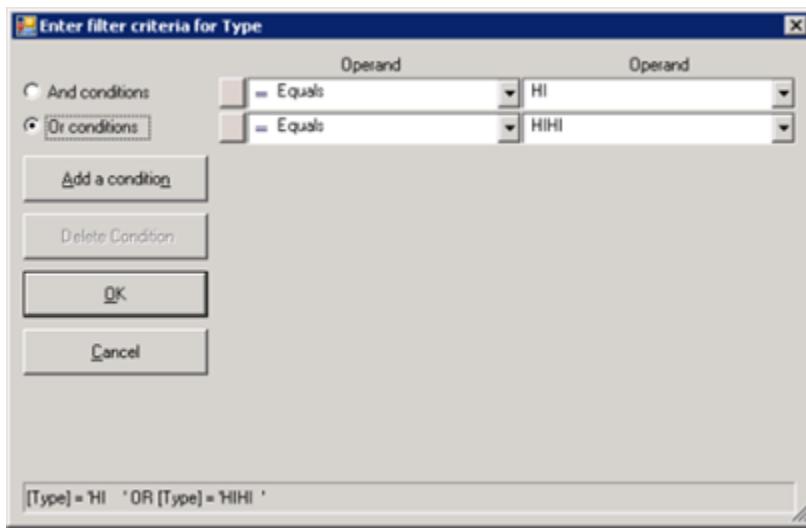


3. Do one of the following:

- Select a different operator for the current condition



- Type or select a different operand for the current condition. The operand can be a value or the different value of a different column in the same row.
- Click Add a condition to add more conditions to the filter.



- Click Delete Condition to delete one or more selected conditions (You can mark the condition by clicking on the button to the left of each condition.)

4. Click **OK**.

Reset the grid

You can reset the AlarmApp's column widths, column order, and names to their last design-time values. When you reset the grid, the query filter is also reset to its default.

To reset the grid

- Right-click the AlarmApp grid and click **Reset**.

Hide alarms

The "hiding" and "unhiding" of alarm records is known in the AlarmApp as "suppressing" and "unsuppressing".

When the AlarmApp is hiding alarms, it ignores certain alarms. If an alarm matches the exclusion criteria, it is not visible.

The actual alarm generation is completely unaffected by hiding. Alarm records are still logged into the alarm history.

You can hide:

- All alarms, including alarms not visible due to the limited space of the AlarmApp
- All visible alarms
- One or more selected alarms
- All alarms with the same provider names and group names of one or more selected alarms
- All alarms with the same provider names, group names, and within the priority ranges of one or more selected alarms
- All alarms with the same provider names, group names, and tag names within the priority ranges of one or more selected alarms

You can also view which alarms are hidden and unhide them.

To hide all alarms

- Right-click the AlarmApp grid, point to **Hide Others**, and click **Hide All**.

To hide all visible alarms

- Right-click the AlarmApp grid, point to **Hide Others**, and click **Hide Visible**.

To hide selected alarms

1. Select one or more alarms in alarm state.
2. Right-click the AlarmApp grid and click **Hide Selected**.

To hide alarms with common parameters

1. Select one or more alarms.

2. Right-click the AlarmApp grid, point to **Hide Others**, and click one of the following:
 - Hide Selected Groups to hide alarms with the same provider names and group names of one or more selected alarms
 - Hide Selected Tags to hide alarms with the same provider names, group names, and tag names within the priority ranges of one or more selected alarms
 - Hide Selected Priorities to hide alarms with the same provider names, group names, and within the priority ranges of one or more selected alarms

To unhide alarms

1. Right-click the AlarmApp grid and click **Hidden**. The **Hidden Alarms** dialog box appears.
2. Select the alarms you want to unhide and click **Unhide**.
3. Click **Close**.

Freeze and unfreeze the alarm grid

You can freeze the AlarmApp to prevent the AlarmApp tree from being updated with any further changes. For example, if new alarms occur while the AlarmApp is frozen, the new alarms are only shown after you unfreeze the AlarmApp.

You can configure a time period after which the AlarmApp automatically unfreezes to avoid the AlarmApp being unknowingly frozen. For example, the operator leaves the workstation and returns without realizing that the AlarmApp is still frozen.

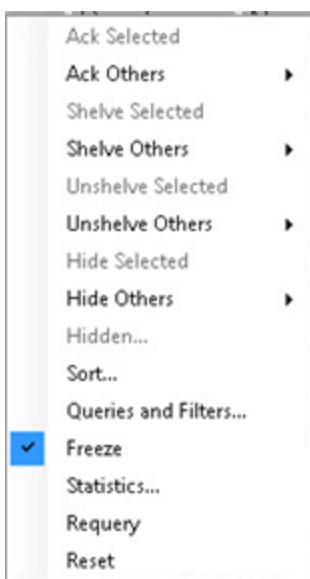
The AlarmApp unfreezes automatically if one of the following changes:

- Alarm Mode
- Alarm Query
- Query Filter

After you unfreeze the AlarmApp, the grid updates with the new alarm records and any other updates while the grid was frozen.

To freeze or unfreeze the AlarmApp grid

1. Right-click the AlarmApp grid. The shortcut menu appears.



A check mark next to the **Freeze** option indicates if the grid is currently frozen.

2. Click **Freeze**. The AlarmApp grid is either frozen or unfrozen.

Show alarm statistics

You can view alarm statistics at run time to see which alarm providers are providing the alarm data.

To show alarm statistics

1. Right-click the AlarmApp grid and click **Statistics**. The **Alarm Statistics** dialog box appears.
If you use an Alarm Hotbackup name as alarm query, you can expand the Hotbackup name in the **Alarm Statistics** dialog box to show the individual percentages of retrieval for the configured primary and backup alarm provider.
2. Click **Update** to update the statistics.
3. Click **Close**.

Set font size for the alarm grid

The font size for the alarm grid control in the AlarmApp can be set using the **FontSize** property.

User	State	Node	Group	Name	Alarm Comment	Type	TimeLCT	Limit	Current	AlarmDuration	Operator	UnAckDuration
2	UNACK...	Naus...	Platform	Platform.RAMAvailable	The Platform represents...	Syst...	11/19/2020 7:38...	100...	272...	000 00:00:...		000 00:00:...
2	UNACK	Naus...	Enterp...	Enterprise.Attribute00...	Enter attribute descr...	Lolo	11/19/2020 7:43...	10.0	0	000 00:00:...		000 00:00:...
2	UNACK	Naus...	Enterp...	Enterprise.Attribute00...	Enter attribute descr...	To	11/19/2020 7:43...	25.0	0	000 00:00:...		000 00:00:...

Typical font sizes for the grid are 18 to 20, and the font size is set by scripting.

```
MyContent.AlarmControl1.FontSize = 18;
```

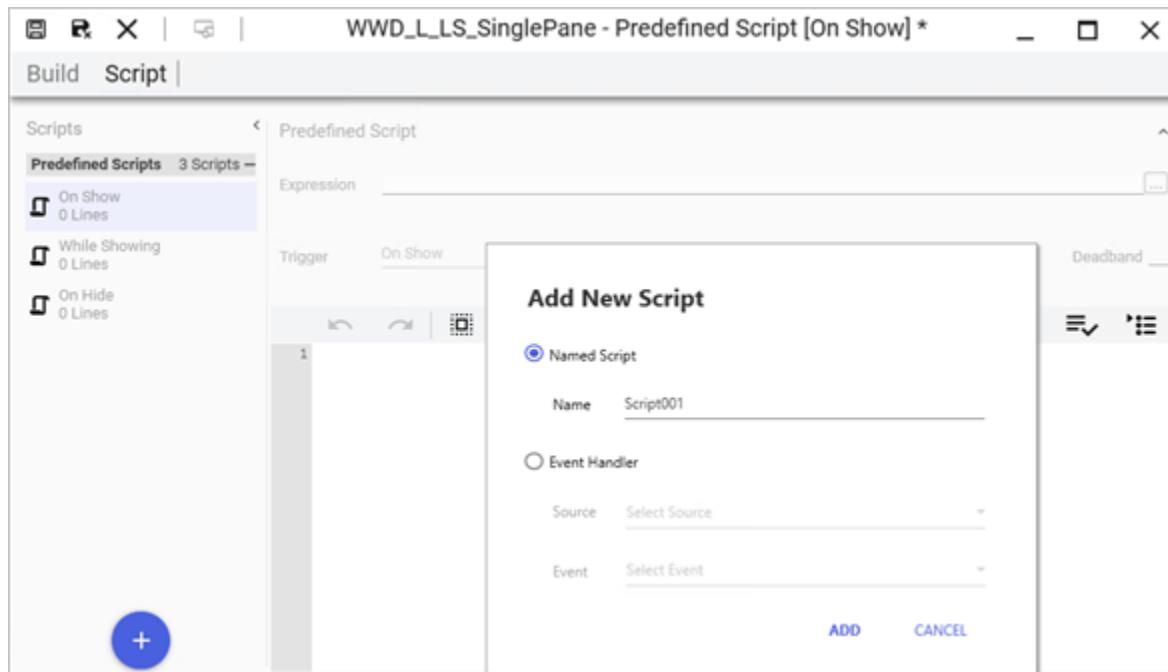
You can also set the font family through scripting, using the `FontFamily` property.

AlarmApp scripts

The AlarmApp can show the underlying .NET Winforms and Windows Presentation Foundation (WPF) control properties, methods, and events in addition to the native properties and methods included with the AlarmApp.

The underlying control properties appear on the Layout editor **Properties** grid after placing the AlarmApp or HistoricalTrendApp onto a layout pane. Because of the large number of underlying control properties, use the Layout editor's search function to find a control property.

AlarmApp methods, many properties (with certain exclusions), and events can be used by the Layout script editor to create named and event handler scripts.



For more information about writing scripts with the Layout editor, see [Build layout scripts](#).

Write a named script for the AlarmApp

You write scripts for the from the Layout editor while the AlarmApp has been placed onto a layout pane. The Layout script editor supports the MyContent namespace, which is intended for all content items placed on a layout pane that can implement named scripts.

The Layout script editor includes the AutoComplete feature, which shows a positional drop-down list of namespaces, content types, and the methods or public properties associated with AlarmApp content. Although all public properties will appear in the autocomplete list, not all of them can be used in scripts. See

Write an AlarmApp named or predefined script

1. Open the layout and show the items listed in the **Toolbox** tab.
2. Select the AlarmApp from the **Toolbox** list to show its preview thumbnail.

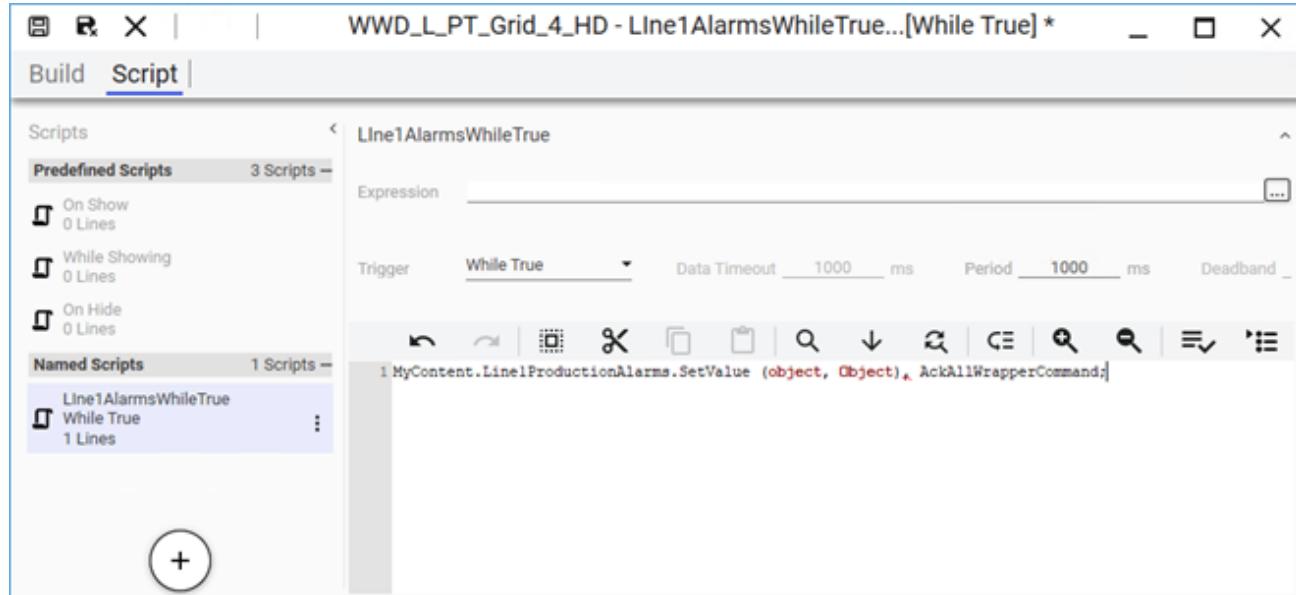
The **Toolbox** pathway to the AlarmApp is:

_Default Content > 4. Apps > AVEVA OMI Apps > AlarmApp

3. Drag and drop the preview thumbnail onto a layout pane.
4. Select **Script** from the menu bar of the Layout editor to open the script editor.
5. Create a Named script.
 - a. Select the plus circle to show the **Add NewScript** dialog box.
 - b. Select **Named Script**, assign a name to the script, and select **Add** to close the dialog.

The new script appears beneath the **Named Scripts** area at the left of the editor and the window focus is set to the script edit area to begin writing script code

6. Enter your script code.



A script intended for an AlarmApp placed on a layout pane uses the MyContent namespace and its content name is the name assigned to it from the **Name** property of the Layout editor. Place a period immediately after the item you selected from the AutoComplete list. AutoComplete shows the next positional list of items based on your previous selections in the script statement.

AutoComplete provides a list of properties and methods that can be included in the script based on the MyContent.AlarmControll prefix of the script statement.

- For .NET methods and properties, refer to .NET documentation on the Microsoft web site.
 - For descriptions of legacy AlarmApp methods and properties, refer to the *Alarm Client Control* help. The document is a PDF file located on the computer hosting the System Platform IDE.
[C:\Program Files \(x86\)\ArchestrA\Framework\Docs\1033\AlarmClientControl.pdf](C:\Program Files (x86)\ArchestrA\Framework\Docs\1033\AlarmClientControl.pdf)
7. Validate your script code by selecting the validation button at the right on the menu bar above the script edit area.
- 
8. After making any script corrections, save and exit from the Layout script editor.

Alarm client control pass-through properties

The following AlarmApp properties can be used in scripts, but are pass-through properties to the underlying Alarm Control. To maximize performance, we recommend that you set these through the underlying Alarm Client Control property when used in a layout script, using the following syntax:

```
MyContent.AlarmControl1.AlarmClientControl.AlarmQuery
```

- AlarmQuery
- AutoScroll
- ClientMode
- FlashUnAckAlarms
- GroupByHeader
- MaxPriority
- MinPriority
- Required
- ShowContextMenu
- ShowGrid
- ShowHeading
- ShowStatusBar

Alarm Client Control Property

The AlarmClientControl property lets you explore the underlying methods and events that can be scripted within the underlying Alarm Client Control, using the following syntax:

```
MyContent.AlarmControl1.AlarmClientControl.<PropertyName>
```

Examples:

```
MyContent.AlarmControl1.AlarmClientControl.AlarmQuery
MyContent.AlarmControl1.AlarmClientControl.ClientMode
MyContent.AlarmControl1.AlarmClientControl.AckSignature.MaxPriority
MyContent.AlarmControl1.AlarmClientControl.ShowGroupByHeader
```

AlarmApp scriptable properties

The AlarmApp by itself contains only a single public property (ShowAreaHierarchy) for use in scripting. All other

properties are exposed by the underlying control. Although some of these pass-through properties can be seen in autocomplete, you should use the underlying property to maximize script performance.

The syntax for invoking the built-in AlarmApp property is:

```
MyContent.AlarmControl1.ShowAreaHierarchy
```

The syntax for invoking pass-through properties is as follows:

```
MyContent.AlarmControl1.AlarmClientControl.AlarmQuery
```

Scriptable Properties

Note: Since most scriptable properties in the AlarmApp are exposed by the underlying control, this list is very similar to the list of Alarm Client Control pass-through properties listed in the previous section.

- AlarmQuery
- AutoScroll
- ClientMode
- FlashUnAckAlarms
- GroupByHeader
- MaxPriority
- MinPriority
- Required
- ShowAreaHierarchy (built-in)
- ShowContextMenu
- ShowGrid
- ShowHeading
- ShowStatusBar

Non-Scriptable Properties

The remainder of the properties exposed by autocomplete in the Script Editor are public, but are not scriptable.

Write an event handler script

You can write event handler scripts by two methods after placing the AlarmApp onto a layout pane.

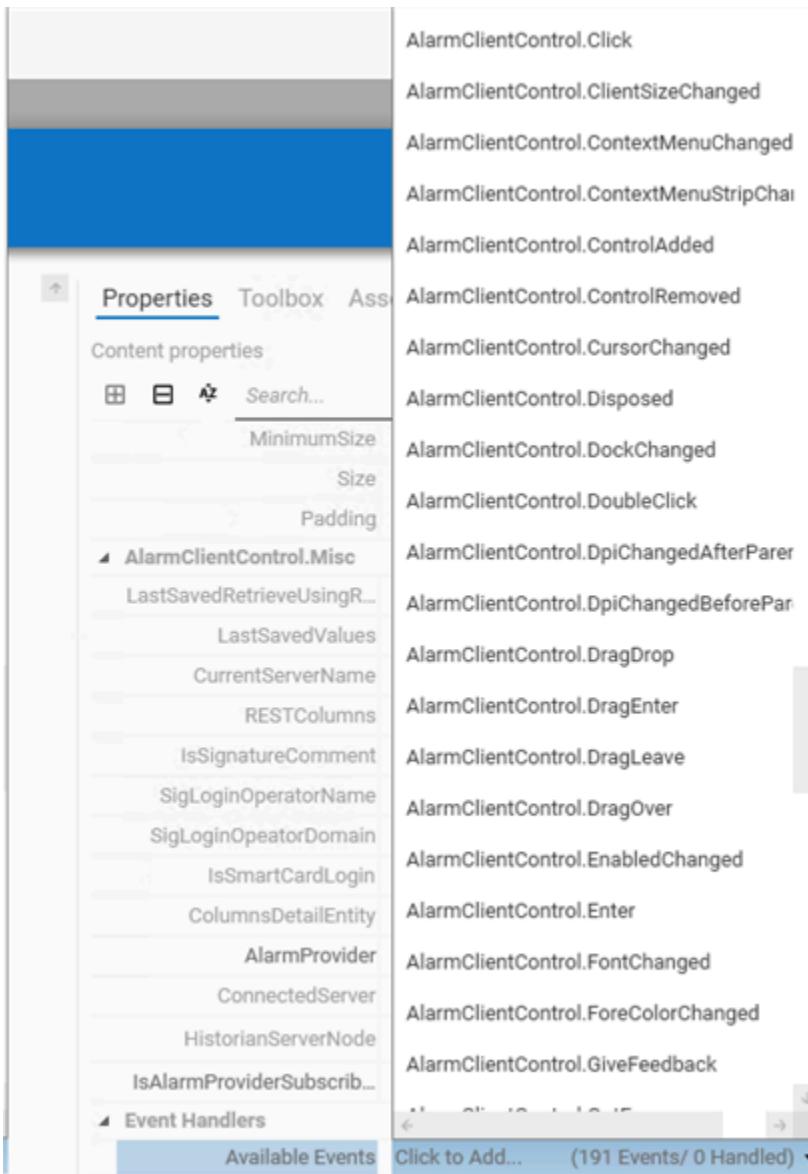
- Assigning values to the **Available Events** property on a selected control's **Properties** grid.
You select an event from a list of all available events of the control to create an event handler.
- Add a new event handler script from the Layout script editor's **Add New Script** dialog.

The following topics describe the workflows to create an event handler script by these two methods.

Write an event handler script from the Properties grid

The underlying control of the AlarmApp includes a set of public properties and events that can be viewed from

the Layout editor's **Properties** grid. The **Available Events** property near the bottom of the **Properties** list shows a list of AlarmApp events when it is selected. The listed events are prefaced with the name of the underlying control's property, as declared by the app.



After selecting a public AlarmApp event from the list, the Layout script editor opens to write an event handler script.

Write an Event Handler script using the Available Events property of the AlarmApp

1. Open the Layout editor and show the items listed in the **Toolbox** tab.
2. Select the AlarmApp from the **Toolbox** list to show its preview thumbnail.

The **Toolbox** pathway to the AlarmApp is:

_Default Content > 4. Apps > AVEVA OMI Apps > AlarmApp

The preview thumbnail of the AlarmApp appears beneath the list of **Toolbox** items.

3. Drag and drop the preview thumbnail of the AlarmApp onto a layout pane.
4. Select the pane where the AlarmApp has been placed and select the **Properties** grid.

AlarmApp properties appear in the grid. AlarmApp events appear in a list of the **Available Events** property.

5. Select the data entry field of the **Available Events** property to show a list of events.

Events are listed in alphabetic order.

6. Select one or more events from the list.

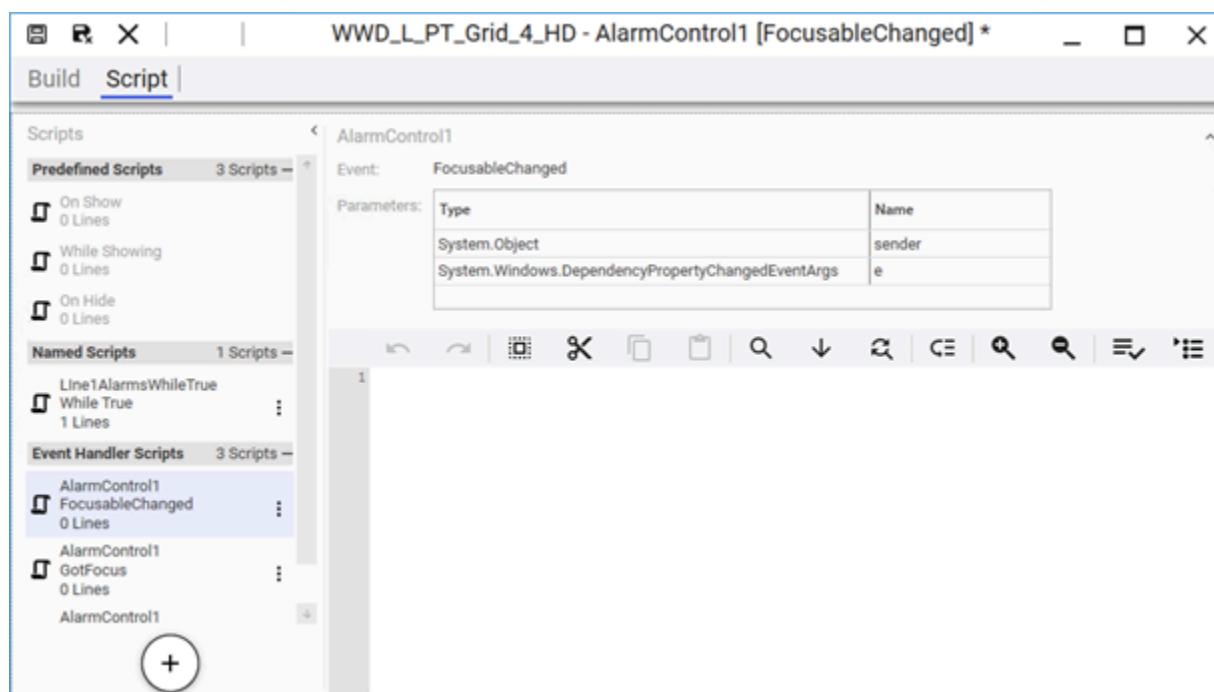
A selected handled event name appears in bold text.

The **Available Events** property updates to show each selected handled event with a data entry field.

All handled events without any script lines are listed in alphabetic order.

7. Select the button at the right of the selected event's data entry field.

The Layout script editor opens with the selected handled events listed beneath the **Event Handler Scripts** column. The focus of the Layout script editor will be the event handler you selected from the **Available Events** property.



The **Event** field shows the selected event script from the list.

8. Enter script code for each event handler.

The number of lines in each event handler script are shown in the **Available Events** property of the Layout editor.

9. Validate your script code by selecting the validation button at the right on the menu bar above the script edit area.



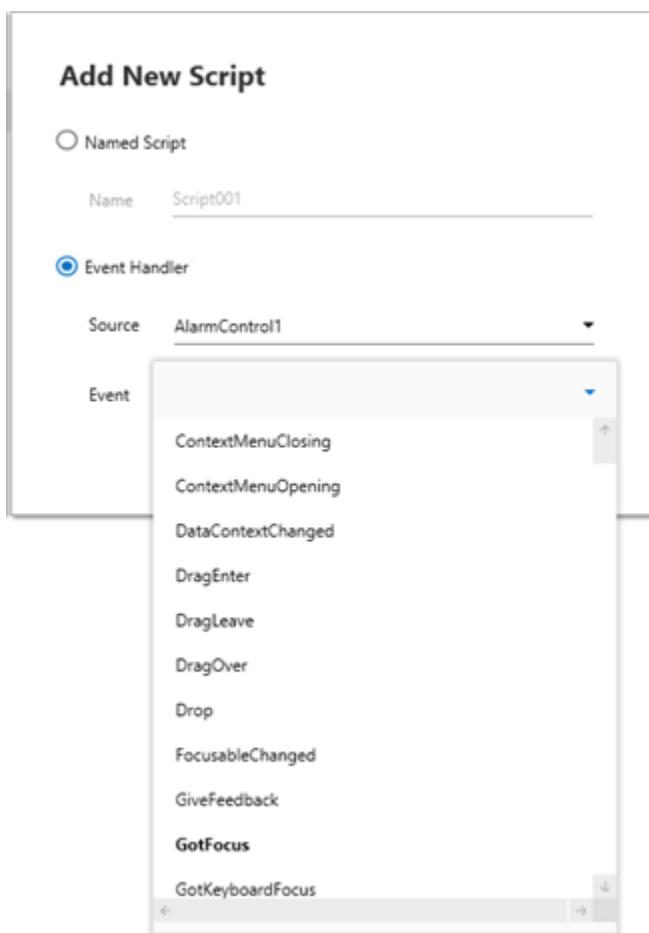
10. Save your scripts and exit from the Layout script editor.

Write an event handler script from the Layout script editor

Event Handler scripts can be created with the **Add New Script** dialog box of the Layout script editor. Handled events can be selected for the **Event** list of the **Add New Script** dialog box.

Write an AlarmApp event handler script from the control Properties page

1. Open the layout and show the items listed in the **Toolbox** tab.
2. Select the AlarmApp from the **Toolbox** list to show its preview thumbnail.
The **Toolbox** pathway to the AlarmApp is:
_Default Content > 4. Apps > AVEVA OMI Apps > AlarmApp
3. Drag and drop the preview thumbnail onto a layout pane.
4. Select **Script** from the menu bar of the Layout editor to open the script editor.
5. Create an event handler script.
 - a. Within the Layout script editor, select the red plus circle to show the **Add NewScript** dialog box.
 - b. Select **Event Handler**.
 - c. Select the name of the AlarmControl placed on the layout pane from the **Source** list.
 - d. Select the data field of **Event** to show a list of event handlers.



- e. Select one or more event handlers from the list.
- f. Select **Add** to close the **Add New Script** dialog box and return to the Layout script editor.

The **Event Handler Scripts** area shows the selected event handlers in alphabetic order.

6. Select an event handler from the list and enter script code for each event handler.

The number of lines in each event handler script are shown in the **Available Events** property of the Layout

editor.

7. Validate your script code by selecting the validation button at the right on the menu bar above the script edit area.

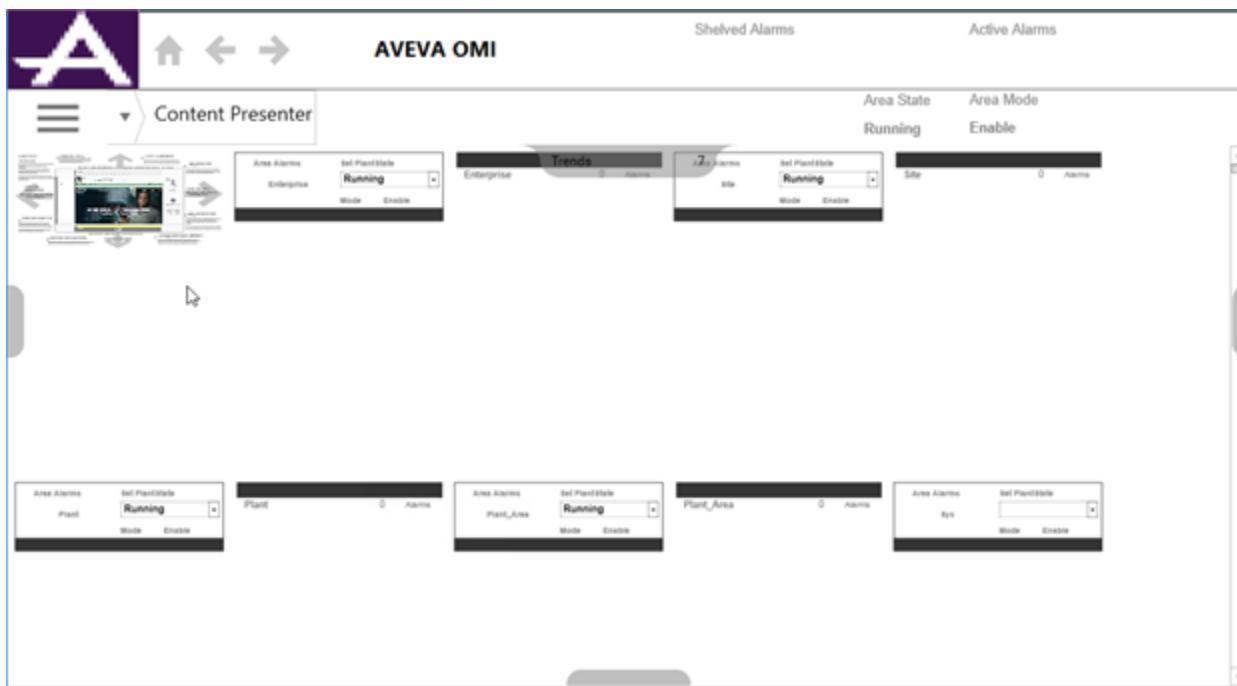


8. Save your scripts and exit from the Layout script editor.

The new event handler scripts appears beneath the **Event Handler Scripts** area at the left of the editor and the window focus is set to the script edit area to begin writing script code.

ContentPresenterApp

The ContentPresenterApp shows graphics from one or more selected nodes of a ViewApp navigation model. The pane in which the ContentPresenterApp has been placed shows a set of graphic thumbnails that match the ContentPresenterApp filter criteria. Graphics shown by the ContentPresenterApp can be filtered by content type, name, and location within the ViewApp navigation model.

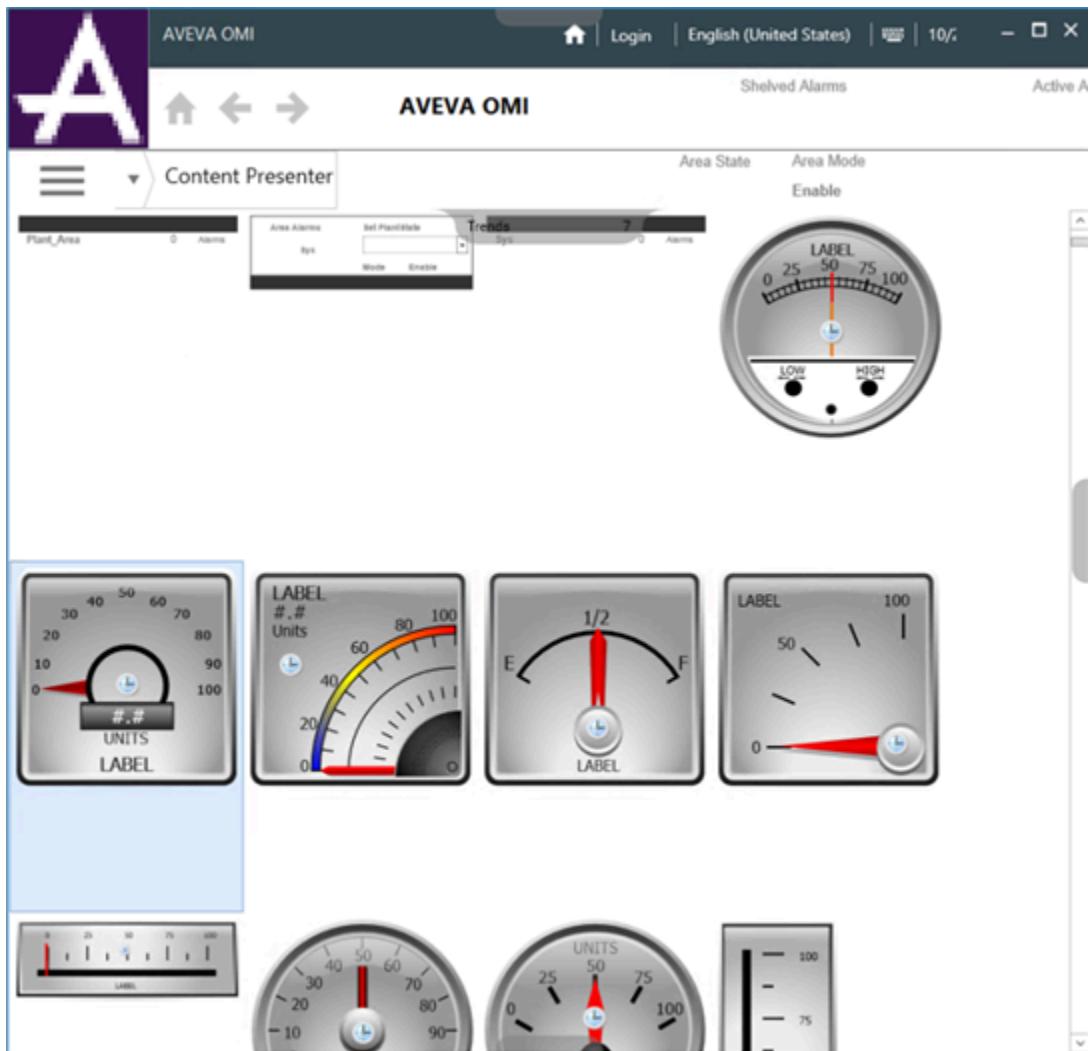


Other ContentPresenterApp properties determine the size and layout of the graphic thumbnails. An **Interactive Content** property enables users to interact with the scripts and animation of selected graphic items shown by the ContentPresenterApp. Finally, users can select a graphic shown by the ContentPresenterApp to set the current focus of a ViewApp to the navigation item containing the graphic.

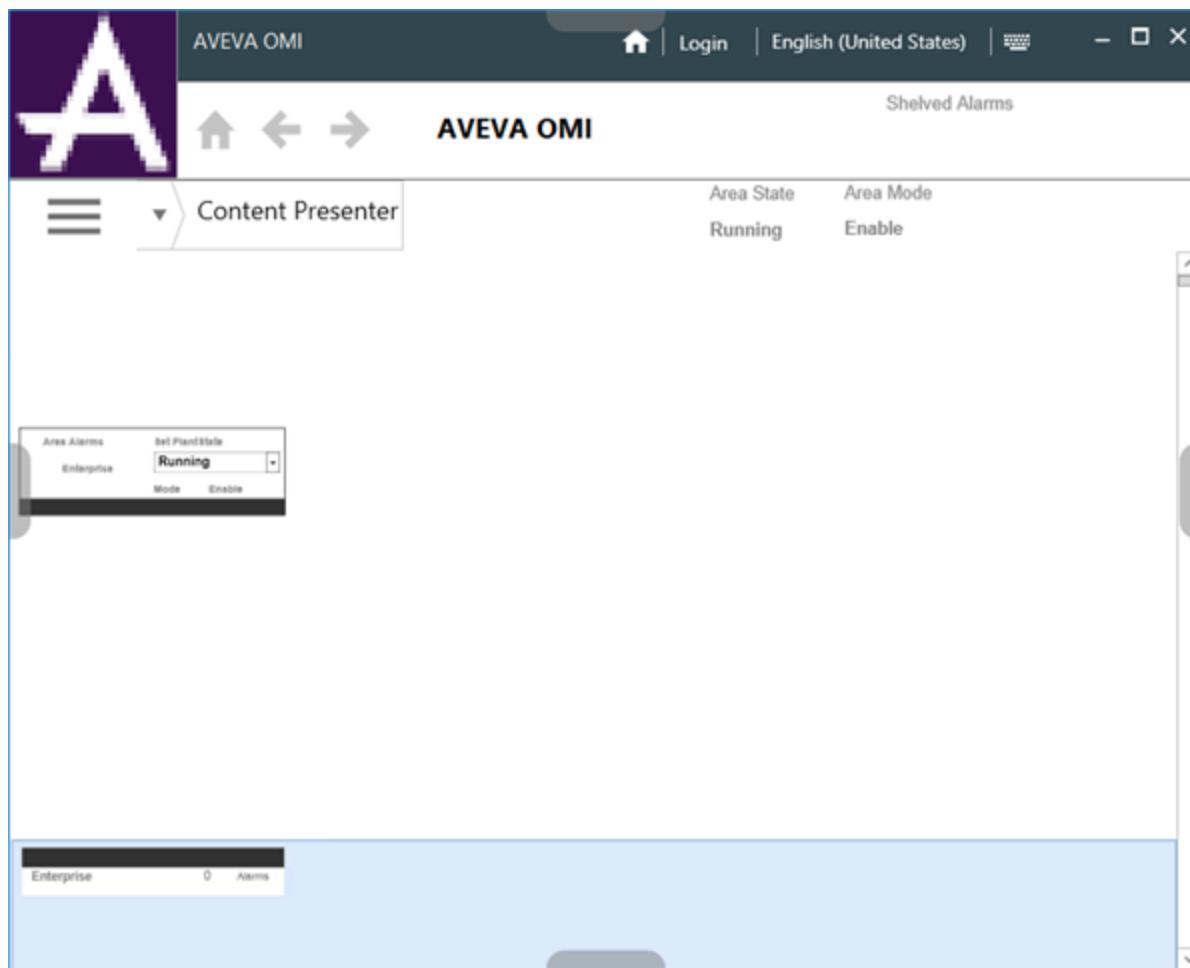
Details of the ContentPresenterApp display

During configuration, a ContentPresenterApp is placed in a layout pane. The pane itself becomes the view port of the ContentPresentApp to show the graphics incorporated in the running ViewApp. The following figure shows the display of the ContentPresenterApp during runtime. In this example, graphics are listed in horizontal order and wrap to the next row of the view port when a graphic cannot completely fit within the right view port border.

The size of the view port can be changed by adjusting the size of the layout pane to show more or fewer graphics on each row. Graphics continue to wrap to the next row until they extend beyond the bottom of the view port. Scroll bars appear to move the view port vertically and show the remaining graphics.



This example shows a graphic display when the **Fill** property is set to **Horizontal** and the **View Mode** property is set to **WrapContent**. Both properties include other options to list graphics vertically and show the graphics in continuous sequential order without wrapping.



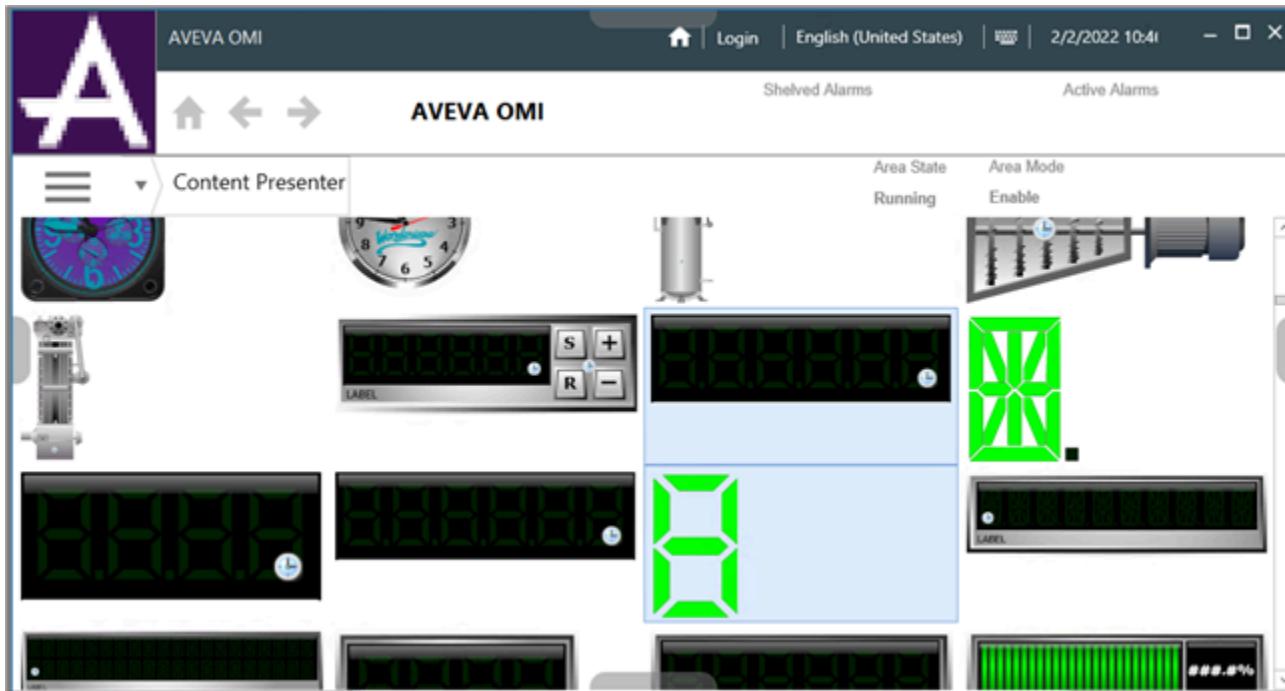
The blue background shown in both examples represents a cell within a view port. Each view port cell contains a single graphic. The ContentPresenterApp includes a set of **Layout** and **Size** properties that determine the size of the cell, the blank space around the graphic, and the placement of the graphic within the cell.



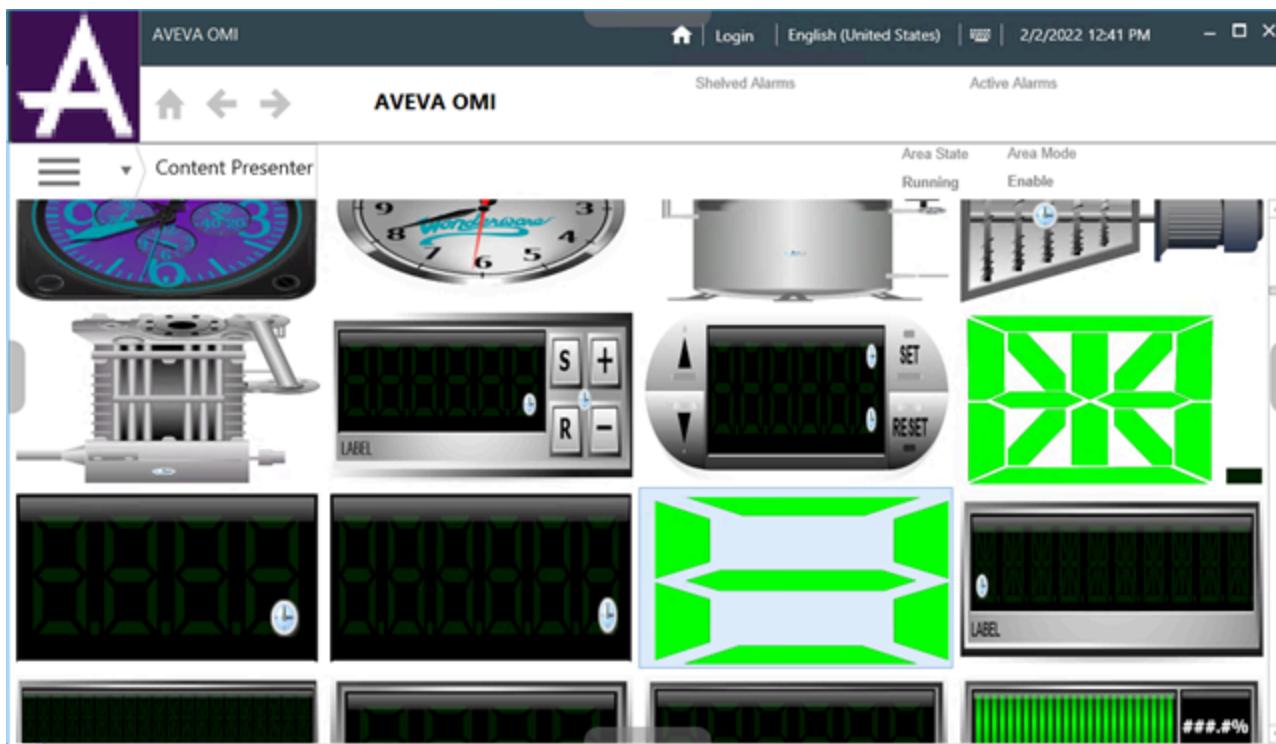
The figure above shows a fixed size cell with five pixels of padding around the graphic, which is placed at the top

center of the cell. All cells in the view port have the same fixed width and height.

The **Layout** and **Size** properties include other options to scale the size of view port cells to fit the size of graphics and specify the number of cells that appear in each row or column of the view port. The following example shows cells that are scaled to the width and height of the graphics placed within them. The **Display Columns** property has been set to 4 to always shows four columns of graphics in the view port.



Here is the same example again with the **Stretch** property enabled. The aspect ratio of a graphic is ignored and the graphic is enlarged in its non-constraining dimension of the cell. For example, the number 8 LED shown above is constrained by its height within its cell. When **Stretch** is enabled, the graphic width increases to the full width of the cell, as shown below.

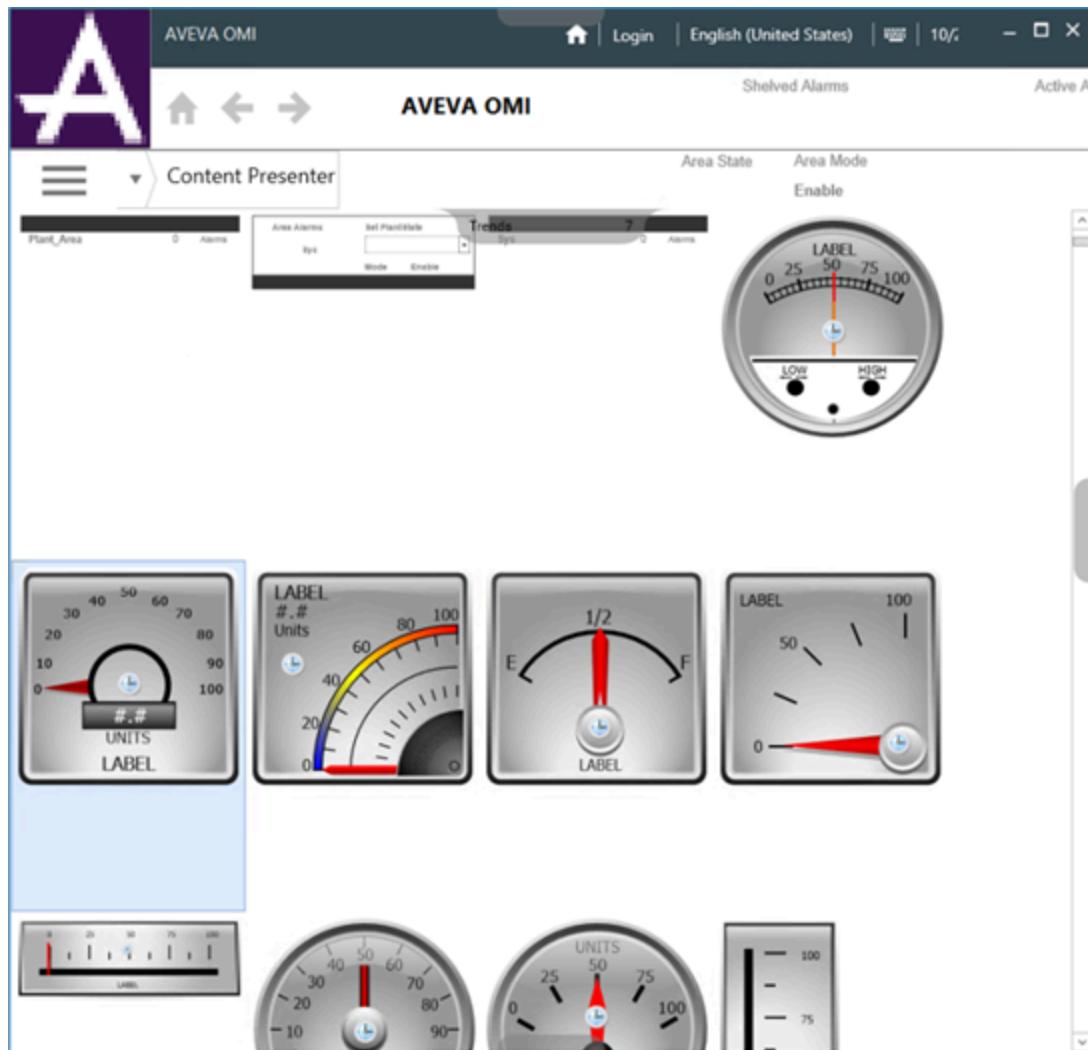


See [More information about ContentPresenterApp properties](#) for a description of the configuration procedure and how the various property options affect the visual characteristics of graphics within a running ViewApp.

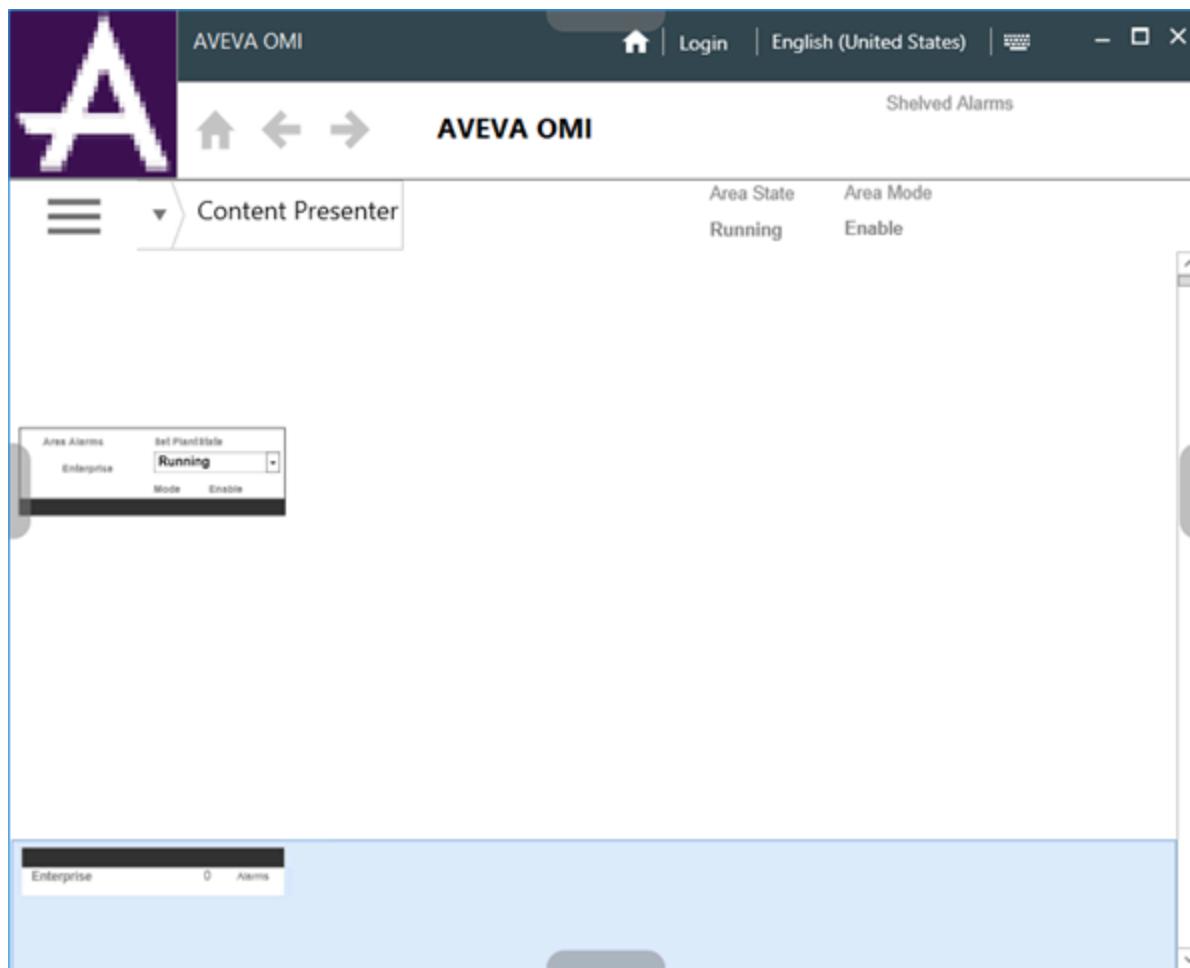
ContentPresenterApp runtime examples

During configuration, a ContentPresenterApp is placed in a layout pane. The pane itself becomes the view port of the ContentPresentApp to show the graphics incorporated in the running ViewApp. The following figure shows the display of the ContentPresenterApp during runtime. In this example, graphics are listed in horizontal order and wrap to the next row of the view port when a graphic cannot completely fit within the right view port border.

The size of the view port can be changed by adjusting the size of the layout pane to show more or fewer graphics on each row. Graphics continue to wrap to the next row until they extend beyond the bottom of the view port. Scroll bars appear to move the view port vertically and show the remaining graphics.



This example shows a graphic display when the **Fill** property is set to **Horizontal** and the **View Mode** property is set to **WrapContent**. Both properties include other options to list graphics vertically and show the graphics in continuous sequential order without wrapping.



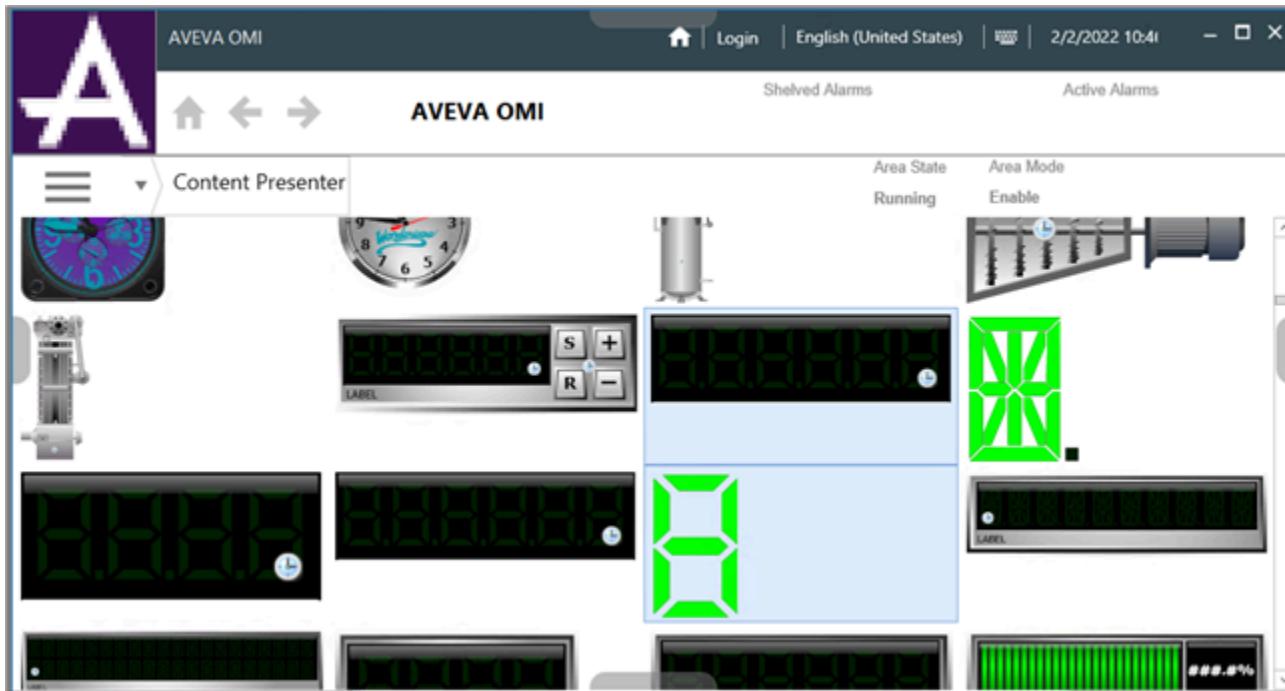
The blue background shown in both examples represents a cell within a view port. Each view port cell contains a single graphic. The ContentPresenterApp includes a set of **Layout** and **Size** properties that determine the size of the cell, the blank space around the graphic, and the placement of the graphic within the cell.



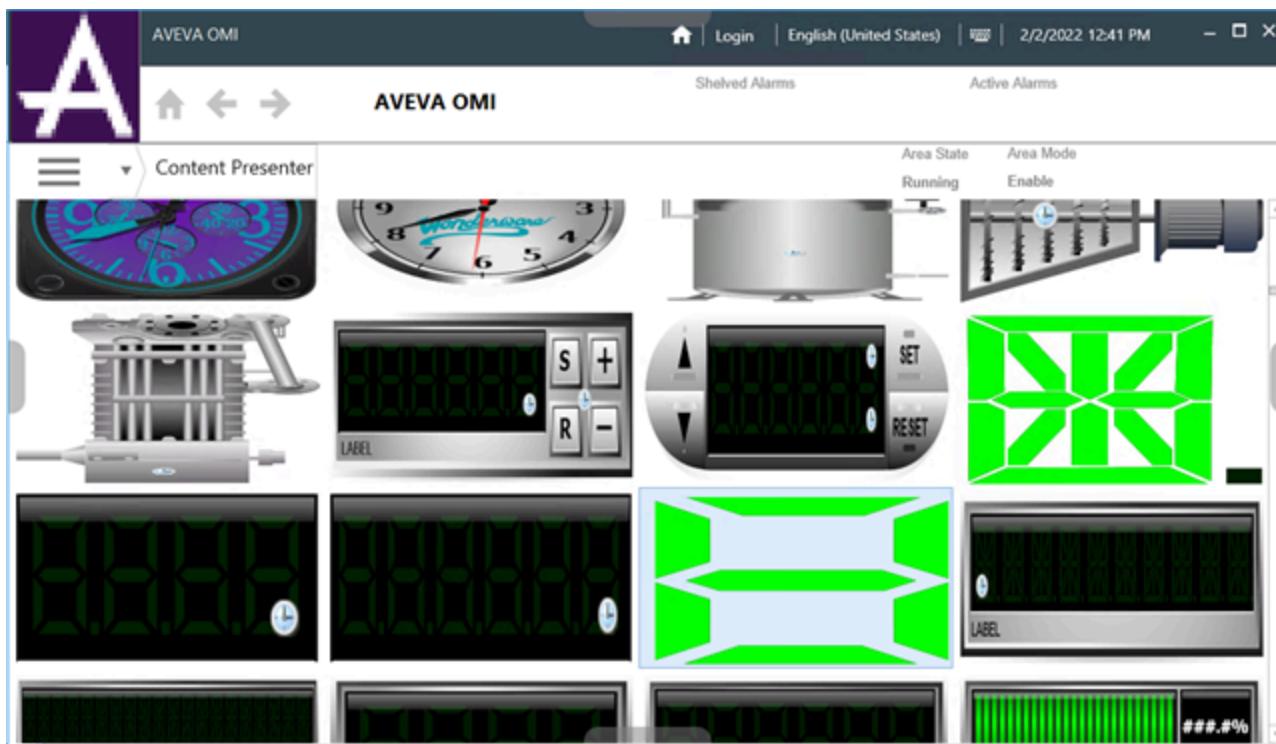
The figure above shows a fixed size cell with five pixels of padding around the graphic, which is placed at the top

center of the cell. All cells in the view port have the same fixed width and height.

The **Layout** and **Size** properties include other options to scale the size of view port cells to fit the size of graphics and specify the number of cells that appear in each row or column of the view port. The following example shows cells that are scaled to the width and height of the graphics placed within them. The **Display Columns** property has been set to 4 to always shows four columns of graphics in the view port.



Here is the same example again with the **Stretch** property enabled. The aspect ratio of a graphic is ignored and the graphic is enlarged in its non-constraining dimension of the cell. For example, the number 8 LED shown above is constrained by its height within its cell. When **Stretch** is enabled, the graphic width increases to the full width of the cell, as shown below.



See [More information about ContentPresenterApp properties](#) for a description of the configuration procedure and how the various property options affect the visual characteristics of graphics within a running ViewApp.

Configure the ContentPresenterApp

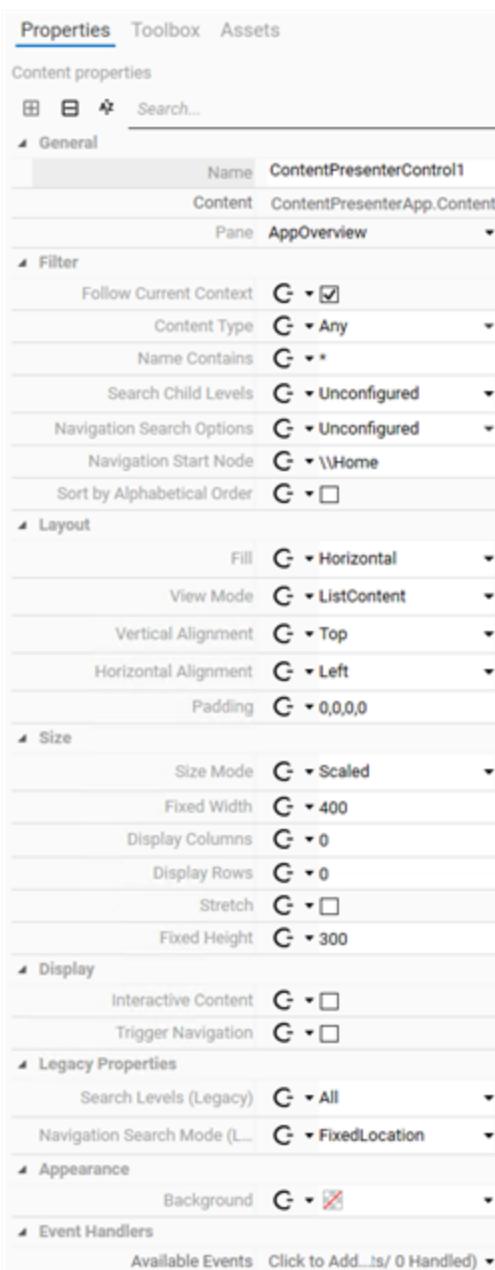
The ContentPresenterApp includes several sets of related properties that determine the search criteria to select the displayed graphic items, how they are items organized within a pane, their size, and how they behave when selected.

To configure the ContentPresenterApp

1. Open the Layout or ViewApp Editor and show the items listed in the **Toolbox** tab.
2. Open the folder in the **Toolbox** where the ContentPresenterApp is located. Typically, this is the **OMI Apps** folder.
3. Select ContentPresenterApp from the **Toolbox** list.
4. Drag and drop the ContentPresenterApp thumbnail onto a pane.
5. Select the ContentPresenterApp thumbnail and select the **Properties** tab.

The **Properties** tab shows the properties of the ContentPresenterApp. The following figure shows the default property values.

See [More information about ContentPresenterApp properties](#) for the details of each property.



- Set the property values of the ContentPresenterApp.
- Save your changes.

More information about ContentPresenterApp properties

ContentPresenterApp properties can be shown in the Layout or ViewApps editors by placing the ContentPresenterApp on a layout pane. After selecting the app, its properties appear on the **Properties** page by the functional groups shown in the following table.

Some combinations of property values are invalid. When a configuration error occurs, an error dialog appears when the ViewApp is deployed and starts running. The ContentPresenterApp view port never appears in the layout pane of the running ViewApp. For more information, see [ContentPresenterApp property validation](#).

Filter Properties	
Follow Current Context	<p>Determines if the currently selected node will be used as starting point for searching assets at run time.</p> <ul style="list-style-type: none"> • If enabled (default), the asset corresponding to the currently selected node is used as the starting point for search. • If disabled, the configured Navigation Start Node is used as the starting point for search. <p>Note: You must configure Navigation Start Node if this property is disabled.</p> <hr/> <p>Data Type: Boolean ½ Default Value: True ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Content Type	<p>A drop-down list that shows the different types of graphic content that can be shown in the ContentPresenterApp. Any is the default, which displays all graphic content types.</p> <p>If one or more specific content types are selected, the ContentPresenterApp restricts the display to only those graphic items that have a matching ContentType property value assigned from the Graphic Editor.</p> <p>Use a comma as the delimiter to separate multiple content types in the Content Type field, for example: Any, Overview, Detailed, Faceplate</p> <p>See ContentType property for more information.</p> <hr/> <p>Data Type: Enum ½ Default Value: Any ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Name Contains	<p>Filter to restrict the graphic items shown in the ContentPresenterApp to those graphics whose names match the search string entered as the Name Contains value. * is the default, which shows all graphics.</p> <p>Asterisk (*) and question mark (?) wildcard characters can be entered as placeholder character replacements in the search string.</p> <hr/> <p>Data Type: String ½ Default Value: * ½ Runtime</p>

Filter Properties	
	<p>Property: No Scriptable: Yes ½ Bindable: Yes</p>
Search Child Levels	<p>Determines the depth of the child hierarchy to be traversed for content search. This property is only used if “Child Levels” is selected under Navigation Search Options. Options are:</p> <ul style="list-style-type: none"> • Unconfigured: Navigation search options will function in <i>legacy</i> mode, and the legacy search properties Search Levels and Navigation Search Mode will be automatically converted to their current equivalents (Search Child Levels and Navigation Search Options). <p>Note: Legacy properties are deprecated as of the System Platform 2023 release. Legacy mode and legacy search properties are described below, towards the end of this table.</p> <ul style="list-style-type: none"> • All: Searches all levels of the hierarchy below the currently-selected node. • OneLevel: Searches only the level of the hierarchy immediately below the level of the currently-selected node. • TwoLevels: Searches the two levels of the hierarchy immediately below the level of the currently-selected node. • ThreeLevels: Searches the three levels of the hierarchy immediately below the level of the currently-selected node. <p>Note: To include the current level of the hierarchy in searches, you must also enable the "Siblings" option in Navigation Search Options.</p> <hr/> <p>Data Type: Enum ½ Default Value: Unconfigured ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>

Filter Properties	
Navigation Search Options	<p>Determines which navigation search option(s) is/are enabled. Multiple search options can be selected.</p> <ul style="list-style-type: none"> • Unconfigured: Navigation search options will function in <i>legacy</i> mode, and the legacy search properties Search Levels and Navigation Search Mode will be automatically converted to their current equivalents (Search Child Levels and Navigation Search Options). <p>Note: Legacy properties are deprecated as of the System Platform 2023 release. Legacy mode and legacy search properties are described below, towards the end of this table.</p> <ul style="list-style-type: none"> • StartNode: Search begins from a node, either specified by the Current Asset, or by the navigation path configured in the NavigationStartNode property. • Siblings: Search begins from the sibling node of a StartNode and traverses all siblings of the start node. • ChildLevels: Searches Child Nodes to the depth specified by the SearchChildLevels property. <hr/> <p>Data Type: Enum ½ Default Value: Unconfigured ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Navigation Start Node	<p>The node in the ViewApp's navigation model to start the search for graphic items to be shown in the ContentPresenterApp. \\Home is the default starting node.</p> <p>The starting node is always expressed as the navigation item pathway in the form: \\StartNode\\node1\\node2\\node3</p> <p>Navigation Search Mode must be set to FixedLocation to enable the Navigation Start Node property.</p> <p>Navigation Start Node is a bindable/run-time property that can be bound to a data reference, and the value can be changed dynamically at run time. See Navigation Start Node property for more information.</p> <hr/> <p>Data Type: String ½ Default Value: \\Home ½</p>

Filter Properties	
	<p>Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Sort by Alphabetic Order	<p>When selected, the ContentPresenterApp displays graphics in ascending alphabetic order by name.</p> <hr/> <p>Data Type: Boolean ½ Default Value: False ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Layout Properties	
Fill	<p>Options to set vertical or horizontal fill order of graphics shown in the ContentPresenterApp. Horizontal fill order is left to right within the view port. Vertical fill order is top to bottom within the view port. Horizontal is the default.</p> <p>See Layout properties for more information about the fill order of content.</p> <hr/> <p>Data Type: Enum ½ Default Value: Horizontal ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
View Mode	<p>Determines the organization of graphic items shown within the viewer port of the ContentPresenterApp.</p> <ul style="list-style-type: none"> • ListContent: Graphic items are shown in the ContentPresenterApp as a continuous sequential list. Based on the value specified for the Fill property, the list is arranged in vertical or horizontal order. The default view mode. Scroll bars appear when the graphic list extends beyond the borders of the view port. • WrapContent: Graphics are listed in the fill direction until a graphic does not fit within the border of the view port. The graphic then wraps to

Layout Properties	
	<p>the next column or row within the view port. Scroll bars appear when the graphic item list extends beyond the view port borders.</p> <ul style="list-style-type: none">• GridContent: Use of GridContent mode is NOT recommended. This mode has been deprecated, but is retained to support ViewApps developed with an earlier version of the ContentPresenterApp. When specified, graphics wrap in the direction specified by the Fill property. See Updates to the ContentPresenter App for a description of backward compatibility in the ContentPresenterApp when GridContent is selected. <p>See Layout properties for more information about other View Mode options.</p> <hr/> <p>Data Type: Enum ½ Default Value: ListContent ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Vertical Alignment	Vertical alignment (Top , Center , Bottom) of the graphic within a cell of the ContentPresenterApp view port. <hr/> <p>Data Type: Enum ½ Default Value: Top ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Horizontal Alignment	Horizontal alignment (Left , Center , Right) of the graphic within a cell of the ContentPresenterApp view port. <hr/> <p>Data Type: String ½ Default Value: Left ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Vertical Alignment	Vertical alignment (Top , Center , Bottom) of the graphic within a cell of the ContentPresenterApp view port. <hr/> <p>Data Type: Enum ½ Default Value: Top ½ Runtime Property: No</p>

Layout Properties	
	Scriptable: Yes ½ Bindable: Yes
Padding	<p>Number of pixels as blank space padding on the left, right, top, and bottom of a graphic shown in the ContentPresenterApp. 0,0,0,0 is the default, which does not provide any padding.</p> <p>See Padding property for more information.</p> <hr/> <p>Data Type: Integer Array ½ Default Value: 0,0,0,0 ½ Runtime Property: No</p> <p>Scriptable: Yes ½ Bindable: Yes</p>

Size Properties	
Size Mode	<p>Options to set the size of graphics shown in the ContentPresenterApp by a fixed or scaled size. Scaled is the default size mode.</p> <ul style="list-style-type: none"> • Scaled: Graphic size is scaled to the constraining horizontal or vertical dimension of the containing view port cell while maintaining the aspect ratio of the graphic. <p>Cell size is an inverse function of the number of cells specified to appear in the view port. Cell size decreases with an increasing number of rows and columns in the view port. The number of cells shown in a view port is determined by the Display Rows and Display Columns properties.</p> <ul style="list-style-type: none"> • Fixed: The cells shown in the view port are set to a fixed width and height in pixels. Graphic size is determined by its constraining dimension when placed in the fixed fixed width or height of the cell. <p>The width and height values of view port cells are specified from the Fixed Width and Fixed Height properties.</p> <p>See Details of the Size Mode property for more information.</p> <hr/> <p>Data Type: Enum ½ Default Value: Scaled ½ Runtime Property: No</p> <p>Scriptable: Yes ½ Bindable: Yes</p>
Fixed Width	<p>Number of pixels as the width of a cell within the view port of the ContentPresenterApp. The size of a graphic is restricted by the constraining dimension of a cell while maintaining the graphic's aspect ratio.</p> <p>The Size Mode property must be set to Fixed to enable the Fixed Width and Fixed Height properties.</p> <hr/> <p>Data Type: Integer ½ Default Value: 400 ½ Runtime Property: No</p> <p>Scriptable: Yes ½ Bindable: Yes</p>
Fixed Height	Number of pixels as the height of a cell within the view port of the ContentPresenterApp.

Size Properties	
	<p>The Size Mode property must be set to Fixed to enable the Fixed Width and Fixed Height properties.</p> <hr/> <p>Data Type: Integer ½ Default Value: 300 ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Display Columns	<p>Number of columns containing graphics shown in a view port. Column numbers are specified as integers.</p> <p>The ContentPresenterApp view pane includes a horizontal scroll bar to show columns outside of the limit set by the Display Columns property.</p> <p>See Display Columns property for more information.</p> <hr/> <p>Data Type: Integer ½ Default Value: 0 ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Display Rows	<p>Number of rows containing graphics shown in a Content Presenter view port. Row numbers are specified as integers.</p> <p>The ContentPresenterApp view port includes a vertical scroll bar to show rows outside of the limit set by the Display Rows property.</p> <p>See Display Rows property for more information.</p> <hr/> <p>Data Type: Integer ½ Default Value: 0 ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Stretch	<p>When Stretch is enabled, the horizontal/vertical aspect ratio of a graphic changes to fill the available space of the non-constraining dimension within a cell of a Content Presenter view port.</p> <p>The Display Rows and Display Columns properties must be set to non-zero values to enable the Stretch property.</p> <p>See Stretch property for more information.</p> <hr/> <p>Data Type: Boolean ½ Default Value: False ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>

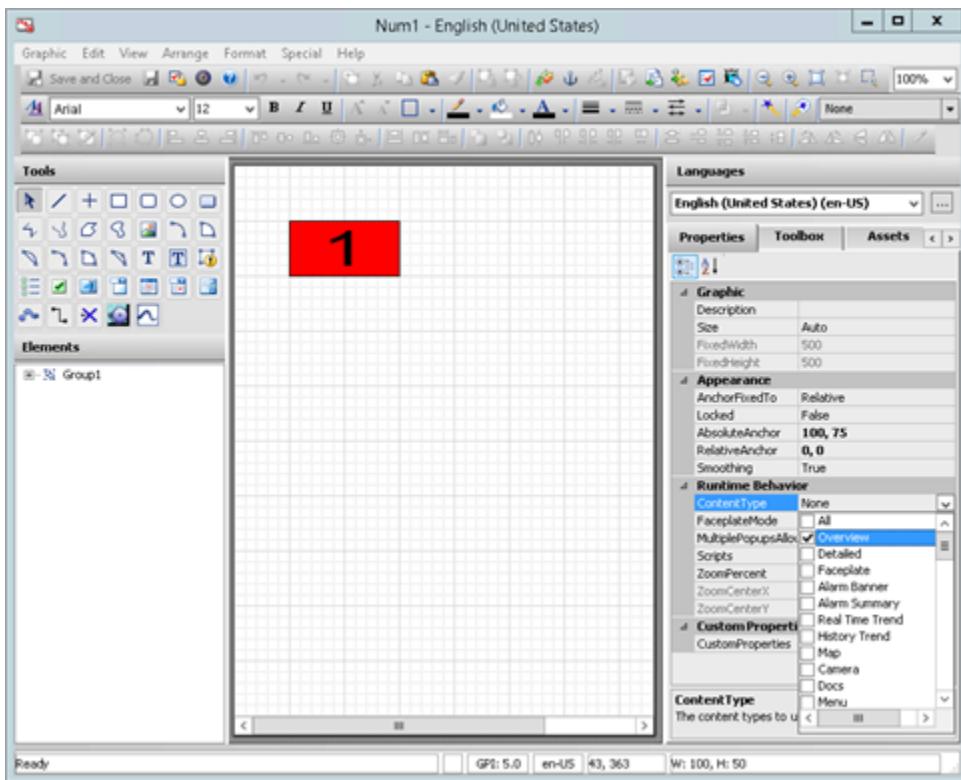
Display Properties	
Interactive Content	<p>Boolean value that determines whether graphics shown by the ContentPresenterApp are interactive or not. When Interactive Content is selected, the scripts and animations of selected graphic items from the ContentPresenterApp are enabled.</p> <p>Both Display properties can be selected simultaneously. In the case when both properties are selected, Trigger Navigation takes precedence and the Interactive Content property value is not honored.</p> <hr/> <p>Data Type: Boolean ½ Default Value: False ½ Runtime Property: No</p> <p>Scriptable: Yes ½ Bindable: Yes</p>
Trigger Navigation	<p>Boolean value that determines whether a selected graphic shown by the ContentPresenterApp can set the focus within the running ViewApp to the navigation item containing the graphic. When the user selects a graphic item shown by the ContentPresenterApp, the current focus of the ViewApp is set to the navigation item containing the graphic.</p> <p>Both Display properties can be selected simultaneously. In the case when both properties are selected, Trigger Navigation takes precedence and the Interactive Content property value is ignored.</p> <hr/> <p>Data Type: Boolean ½ Default Value: False ½ Runtime Property: No</p> <p>Scriptable: Yes ½ Bindable: Yes</p>
Legacy Properties	
Search Levels	<p>This property is deprecated as of the System Platform 2023 release, but is maintained to ensure that existing ViewApps will continue to function as expected, without reconfiguration.</p> <p>Options to set the number of node levels (1,2,3, All) at and beneath the starting node in the hierarchical ViewApp navigation model to search for graphics shown in the ContentPresenterApp. All is the default, which shows all graphics at the starting node and all nodes beneath it in the navigation model.</p> <p>See Search Levels property for more information.</p>

Legacy Properties	
	<p>Data Type: Enum ½ Default Value: All ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Navigation Search Mode	<p>This property is deprecated as of the System Platform 2023 release, but is maintained to ensure that existing ViewApps will continue to function as expected, without reconfiguration.</p> <p>Options to set the boundaries of a search within the node levels of a ViewApp navigation model.</p> <ul style="list-style-type: none">• FixedLocation: The search starts at the navigation node specified by the Navigation Start Node property and continues within the hierarchy of the navigation model based on the number of navigation node levels set by the Search Levels property.• CurrentNav: The search starts from the current selected navigation node of the ViewApp and potentially its children based on the value set for the Search Levels property.• ChildrenAlone: The search starts from the child nodes beneath the current selected navigation node and potentially the children of the children if the Search Levels property is set to search Two Levels or Three Levels. <p>Data Type: Enum ½ Default Value: FixedLocation ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes</p>
Appearance Property	
Background	Background color of the ContentPresenterApp view port. Select the Background field to show a color picker to set the background color. No color is the default background.

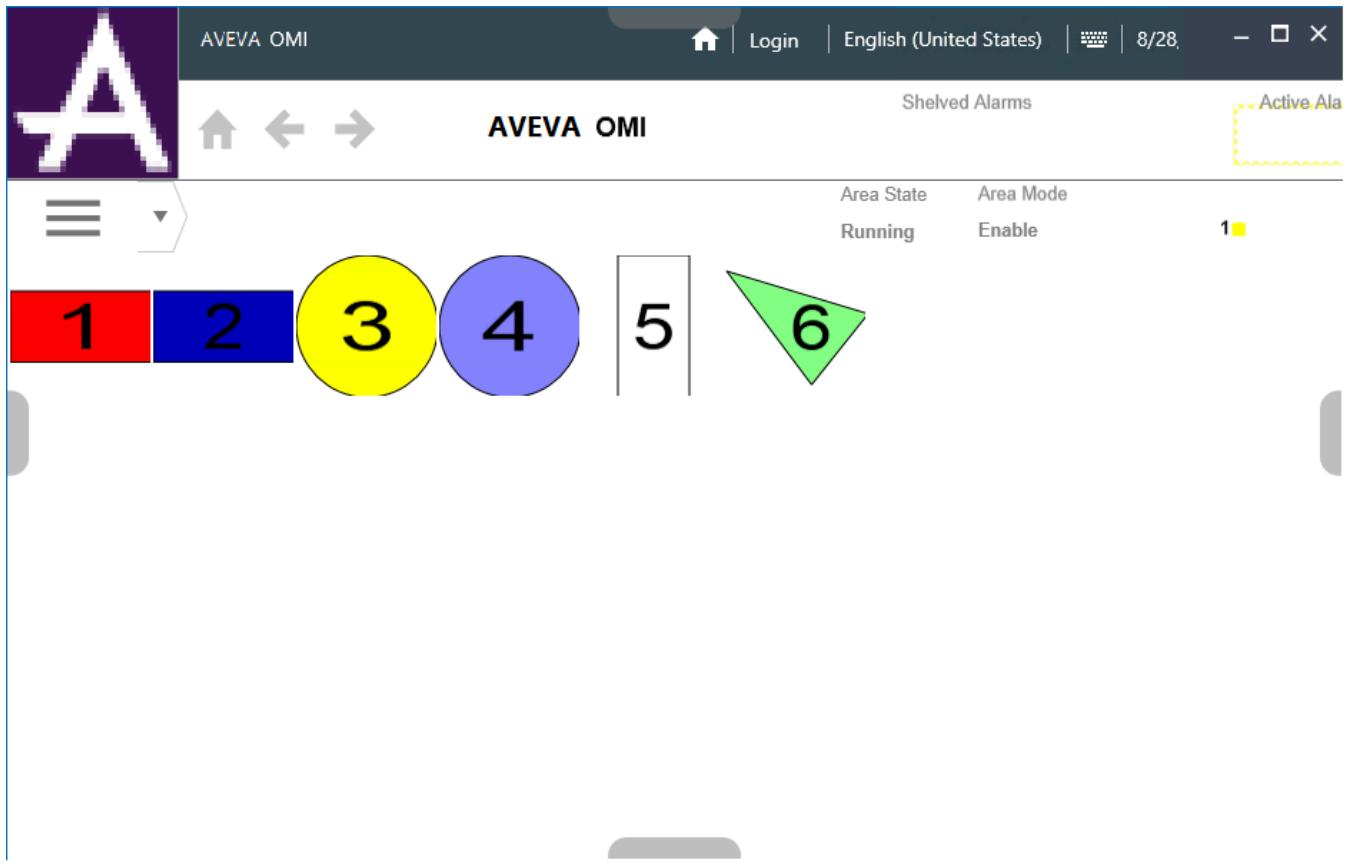
Appearance Property	
	Data Type: Color Picker ½ Default Value: No color ½ Runtime Property: No Scriptable: Yes ½ Bindable: Yes

ContentType property

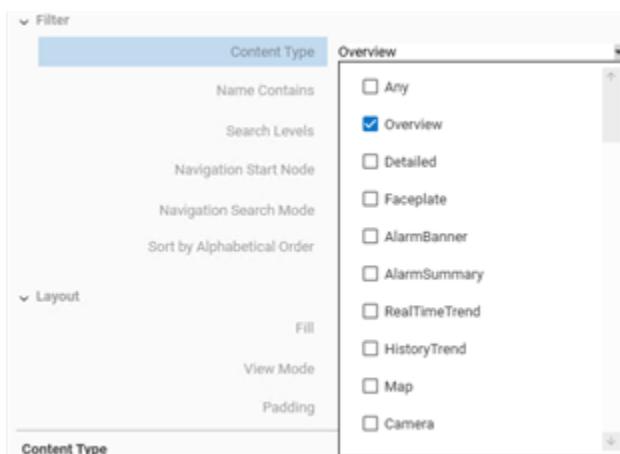
By default, a System Platform graphic is not assigned a value for its **ContentType** property. A graphic can be assigned a **ContentType** property value in the Graphic Editor. The following figure shows the Num1 graphic has been assigned **Overview** as its **ContentType** property value. The ContentPresenterApp can restrict the display of graphics to those that have been assigned a matching **ContentType** property value.



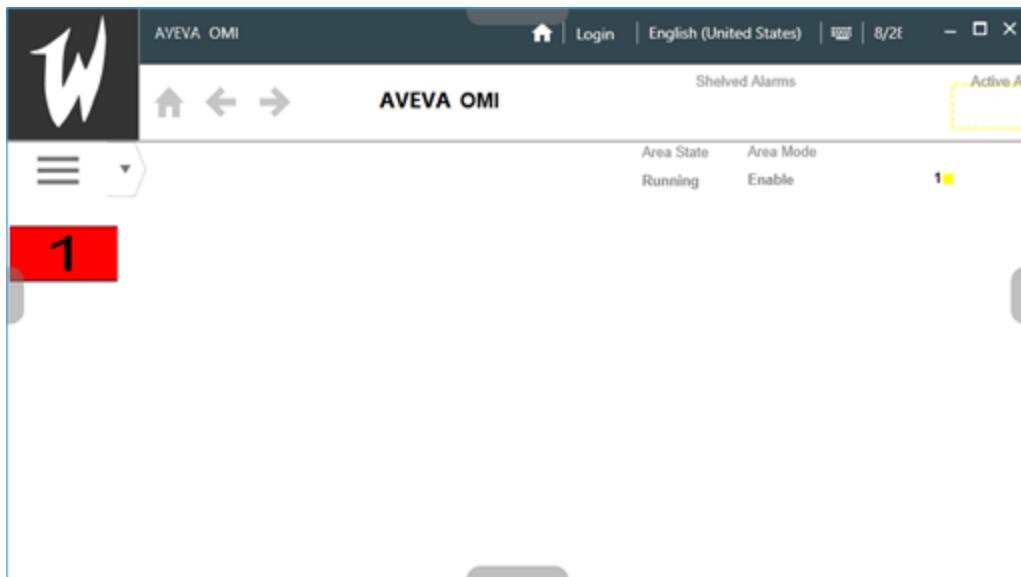
The default value of the **ContentType** property is **Any**, which shows all graphics regardless of each graphic's assigned **ContentType** property.



You can select one or more **ContentType** properties from a drop-down list in the **Properties** area.



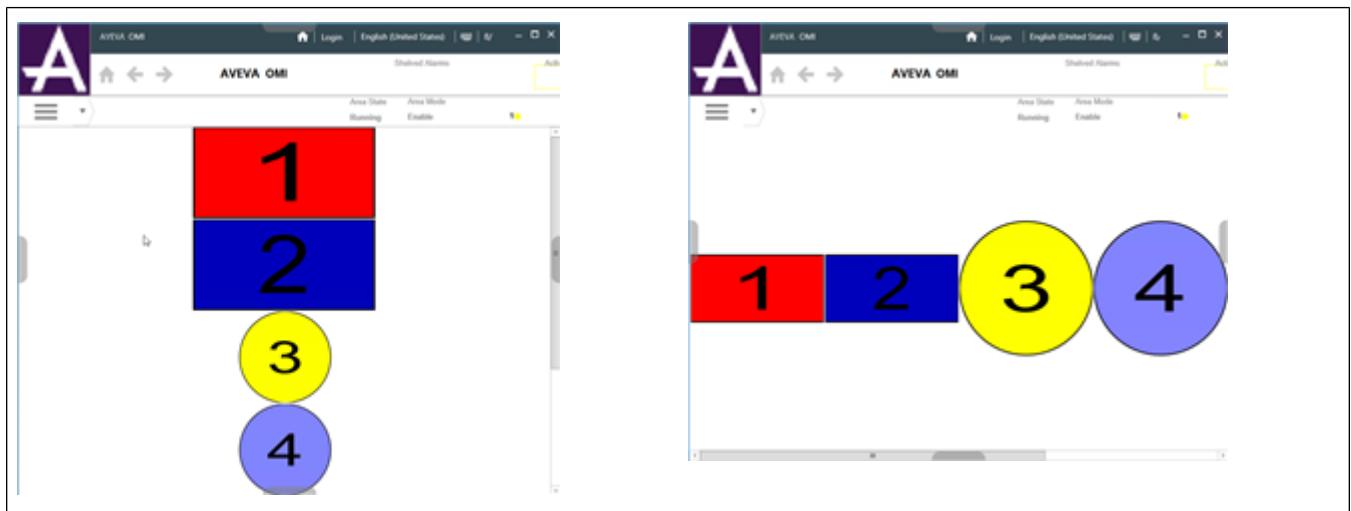
When the **ContentType** property is assigned one or more values, the ContentPresenterApp restricts the display to only those graphics with a matching **ContentType** property



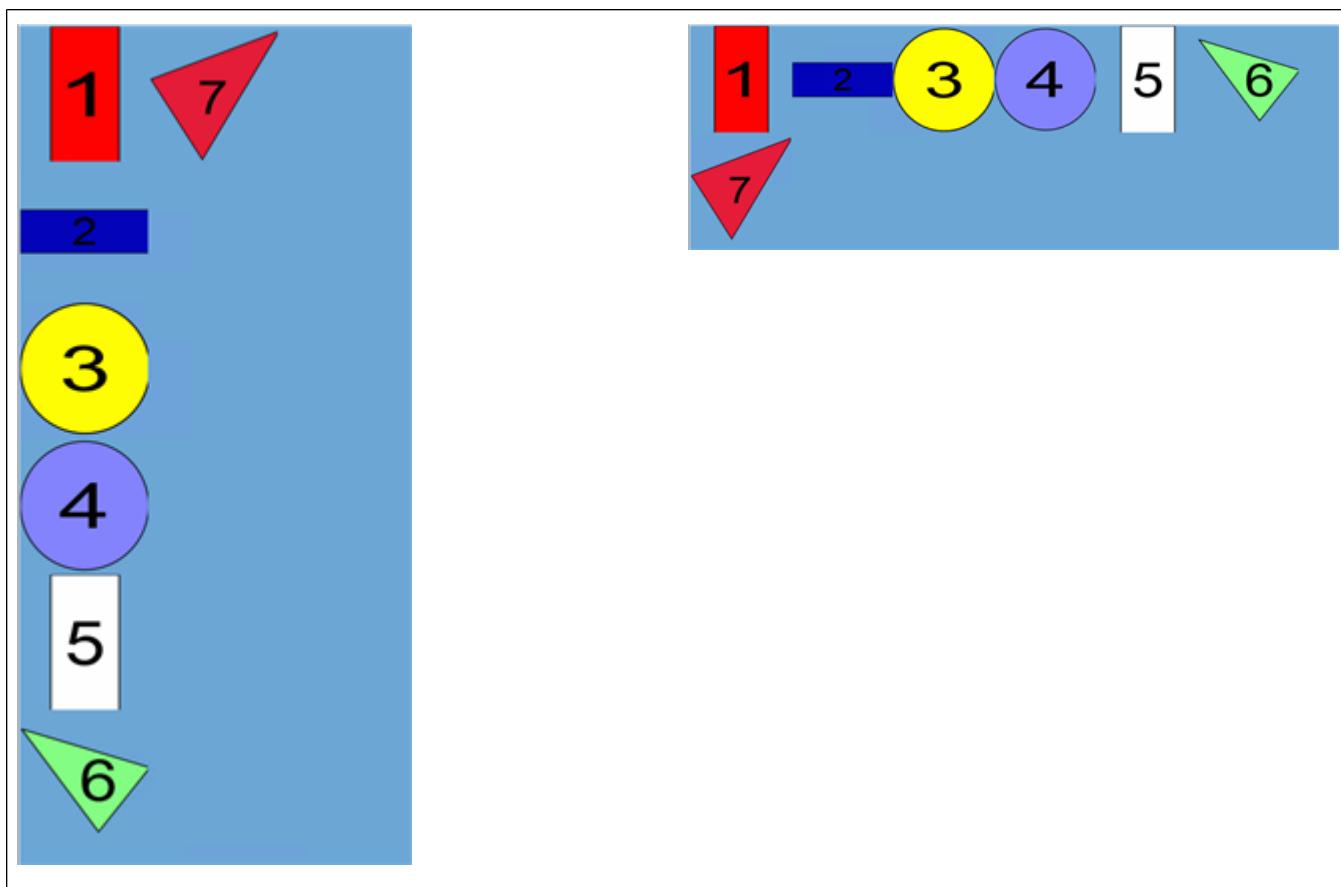
Layout properties

The **Fill** property includes two options to order graphic items in the ContentPresenterApp horizontally or vertically.

When the **ViewMode** property is set to **ListContent**, the viewer pane includes scroll bars to view graphic items that extend beyond the bottom or right border of the viewer pane.



When the **ViewMode** property is set to **WrapContent**, graphics appear on the next row or next column based on whether the **Fill** property value is **Horizontal** or **Vertical**.



Padding property

Padding is the blank space placed around the borders of a graphic when it appears in the ContentPresenterApp while a ViewApp is running.

Padding values are specified by the number of pixels placed at the left, top, right, and bottom borders of a graphic specified as four positional integers separated by commas.

30,0,0,0	0,30,0,0	0,0,30,0	0,0,0,30
Left Padding	Top Padding	Right Padding	Bottom Padding
A blue circle with the number 4 inside, surrounded by a light blue padding area.	A blue circle with the number 4 inside, surrounded by a light blue padding area.	A blue circle with the number 4 inside, surrounded by a light blue padding area.	A blue circle with the number 4 inside, surrounded by a light blue padding area.

The aspect ratio of a graphic is maintained when padding is applied. In the case when a graphic extends beyond a cell border of the ContentPresenterApp window because of padding, the graphic is reduced in size until it fits within the border while including the padding space.

No Padding	Left Padding

Size Mode property

The **Size Mode** property includes two options that determine the size of a graphic shown by the ContentPresenterApp.

- **Scaled**

The size of graphics shown in the ContentPresenterApp are scaled to a percentage of their actual dimensions and still maintain their original aspect ratio. Graphic size is determined by the constraining dimension of cells shown in the ContentPresenterApp view port.

Graphic size is a function of the number of cells shown in the view port. If there are many cells, graphic size must be necessarily smaller to fit within the dimensions of small cells. The **Display Rows** and **Display Columns** properties specify the number of cells shown in the view port.

- **Fixed**

The **Fixed Width** and **Fixed Height** properties specify the pixel width and height of a cell within the view pane of the ContentPresenterApp.

The following example shows the different sizes of the same graphics when the **Fixed Width** property cell width is set to 200 and 400. The constraining vertical or horizontal dimension of a view cell determines the size of a graphic at its original aspect ratio.

Fixed Cell Width 200	Fixed Cell Width 400

Display Columns property

The value assigned to the **DisplayColumns** property determines the number of columns that appear within the ContentPresenterApp viewer window. A non-zero value shows graphics in the specified number of columns and the graphics are scaled to fit the width of each column. The remaining graphics beyond the column limit appear on the next row of the view pane.

A value of zero indicates the **Display Columns** property is ignored and graphics appear in the viewer window in as many columns as calculated based on the values of other ContentPresenterApp properties.

Display Rows property

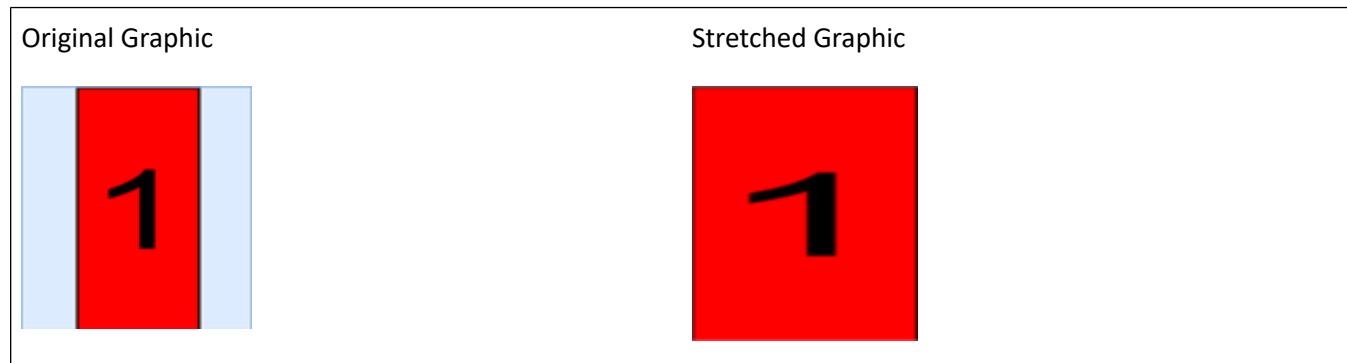
The **Display Rows** property specifies the number of graphic items that can be placed on a row within the ContentPresenterApp view pane.

If a value is specified then Content will be Fit to the height of the Row. A value of Zero indicates that Row specification is ignored and contents will be shown in as many Rows as calculated based on various factors such Contents Size, Fill Mode, Size Mode, Pane size etc.

Graphic items beyond the value set to the **Display Rows** property appear on rows beyond the current view pane.

Stretch property

When enabled, the **Stretch** property increases the size of a graphic within a cell of the ContentPresenterApp view pane in its non-constrained horizontal or vertical dimension. The original aspect ratio of the graphic is ignored.



The **Display Columns** and **Display Rows** properties must be set to non-zero values to enable the **Stretch** property.

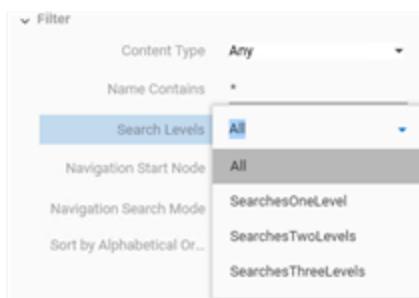
Search Levels property

The **Search Levels** property and the **Navigation Search Mode** property are both deprecated as of the System Platform 2023 release, but remain available in this release to preserve compatibility with applications created in prior versions. To replace and enhance the functionality of these deprecated properties, two new properties, **Search Child Levels** and Navigation **Search Options**, were added. To preserve the functionality of the deprecated properties, set the new properties "**Search Child Levels**" and "**Navigation Search Options**" properties to "**Unconfigured**."

The deprecated **Search Levels** and **Navigation Search Mode** properties have been moved from the **Filter** properties category to the **Legacy** properties category.

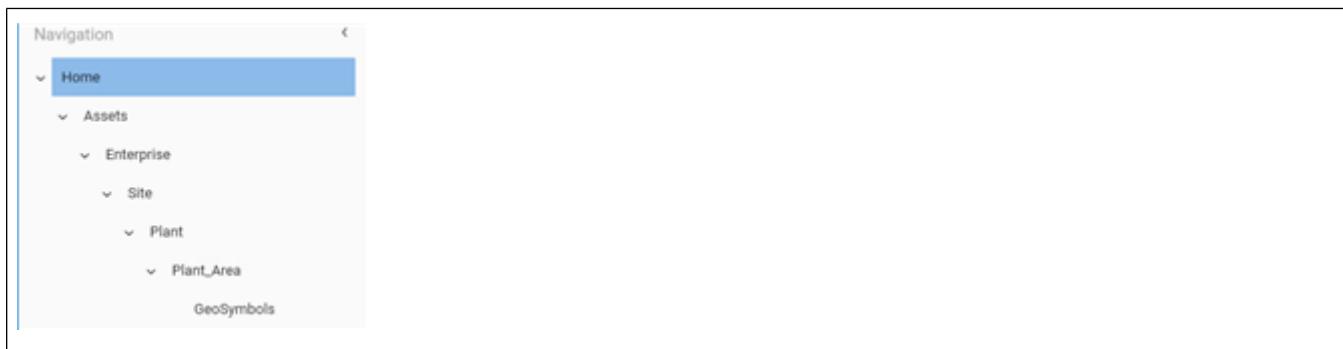
Search Levels Property Behavior

The **Search Levels** property includes four options to search for graphic items in the hierarchy of a ViewApp navigation model starting from the navigation node set as the start node by the **Navigation Start Node** property. **All** is the default option to search all hierarchical node levels, which includes the active node of the navigation model and all of its child nodes beneath it.

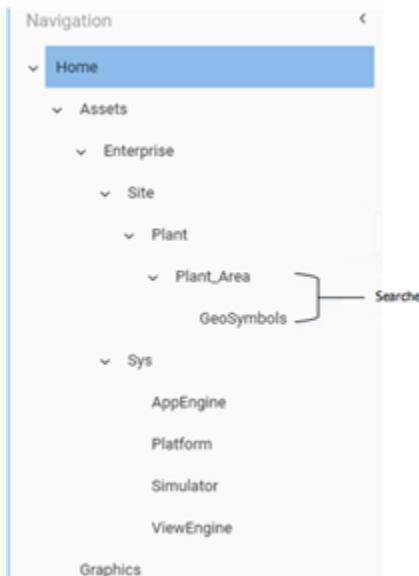


The remaining options of the **Search Levels** property set the scope of the graphic search within a ViewApp navigation model to the active node and the number of child nodes (1, 2, or 3) beneath the active node.

The following figure shows the navigation model of a ViewApp with its hierarchy of navigation nodes. The GeoSymbols node contains a set of geographic graphics.



The **Search Levels** property has been set to **SearchesOneLevel**. The **Navigation Search Mode** property is set to **CurrentNav**. The geographic graphics do not appear in the ContentPresenterApp unless the user sets the focus of the ViewApp to the Plant_Area or GeoSymbols nodes.



Contextualization

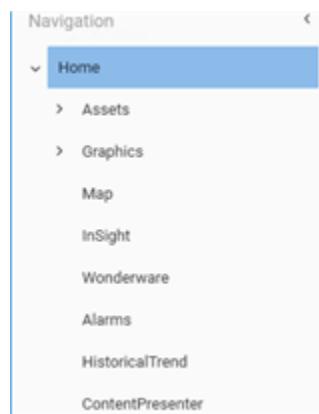
Content can be contextualized within the ContentPresenterApp. Contextualization is enabled by setting the **Follow Current Context** property to true. With contextualization enabled, the asset applicable to the currently-selected asset is used as the starting point for search. This is the default setting. If contextualization is disabled,

you must configure the **Navigation Start Node** property to provide a starting point for navigation.

Navigation Start Node property

\Home is the default value assigned to the **Navigation Start Node** property, which is the top navigation item or node shown by the ViewApp editor in an hierarchical navigation model. Based on default values assigned to other properties, the ContentPresenterApp shows all graphics assigned to a ViewApp because \Home is the top most node in the hierarchy of a navigation model.

Navigation Start Node is a dynamic, scriptable property. If **Follow Current Content** is not selected (disabled), **Navigation Start Node** must be set, either by data binding or through scripting.



The value assigned to the **Navigation Start Node** property restricts the display of graphics starting from a specific node in the hierarchy of the navigation model rather than \Home.

The **Navigation Search Mode** (legacy) property must be set to **FixedLocation** to enable the **Navigation Start Node** property. The graphics belonging to the starting node and the hierarchical child nodes of the starting node can be shown based on the value assigned to the **Search Levels** property.

ContentPresenterApp scripting

The ContentPresenterApp has two scriptable properties: Follow Current Context and Navigation Search Mode. Only one property, Navigation Search Mode is bindable, allowing you to change its value via scripting at run time.

Script statements can be specified using the MyContent Namespace format in layout scripting by the following syntax formats:

MyContent.name of the ContentPresenterApp instance.ContentPresenterApp_Method(parameter_1, parameter_2, parameter_n);

or

MyContent.name of the ContentPresenterApp instance.ContentPresenterApp_Property = xxx

Scripting Methods

The ContentPresenterApp includes the **Refresh()** scripting method for layout scripts.

Content Presenter OMI App and Layout Scripts

The following table summarizes ContentPresenterApp properties, their data types, and their default values. The

scriptable property name Navigation Start Node, when used in a layout script, is entered as a single word without blank spaces.

PI Vision App Properties	Data Type	Default Value	Scriptable	Bindable
Follow Current Context	Boolean	True	No	YES
Content Type	Enum	Any	No	No
Name Contains	String	*	No	No
Search Child Levels	Enum	Unconfigured	No	No
Navigation Search Options	Enum	Unconfigured	No	No
Navigation Start Node	String	\Home	YES	YES
Sort by Alphabetic Order	Boolean	False	No	No
Fill	Enum	Horizontal	No	No
View Mode	Enum	ListContent	No	No
Vertical Alignment	Enum	Top	No	No
Horizontal Alignment	Enum	Left	No	No
Padding	System.Windows.Thickness	0,0,0,0	No	No
Size Mode	Enum	Scaled	No	No
Fixed Width	Double	400	No	No
Display Columns	Integer	0	No	No
Display Rows	Integer	0	No	No
Stretch	Boolean	False	No	No
Fixed Height	Double	300	No	No
Interactive Content	Boolean	False	No	No
Trigger Navigation	Boolean	False	No	No

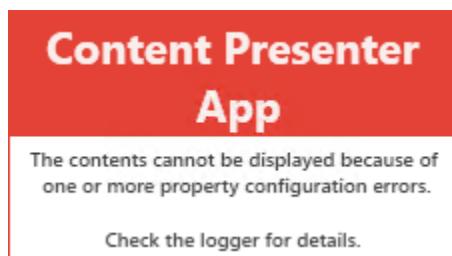
Search Levels	Enum	All	No	No
Navigation Search Mode	Enum	Fixed location	No	No
Background	System.Windows.Media.Brush	No color	No	No

ContentPresenterApp property validation

The properties listed in the ContentPresenterApp property grid are validated during design time to ensure proper configuration.

For most of the properties, design time validation occurs only when the property value is modified in the property grid. If the property is not modified in the property grid, validation will occur when the ViewApp is deployed to run time. Values entered for properties are checked, and if the required minimum configuration is not met, an visual cue indicating the error is shown in the app, and details of the error are sent to the Operations Control Logger.

When invalid ContentPresenterApp property values are detected at deployment, an error message appears when the ViewApp containing the ContentPresenterApp starts. The ViewApp continues running, but the pane containing the ContentPresenterApp shows an error message.



The ContentPresenterApp validates two sets of configuration property values:

- Invalid row or column counts

The **Display Rows** or **Display Columns** properties are set to a negative number.
- Invalid view port cell size

The **SizeMode** property is set to **Fixed** and the **Fixed Width** and **Fixed Height** properties are configured to 0.

Check the Operations Control Logger for one or more messages that describe the specific configuration error.

Graphic displays during runtime

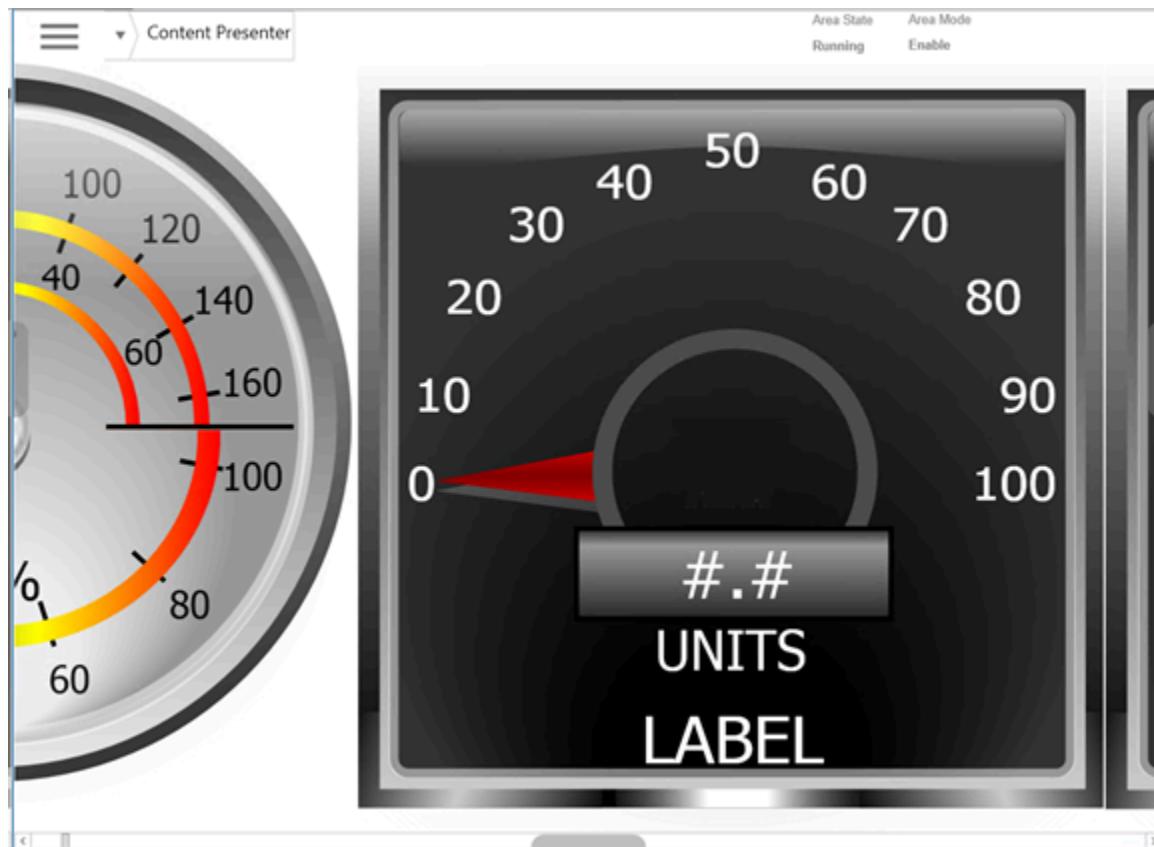
This section describes some typical graphic displays by configuring specific combinations of ContentPresenterApp properties. Each section shows an example screen shot of the ContentPresenterApp view port with specific combinations of property values.

List Graphics Horizontally and Vary the Number of Displayed Rows and Columns

The following examples show horizontal and scaled sequential lists of graphics when the number of displayed rows and columns vary.

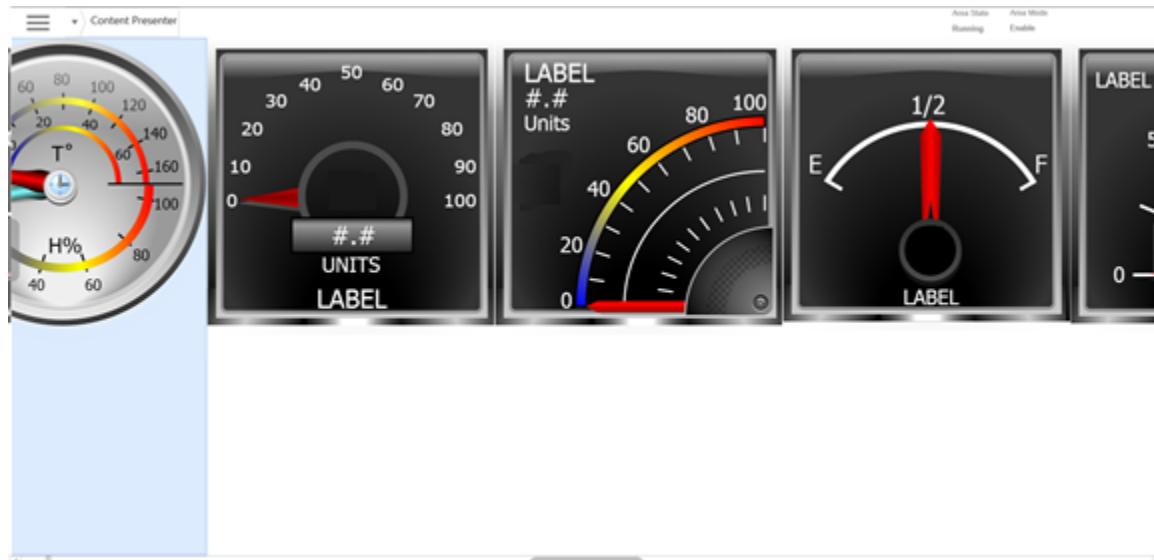
Horizontal List With No Specified Rows or Columns

Properties	Assigned Values
Fill	Horizontal
View Mode	ListContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Scaled
Display Rows	0
Display Columns	0
Stretch	Disabled
Fixed Width	200 ignored because Size Mode is set to Scaled
Fixed Height	300 ignored because Size Mode is set to Scaled



Horizontal List With Multiple Rows and Columns

Properties	Assigned Values
Fill	Horizontal
View Mode	ListContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Scaled
Display Rows	3
Display Columns	4
Stretch	Disabled
Fixed Width	200 ignored because Size Mode is set to Scaled
Fixed Height	300 ignored because Size Mode is set to Scaled



Graphics are smaller because the view port shows four columns of cells specified by the **Display Columns** property.

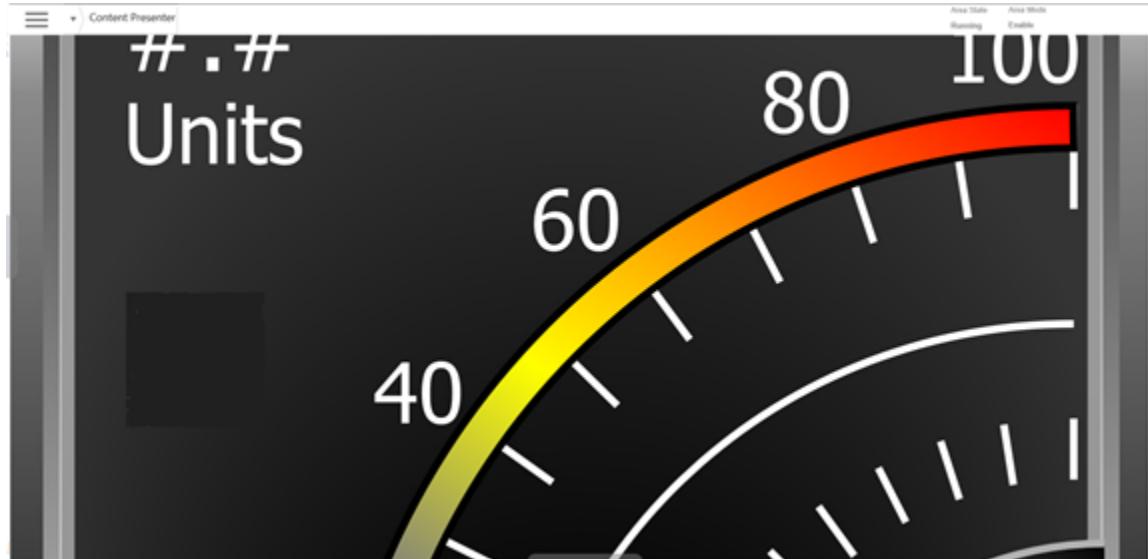
List Graphics Vertically and Vary the Number of Displayed Rows and Columns

The following examples show vertical and scaled sequential lists of graphics when the number of displayed rows and columns vary.

Vertical List With No Specified Rows or Columns

Properties	Assigned Values
Fill	Vertical

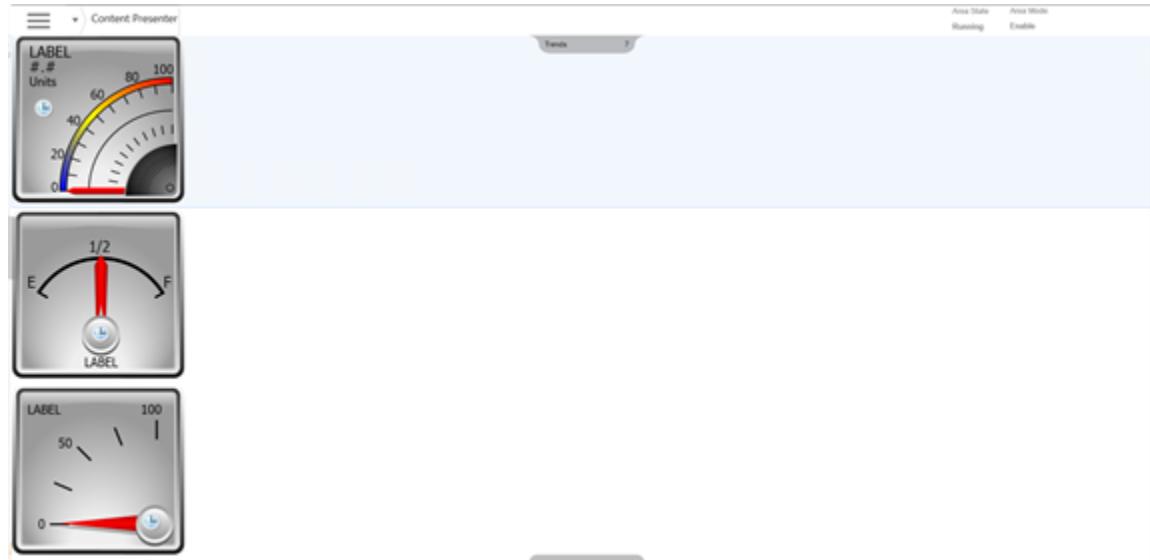
View Mode	ListContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Scaled
Display Rows	0
Display Columns	0
Stretch	Disabled
Fixed Width	200 ignored because Size Mode is set to Scaled
Fixed Height	300 ignored because Size Mode is set to Scaled



Vertical List With Multiple Rows and Columns

Properties	Assigned Values
Fill	Vertical
View Mode	ListContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Scaled

Display Rows	3
Display Columns	4
Stretch	Disabled
Fixed Width	200 ignore because Size Mode is set to Scaled
Fixed Height	300 ignore because Size Mode is set to Scaled



Graphics are smaller because the view port shows three of cells specified by the **Display Rows** property.

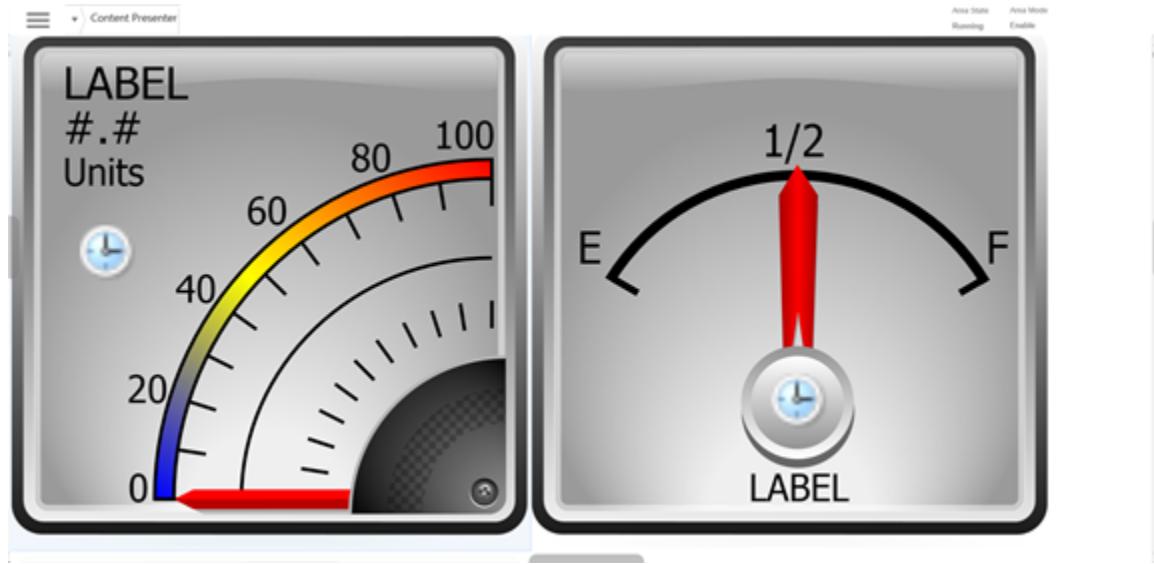
Wrapped Horizontal List and Vary the Number of Display Rows and Columns

The following examples show wrapped and scaled horizontal lists of graphics when the number of displayed rows and columns vary.

Wrapped Horizontal List With No Specified Rows or Columns

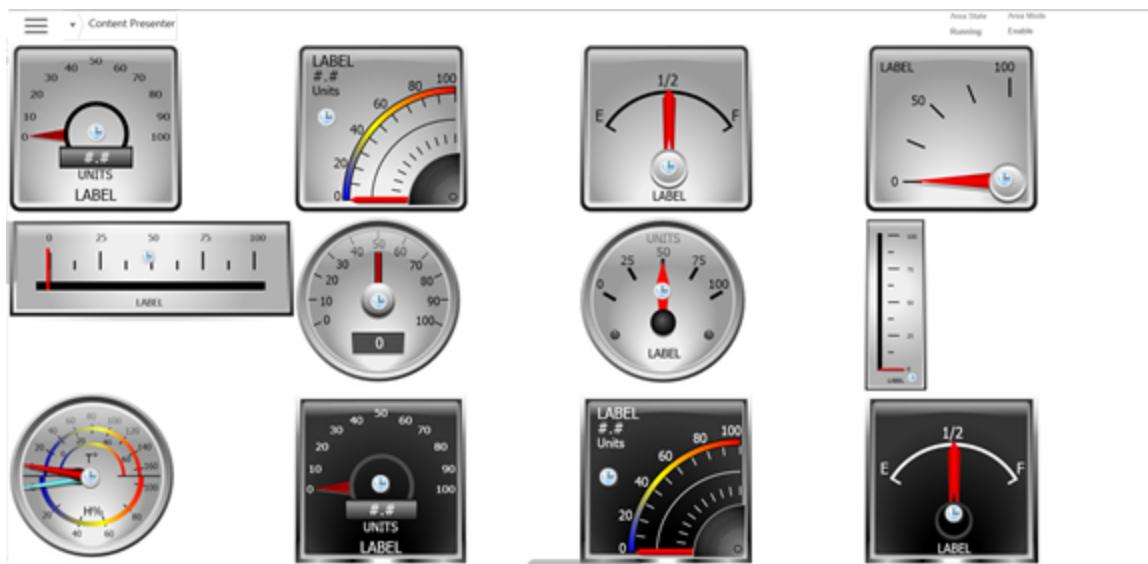
Properties	Assigned Values
Fill	Horizontal
View Mode	WrapContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Scaled
Display Rows	0
Display Columns	0

Stretch	Disabled
Fixed Width	200 ignore because Size Mode is set to Scaled
Fixed Height	300 ignore because Size Mode is set to Scaled



Wrapped Horizontal List With Multiple Rows and Columns

Properties	Assigned Values
Fill	Horizontal
View Mode	WrapContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Scaled
Display Rows	3
Display Columns	4
Stretch	Disabled
Fixed Width	200 ignore because Size Mode is set to Scaled
Fixed Height	300 ignore because Size Mode is set to Scaled



Graphics are smaller because the view port window has been configured to show four columns and three rows of cells. Graphic size adjusts to the constraining horizontal or vertical dimension of a cell while maintaining its original aspect ratio.

Wrapped Vertical List and Vary the Number of Display Rows and Columns

The following examples show wrapped and scaled vertical lists of graphics when the number of displayed rows and columns vary.

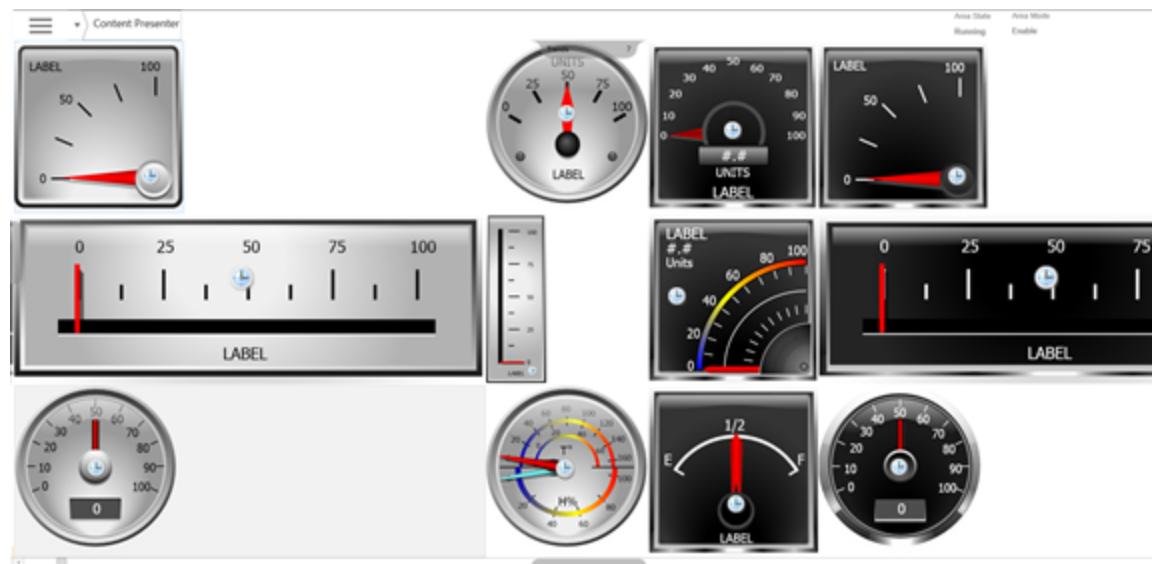
Wrapped Vertical List With No Specified Rows or Columns

Properties	Assigned Values
Fill	Vertical
View Mode	WrapContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Scaled
Display Rows	0
Display Columns	0
Stretch	Disabled
Fixed Width	200 ignore because Size Mode is set to Scaled
Fixed Height	300 ignore because Size Mode is set to Scaled



Wrapped Vertical List With Multiple Rows and Columns

Properties	Assigned Values
Fill	Vertical
View Mode	WrapContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Scaled
Display Rows	3
Display Columns	4
Stretch	Disabled
Fixed Width	200 ignore because Size Mode is set to Scaled
Fixed Height	300 ignore because Size Mode is set to Scaled



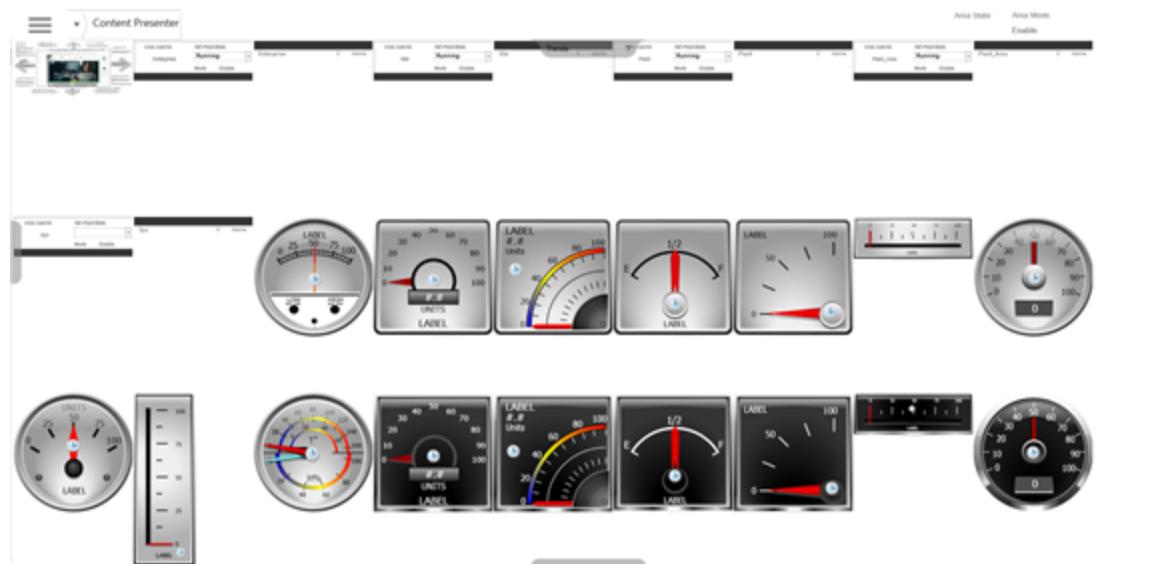
Cells are listed in vertical order by four columns and three rows. Graphic size adjusts to the constraining horizontal or vertical dimension of a cell while maintaining its original aspect ratio.

Wrapped Horizontal List with Fixed Size Cells and Vary the Number of Rows and Columns

The following examples show wrapped and horizontal lists of graphics in fixed size cells when the number of displayed rows and columns vary.

Wrapped Horizontal List With a Fixed Cell Size and No Specified Rows or Columns

Properties	Assigned Values
Fill	Horizontal
View Mode	WrapContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Fixed
Display Rows	0
Display Columns	0
Stretch	Disabled
Fixed Width	200
Fixed Height	300



View port cells are arranged horizontally. When the next cell cannot fit within the view port area, the horizontal list wraps to the next row. Users can scroll outside of the view port using the vertical scroll bar.

Wrapped Horizontal List With a Fixed Cell Size and Multiple Rows and Columns

Properties	Assigned Values
Fill	Horizontal
View Mode	WrapContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Fixed
Display Rows	3
Display Columns	4
Stretch	Disabled
Fixed Width	200
Fixed Height	300



Graphics are placed in fixed size view port cells. Graphics are arranged in the view port horizontally in four columns and three rows.

Wrapped Horizontal List With Scaled Graphics in Stretch Mode

Properties	Assigned Values
Fill	Horizontal
View Mode	WrapContent
Padding	0,0,0,0
Horizontal Alignment	Left
Vertical Alignment	Top
Size Mode	Fixed
Display Rows	3
Display Columns	4
Stretch	Enabled
Fixed Width	200
Fixed Height	300



Graphics are placed in fixed size view port cells and stretched in their non-constrained dimension to completely fill the cell. The aspect ratio of each graphic is ignored.

Upgrade and migration of the ContentPresenterApp

The ContentPresenter app has been updated for the AVEVA System Platform 2023 release. The following new properties were added to normalize navigation and search behavior with other AVEVA OMI apps:

- **Search Child Levels:** This property lets you specify how many child levels down in the hierarchy from the current level are searched. Options are Unconfigured, All, OneLevel, TwoLevels, and ThreeLevels. The default is **Unconfigured**, which allows a migrated app to maintain its existing behavior by using deprecated properties options to determine how the search is conducted. This property only has meaning if **Navigation Search Options** has **ChildLevels** enabled.
- **Navigation Search Options:** This property lets you specify how searches within the app should begin, and what nodes should be included in the search. Options are Unconfigured, StartNode, Siblings, and ChildLevels. The default is **Unconfigured**, which allows a migrated app to maintain its existing behavior by using deprecated properties options to determine how the search is conducted. Multiple options can be enabled. Thus, to allow a search across the three child levels of the hierarchy from the search start point plus the start point and its siblings, you would enable ThreeLevels for the **Search Child Levels** property, and enable StartNode, Siblings, and ChildLevels for the **Navigation Search Options** property.

The following existing properties were deprecated. The deprecated properties are maintained to ensure backward compatibility with prior releases, and are now listed under the property category "Legacy Properties."

- **Search Levels:** This property is only used by the app if **Search Child Level** and **Navigation Search Options** are left in their default **Unconfigured** state. The default remains as **All**.
- **Navigation Search Mode:** This property is only used by the app if **Search Child Level** and **Navigation Search Options** are left in their default **Unconfigured** state. The default remains as **FixedLocation**.

The ContentPresenterApp can be migrated to subsequent versions of System Platform as they become available. You can export the ContentPresenter app to an .aaPKG file and then import the aaPKG file into a future release of System Platform.

DocViewerApp

The DocViewerApp shows a Word document within a pane of a ViewApp while a ViewApp is running or displayed in preview mode. The DocViewerApp initially shows the first page of the document at 100 percent zoom scale.

The DocViewerApp provides functionality similar to viewing a Word document in read-only mode. A document can be searched and the document view can be enlarged or reduced in size. But, no permanent editing changes can be made to a Word document while displayed by the DocViewerApp.

Standard Operating Procedure for:

Use and maintenance of vacuum pumps

Introduction
This SOP covers the setup, basic use, and maintenance of rotary vane vacuum pumps. These pumps can be used for many applications. However, check with appropriate documentation before using a pump for a new application.
If flammable materials are to be pumped A SEPARATE RISK ASSESSMENT IS REQUIRED. This risk assessment and SOP do not cover use of gas ballast. Gas ballast is used when pumping condensable vapours in high concentrations.
Ensure appropriate PPE is worn during maintenance **including laboratory coat, safety glasses and nitrile gloves**.

Before use
These are general points to consider before operation of a vacuum pump. The list is not comprehensive and every situation should be assessed on its own merits.

- 1) Make sure pump has an up to date PAT certificate
- 2) Check that the pump oil-level is between the MAX and MIN marks on the bezel of the oil-level sight-glass; if it is not, DO NOT USE.
- 3) Ensure that vessel to be evacuated is appropriately set up. If unsure **ALWAYS ASK**. Never set up a new vacuum system (especial glass) on your own unless experienced.
- 4) Ideally use a suitable inlet-valve to isolate the pump from your vacuum system, important:
 - i) if you need to allow the pump to warm up before you pump condensable vapours.
 - ii) if you need to maintain vacuum when the pump is switched off.
- 5) Avoid high levels of heat input to the pump from the process gases.
- 6) Ensure that any heating of the pump body will not affect any other equipment or cause a danger to others.
- 7) Make sure that the exhaust pipeline is either vented to a safe location or filtered. Also ensure that it cannot become blocked. **Check expiry date on filter, if it has expired DO NOT USE.**
- 8) Make sure pump is on a firm, level platform and it is located so that the oil-level sight-glass is visible.

Maintenance schedule

Operation	Frequency	Operation	Frequency
Check oil level	At least monthly	Clean the motor fan-cover and enclosure*	Yearly
Replace Oil	3000 hours or yearly	Clean and overhaul the pump (test motor)*	15000 hours
Inspect and clean the inlet or outlet filter	Yearly	Fit new blades*	30000 hours

*not covered in this SOP

Pump oil

PPE required:

-
-
-

Hazard symbols:
See individual users risk assessment

Significant hazards:

- High temperature

Hazard phrases (H):
See individual users risk assessment

Can it be done out of hours?
Vacuum pumps can be operated out of hours but maintenance should not be

PAGE: 1 OF 0 | 969 Words | 100% 0

The DocViewerApp includes a set of visual controls that appear in the command bar above the displayed document. These controls can be used to change the zoom level of a document while it is being displayed and search for a word or phrase within the document.



The DocViewerApp also includes a slider zoom control in the status bar beneath the displayed document.

Moving the slider left or right zooms the document view in or out.



Change the zoom level of a document

The DocViewerApp provides several methods to set the zoom level of a document page shown in preview mode or a running ViewApp. A page can be zoomed out to a minimum of nine percent of its original size and zoomed in to a maximum of 500 percent.

- Zoom in and zoom out controls



These controls appear above the displayed document on the menu bar. Each time the user selects a zoom in or zoom out control, the view magnification increases or decreases by ten percent.

- Zoom slider control



The slider control appears beneath the displayed document on the status bar. Moving the slider right or left increases or decreases the magnification of the document view.

Users can also select the + or - controls at the ends of the slider. Each press of the + control increments magnification of a document by 10 percent. Each press of the - control decrements magnification of a document by 10 percent.

Touch support

The DocViewerApp supports single finger gestures on touch screens or portable devices.

- A Word document can be swiped or flicked vertically to show the previous or next pages of a document.
- A single finger tap can be used to select the visual icons shown within the viewer window of the DocViewerApp.

See [Optimize the DocViewerApp for a touch device](#) for instructions to enlarge the size of the visual icons to make it easier to select them while using touch gestures.

Optimize the DocViewerApp for a touch device

The visual icons shown within the viewer window of the DocViewerApp can be enlarged to make them easier to select on touch screens or portable devices.

The following screen captures show the default size of the DocViewerApp's visual icons and their size after they have been enlarged for touch. AVEVA OMI includes the `OptimizeForTouch` Boolean attribute that belongs to the `MyViewApp.Settings` namespace. The default value of `OptimizeForTouch` is false. When `OptimizeForTouch` is set to true, the icons are enlarged.

OptimizeForTouch Attribute Value	Icon Size	Example Icons
MyViewApp.Settings.OptimizeForTouch=False	Regular Size (Non-Touch)	  
MyViewApp.Settings.OptimizeForTouch=True	Optimized for Touch	  

A common way to dynamically adjust the size of the visual icons in a running ViewApp is to place a graphic element like a button on a pane and then associate user input animation to toggle the state of the OptimizeForTouch attribute.

Compatibility of the DocViewerApp

The DocViewerApp includes compatibility with current and future versions of System Platform.

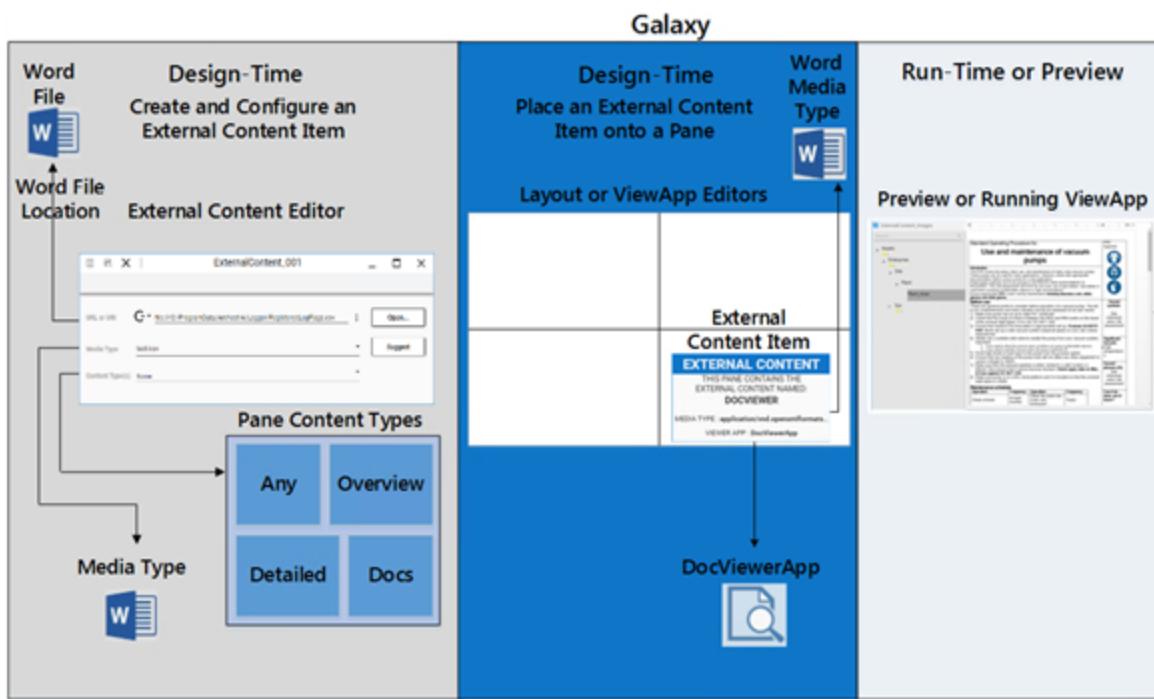
- The DocViewerApp package must be installed on a computer running System Platform 2017 Update 3. The DocViewerApp cannot be installed on any other version of System Platform except Update 3.
- After the DocViewerApp is installed on a System Platform 2017 Update 3 system, the DocViewerApp will be able to migrate to future versions of System Platform InTouch OMI when they become available.

Configure the DocViewerApp to show Word documents

The following figure summarizes using an External Content item to point to a remote computer containing a Word document. An External Content item is configured during design time to identify the location of the Word document and its media type.

Note: The DocViewerApp itself is not configured to show a Word document in a ViewApp.

The External Content item is associated with an asset or a user defined object. The asset or object is selected from the **Toolbox** and placed onto a layout pane. During runtime, the user selects an asset associated with a navigation item to view training procedures described in the Word document.



The major steps to implement the DocViewerApp to show Word documents in a ViewApp are summarized in the following list:

1. Place a Word document on a computer accessible from your Galaxy.
2. Create an External Content item.
3. Configure the External Content item that identifies the location and media type of the Word document.
4. Associate an External Content item to a graphic or object.
5. Place an asset or graphic on a ViewApp pane associated with the External Content item.

Before you begin

Before starting the steps to configure an External Content item, ensure the following prerequisites have been met:

- If you are placing the Word document on a remote computer in your network, ensure connectivity between the computer where the ViewApp will run and the computer where the remote Word document has been saved.

Mapping a network drive is a typical way to connect to a remote computer hosting documents that you want show in a running ViewApp. For this configuration to work successfully, it is important that the computer to which the ViewApp is launched from has the same drive letter mapped to a location having the referenced document.

The DocViewerApp shows an error message to the user if the specified URI connection to the Word file is bad. For more information about the error message, see [DocViewerApp error message](#).

- Save the Word document in the presentation format that you want to display in the DocViewApp.

Typically, you want to format the Word document in its intended final form without showing any revision markup.

Important: The DocViewerApp supports both the legacy .doc Word file type and the current docx file type.

DocViewerApp error message

An External Content item includes an **URL or URI** option to specify the location of the Word file to be shown by the DocViewerApp during runtime or in preview mode. The specified location of the file must be complete and accurate. If the file is moved, deleted, or renamed, the DocViewerApp shows an error message to the user during runtime, but the ViewApp continues to run.



Create an External Content item

You create an External Content item from the Visualization folder of the System Platform IDE. Initially, the External Content item defaults to a set of values that can be changed using the External Content editor.

To create an External Content item

1. Open the IDE and select the **Visualization** tab.
2. Select a folder within the **Visualization** folder if you want to create a new External Content item at a specific location.
3. On the **Home** ribbon, in the **Create** area, select **External content**.

You can also create an External Content item by other methods:

- **Keyboard Shortcut**

Press Ctrl + Shift + C

- **Shortcut Menu**

Right-click a folder of the **Visualization** folder to show a shortcut menu. Select the **New** option, and then select **External Content**.

A new **External Content** item is created in the **Visualization** folder.

 ExternalContent_001 * * *

The name of the new item follows a default naming convention of appending a three-digit number to the word **ExternalContent**.

4. Rename the External Content item.

After you create an External Content item, it must be configured to specify the remote location of the content and its media type.

Configure an External Content item

Each External Content item has three properties that must be assigned values:

- **URL or URI**
- **Media Type**
- **Content Type(s)**

An External Content item is modified from the External Content editor. The editor can be opened by double-clicking on an item from the Visualization folder. Also, an External Content item that has been placed onto a layout pane can be edited from the Layout and ViewApp editors by selecting the item from the Actions list.

To configure an External Content Item

1. Open the IDE and select the **Visualization** folder to show the External Content items available in the Galaxy.
2. Double-click an External Content item to open it in the External Content editor.
For more information about each field in an External Content item, see [Details of the External Content editor](#).
3. Enter a URI string where the Word document is saved.

The value you enter is the location of the Word file specified by a URI-formatted string. A media location must be specified.

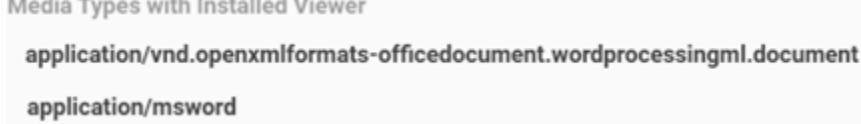
You can browse for the Word file by selecting the vertical dots icon to the right of the data entry field. A drop-down list includes a **Browse for file** option that enables you to browse your network and select a media file. The **URL or URI** field updates and shows a formatted URI path to the file.



See [More information about specifying a URI](#) for information about the format of a URI string.

4. If you want, select the **Open** button to validate the URI you entered.
The **Open** button launches the appropriate application associated with the doc or docx file types, which is usually Word.
5. Select the type of media associated with Word documents from the drop-down list of the **Media Type** field.
The value you enter is the MIME type of a Word document. The DocViewerApp provides two Word MIME types, as shown in the following screen capture.

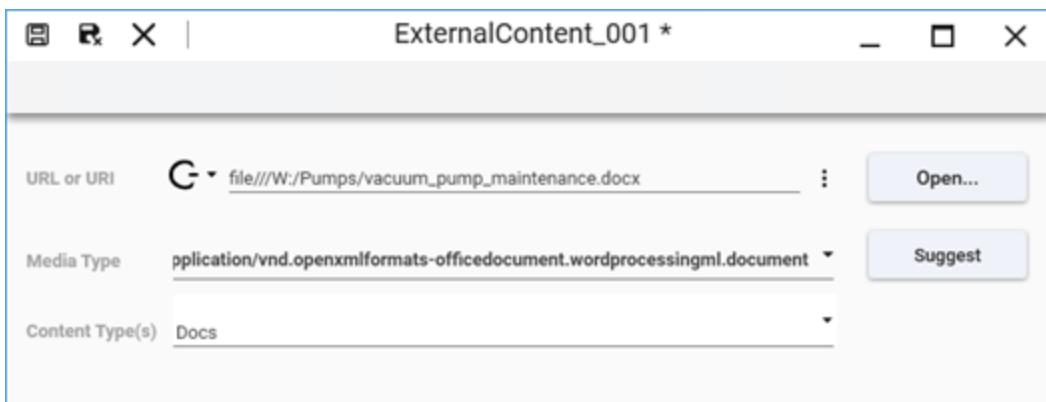
- The application/msword Media Type is for legacy .doc Word files.
- The application/vnd.openformats-officedocument.wordprocessingml.document Media Type is for .docx Word files.



See [More information about media types](#) for information about supported media types.

6. Select the type of content from the drop-down list of the **Content Type(s)** field.
The value you enter is used by content-placement algorithms within AVEVA OMI during run time to determine the placement of content when AutoFill or ShowContent calls are invoked.

The configured External Content item for a Word document should look like the following screen capture.



7. Save your changes and exit from the External Content item.

Link an External Content item to an object or template

There are three ways to integrate an External Content in a ViewApp.

- Associate an External Content item to an object template, which is described in the following procedure
- Associate an External Content item to a custom navigation item that appears in the navigation model of the ViewApp.
- Place the External Content item on a layout pane, which is the simplest way and is described in [Adding an External Content item to a ViewApp](#).

To link an External Content item to an object template

After you have created and configured an External Content item, you can link it to an object template. External Content items can be linked only to objects as opposed to graphics, which can be linked to or owned by an object. All External Content items reside in the Visualization folder. A single External Content item can be linked to multiple objects.

1. Open an object template in the **Object Editor**.
2. Select the **Attributes** tab.
3. In the **Content** pane, select the **Link Content**  button.

The Galaxy Browser opens.

4. Navigate to the folder that contains the **External Content** item to be linked.
5. Select the item, then click **OK**.

The item is added to the object and appears in the **Content** tab.

Note: After linking an External Content item, you can modify it by selecting it and then pressing the **Edit** button. However, any changes you make will apply to all objects that link to the item.

6. Save your editing changes to the object template and exit from the Object Editor.
7. Create an instance of the linked object template.
8. Create a layout which contains pane with a Content Type that matches the content type of the External Content item.
9. Add a navigation tree to the layout
10. Create a ViewApp that incorporates the layout.

11. Deploy the ViewApp.

In runtime, navigate to the object or any instance created from the template. External Content item will auto-fill the matching pane to show the external content.

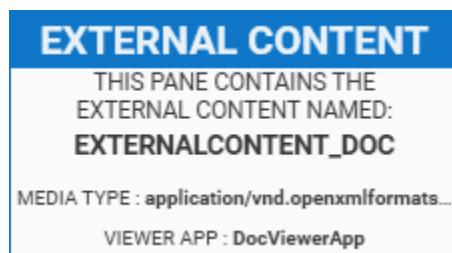
Associate an External Content item with a ViewApp

You can place an External Content item directly onto a pane from either the Layout or ViewApp editors . External Content items are listed in the **Toolbox** or **Assets** tabs of either editor.

To associate external content to a ViewApp

1. Open the System Platform IDE and select the **Visualization** folder to see the list of layouts.
2. Select a layout that you want to insert external content and open it in the Layout editor.
3. Select the **Toolbox** tab of the Layout editor to show the list of content accessible from the Galaxy.
4. Locate the External Content item you want to place in the layout you selected.
External Content thumbnails appear beneath of the list of content shown in the **Toolbox** list.
5. Select an External Content thumbnail from the **Toolbox** area, and then drag and drop it onto a pane of a layout incorporated into a ViewApp.

The External Content thumbnail appears in the pane containing the name of the External Content item, the media type of the external content, and the viewer app that will display the media during runtime. Also, the layout **Actions** area shows the name of the External Content item and the pane that it was placed in.



Editing an External Content Item After It Is Placed on a Pane

An External Content item can be modified after being placed in a layout or a ViewApp, which automatically propagates the changes

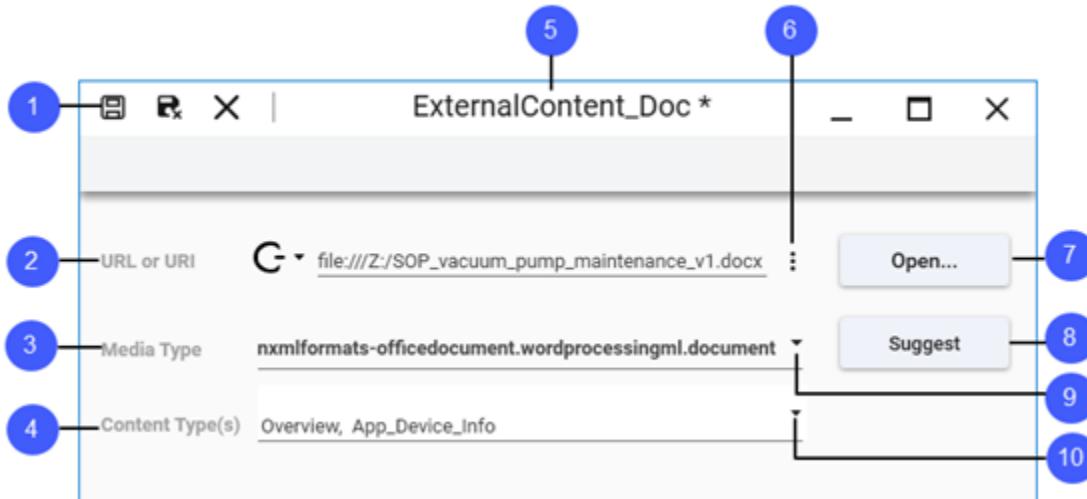
To edit an External Content item in a layout

1. If necessary, open the layout containing the External Content item.
2. Select the **Options** icon to the right of the item in the **Actions** list.
A drop-down list shows an **Edit External Content** command.
3. Select **Edit**.
The External Content editor opens the item for editing.
4. Update the values assigned to the item and save your changes.
The External Content thumbnail placed in a pane updates to reflect the changes made to the item.

Details of the External Content editor

You use the External Content editor to configure an External Content item for a document. Before you configure an External Content item, you must know:

- Network location of a document
- Media type of the document
- Content type(s) of the layout pane that will host the External Content item



1	Commands to save, save and close, and close the External Content editor.
2	URL or URI that specifies the location of external media using a standard format. See More information about specifying a URI for the format of a URI or URL A warning message appears if the document cannot be located during runtime.
3	A media type is a two-part identifier that specifies the type of application required to process or view remote content. A media type can be entered in the field or selected from a drop-down list. See More information about media types for the format of a Media type value. A warning message appears if a media type is specified that does not have an associated viewer application.
4	Content type assigned to the external media that enables ViewApp algorithms to place content in specific panes during runtime.

5	Name of the External Content item.
6	Drop-down list with options to browse for an external content file or specify a URL using HTTP or HTTPS.
7	Validates the location of external media specified in the URL or URI field. An attempt is made to display the external media in an application assigned as the default by the operating system, not the app specified for the external content media type. A warning message appears if the content cannot be found at the location specified in the URL or URI field.
8	Suggest is optional. When selected, it automatically selects a media type based on the entry in the URL or URI field. The entered value can be changed if the suggestion does not match the expected media type. Note: Ensure the suggested media type is correct. The external media does not appear if the media type is incorrect.
9	Drop-down list of commonly used media types. Media types that have apps in the Galaxy capable of servicing them are shown in bold text.
10	Drop-down list of content types that can be used at preview\runtime to assign the external content to panes of a matching content type. Note: None is not a content type. Instead, it simply means a content type has not been assigned to the pane.

More information about media types

A media type is a standard two-part string analogous to a MIME type that identifies external content file types and their format. System Platform uses a media type to identify external content and the type of app required to display media during runtime.

A media type consists of a type and a subtype, which is further structured into a tree.

type "/" [tree "."] subtype ["+" suffix] *[";" parameter]

For more information about the format of media types, see <https://www.iana.org/assignments/media-types/media-types.xhtml>

System Platform provides several default media types that include associated apps to display media. The drop-

down list of the **Media Type** field shows the default media types in bold text to indicate a viewer app is available in the Galaxy that can service external content with these media types. Other listed media types that appear in plain text require an app to be created and imported into a Galaxy to play the specified media.

More information about specifying a URI

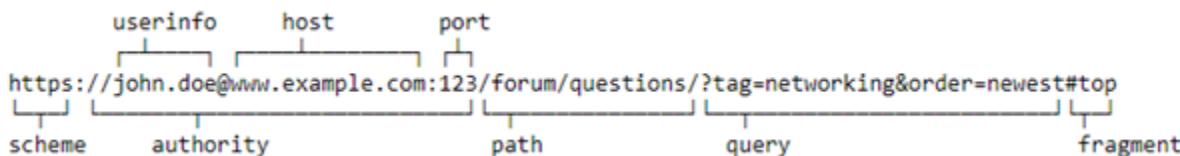
In the context of External Content, a Uniform Resource Identifier (URI) is a string that identifies the pathway to specific content that can appear in a running ViewApp.

All URIs adhere to predefined syntax rules.

- Generic URI syntax consists of a hierarchical sequence of components

```
URI = scheme:[//authority]path[?query][#fragment]
```

Example



- Each URI begins with a scheme name followed by a colon
Examples of common scheme names include http:, https:, and file:
 - An optional authority component preceded by two slashes (//)
 - Use information component consisting of a user name and optional password preceded by a colon followed by an at symbol (@)
`//username:password@`
 - Host subcomponent consisting of a registered hostname or IP address
 - A path component consisting of a sequence of path segments separated by a slash (/). A path is always defined for a URI, though the defined path may be empty
 - An optional query component preceded by a question mark (?), containing a query string of non-hierarchical data.

For more information about the format of a URI, see <https://tools.ietf.org/html/rfc3986>

The following examples show common URIs that identify External Content within a Galaxy

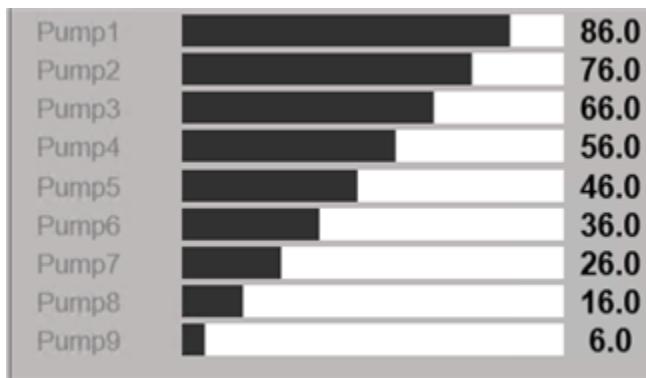
- `ftp://ftp.is.co.za/rfc/rfc1808.txt` -- ftp scheme for File Transfer Protocol services
- `file:///Z:/Media/Runtime%20Language%20Switching.mp4`
- `http://www.exampleserver.com/documents/index.html`

AVEVA GraphicRepeaterApp

The AVEVA™ GraphicRepeaterApp shows a graphic that is repeated multiple times based on custom properties assigned to different references. During runtime, the graphic appears in a repeatable list sorted by a single referenced custom property value common to each graphic in the list. Hereafter, the AVEVA™ GraphicRepeaterApp is referred to as the GraphicRepeaterApp.

The GraphicRepeaterApp incorporates the functionality of the SetCustomPropertyValue() method API to change the expression or reference value of a repeatable graphic custom property. No user configuration of the method is required.

The GraphicRepeaterApp is particularly useful for visualizing a common process value across multiple pieces of equipment simultaneously. The following figure shows the GraphicRepeaterApp displaying a common process value for a set of pumps.



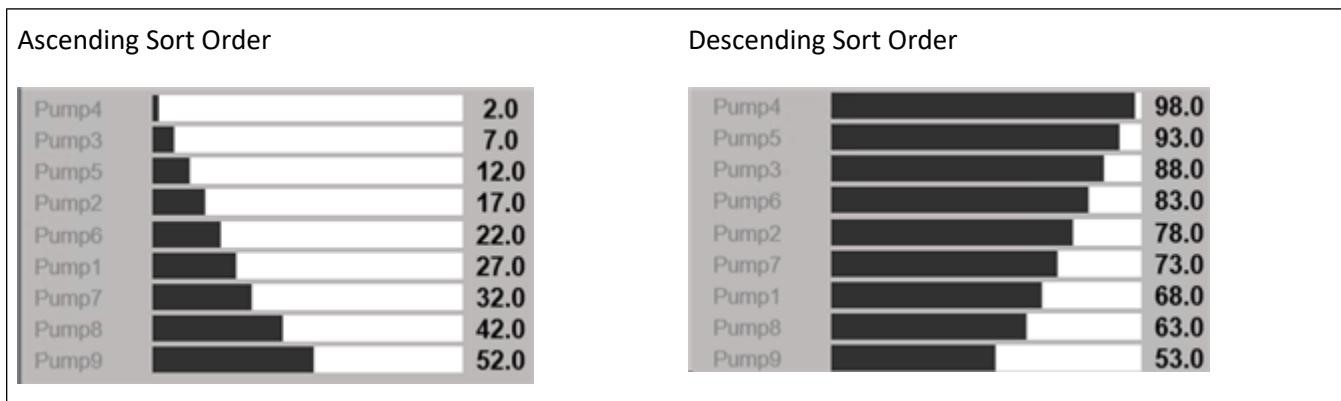
Visual and Functional Characteristics of the GraphicRepeaterApp

The GraphicRepeaterApp includes layout properties that modify the appearance of repeatable graphics during runtime and functional properties that filter and sort the list of repeatable graphics. For a complete list of properties that set visual and functional characteristics of the GraphicRepeaterApp, see [Configure GraphicRepeaterApp properties](#).

Visual Characteristics

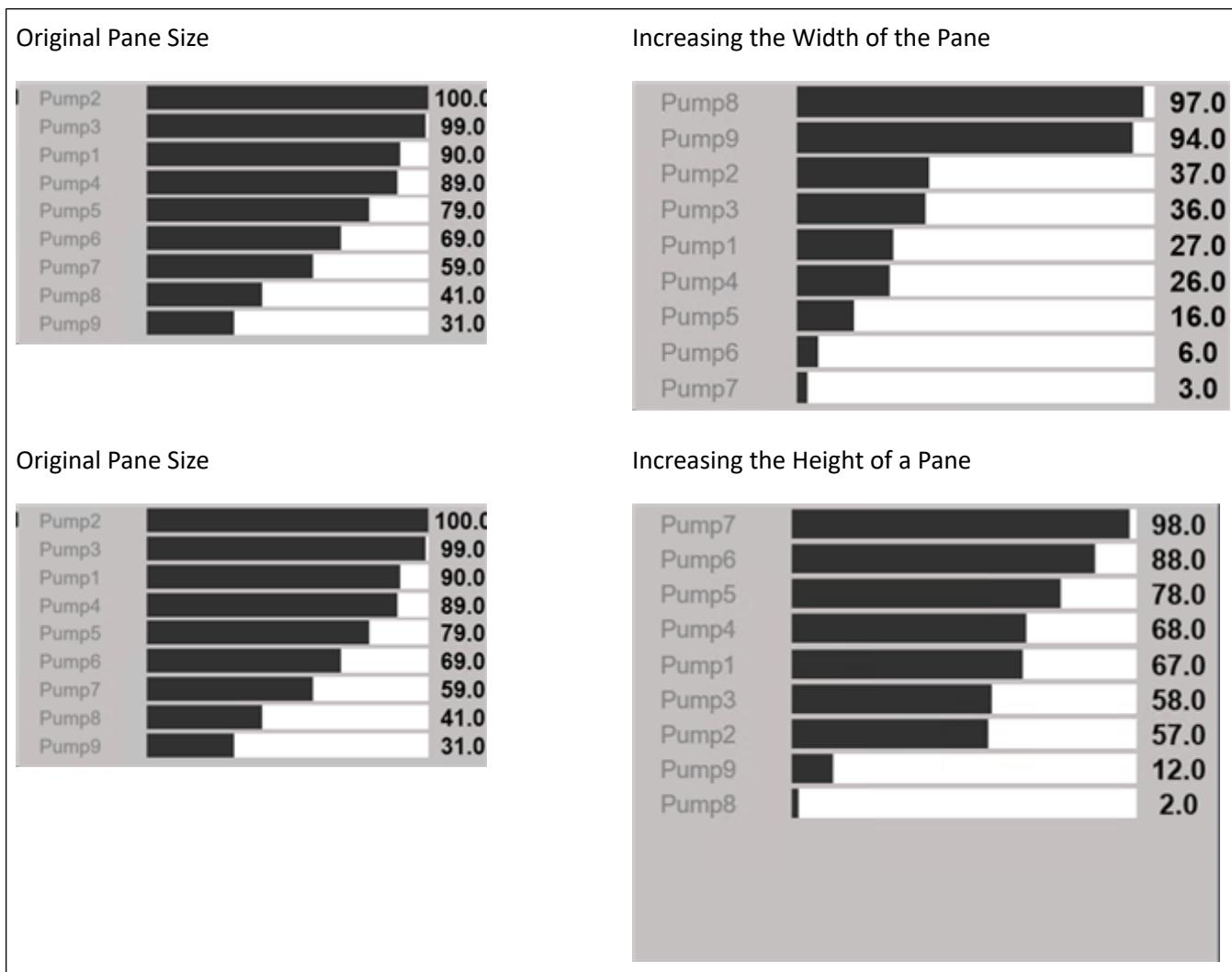
The GraphicRepeaterApp includes a **Layout** group of properties that determine the visual appearance of repeatable graphics during runtime.

The GraphicRepeaterApp includes the **Sort Order** property to arrange repeatable graphics to appear in ascending or descending order based on the value of the custom property specified for the **Sort By** property.



The size of the GraphicRepeaterApp shown during runtime is based on the size of the layout pane in which it has been placed. Changing the vertical or horizontal dimensions of a pane during runtime changes the size the app.

The following figures show how the GraphicRepeaterApp changes size by increasing the width and height of the pane in which it is placed. In these examples, the fill style of the repeatable graphics are arranged vertically.



The GraphicRepeaterApp includes a **Fill Style** property, which includes options to arrange repeatable graphics horizontally or vertically. When **Fill Style** is set to vertical, repeatable graphic size adjusts to the width of the pane. When **Fill Style** is set to horizontal, repeatable graphic size adjusts to the height of the pane.

Repeatable graphics can be shown in specified numbers of rows and columns by assigning values to the **Display Columns** and **Display Rows** properties. The following examples show repeatable graphics arranged in two columns and three rows.

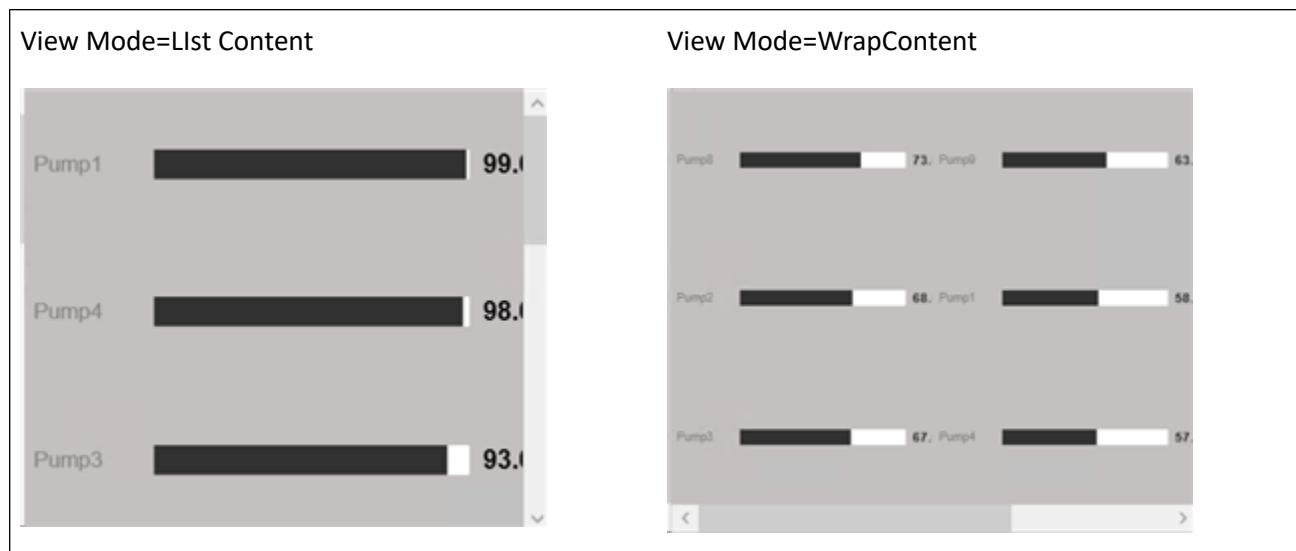
However, the value of the **View Mode** property determines how the columns and rows appear in the pane.

- **View Mode=ListContent**

Graphic items are shown in the GraphicRepeaterApp as a continuous sequential list. The GraphicRepeaterApp includes scroll bars to view graphics that extend beyond the bottom or right border of the pane.

- **View Mode =WrapContent**

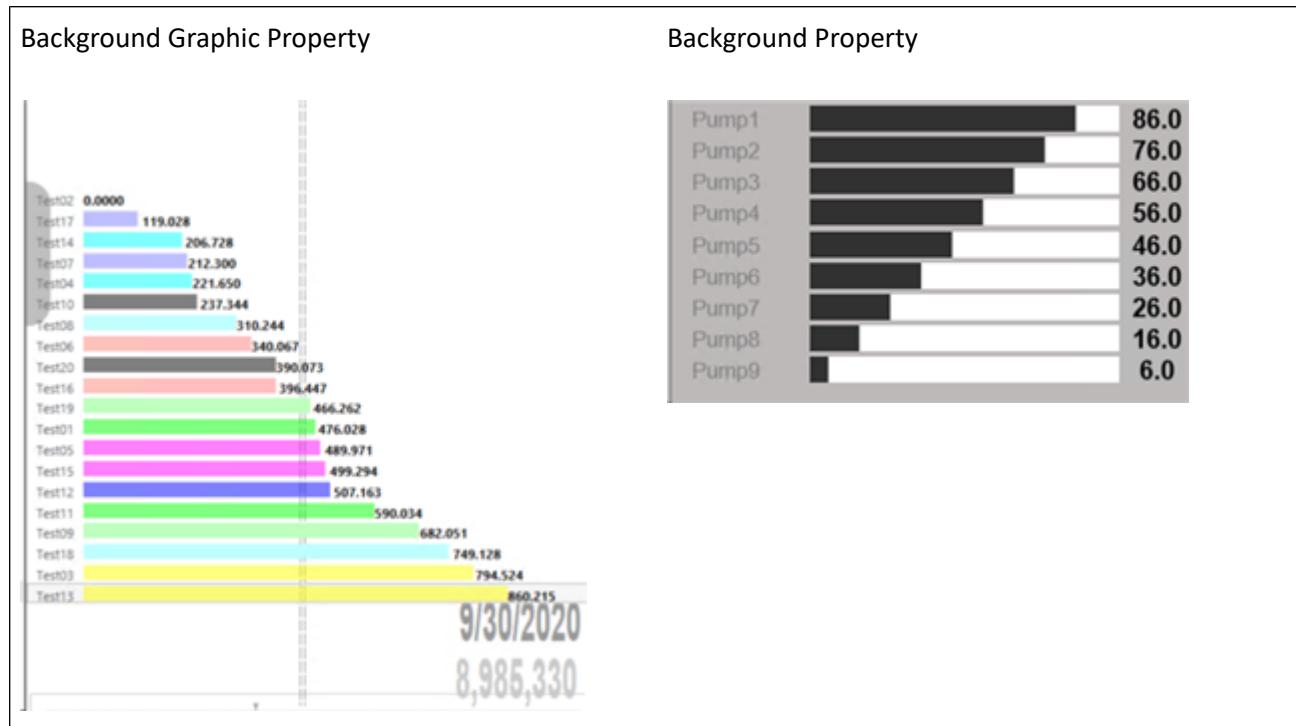
Repeating graphics are listed in the fill direction until a graphic does not fit within the border of the pane. The graphic then wraps to the next column or row within the pane.



Scrollbars are included when repeatable graphics appear outside the pane. If the GraphicRepeaterApp runs on a touch monitor or smart phone, the pane can be scrolled horizontally or vertically with a flick gesture left and right or up and down.

The GraphicRepeaterApp provides two properties to set a background behind the list of repeatable graphics during runtime.

- The **Background Graphic** property provides the capability to specify a graphic that is placed behind the repeatable graphic and serves as the background
- The **Background** property includes a color picker to set a background color that appears over the entire surface of the pane in which the GraphicRepeaterApp has been placed.



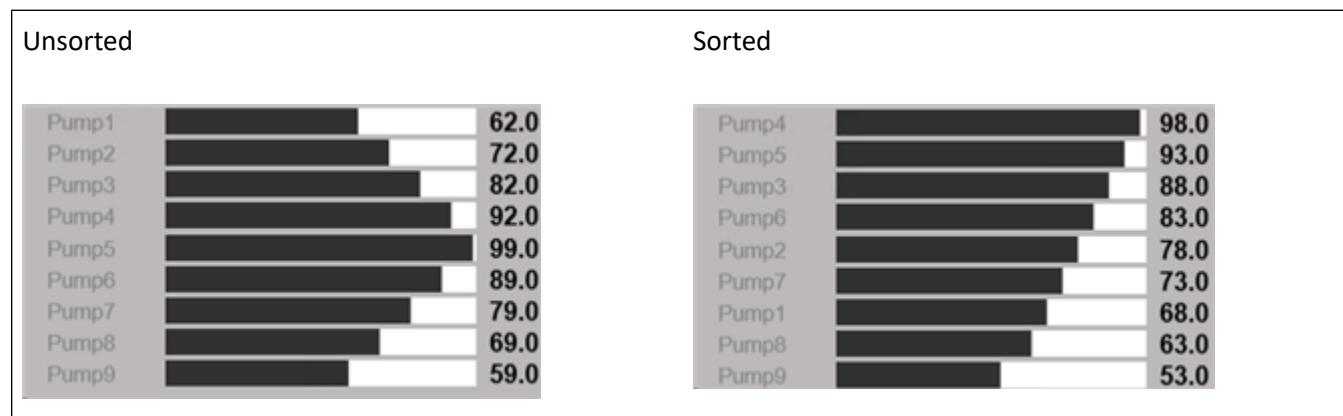
The background properties are not mutually exclusive. Both properties can be specified simultaneously for an

instance of the GraphicRepeaterApp.

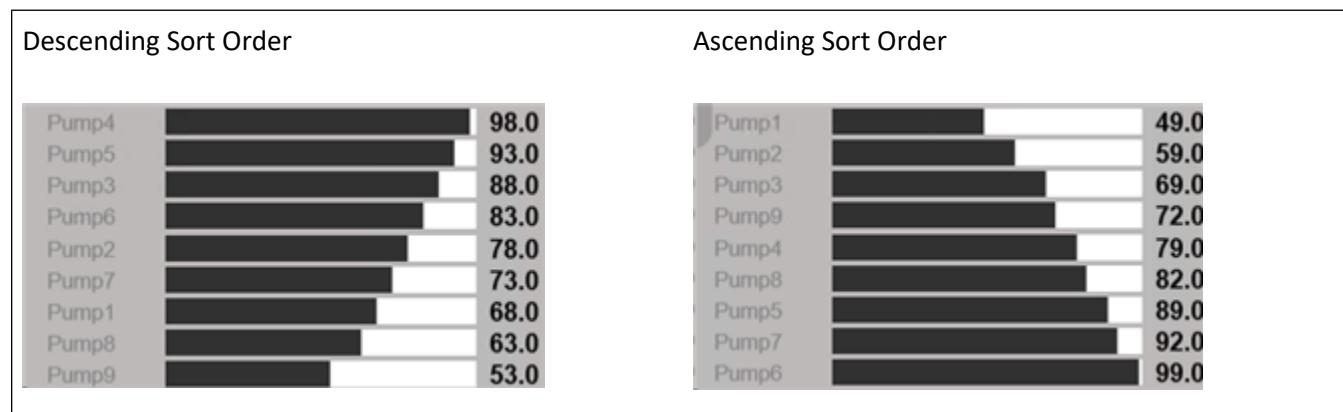
Functional Characteristics

The GraphicRepeaterApp includes a **Sort** group of properties that determine if the list of repeatable graphics are sorted during runtime, their sort order, and if filtering is applied to show only a specified number of graphics with the highest or lowest values.

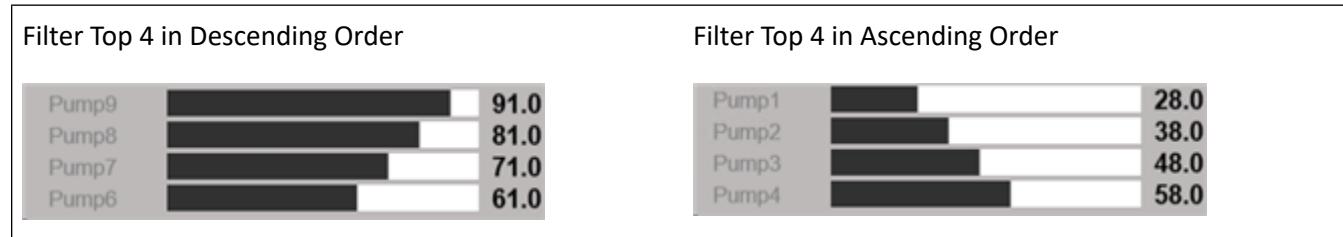
By default, the list of repeatable graphics is not sorted. Graphics appear by name order. Sorted graphics show repeatable graphics in order by lowest or highest values based on the value of the **Sort Order** property.



A descending list of repeatable graphics places the highest values at the top. An ascending sort order places the lowest values at the top.



The repeatable list of graphics can be filtered to restrict the display to a specified number of graphics. Based on the sort order, a descending list shows the highest values. An ascending sort order shows the lowest values.



Configure the GraphicRepeaterApp

Preparing the GraphicRepeaterApp to run in an AVEVA ViewApp includes major workflows to create a repeatable graphic and configure the app's properties. The following section includes an example of preparing the Graphic

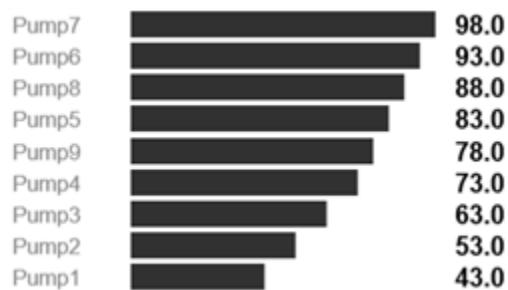
Repeater app for use with an AVEVA OMI ViewApp. After that, [Configure GraphicRepeaterApp properties](#) includes the details of assigning values to the app's configuration properties.

GraphicRepeaterApp example

Preparing the GraphicRepeaterApp to run in a ViewApp includes two major workflows of creating a repeatable graphic and assigning values to the app's properties. This topic provides a quick over view of the major steps in both workflows.

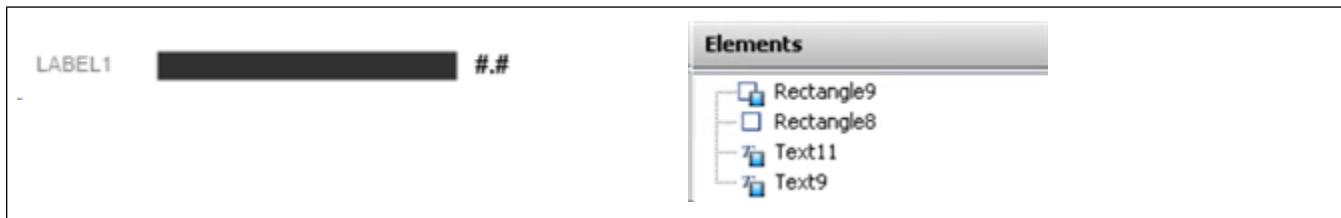
Create a repeatable graphic

The following figure shows a GraphicRepeaterApp running in a pane of a ViewApp. The repeatable graphic is a horizontal bar with a label on the left and the current value of a custom property on the right.

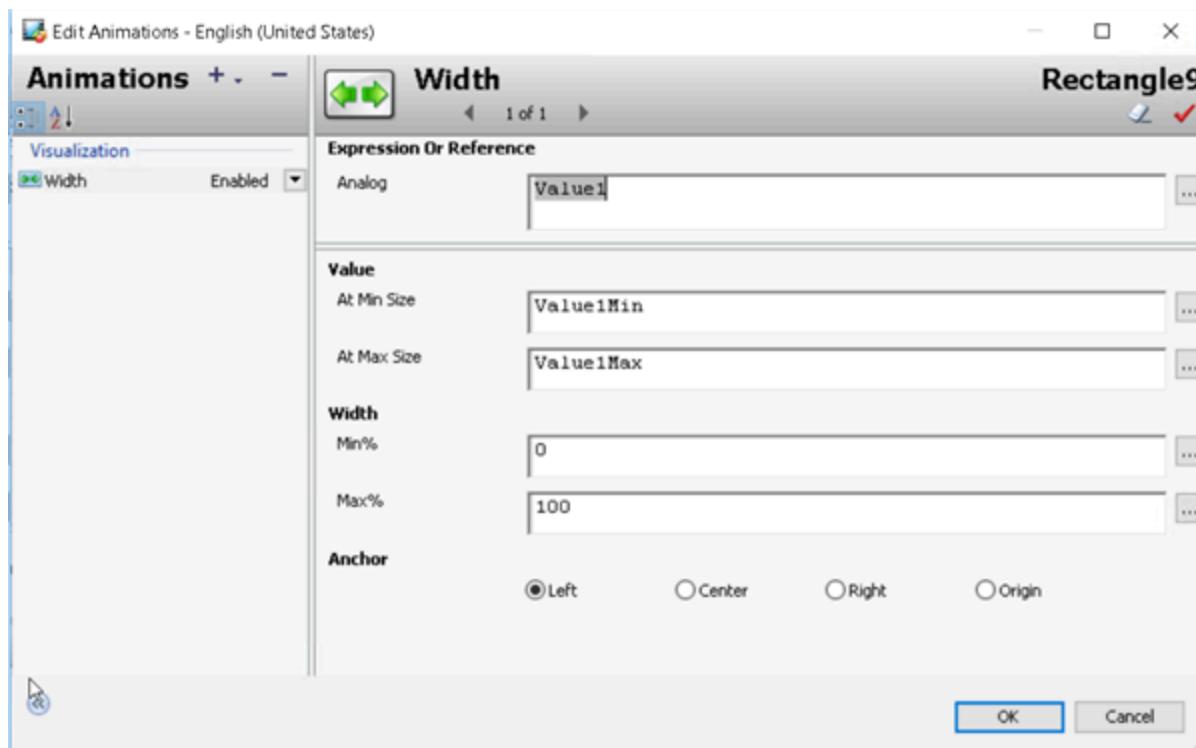


The GraphicRepeaterApp is configured to show graphics horizontally in descending order by the current values of each graphic. There are nine repeatable graphics shown by the app.

This is what the repeatable graphic looks like when it is open in the Industrial Graphic Editor. The graphic consists of two rectangle graphic elements and two text elements.

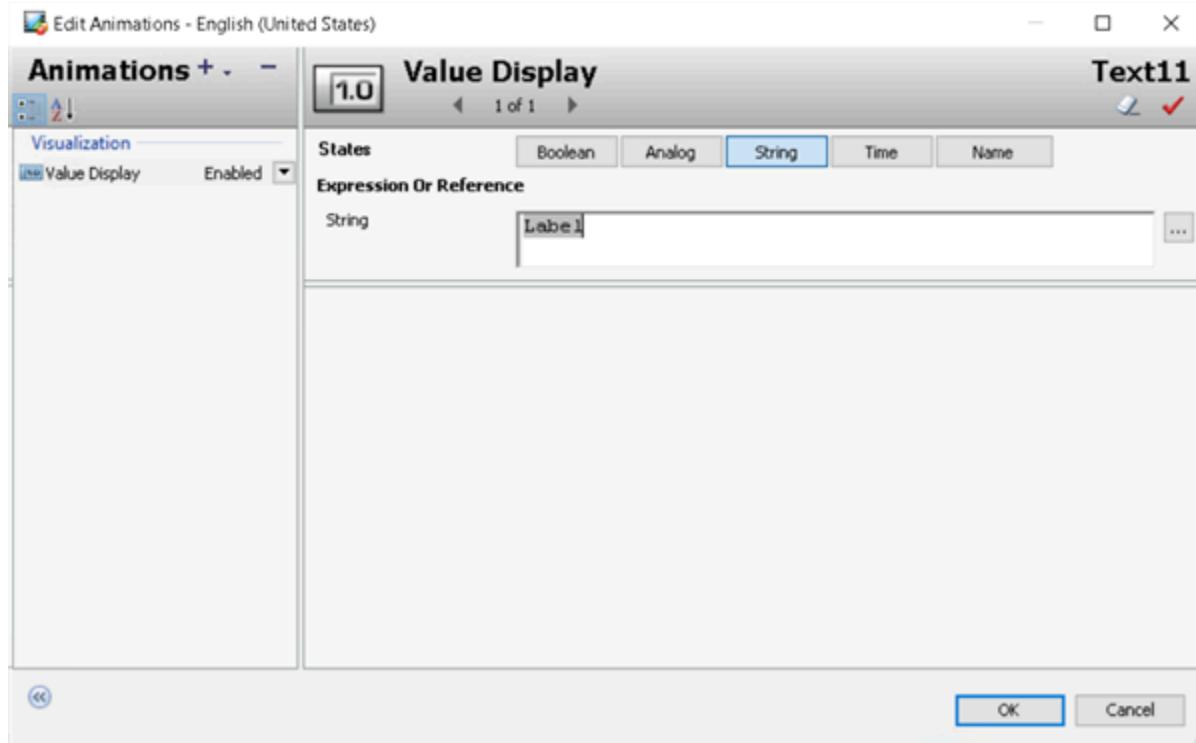


Width animation is set to custom property Value1for Rectangle9.

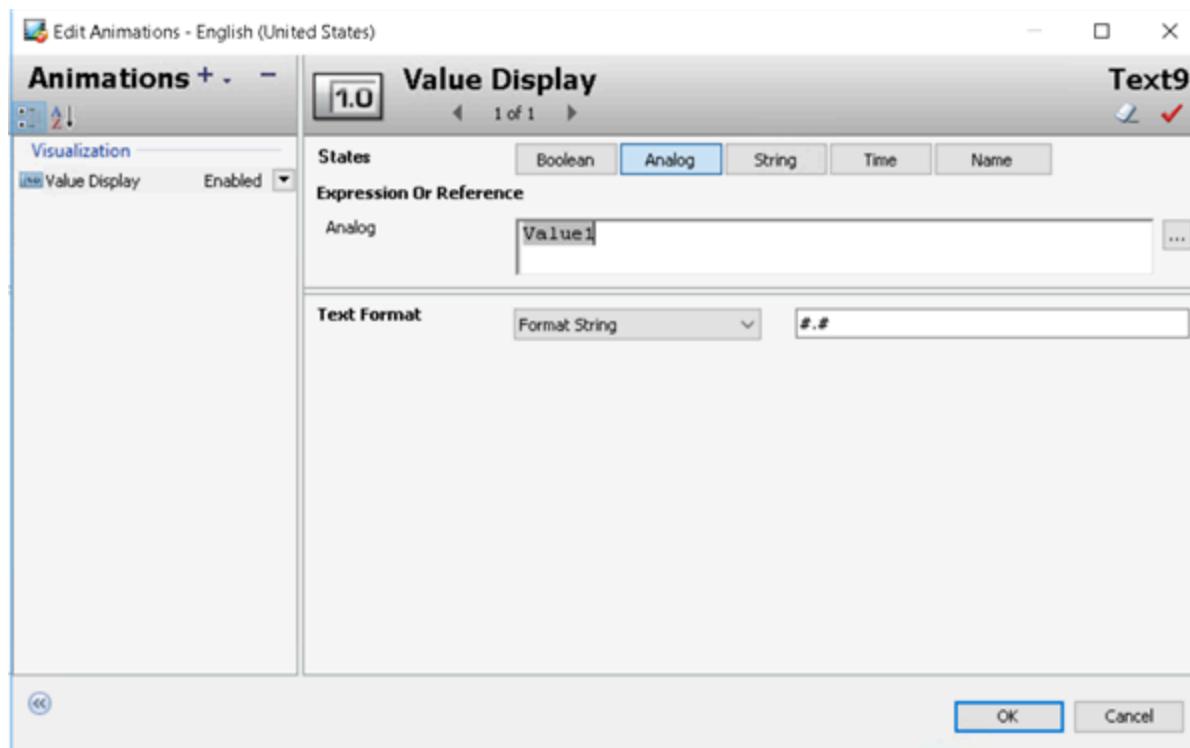


Rectangle8 has no assigned animation.

Text11 associates custom property Label1 to Value Display animation.



Text9 associates custom property Value1 to Value Display animation.



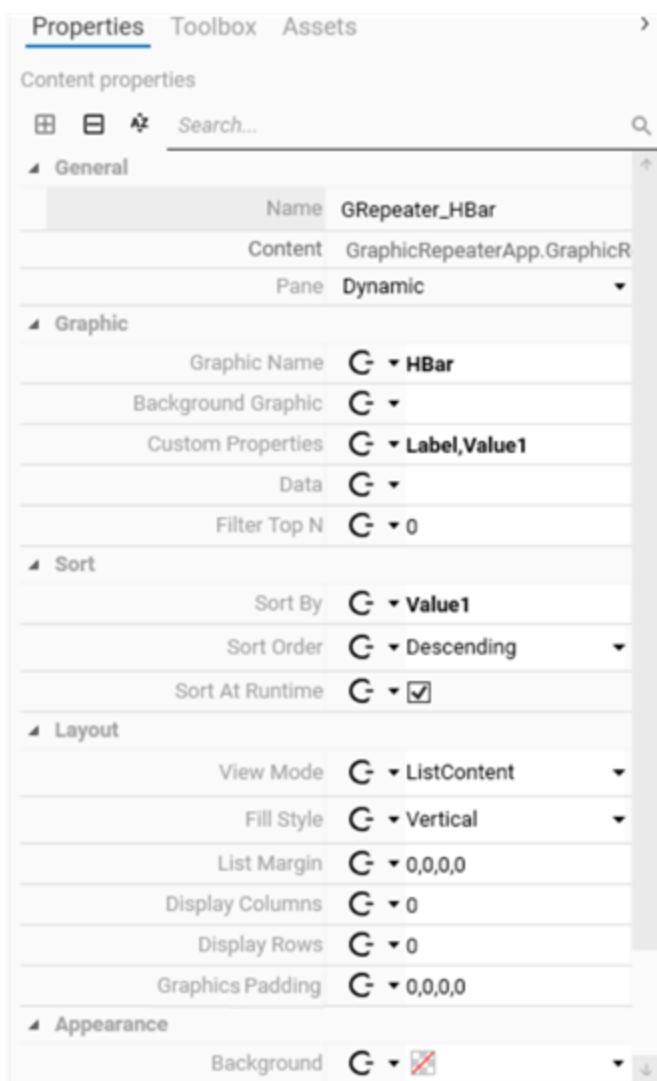
The width of the repeatable graphic bar is determined by the current value of its Value1 custom property. A static string will be associated with the Label1 custom property of the Text11 graphic element.

Assign values to GraphicRepeaterApp properties

The next section of the chapter describes the complete procedure to place the GraphicRepeaterApp onto a layout pane, and then assign values to properties. This section summarizes some of the major property configuration steps.

The following figure shows the values assigned to GraphicRepeaterApp properties for this example repeatable graphic.

- Label1 and Value1 are the custom properties of the HBar graphic.
- The repeatable graphics will be sorted by the current value of the Value1 custom property.
- The graphics will be oriented vertically and in descending order.
- The name of the instance of the GraphicRepeaterApp is GRrepeater_HBar.



Write a layout script

The following example layout script provides values to the Label1 and Value1 custom properties that belong to the HBar repeatable graphic.

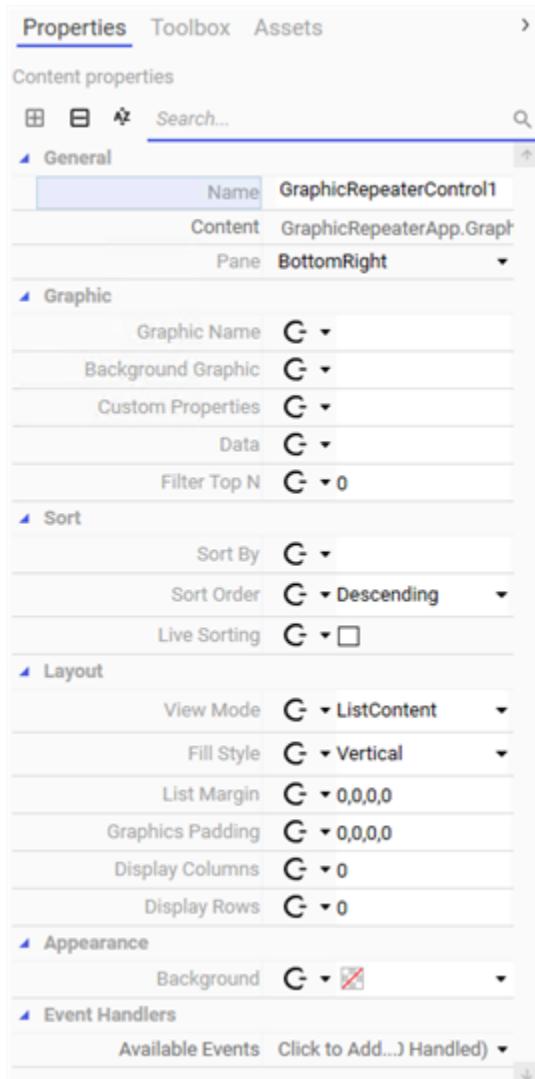
The **Data** property associated with the GRepeater_HBar instance of the GraphicRepeaterApp assigns values to the Label1 and Value1 custom properties in their comma delimited order specified in the **Custom Properties** property. Label1 is assigned a static string enclosed in double quotation marks. Value 1 is assigned values by object references. The script assigns values to nine instances of the HBar graphic.

```
MyContent.GRepeater_HBar.Data = " ""Pump1"" ,PLCSim.Triangle1, " +
" ""Pump2"" ,PLCSim.Triangle2, " +
" ""Pump3"" ,PLCSim.Triangle3, " +
" ""Pump4"" ,PLCSim.Triangle4," +
" ""Pump5"" ,PLCSim.Triangle5," +
" ""Pump6"" ,PLCSim.Triangle6," +
" ""Pump7"" ,PLCSim.Triangle7," +
" ""Pump8"" ,PLCSim.Triangle8," +
" ""Pump9"" ,PLCSim.Triangle9";
```

For more detail about layout scripts, see [Example layout script with the GraphicRepeaterApp](#).

Configure GraphicRepeaterApp properties

GraphicRepeaterApp properties can be shown in the Layout or ViewApp editors by placing the GraphicRepeaterApp on a layout pane. After selecting the app on a pane, its properties appear on the **Properties** page by functional groups. The following figure shows the default property values after initially opening the app.



Property value validation checking

As you enter values for properties, the GraphicRepeaterApp validates the input of some properties. If an invalid input value is detected, a red rectangle appears around the data entry field.



Placing your mouse directly over the data entry field containing the invalid data shows a tooltip with more information about the invalid data.



The table following the property configuration procedure includes a list of property value requirements for those properties whose input is validated by the GraphicRepeaterApp.

Further error checking occurs when the ViewApp containing a GraphicRepeaterApp is deployed and starts running. See [Validity checking during runtime](#).

To configure GraphicRepeaterApp properties

1. Open the Layout or ViewApp Editor and select the **Toolbox** tab.
 2. Open the folder in the **Toolbox** where the GraphicRepeaterApp was placed when it was imported.
 3. Select the GraphicRepeaterApp.
 4. Drag and drop the GraphicRepeaterApp thumbnail onto a layout pane.
 5. Select the GraphicRepeaterApp thumbnail and select the **Properties** tab.
- The **Properties** tab shows the properties of the GraphicRepeaterApp.
- The following table describes each property.
6. Assign values to properties and save your changes.

Note: The GraphicRepeaterApp can be shown in a preview session to quickly assess your property changes without deploying the ViewApp. A preview session launches immediately and runs the ViewApp containing the GraphicRepeaterApp in a separate window.

Properties	Descriptions
Graphic Name	<p>Specifies the graphic to repeat by the app. If it is a graphic associated with an object, specify the name as <i>ObjectName.GraphicName</i>.</p> <p>Embedded graphics are supported. In the following example, the <i>basesymbol</i> specified from the Graphic Name property contains <i>embedsymbol1</i>. The <i>cpint</i> custom property of the embedded graphic is specified as a custom property in the form <i>embedded_graphic_name.custom_property</i></p>  <p>The number of repeatable graphics shown during runtime is based on the following formula:</p> $\# \text{ graphics} = \# \text{ data groups} / \# \text{ custom properties}$ <p>Property Value Requirements</p> <ul style="list-style-type: none"> • A single graphic name must be assigned to the Graphic Name property. The Graphic Name property cannot be left blank.

Properties	Descriptions
	<ul style="list-style-type: none"> Graphic name must contain at least one alphanumeric character or special characters \$, #, _ or a period (.) Graphic name cannot start with a dollar sign (\$) or period (.) Graphic name cannot end with a period (.) or contain more than one period in the name.
Background Name	<p>Specifies the name of an optional graphic to appear behind the repeating graphic list during runtime. The background graphic provides header and footer information for the app.</p> <p>A background graphic stretches to fit the full width and height of the pane in which the GraphicRepeaterApp has been placed. Use the List Margin property to adjust the offset between the graphic list and the background graphic.</p> <p>For more information about specifying a background for the GraphicRepeaterApp, see Details of background properties.</p> <p>Property Value Requirements</p> <p>Same as the Graphic Name property.</p>
Custom Properties	<p>Specifies one or more custom properties that belong to the graphic specified by the Graphic Name property.</p> <p>Multiple custom properties must be specified in a comma delimited list.</p> <p>cp1,cp2,cp3</p> <hr/> <p>Note: Regardless of the regional setting of the computer, a comma is used as the data delimiter.</p> <p>For more information about the rules of comma delimited data, see Details of comma-delimited data.</p> <p>Custom properties specified by the Custom Properties property must match the corresponding Data values one to one in the list.</p> <p>Property Value Requirements</p> <ul style="list-style-type: none"> At least one custom property must be specified and the property cannot be left blank. Custom property names must contain at least one

Properties	Descriptions
	<p>alphanumeric character or special characters \$, #, _ or a period (.).</p> <ul style="list-style-type: none"> Number of specified custom properties cannot exceed 20.
Data	<p>A list of data references matched to custom properties in a comma delimited list that maps one to one with the custom properties list.</p> <hr/> <p>Important: If there is a mismatch between custom properties and data during runtime, the app will discard unmatched data points and an error message appears in the logger.</p> <p>A data value can be a reference or a constant and can be modified during runtime. For more information about the Data property, see Details of the Data property.</p> <p>Client and layout scripts can be used to set data references for graphic custom properties.</p> <p>Property Value Requirements</p> <ul style="list-style-type: none"> Data values must be provided <p>For more information about the number of data points that can be shown by the GraphicRepeaterApp based on the value of the Live Sorting property, see Details of the Live Sorting property.</p> <ul style="list-style-type: none"> Passing data to custom properties using relative references (eg: Me.references) is not supported.
Filter Top N	<p>Filters the list of repeatable graphics shown during runtime to a specified number (<i>N</i>). The default is 0 to show all graphics.</p> <p>Property Value Requirements</p> <ul style="list-style-type: none"> <i>N</i> value must be an integer. The Filter Top N value must be less than or equal to the number of data points. If the Filter Top N

Properties	Descriptions
	value is greater than the number data points, then the Filter Top N value is ignored and all data points are shown without filtering.
Sort By	<p>Specifies the name of a custom property whose current value is used to sort graphics during runtime based on the configured sort order.</p> <p>For more information about sorting data, see Details of sorting properties.</p> <p>Property Value Requirements</p> <p>Entered custom property name must match one of the graphic custom properties specified for the Custom Properties property.</p>
Sort Order	Identifies the repeating graphics sort order: Ascending or Descending by custom property value. The default sort order is Descending .
Live Sorting	<p>Boolean property to enable or disable graphics sorting at runtime. The default is false to disable sorting.</p> <p>The Live Sorting property can be modified during runtime to enable or disable graphic sorting.</p> <p>For more information, see Details of the Live Sorting property.</p>
View Mode	<p>Determines the organization of repeating graphics within a layout pane.</p> <p>ListContent</p> <p>Graphic items are shown in the GraphicRepeaterApp as a continuous sequential list. Based on the specified value of the Fill Style property, the sequential list is arranged in vertical or horizontal order. The default view mode.</p> <p>When View Mode property is set to ListContent, the GraphicRepeaterApp includes scroll bars to view graphics that extend beyond the bottom or right border of the pane.</p> <p>WrapContent</p> <p>Repeating graphics are listed in the fill direction until a graphic does not fit within the border of the pane. The graphic then wraps to the next column or row within the pane. Scroll bars appear when the graphic item list extends beyond the pane borders.</p>

Properties	Descriptions
	When View Mode is set to WrapContent , repeating graphics appear on the next row or next column based on whether the Fill Style property value is Horizontal or Vertical.
Fill Style	<p>Options to set vertical or horizontal fill order of repeating graphics.</p> <ul style="list-style-type: none"> • Horizontal fill order is left to right within the pane. • Vertical fill order is top to bottom within the pane. Vertical is the default. <p>For more information about graphic size based on the selected Fill Style options, see Details of the Fill style property.</p>
List Margin	<p>Specifies the pixel offset between a background graphic and the list of repeating graphics. The offset is specified as blank space padding on the left, top, right, and bottom of the reference repeating graphic.</p> <p>List Margin values are specified in a comma delimited ordinal list of integers as Left,Top,Right,Bottom.</p> <p>0,0,0,0 is the default, which does not include a margin offset between the repeating graphics and background graphic.</p>
Graphics Padding	<p>Number of pixels as blank space padding on the left, right, top, and bottom of a graphic shown in the GraphicRepeaterApp.</p> <p>Padding values are specified in a comma delimited ordinal list of integers as Left,Top,Right,Bottom.</p> <p>0,0,0,0 is the default, which does not provide any padding.</p> <p>For more information, See Details of the Padding property.</p>
Display Columns	<p>Number of columns containing repeating graphics during runtime. Column numbers are specified as an integer and 0 is the default value.</p> <p>The GraphicRepeaterApp includes a horizontal scroll bar to show columns outside of the limit set by the Display Columns property.</p> <p>Important: If the product of the Display Columns and Display Rows values exceed 50, an error occurs at runtime.</p>

Properties	Descriptions
	<p>For more information, see Details of the DisplayColumns property.</p> <p>Property Value Requirements</p> <ul style="list-style-type: none"> • Display Columns value must be an integer. • Display Columns value cannot exceed 50.
Display Rows	<p>Number of rows containing repeating graphics shown during runtime. Row numbers are specified as integers and 0 is the default value.</p> <p>The GraphicRepeaterApp includes a vertical scroll bar to show rows outside of the limit set by the Display Rows property.</p> <p>Important: If the product of the Display Columns and Display Rows values exceed 50, an error occurs at runtime.</p> <p>For more information, see Details of the Display Rows property.</p> <p>Property Value Requirements</p> <ul style="list-style-type: none"> • Display Rows value must be an integer. • Display Rows value cannot exceed 50.
Background	<p>Optional background color that appears in the entire pane area in which the GraphicRepeaterApp has been placed. No color is the default background.</p> <p>Select the Background entry field to show a color picker control to select a background color.</p>

The GraphicRepeaterApp includes two read-only properties that show the current minimum and maximum values of the custom property associated with the **Sort By** property. These two properties can be used only in scripts.

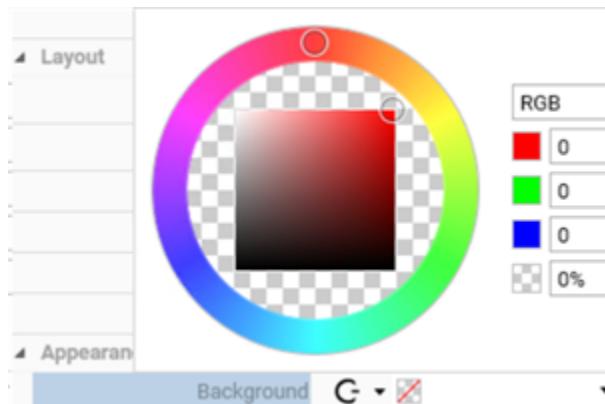
Important: In scripts, you must be careful in typecasting to the appropriate datatype and include checking for null values.

MaxValue	Gets the current maximum value of the Sort-By custom property from the set of repeatable graphics. This property can be accessed only by script. Example MyViewApp.VAN.Max = MyContent.GraphicRepeaterControl1.MaxValue;
MinValue	Gets the current minimum value of the Sort-By custom property from the set of repeatable graphics. This property can be accessed only by script. Example MyViewApp.VAN.Min = MyContent.GraphicRepeaterControl1.MinValue;

Details of background properties

A background that appears behind the list of repeating graphics can be specified as a solid fill color specified by the **Background** property, or another graphic specified by the **Background Graphic** property. The background properties are not mutually exclusive. You can use both properties to assign a background to a GraphicRepeaterApp.

The **Background** property includes a color picker control that appears when you select the data entry field with your mouse. The selected background color appears in the pane with the repeating graphics list in the foreground. Choose a background color that will provide sufficient contrast to the colors used in the repeating graphics.



The graphic specified by the **Graphic Background** property stretches to fit the full width and height of the pane in which the GraphicRepeaterApp has been placed. The graphic does not maintain its original horizontal and vertical dimensions. As a result, text in the graphic can be skewed and captions no longer align with the elements of graphics.

The **List Margin** property can compensate by setting offset padding between the background graphic and the list of repeating graphics for better alignment of the text in the background graphic. To reduce the visual problems caused by stretched background graphics, create a background graphic that is similar in size to the pane in which the GraphicRepeaterApp is placed.

Details of comma-delimited data

Commas are used as delimiters to separate multiple values specified for GraphicRepeaterApp custom properties. A comma is always used as the data delimiter regardless of the regional setting of the computer.

The following table shows examples of comma-delimited strings and how the entered data is parsed.

Comma Delimited Strings	Parsed Data
10,"Hello There",30,"Second"	10 "Hello There" 30 "Second"
40,"Hello",50 , "World", 100	40 "Hello" "50" "World" "100"
40,"Hello, How "are" you",50 , "World",100	40 "Hello, How "are" you" 50 "World" 100
40,"Hello, How "are" you",50 "World",100	40 "Hello, How "are" you" 50 "World" 100
40,"Hello,,,,,Th""ere","",50"Se"co"nd	40 "Hello,,,,,Th""ere" 50"Se"co"nd
35,Obj.Attribute1,"Pump1 output,36,Obj.Attribute2	35 Obj.Attribute1 "Pump1 output,36,Obj.Attribute2
35,Obj.Attribute1,Tank2 Volume Attribute,",36,Obj.Attribute2	35 Obj.Attribute1 Tank2 Volume Attribute ,36,Obj.Attribute2

Data property values can be entered as free text, making data parsing more complex. The following rules are followed to parse a comma delimited string containing many quotation marks and commas.

- If data starts without a quotation mark, the entered data is evaluated as a single value until a comma appears in the string.
- If data starts with a quotation mark, the entered data is evaluated as a single value until the next quotation mark and comma (",") appears in the string.

Details of the Data property

The Data property is the only property in the GraphicRepeaterApp that can be modified at runtime through scripting. See [Example layout script with the GraphicRepeaterApp](#) for more information.

The following table shows the format of data entered as GraphicRepeaterApp property values.

Data Type	Format	Example
Integer	Value	22
String	"Value"	"Inlet Pump"
Boolean	Value	True or False
Float	Value	117.33
Double	Value	222.09
DateTime	"T:value"	"T:7/15/2012 2:13:41.978 AM"
Elapsed Time	"E:value"	"E:09:32:00.0000000"

Any data value placed between quotation marks are considered as a string with the exception of DateTime and ElapsedTime data types.

All object data references should be formatted as:

Object_Name.Attribute_Name

Calculate the number of repeatable graphics

The number of repeatable graphics rendered by the GraphicRepeaterApp is the quotient of the number of data groups specified by the **Data** property divided by the number of custom properties.

Number of repeatable graphics = Number of data groups / Number of custom properties

If the number of data points in a data group does not match the number of custom properties, the data group is discarded and a message appears in the logger.

Example:

Custom Properties: level,state,

Data: 100,"California",90,"Nevada",50,"Oregon",45

In this example, three repeatable graphics are shown and the fourth partial data point (45) is discarded.

Example layout script with the GraphicRepeaterApp

Layout scripts can be used to provide reference values for the custom properties of the GraphicRepeaterApp.

The following example shows a simple layout script that provides a string and an object reference for the two **Name** and **Value** custom properties of a repeatable graphic.

```
MyContent.GraphicRepeaterControl1.Data = " """Pump1""",PLCSim.Triangle1," +
    " """Pump2""",PLCSim.Triangle2," +
    " """Pump3""",PLCSim.Triangle3," +
    " """Pump4""",PLCSim.Triangle4,"+
    " """Pump5""",PLCSim.Triangle5," +
    " """Pump6""",PLCSim.Triangle6," +
    " """Pump7""",PLCSim.Triangle7," +
    " """Pump8""",PLCSim.Triangle8,"+
    " """Pump9""",PLCSim.Triangle9";
```

The following table describes some of the details of the layout script.

Script Components	Description
MyContent.GraphicRepeaterControl1	Instance name of the GraphicRepeaterApp specified by the Name property on the app's Property page.
Data	Data property of the GraphicRepeaterApp. The Data property is the only property that can be changed at runtime by scripting.
"""Pump1"""	Static string enclosed by double quotation marks for the Name custom property of the repeatable graphic
PLCSim.Triangle1	Object reference for the Value custom property of the repeatable graphic.
" """Pump1""",PLCSim.Triangle1," + ... " """Pump9""",PLCSim.Triangle9";	Script shows the repeatable graphic nine times with different assigned string and object reference values.

Comparison of GraphicRepeaterApp properties and scripting attributes

GraphicRepeaterApp properties can be incorporated into layout scripts as attributes. In most cases, app properties and script attributes have similar names with the exception of no blank spaces in attribute names. However, the GraphicRepeaterApp does have some properties that are very different than their corresponding attributes names.

The following table shows the differences between GraphicRepeaterApp properties and layout script attributes.

GraphicRepeaterApp Properties	GraphicRepeaterApp Scripting Attributes
Graphic Name	GraphicName
Background Graphic	BackgroundGraphicName
Custom Properties	CustomProperties
Data	Data
Filter Top N	FilterTopN
Sort By	SortBy
Sort Order	SortOrder
Live Sorting	DynamicSort
View Mode	ViewMode
Fill Style	FillStyle
List Margin	ListMargin
Graphics Padding	GraphicsPadding
Display Columns	DisplayColumns
Display Rows	DisplayRows
Background	Background

Details of sorting properties

The GraphicRepeaterApp includes three properties that determine how repeatable graphics are sorted during runtime:

- **Sort By**

Specifies a custom property whose current value is used to sort repeatable graphics.

If the entered custom property name binds to a reference during design time ,the datatype of **Sort By** is initially evaluated as a string After receiving the subscribed value during runtime, the **Sort By** datatype changes to that of the subscribed value.

- **Sort Order**

Specifies if repeatable graphics are sorted by their current custom property values in ascending or descending order.

- **Live Sorting**

Boolean value that determines if repeatable graphics are sorted during runtime. A script can be used to enable or disable sorting during runtime by changing the Boolean value of **Live Sorting**.

Details of the Live Sorting property

The GraphicRepeaterApp uses virtualization to reduce memory load and maintain acceptable performance during runtime. Virtualization minimizes memory usage by loading only those repeatable graphics that appear in the layout pane. Virtualization removes those repeatable graphics from memory that are not shown in the visible layout pane. Repeatable graphics are reconstructed again when the user scrolls the pane to show the graphics in the pane. Those graphics moved out of the pane view by scrolling are removed from memory.

The GraphicRepeaterApp disables virtualization when the **Live Sorting** property is active. Graphics that do not appear in the layout pane are still loaded in memory. As a result, performance can be affected if the GraphicRepeaterApp includes a large number of repeatable graphics.

Follow these recommendations for the number of repeatable graphics based upon whether sorting will be active or not during runtime.

- When **Live Sorting** is active, restrict the number of repeatable graphics to a maximum of 50.
- When **Live Sorting** is inactive, restrict the number of repeatable graphics to a maximum of 200.

Other factors affect the performance of the GraphicRepeaterApp during runtime. The maximum limits listed above are for a single instance of the GraphicRepeaterApp in a running ViewApp. If a ViewApp includes multiple instances of the GraphicRepeaterApp that run simultaneously, the number of repeatable graphics displayed by each instance is reduced to the maximum limit divided by the number of instances.

Details of the Fill style property

The Fill property includes two options to order graphic items in the Content Presenter app horizontally or vertically.

In the case when the ViewMode property is set to ListContent, the viewer pane includes scroll bars to view graphic items that extend beyond the bottom or right border of the viewer pane.

In the case when the ViewMode property is set to WrapContent, graphics appear on the next row or next column based on whether the Fill property value is Horizontal or Vertical.

Details of the Padding property

Graphics padding is the blank space placed around the borders of repeating graphics during runtime. Padding values are specified by the number of pixels placed at the left, top, right, and bottom borders of a graphic, which are specified as four positional integers separated by commas.

L,T,R,B

No padding is the default, which is specified as 0,0,0,0.

The aspect ratio of the repeating graphic is maintained when padding is applied. In the case when a graphic extends beyond the pane border because of padding, the repeating graphic is reduced in size until it fits within the pane while including the padding space.

Details of the DisplayColumns property

The value assigned to the **DisplayColumns** property determines the number of columns that appear within the GraphicRepeaterApp during runtime. A non-zero value shows graphics in the specified number of columns and the graphics are scaled to fit the width of each column. The remaining graphics beyond the column limit appear

on the next row of the view pane.

A value of zero indicates the **DisplayColumns** property is ignored and graphics appear in the viewer window in as many columns as calculated based on the values of other GraphicRepeaterApp properties.

Details of the Display Rows property

The **Display Rows** property specifies the number of graphics that can be placed on a row within the pane in which the Grapher Repeater app has been placed.

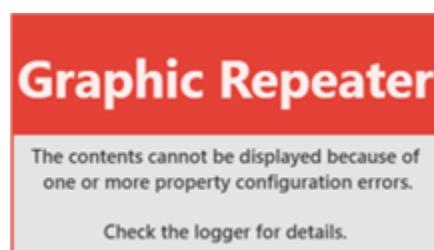
A value of Zero indicates that row specification is ignored and repeatable graphics appear in as many rows as calculated based on various factors such Contents Size, Fill Mode, Size Mode, Pane size etc.

Validity checking during runtime

The GraphicRepeaterApp validates the following property values when a ViewApp starts running that contains an instance of the GraphicRepeaterApp

Properties	Validation Checking
Graphic Name	No graphic name assigned
Custom Properties	Number of assigned custom properties is zero or greater than 20.
Filter Top N	Filter Top N is assigned a number greater than the number of data points
Display Rows	Display Rows is assigned a number greater than 50. The product of assigned Display Rows and Display Columns values cannot exceed 50.
Display Columns	Display Columns is assigned a number greater than 50. The product of assigned Display Rows and Display Columns values cannot exceed 50.

An error icon appears when the ViewApp containing starts if invalid GraphicRepeaterApp property values are detected. The ViewApp continues running, but the pane containing the GraphicRepeaterApp shows the error icon.

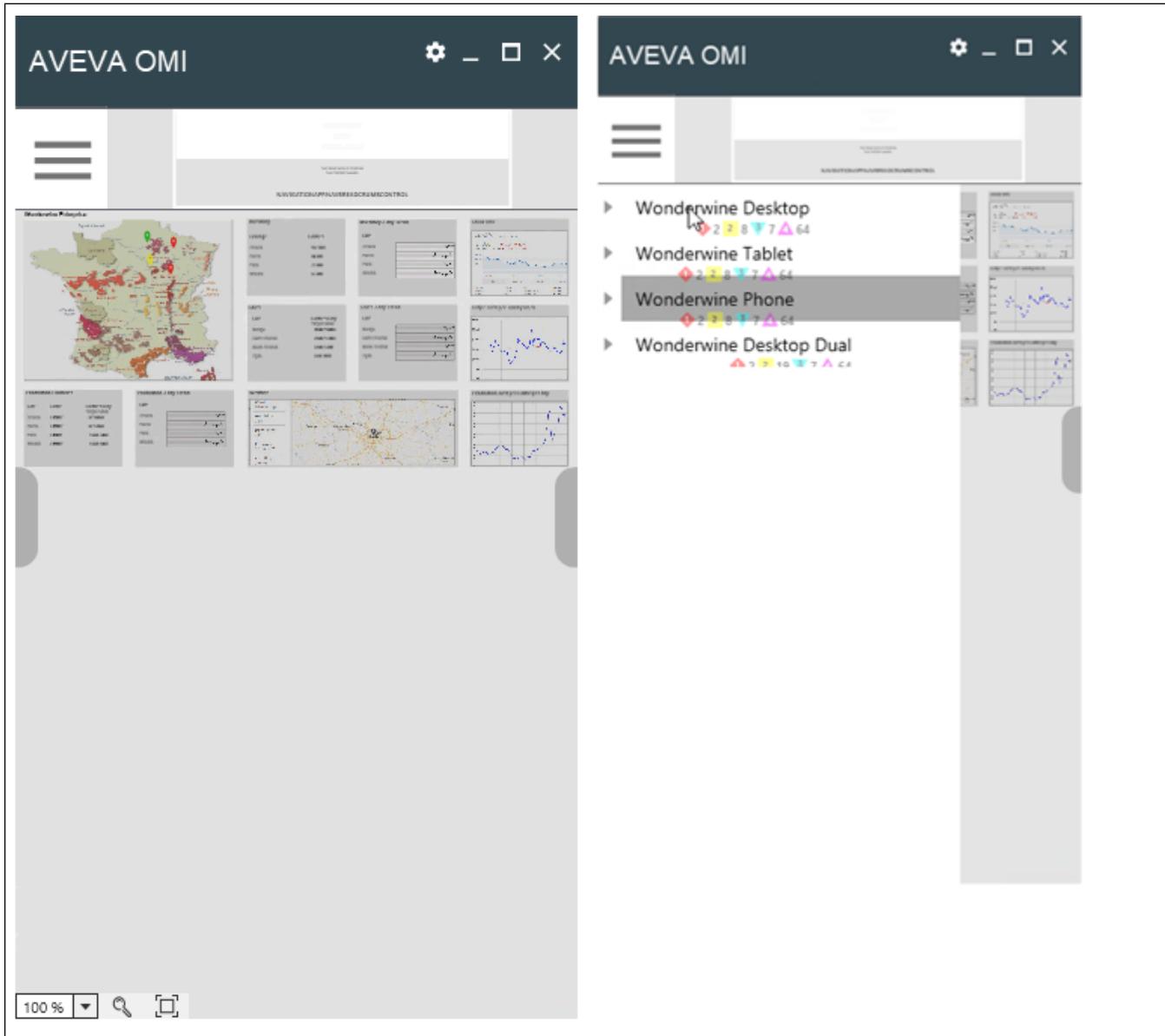


One or more error messages appear in the Operations Control Logger with descriptions of the problems.

Data property values are not validated when the ViewApp starts. If Data values are out of range, the pane containing the GraphicRepeaterApp is blank without displaying an error icon. But, error messages are written to the Operations Control Logger.

HamburgerApp

The HamburgerApp controls the movement of a selected layout slide-in pane. During run time, the HamburgerApp appears as a button consisting of three parallel horizontal lines. When the user selects the button, the selected slide-in pane appears. Users can toggle the hamburger button again or select an area outside of the slide-in pane to hide the pane.



The HamburgerApp is typically placed near the top corner of a window to save screen space for ViewApps designed to run on cell phones or tablets.

Configure the HamburgerApp

The HamburgerApp includes properties to select a slide-in pane that slides out when the app is selected and the foreground and background colors of the hamburger button.

Each property contains a drop-down list to reset the property to its default value by selecting the visual icon next to the data entry field.



To configure a HamburgerApp

1. Open the Layout or ViewApp Editor and show the items listed in the **Toolbox** tab.
2. Select HamburgerApp from the **Toolbox** list to show its preview thumbnail.
3. Drag and drop the preview thumbnail on a layout pane.
4. Select the preview thumbnail on the pane and select the **Properties** tab.

The **Properties** tab shows a list of HamburgerApp properties.

The screenshot shows the AVEVA Properties tab interface. At the top, there are tabs for 'Properties', 'Toolbox', and 'Assets'. Below the tabs is a search bar labeled 'Search...'. The main area displays a list of properties under sections like 'General', 'Misc', 'Appearance', and 'Event Handlers'. The 'General' section includes fields for 'Name' (HamburgerButton1), 'Content' (HamburgerApp.HamburgerB), and 'Pane' (TopLeft). The 'Misc' section includes 'SlideInPanePosition' set to 'Left' and 'HamburgerColor' set to '#FF000000'. The 'Appearance' section includes a 'Background' field with a color picker. The 'Event Handlers' section shows 'Available Events' and 'Click to Add...'.

5. Set the **Background** and **HamburgerColor** color properties.

Both properties include a color picker. Select the color property and then select the small triangle at the right of both property fields to show the color picker.

6. Select the **SlideinPanePosition** field to show a list of slide-in pane positions (**Left**, **Top**, **Right**, **Bottom**) to activate on the layout. **Left** is the default slide-in position.
7. If you want to create event handler scripts, select the **Available Events** data entry field to show a list of HamburgerApp events.
8. Save your property selections.

Size the HamburgerApp to the Layout pane

It can occur that the HamburgerApp does not adjust its size at runtime to adapt to the size of the hosting layout pane. There are several methods to resolve HamburgerApp and pane relative size issues. One recommended method is to adjust pane properties to your layout. For example, the width and height pane properties can be adjusted to fixed sizes to fit your specific layout. You can do this with a quick calculation of the HamburgerApp container pane size as a percentage of total layout width and height. Adjust that number to accommodate the

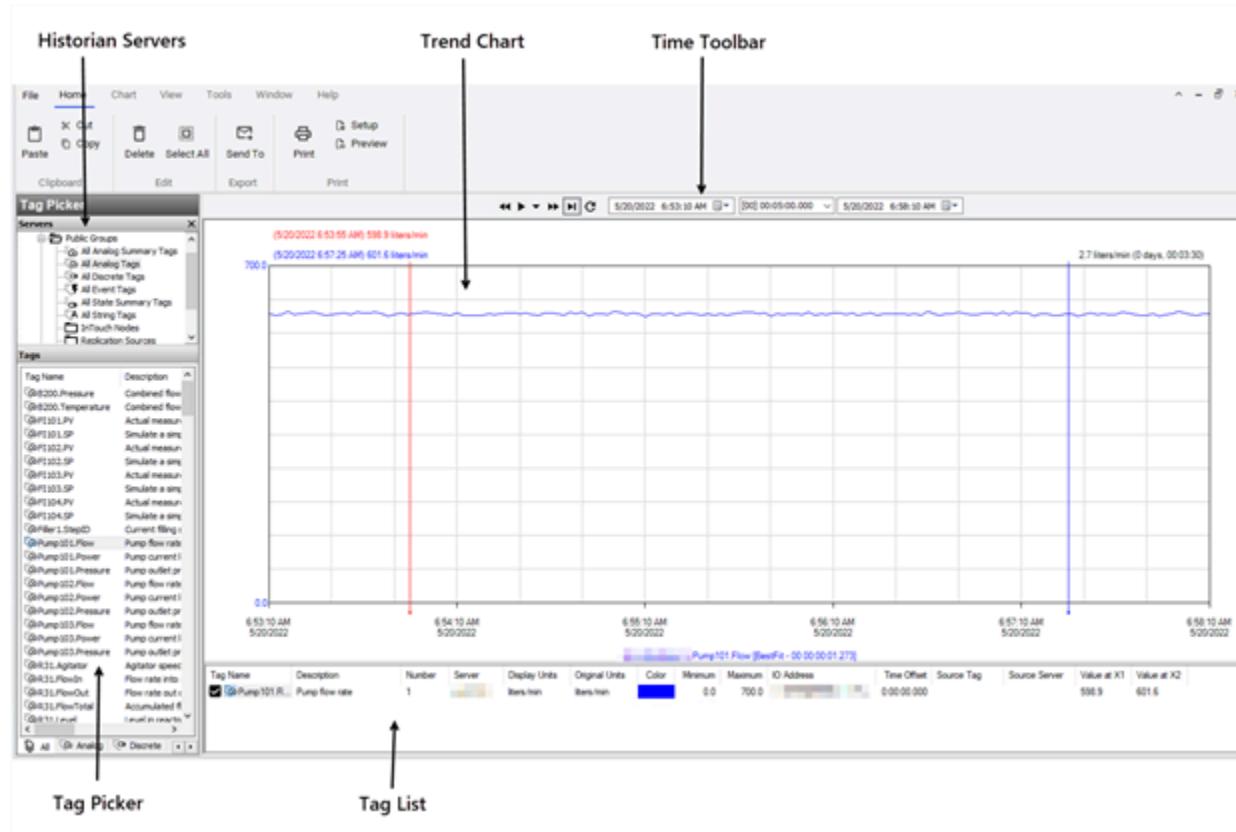
HamburgerApp dimensions.

AVEVA Historian Client Trend

AVEVA Historian Client Trend is a stand-alone app which contains a set of properties that can be configured in both design-time and run-time. The design-time properties are set to default values that enable you to show trends in your ViewApps with minimal configuration work. Changes made to a property during run time override the value set to the same property during design time.

When you start the Historian Client Trend for the first time, you must log on and connect to a Historian server. If you are opening an existing Trend file that includes at least one server configuration and the log in was successful, you are not prompted to log on. For more information about connecting to a Historian, see [Creating a New Server Connection](#).

After you establish a connection with the server, the Trend main window appears.

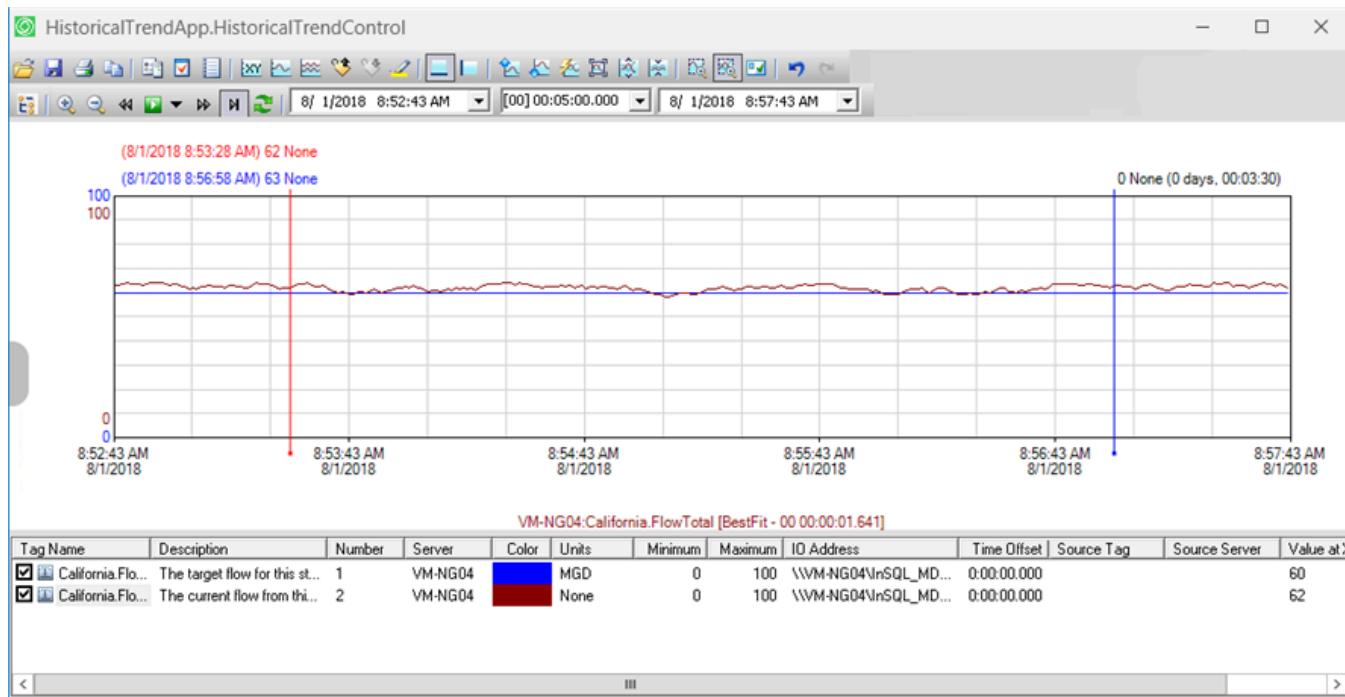


- For information about assigning values to properties, see [Configure trend file properties](#).
- For information on using the Tag Picker and the Time picker, see [Configure a trend](#). To show or hide tool bars or components, use the corresponding commands on the **View** menu.

HistoricalTrendApp properties

The HistoricalTrendApp includes a set of design-time properties that enable or disable visual or functional components of the Historical Trend during run time. All properties are assigned default values to enable the HistoricalTrendApp to run immediately in a ViewApp with minimal run-time configuration.

The following figure shows the Historical Trend control running in a ViewApp after accepting the default values assigned to its **Run-Time Behavior** properties.



Configure HistoricalTrendApp properties

The HistoricalTrendApp shows two different types of properties in the **Properties** area of the Layout editor. The **Run-Time Behavior** properties are the native properties of the HistoricalTrendApp. This topic provides descriptions of these properties.

The remaining properties are from the underlying .NET trend control that are exposed and can be assigned values from the Layout editor or included in scripts. .NET property groups are identified by the **TrendControl** prefix in their titles that appear on the **Properties** page. For example, the **TrendControl.Accessibility** title appears immediately beneath the **Run-Time Behavior** group of properties.

Descriptions of .NET properties are accessible from the Microsoft Web site and are not described in System Platform documentation. For more information about the .NET properties that appear in the HistoricalTrendApp, see .NET property descriptions at the Microsoft web site.

A .NET control enables developers to include a description using the **DescriptionAttribute**, as shown in the following example.

```
[DescriptionAttribute("Represents a double-precision 64-bit number ")]
```

The text shown as part of the **DescriptionAttribute** appears in the **Description** area of the **Properties** tab in Layout and ViewApp editors.

To configure the properties of theHistoricalTrendApp

1. Open the Layout and show the items listed in the **Toolbox** tab.
2. Select the HistoricalTrendApp from the **Toolbox** list to show its preview thumbnail.

The **Toolbox** pathway to the Historical Trend App is:

_Default Content > 4. Apps > AVEVA OMI Apps > HistoricalTrendApp

3. Drag and drop the preview thumbnail onto a layout pane.
4. Select the preview thumbnail on the pane and select the **Properties** tab.

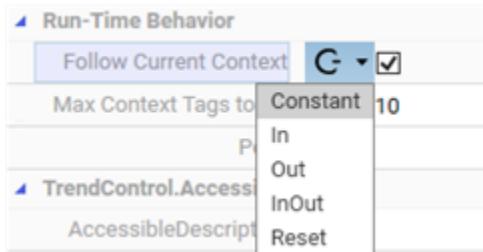
The **Properties** tab shows a list of HistoricalTrend app properties. Use the Layout editor search function to search for a property by name.

General properties show the name of the control, its content type, and the name of the pane in which the HistoricalTrendApp is placed.

5. Set values to the **Run-Time Behavior** properties.

Important: See [Upgrading HistoricalTrend App Runtime Properties](#) for information about the **Run-Time Behavior** properties of the HistoricalTrendApp after upgrading to System Platform 2020.

Most HistoricalTrendApp properties include a user control with a drop-down list of binding options to configure static or dynamic value based binding between the property and its associated reference.



Constant	Static binding to the specified property value. Enter a value in the property data entry field.
In	Dynamic property value with read only binding by the control to a reference.
Out	Dynamic property value with write only binding by the control to a reference.
InOut	Dynamic property value with read/write binding by the control to a reference.
Reset	Reset the property binding value to its default.

A check box to the right of some properties indicates the initial default value of a Boolean property. When the check box is selected, the property is set to true.

Run-Time Behavior Properties

Follow Current Context	Boolean value that determines if the trend shows values of the attributes belonging to the selected asset. True (default value) shows the selected asset's attribute values on the trend. The historized tags are picked from the currently selected asset in alphabetical order. The number of tags that can be shown from an asset is set by the Max Content Tags to Show property.
-------------------------------	---

	<p>With the property set to true, trend pens are auto-discovered and loaded asynchronously. This can cause unpredictable results if you will use the ShowContent() method to place the Historical Trend control into a pane at run time, or if the Follow Current Context property can be dynamically modified at run time. In these cases, set Follow Current Context to FALSE.</p> <p>Note: If Follow Current Context is set to False, the Show Tag Picker property must be set to True.</p> <p>IMPORTANT: In prior releases, Historical Trend App properties were not exposed. System Platform exposes a number of configurable properties and sets some to a default value. For example, in Updates 1 and 2, the TagPicker was visible when you used the Historical Trend App. In Update 3, the TagPicker is not visible by default, since tags from the current context are shown when Follow Current Context is true.</p> <p>For more information about automatically selecting tags to show in the trend based on the selected asset from the navigation model, see Follow Current Context Property.</p>
Max Context Tags to Show	The number of attribute values from the selected asset that can be shown on the trend when the Follow Current Context property is set to True. 10 is the default maximum number of tag values shown on a trend, and is the recommended maximum. If more than 10 tags are shown, performance may be affected. <p>Note: When Follow Current Context is set to False, the Max Auto Content Tags to Show property is inactive. The entered number has no meaning.</p>
Pens	The Pens property enables you to add specific tags to be shown in the view application at run time. Note that tag names are not validated. The tags you add are in addition to the Current Context tags. See Follow Current Context for more information.

- To add tags using static binding, set the value of this property to be one or more fully qualified tag names. Multiple tags must be separated by a comma. The following example shows three tag names, each using an application object name (Pump_00n) and attribute name (PV, CV, and Pressure):

```
Pump_001.PV, Pump_002.CV,  
Pump_001.Pressure
```

- To add tags using dynamic binding, create a string attribute that contains the tag name(s) that you want to add as trend lines. Enter the reference in the following format:

```
MyViewApp.<ViewAppNameSpaceName>.<Attribute Name>
```

For example:

```
MyViewApp.MyTrendSettings.AdditionalP  
ens
```

The reference attribute must contain the tag name(s), formatted the same way as with static binding (ObjectName.AttributeName). As with static binding, separate multiple tags with a comma.

If you use an application object reference instead of a ViewAppNamespace reference, enter the reference as the application object name followed by the attribute name, separated by a dot (period)

Name: PumpTrends

Description:

Data type: String

Writeability: User writeable

Initial value: Pump_001.PV, Pump_001.CV, Pump_002.Pressure

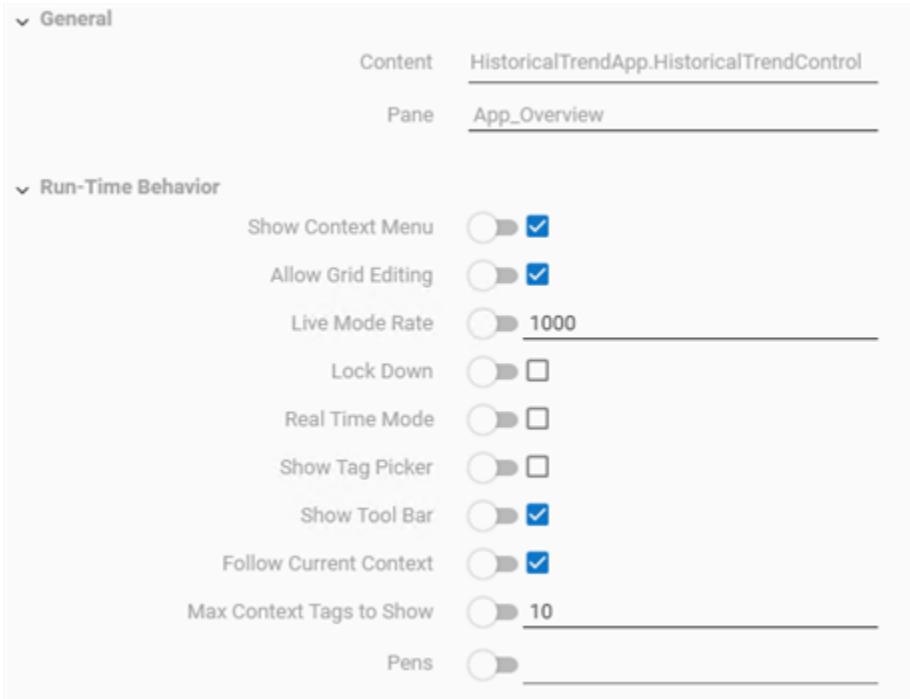
Available features: I/O, History

Upgrading HistoricalTrend App Runtime Properties

If you are upgrading from System Platform 2017 Update 3 Service Pack 1 or an earlier version, the HistoricalTrendApp included a set of runtime behavior properties that enabled its visual components to be shown or hidden during runtime. Other properties determined some aspects of the HistoricalTrendApp's

behavior.

Most of the HistoricalTrendApp **Run-Time Behavior** properties have been removed because the underlying control properties are now exposed and configurable through the **Properties** grid of the Layout editor. The following screen capture shows the properties of the HistoricalTrendApp that were available in System Platform 2017 Update 3 Service Pack 1.



Below is a screen capture that shows the properties as they now exist in the HistoricalTrendApp.



Upgrading from an earlier release of System Platform to the current version retains the values assigned to the legacy properties that were removed and re-directed directly to the underlying control's properties. Once you have upgraded to the current version of System Platform, the HistoricalTrendApp should retain the original functionality that was configured in the prior version of System Platform, without the need to reconfigure any properties.

Follow Current Context Property

When set to True (default value), shows attribute values for up to 10 tags from the selected asset on the trend. The historized tags are picked from the currently selected asset in alphabetical order, up to the maximum set by the **Max Content Tags to Show** property. Tags are shown only from the first connected Historian Server.

With the property set to true, trend pens are auto-discovered and loaded asynchronously. This can cause unpredictable results if you will use the `ShowContent()` method to place the Historical Trend control into a pane at run time, or if the Follow Current Context property can be dynamically modified at run time. In these cases, set **Follow Current Context** to FALSE.

Note: If **Follow Current Context** is set to False, the **Show Tag Picker** property must be set to True.

Configure a trend

When you configure a trend, you must select the tag(s) for which you want to query the trend data. This data is queried from the Historian database(s) to which you are currently logged on. After you select tags for the trend, you can set the start date and end date for the trend. The default settings in the Trend app are generally adequate to enable you to begin tracking trends.

The following list summarizes the key steps to configure the HistoricalTrendApp during runtime.

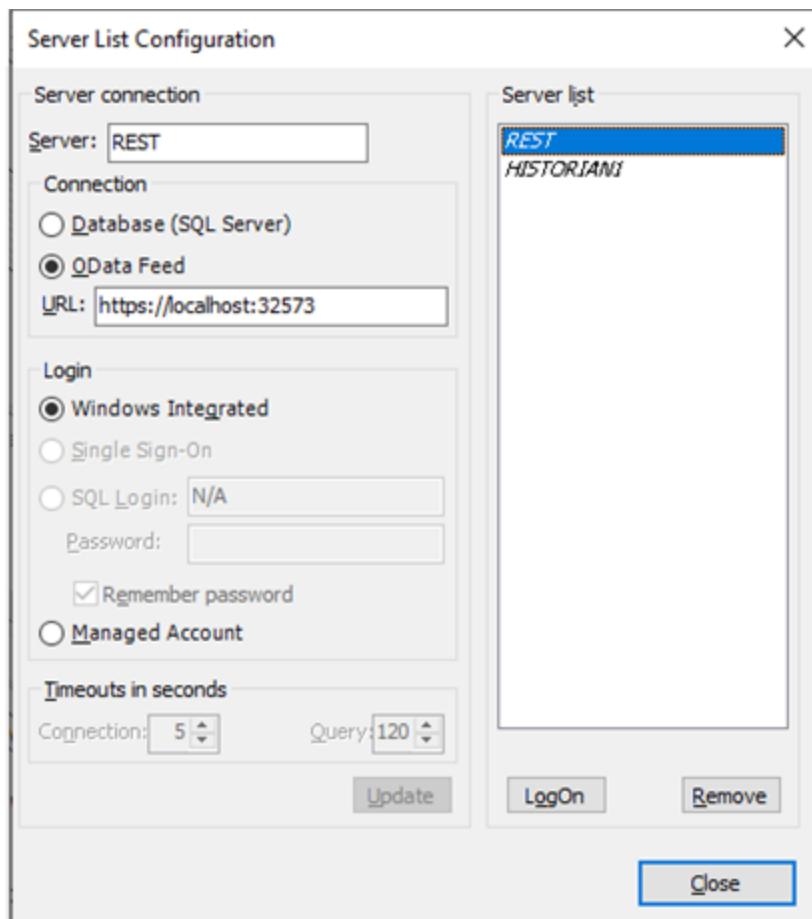
1. Place the HistoricalTrendApp on a layout pane.
2. Configure the HistoricalTrendApp properties. You can use the default settings, or change them as needed.
Default settings are as follows:
 - **Follow Current Context:** Enabled
 - **Max Context Tags to Show:** Up to 10 context tags are shown by default. If the asset you are viewing has more than 10 tags, the first 10 that are listed alphabetically are shown. Showing more than 10 may impact performance.
 - **Pens:** Empty (null). You can add additional context tags to show their trend data by entering the static value or bind to a reference. The pens configured here are in addition to the max context tags setting. List additional pens by entering fully qualified attribute names (for example, ObjectName.AttributeName). Use a comma to separate attribute names. You can also bind a reference to your pen list. To create a reference:
 - a. Create a ViewApp Namespace object in the Visualization folder and rename it (for example, "TrendSettings").
 - b. Create an attribute of string type in the ViewApp Namespace. In the following example, the attribute name is "MyPens."
 - c. Add the fully qualified attribute names you want to add to the trend. Multiple names must be separated by commas.
 - d. For the Pens property, create the dynamic binding read-only reference to the ViewApp Namespace (for example, MyViewApp.TrendSettings.MyPens).
3. Deploy and start the ViewApp containing the HistoricalTrendApp.
4. Log in to the Historian Server that contains the data to you want to show in a trend.

Creating a New Server Connection

The first time you start a Historian Client application, you must create a server connection to the Historian. To create a connection, you need your assigned Historian username and password.

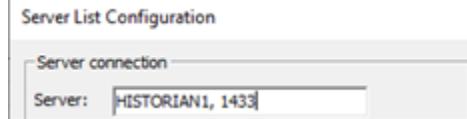
To create a Historian connection

1. From the Historian folder in the Windows Start menu, open the **Trend** application.
2. On the **Tools** menu, select **Servers**. The **Server List Configuration** dialog box appears, which shows a list of connected servers or servers that were configured in the past.



3. In the **Server** field, enter the name of the server to which you want to connect.
 - For an On-Premises Historian, enter the name of the server where it is hosted.
 - For a Managed Historian, use a name of your choice.
 - For a SQL Server named instance, use the format <server_name>\<instance_name>.

Note: If the default port for the SQL Server is changed, the Historian Client server configuration should be given as <server_name>[\<instance_name>], <port_number>. For example:



4. In the **Connection** section, specify the type of connection you want to use:

Note: Support for **runtime language switching** requires a REST configuration, and the server connection type must be configured as **OData Feed**. This configuration supports only <http://localhost:32569> or <https://localhost:32573>.

- **Database (SQL Server):** Choose this option to connect to the SQL Server database.
- **OData Feed:** Choose this option to connect to the open data protocol feed. Then specify the URL and continue to step 7. You can connect using this option only if you are using a Managed Historian connection. ODATA Feed is required to support runtime language switching (see note, above).

Notes: If you select OData Feed to connect to the Historian Server, all the fields related to Login section become unavailable and the Managed Account option becomes deselected automatically.

By default, the Historian's OData server supports unencrypted HTTP connections. To use an encrypted (HTTPS) OData connection, you will first need to install and configure an SSL client certificate. Contact [AVEVA Technical Support](#) for details.

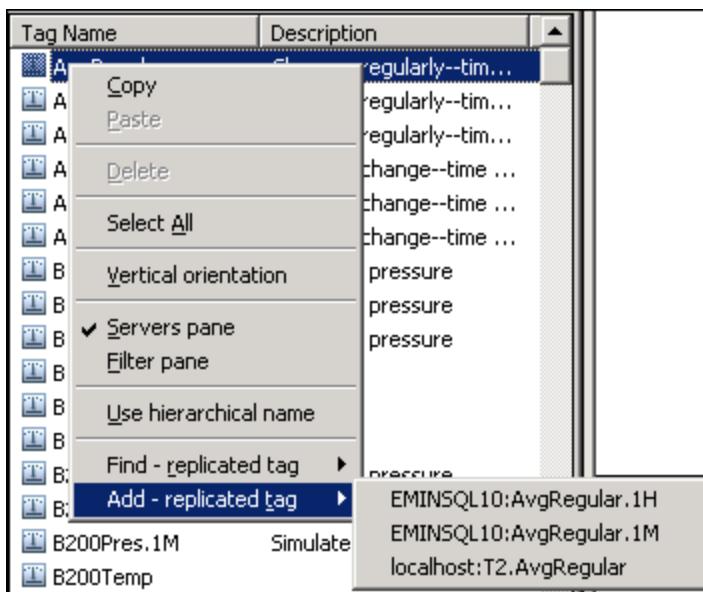
5. In the **Login** section, select one of the following:
 - **Windows Integrated:** Authenticate with Windows integrated security.
 - **Single Sign-On:** Authenticate with AVEVA Identity Manager. You must be using the connected experience for this option to be enabled. See the *Configure Operations Control with connected experience* section in the *AVEVA Service Platform Installation Guide* for details.
 - **SQL Login:** Authenticate with SQL Server credentials, click **SQL Login** and configure the following login details. Then go to step 7.
 - **Login ID:** Type your assigned Historian username. If your system administrator has not assigned you a username and password, you may use one of the default user accounts, which are automatically configured during a typical Historian installation.
 - **Password:** Type the password that is associated with the username. Select the **Remember password** check box to specify for the system to remember your password.
6. In the **Timeouts in seconds** area, configure the time allocated for the database connection and the query execution.
 - **Connection:** The connection time-out period in seconds. Valid values are 1 to 600.
 - **Query:** The query time-out in seconds. Valid values are 1 to 600.
7. Select **Add**.
After several seconds, the server connection should be established.
8. Click **Close**. An error message appears if a connection cannot be made to the Historian server.

Add a source tag to a trend

You can select a source tag or replicated tag from the Tag Picker to add to the active trend chart. For more information about the different components of the Tag Picker, see [Tag Picker](#).

To add a source tag or replicated tag

1. Select a tag in the Tag Picker.
2. If the selected tag is a source tag, do the following:
 - In the **Tags** pane, right-click the selected tag, point to **Add - replicated tag**, and then click the tag that you want to add to the trend chart.



The corresponding replicated tag is added to the active trend chart.

3. If the selected tag is a replicated tag, do the following:
 - In the **Tags** pane, right-click the selected tag, and then click **Add - source tag**.

The corresponding source tag is added to the active trend chart.

The **Add** command is not available if:

- Multiple tags are selected in the Tag Picker.
- A normal tag that is neither a source tag nor a replicated tag is selected in the Tag Picker.

Note: You cannot execute the **Add** command if a source tag is deleted but its replication configuration still exists in the Historian.

The replicated tags are not listed in the context menu if:

- The replicated tags are not committed in the Historian.
- The replication schedule is removed from the Historian. For example, you are connected to a Historian 10.0 server and you create a tag called 'MyTag'. 'MyTag' is replicated as a simple tag called 'MyServer.MyTag'. When you execute the **Add - replicated tag** command, the 'MyServer.MyTag' tag is shown. When you execute the **Add - source tag** command, the 'MyTag' tag is shown. At this instance, if the replication link between 'MyTag' and 'MyServer.MyTag' is removed and if you execute the **Add - replicated tag** command, the 'MyServer.MyTag' tag is not shown in the list of replicated tags.

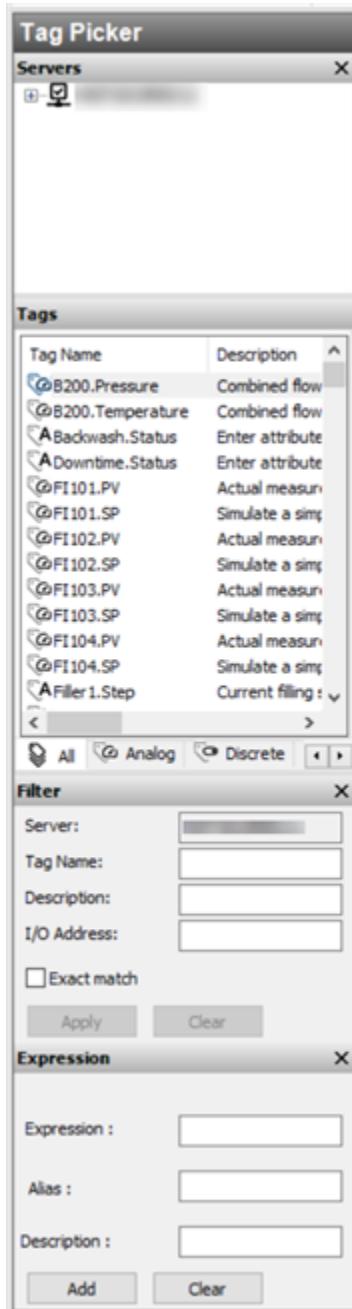
However, if you execute the **Add source tag** command, the 'MyTag' tag is shown as 'MyTag'. If 'MyServer.MyTag' is the only replicated tag, 'MyTag' is considered as a normal tag.

This scenario holds true if the entire replication schedule is removed in the Historian. If only one replication is removed, the list shows the remaining replicated tags.

Tag Picker

The Tag Picker shows which tag groups and tags exist in the database. It shows all of the tags that are visible to the currently logged on user based on his or her permissions.

Using the Tag Picker, you can quickly search the database for tags of a certain type, for tags that match a particular search pattern, or create an expression. You can then select the items you want to include for the client application or control.

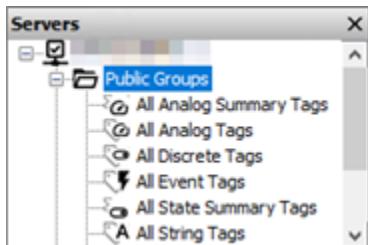


The Tag Picker comprises the following panes:

- Servers Pane
- Tags Pane
- Expression Pane
- Filter pane

Servers pane

The **Servers** pane shows a list of Historian folders. The **Servers** pane enables you to navigate through the folder structure (namespace) of one or more Historian servers and select a group (folder) of tags.



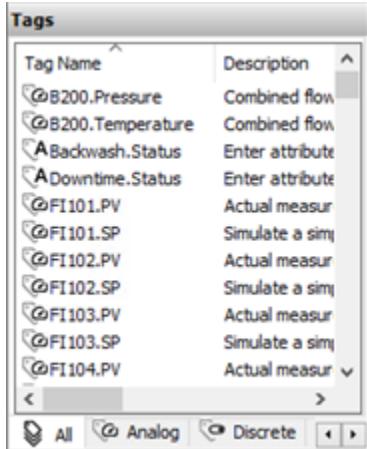
Note: When you select a parent item (such as Public Groups in the illustration above), the Tag Pane shows items from all of the child items (such as All Analog Summary Tags and its peers above).

The **Servers** pane shows the following items:

Category	Description
Servers	All objects that make up the basic Historian system, such as tags, I/O Servers, defined engineering units, storage locations, and so on.
Public Groups	All objects that are visible to all clients. If you have administrative permissions, you can create, rename, and delete groups in the public groups folder.
Private Groups	All objects that are visible to the user that is currently logged on. Users can create, rename, and delete groups in the private groups folder.

Tags pane

The **Tags** pane shows all the tags for the currently selected group in the **Servers** pane.

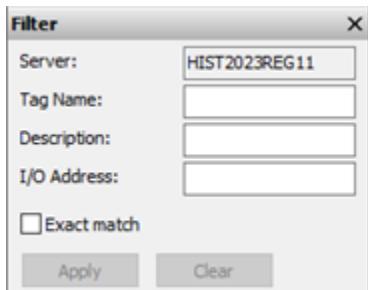


- To select multiple tags from the list, hold the CTRL and/or SHIFT key while clicking.

- To view only tags of a certain type, click the appropriate tab at the bottom of the pane.
- To sort the table by a particular column, click the column heading.

Filter pane

Use the **Filter** pane to filter the tags listed in the **Tags** pane according to criteria that you specify. You can filter tags by name, description, and I/O address.



The following wildcard characters are supported:

Wildcard Character	Filter Function
%	Any string of zero or more characters.
-	Any single character.
[]	Any single character within the specified range or set. For example: <ul style="list-style-type: none">• [a-f]• [abcdef]
[^]	Any single character not within the specified range or set. For example: <ul style="list-style-type: none">• [^a - f]• [^abcdef]

For example, to find all tag names ending with "level," enter "%level." Filter criteria are not case-sensitive.

Note: Wildcard characters cannot be treated literally in a filter; they will always act as wildcard characters.

When the **Servers** pane and the **Filter** pane are both visible, the filter conditions apply to the selected group in the **Servers** pane. When the **Servers** pane is hidden, the filter applies to all of the tags for the selected Historian.

To apply a filter

1. In the **Server** field, specify or verify the server.
This field is not available if the **Servers** pane is visible.
2. In the **Tag name** field, enter the string to match for the tagname.

3. In the **Description** field, enter the string to match for the description.
4. In the **I/O Address** field, enter the string to match for the I/O address.
5. Select **Exact match** to search for tags that exactly match the entire string that you provided for the tagname and/or description fields.

For example, if you specify "level" as the tagname and do not select **Exact match**, any tagname that contains the string "level" appears. This will include tagnames such as "ReactLevel," "ProdLevel," and "\$AccessLevel."

6. Click **Apply** to apply the filter criteria.
7. Click **Clear** to clear the filter criteria.

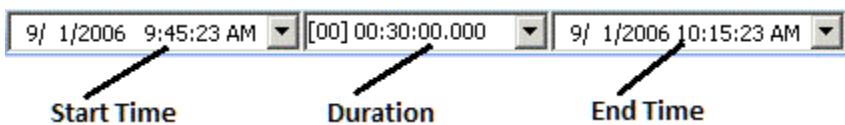
Time picker

The time picker enables you to select a time range by specifying a start time, end time, and/or duration.

An error appears next to the start or end time controls if you specify an invalid time period. For example, an end time before a start time.

To specify a time period

1. On the Time toolbar, specify the start time, end time, and/or duration. To select a date from a calendar, click the down arrow on the start time or end time list. To select a predefined duration, click the down arrow on the duration list.



When you change one of the options, one of the other options is recalculated automatically. While you change the option, a blue frame appears around the option that will be recalculated as a result of the change.

The relation between changed and updated options is as follows:

You change...	The time picker updates...
Start time	End time (based on duration)
End time	Start time (based on duration)
Duration	Start time (based on end time)

If you change multiple options in a row, which option is updated depends on which two other options you changed last. For example, if you change the start time and then the end time, the duration is calculated accordingly. If you change the start time and then the duration, the end time is calculated, and so on.

2. Press Enter.

To specify a time period relative to the current time

1. Do one of the following:
 - On the **Chart** menu, click **Update to Current Time** so that a check mark appears.
 - Click the **Update to Current Time** toolbar button so that it is highlighted.

2. In the duration list of the Time toolbar, click a duration or type one manually.

The start time is automatically calculated as the current time minus the duration you selected, and the trend display is updated with the new time period.

Configure trend application options

The trend options allow the user to configure the trend application. These options apply to all saved trend files. Categories of trend options that can be set include:

- [Configuring retrieval options](#)
- [Configuring color options](#)
- [Configuring time zone options](#)
- [Configure miscellaneous options](#)
- [Configuring other options](#)

Configure retrieval options

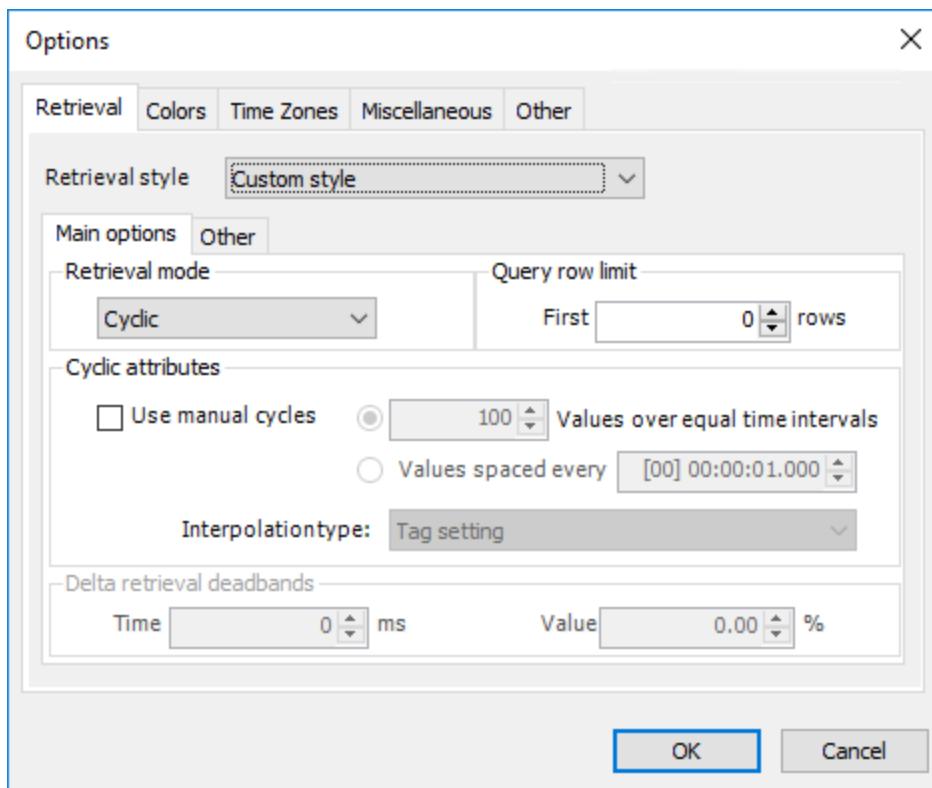
You can define data retrieval options at the application level. These options are used for all tags that do not have their own settings defined.

Application-level retrieval options are not saved in trend files. Therefore, trend files with tags that rely on application-level retrieval settings may look different depending on the retrieval options that are configured in the Trend application that they are opened in. To make sure that your tags are using specific retrieval options, define these options individually for each tag. For more information, see [Configure trend options for a tag](#).

Most retrieval settings that you configure here only apply if you are retrieving data from a Historian with a version of 9.0 or later. If you are using an earlier Historian version, see [Configure other options](#) and [Work with retrieval styles](#) for details.

To configure retrieval options

1. On the **Tools** menu, click **Options**. The **Options** dialog box appears with the **Retrieval** tab selected.



2. Do one of the following:
 - To use a predefined retrieval style, click its name in the **Retrieval style** list. For more information on retrieval styles, see [Work with retrieval styles](#).
 - To use custom retrieval settings, click **Custom style** in the **Retrieval style** list.
3. Specify any additional settings required.
 - If you are using custom retrieval settings, select a retrieval mode and specify all the settings that are relevant to it. For more information, see [Understanding retrieval modes](#).
 - If you are using one of the predefined styles, you can edit all settings that are not covered by the style definition. For information on which settings are covered by style definitions, see [Work with retrieval styles](#).

Because a style definition can contain multiple sets of retrieval settings with different retrieval modes, some of the settings available for editing here may turn out to be irrelevant for the retrieval mode that actually gets used for a given query. However, because there is no way to know in advance whether this will be the case, the settings are still available for editing.

For more information on the various retrieval options, see [Understanding retrieval options](#).

By default, the retrieval settings that you specify here are used for all tags on all trend charts. However, you can override these settings individually for each tag. For more information, see [Configure trend options for a tag](#).

Configure color options

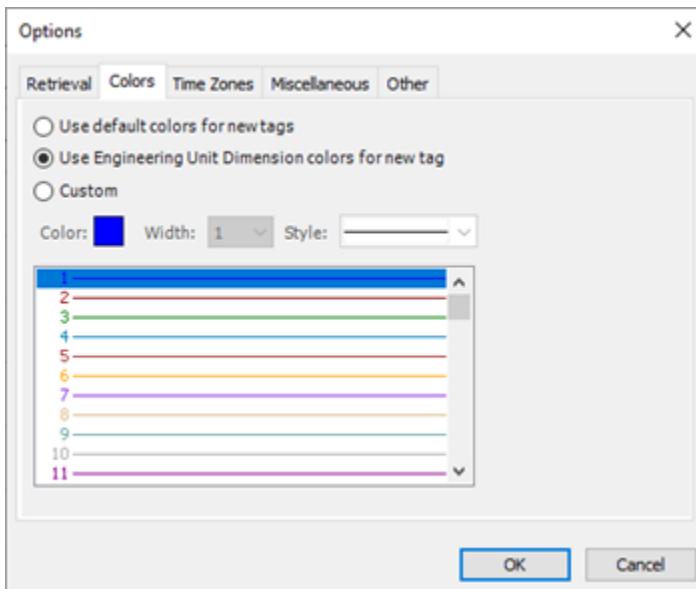
The trend color options control how the trend pen looks for each new tag as it is added to the chart. By default, Trend includes 256 different pen styles, which are numbered from 1 to 256. An unused style is applied each time you add a tag to the trend chart. The trend assigns the lowest pen style that is available. For example, the first

pen style is a solid red line, so the first tag you place in a chart has this style. You can change the default pen styles.

Changing the options does not affect tags that are already in the trend chart.

To configure color options

1. On the **Tools** menu, click **Options**. The **Options** dialog box appears.
2. Click the **Colors** tab.



3. To use the default pen styles for the tags in a trend, select either **Use default colors for new tags** or **Use Engineering Unit Dimension colors for new tag**. Go to step 10.
4. To configure one or more pen styles, select **Custom**.
5. Select a pen number from the list.
6. Click the **Color** box and select or configure a color for the pen line.
7. In the **Width** list, select the width, in pixels, of the pen line.
8. In the **Style** box, select the style of the pen, either a solid line or one of a variety of dashes.
9. Repeat steps 5 through 8 for each pen style you want to configure.
10. Click **OK**.

Configure time zone options

You can configure a trend so that data appears with time stamps that reflect any time zone. For example, you may want to configure the trend to the same time as the location of the server.

To configure time zone options:

1. On the **Historian** tab, in the **Publish** group, click **Options**, and then click **Options**. The **Options** dialog box appears.
2. Select the **Time Zone** tab. The grid displays the current time zone and daylight saving time settings for the following entities:

Entity	Description
Application	The Historian Client Workbook application. You can select the time zone for the data as it appears in the Workbook application.
Client	The physical computer on which the Workbook application is installed. The time zone displayed for the client is for informational purposes only and cannot be changed using the Workbook application.
<Server>	The Historian(s) to which the Workbook application is currently connected. The time zone displayed for the server(s) is for informational purposes only and cannot be changed using the Workbook application.

3. In the **Time zone** list, click the name of the time zone to use for the Workbook application.

The time zone for the Workbook application in the grid displays the new time zone picked.

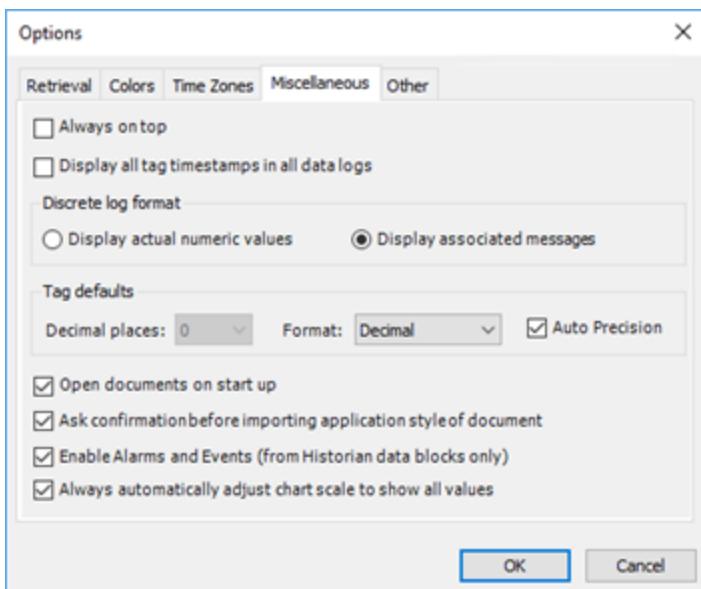
For example, consider a SCADA application that monitors a pipeline between Houston, Texas and Lake Forest, California. The Workbook application is installed on a computer located in Houston, Texas. Therefore, the time zone entry for the Client entity displays Central Standard Time. The server is also located in Houston, Texas. The time zone entry for the Server entity also displays Central Standard Time. You want to send a Workbook file to an engineer located at the start of the pipeline in Lake Forest to aid in troubleshooting a problem. You can set the time zone of the Workbook application to reflect the time of Lake Forest, California (Pacific Standard Time), so that the workbook that you send to the engineer displays data in a time zone that is relevant to him/her.

4. Click **OK**.

Configure miscellaneous options

To configure miscellaneous options

1. On the **Tools** menu, click **Options**. The **Options** dialog box appears.
2. Click the **Miscellaneous** tab.



3. Select **Always on top** to always display Trend as the top-most program on the computer desktop.
4. Select **Display all tag timestamps in all data logs** to include the time stamps for all tags in the data log.
5. In the **Discrete log format** area, configure how the values for discrete tags appear in the data log. Select **Display actual numeric values** to show the numeric value for the discrete tag, either 1 for the TRUE state or 0 for the FALSE state. Select **Display associated messages** to show the text associated with the TRUE or FALSE state of the discrete tag. For example, "On" or "Off," "Started" or "Stopped."
6. In the **Tag defaults** area, configure how tag values appear in the chart. Changes to these settings are not applied until the next tag is added to the chart.

<ul style="list-style-type: none"> • Decimal places • Format • Auto Precision 	<p>The number of values that appear to the right of the decimal period.</p> <p>The format for tag values, either decimal format or scientific format. For the scientific format, the value appears with an E denoting the exponent.</p> <p>If selected, the number of decimal places is automatically set based on the value range. When Auto Precision is selected, the Decimal places field is read-only.</p>
--	---

7. Configure the following additional options:

<ul style="list-style-type: none"> • Open documents on start up • Ask confirmation before importing application style of document 	<p>If selected, when you start the Trend application, it automatically reopens the trend files that were open when you closed it. Otherwise, a new trend file is created.</p> <p>Some style information is saved in trend files, and this may differ from the default styles configured in the Trend application. If this option is selected, when you open a trend file you will be prompted to</p>
---	--

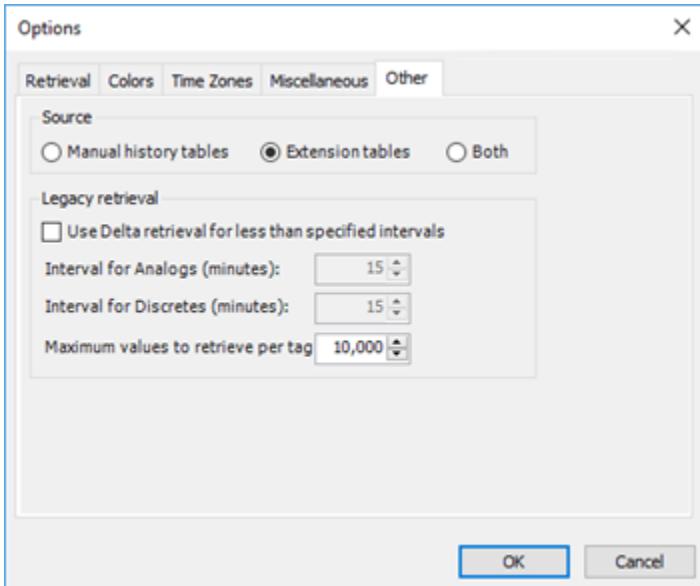
	confirm whether or not to override the default settings with the settings from the trend file. Otherwise, the settings from the trend file will override the default settings automatically, with no confirmation required.
<ul style="list-style-type: none">• Enable Alarms and Events (from Historian data blocks only)• Always automatically adjust chart scale to show all values	Select this option to enable the alarms and events overlay in the trend chart. If this option is selected, when trend values cross the chart boundaries, the Trend application will automatically adjust the chart scale so that all values are displayed.

8. Click **OK**.

Configure other options

To configure other options

1. On the **Tools** menu, click **Options**. The **Options** dialog box appears.
2. Click the **Other** tab.



3. In the **Source** area, specify the Historian tables from which data will be retrieved.

<ul style="list-style-type: none">• Manual history tables• Extension tables	Normal SQL Server tables that are used to store data. These are the <code>ManualAnalogHistory</code> and <code>ManualDiscreteHistory</code> tables.
	Logical tables that are populated from the Historian data files. These tables support the Historian time domain extensions for handling data.

4. In the **Legacy retrieval** area, specify the retrieval mode for data that is retrieved from the Historians with a version earlier than 9.0.

For information on how these settings interact with a retrieval style that you may have selected, see [Working with Retrieval Styles](#).

<ul style="list-style-type: none">• Both	Select this option to retrieve data from both the manual and extension tables.
<ul style="list-style-type: none">• Use Delta retrieval for less than specified intervals	Select this check box to use delta retrieval mode for query time periods that are less than a specified amount.
<ul style="list-style-type: none">• Interval for Analogs	The time period, in minutes, for which delta values are retrieved for analog tags. For greater time periods, cyclic retrieval is used instead. Valid values are 0 to 250,000. The default value is 15.
<ul style="list-style-type: none">• Interval for Discretes	The time period, in minutes, for which delta values are retrieved for analog tags. For greater time periods, cyclic retrieval is used instead. Valid values are 0 to 10,000. The default value is 15.
<ul style="list-style-type: none">• Maximum values to retrieve per tag	The maximum number of values to return per tag. Valid values are 0 to 30,000. The default value is 10,000.

5. Click **OK**.

Configure trend file properties

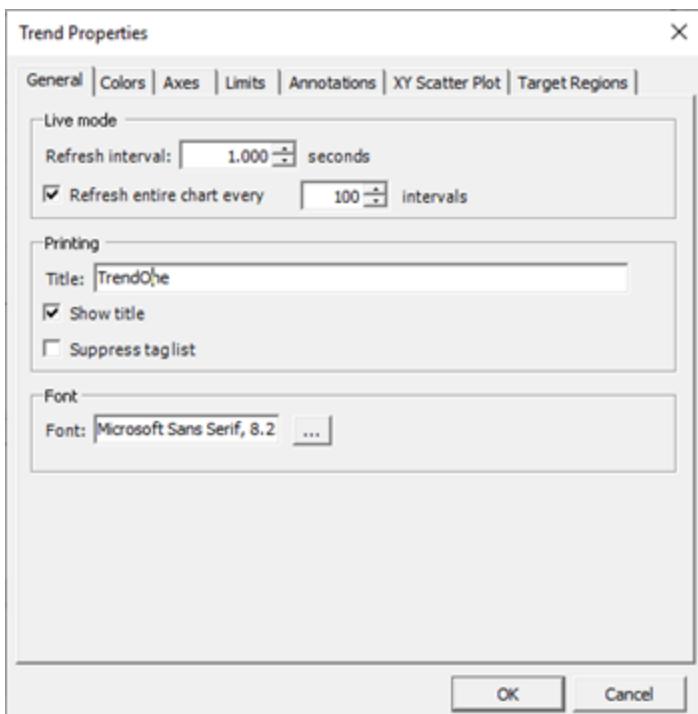
The trend properties enable you to configure a trend file. Trend file properties are saved with the trend file. Categories of trend properties that can be configured include:

- [Configuring General Properties](#)
- [Configuring Color Properties](#)
- [Configuring Axis Properties](#)
- [Configuring Limit Properties](#)
- [Configuring Annotation Properties](#)
- [Configuring Target Region Properties](#)

Configure general properties

To configure general properties

1. On the **Chart** menu, click **Properties**. The **Trend Properties** dialog box appears.
2. Select the **General** tab.

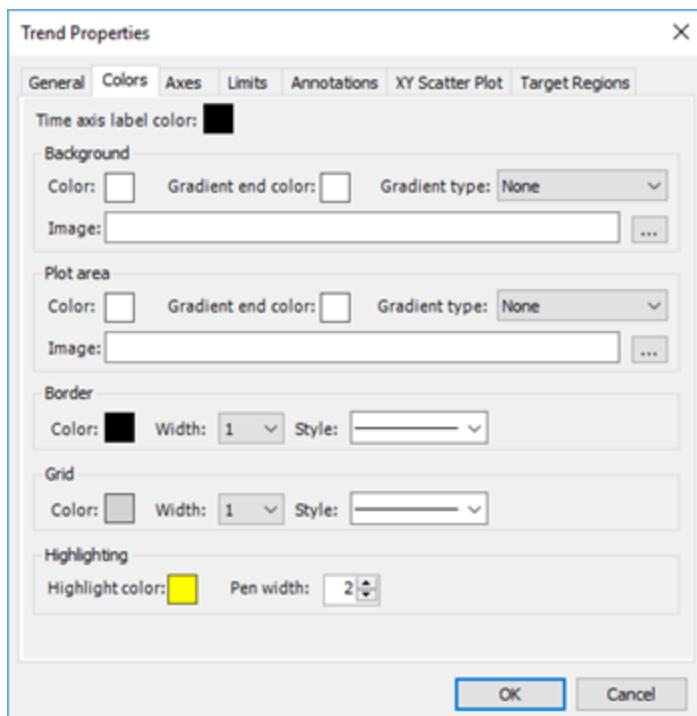


3. In the **Refresh interval** box, specify the time period, in seconds, at which the chart is refreshed if set to live mode. Valid values are 0.25 to 300. The default value is 1.
4. In the **Refresh entire chart every XX intervals** box, specify the number of refresh intervals after which the entire chart is refreshed. The chart is not only refreshed with the new live data, but all the data in the chart is refreshed. Valid values are 1 to 100,000. The default is 100.
5. In the **Printing** area, configure options for chart printing.
 - **Title:** The title of the chart.
 - **Show title:** Show the title in the printout.
 - **Suppress tag list:** Do not include the tag list in the printout.
6. In the **Font** area, click the **Font** icon to select the name, style, and size of the font that is to be displayed on the chart and Tag List.
7. Click **OK**.

Configure color properties

To configure color properties

1. On the **Chart** menu, click **Properties**. The **Trend Properties** dialog box appears.
2. Click the **Colors** tab.



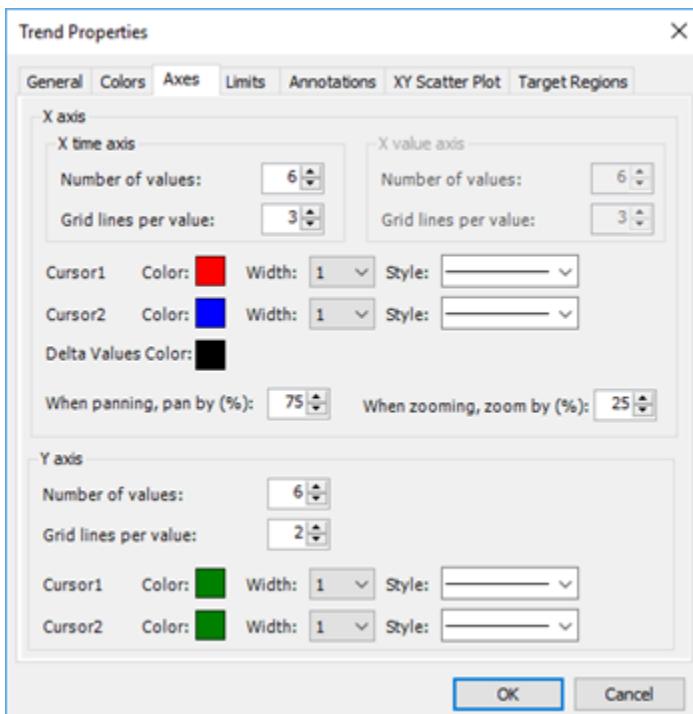
3. Click the **Time axis label color** box to select or configure the color for the time labels that appear at the bottom of the chart.
4. In the **Background** area, configure the colors or image to use for the background of the entire chart area.
 - **Color:** Click to select or configure a main color. If you are using a gradient fill, this is the starting color for the gradient.
 - **Gradient end color:** Click to select the ending color for the gradient. The gradient starts with the main color and fade to the gradient end color.
 - **Gradient type:** The starting point for the flow of the gradient. Valid values are Center, Diagonal Left, Diagonal Right, Horizontal Center, Left Right, Top Bottom, and Vertical Center. For example, if you select green as main color, white as the gradient end color, and center as the gradient type, the center of the chart is green and fades to white towards the surrounding edges.
 - **Image:** The name of the image to use as the background. The image is resized to fit within the chart area. The color of the pixel in the lower left corner of the image is used as the transparency mask for the image. Click the ellipses button to browse for and select an existing image.
5. In the **Plot** area, configure the colors or image to use for the chart plotting area. Options are the same as for the **Background** colors.
6. In the **Border** area, configure the appearance of the chart's border.
 - **Color:** Click to select or configure a color.
 - **Width:** The width, in pixels, of the border line.
 - **Type:** The style of the border line.
7. In the **Grid** area, configure the color for the grid lines of the chart. Options are the same as for **Border**.
8. In the **Highlighting** area, configure the color and pen width to be used for tag highlighting.
 - **Highlight color:** Click to select or configure a color for highlighting the tag curve.
 - **Pen width:** Specify how wide (in pixels) a highlighted curve should be.

9. Click **OK**.

Configure axis properties

To configure axis properties

1. On the **Chart** menu, click **Properties**. The **Trend Properties** dialog box appears.
2. Select the **Axes** tab.



3. In the **X axis** area, configure the properties for the horizontal axis.
 - **Number of values:** The number of values that are shown along the time axis. The values are shown at evenly-spaced points along the axis. The number of values remain the same even if you zoom in and out. The valid range is from 2 to 15, with a default value of 6.
 - **Grid lines per value:** The number of grid lines that appear between each tag value plotted on the chart. The valid range is from 1 to 20, with a default value of 3.
 - **Cursor1 Color, Cursor2 Color:** Click to select or configure the color for each time axis cursor.
 - **Delta Values Color:** Click to select or configure the color of the Delta cursor label displayed above the plot area in the upper-right corner.
 - **Width:** The width of each time axis cursor.
 - **Style:** The line style for each time axis cursor.
 - **When panning, pan by:** The percentage used for the panning scale. The panning scale range is from 1 to 100.
 - **When zooming, zoom by:** The percentage used for the zoom. The zoom factor range is from 1 to 100.
4. In the **Y axis** area, configure the properties for the vertical axis.
 - **Number of values:** The number of values that are shown along the value axis. The time stamps are shown at evenly-spaced points along the axis. The number of values remain the same even if you zoom

in and out. The valid range is from 2 to 15, with a default value of 6.

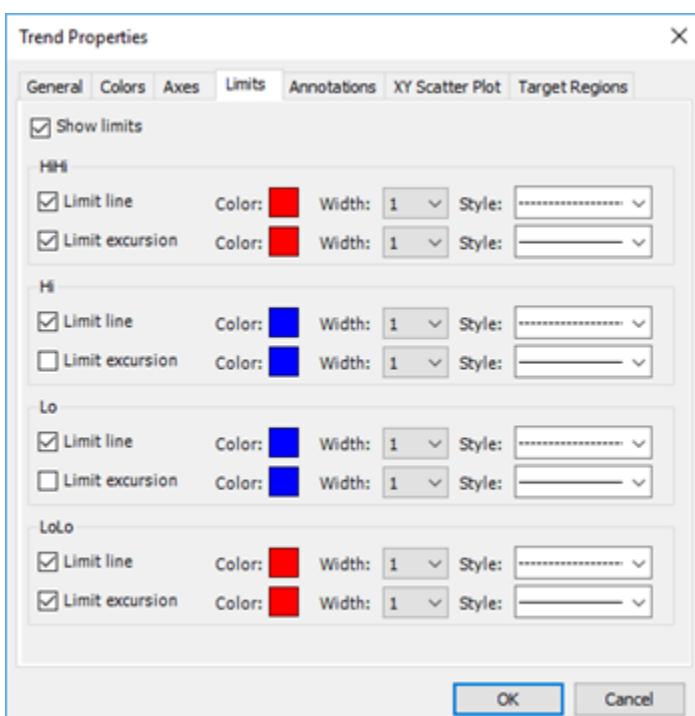
- **Grid lines per value:** The number of grid lines appearing between each tag value that is plotted on the chart. The valid range is from 1 to 20, with a default value of 3.
- **Color:** Click to select or configure the color for each value axis cursor.
- **Width:** The width of each value axis cursor.
- **Style:** The line style for each value axis cursor.

5. Click **OK**.

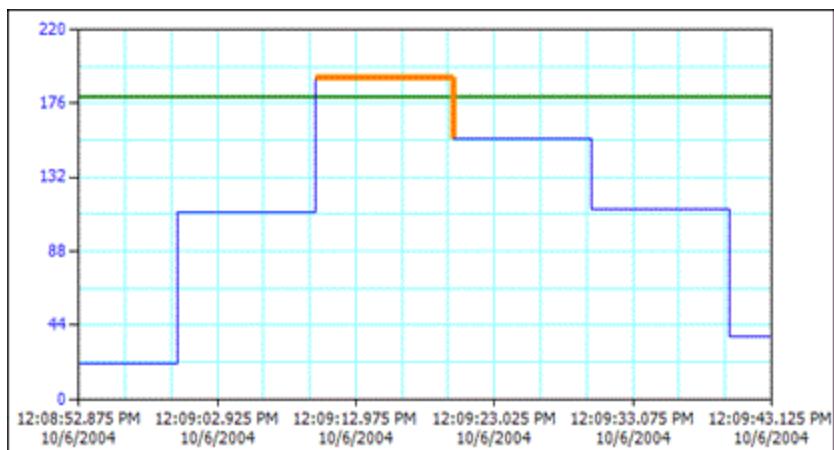
Configure limit properties

To configure limit properties

1. On the **Chart** menu, click **Properties**. The **Trend Properties** dialog box appears.
2. Select the **Limits** tab.



3. Select the **Show Limits** check box to show horizontal lines on the chart at the limit values configured for analog tags.

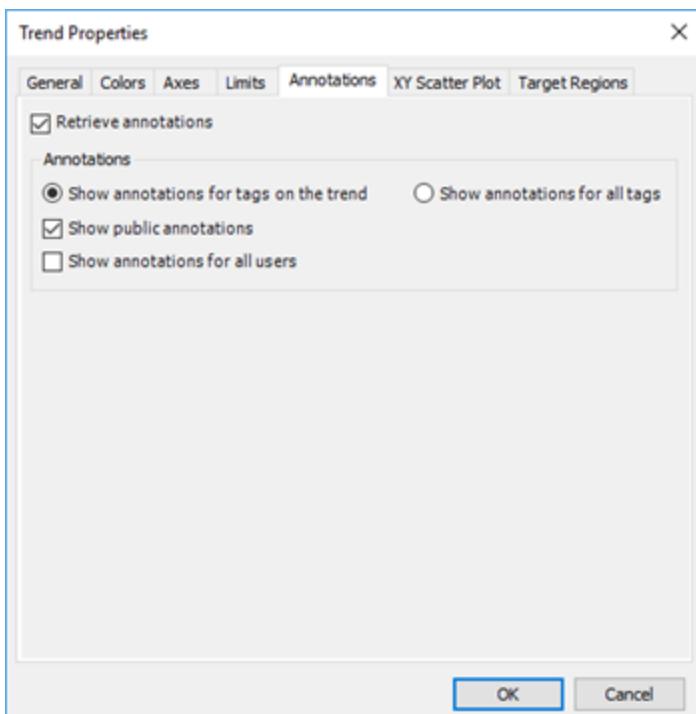


4. For each type of limit (HiHi, Hi, Lo, and LoLo), configure the properties of the line.
 - **Limit line:** Select to include a line on the chart for the limit value. For example, if an analog tag has a Hi limit of 1800, a horizontal line is drawn at the 1800 mark on the vertical axis.
 - **Limit excursion:** Select to indicate the portion of the trace that is outside of the limit.
 - **Color:** The color of the line.
 - **Width:** The width of the line.
 - **Style:** The style of the line.
5. Click **OK**.

Configure annotation properties

To configure annotation properties

1. On the **Chart** menu, click **Properties**. The **Trend Properties** dialog box appears.
2. Select the **Annotations** tab.



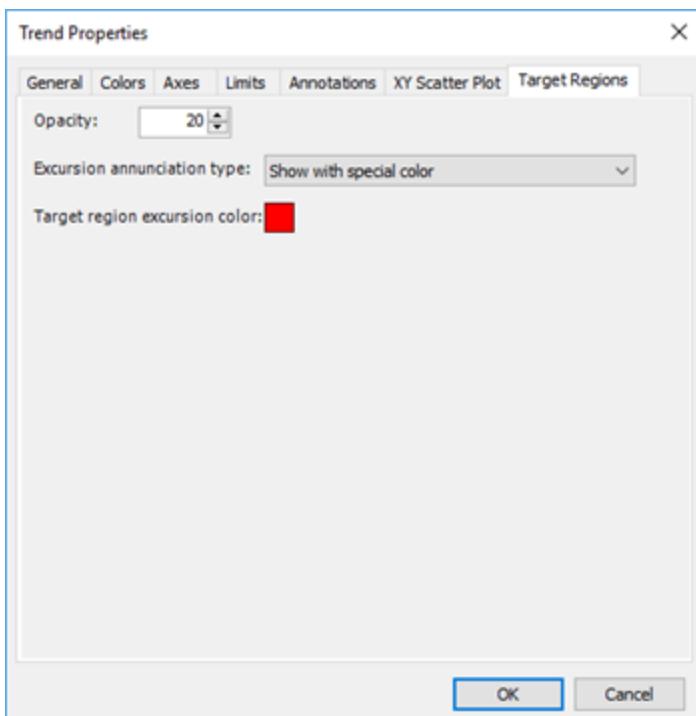
3. Select the **Retrieve annotations** check box to retrieve annotation information and show them on the chart.
4. In the **Annotations** area, configure how annotations are shown on the chart.
 - **Show annotations for tags on the trend:** Show only the annotations for the tags currently charted in the trend.
 - **Show annotations for all tags:** Show all annotations for all tags. For those tags not currently charted on the trend, the annotation marker appears at the top of the chart at the point in time on chart at which the annotation was made.
 - **Show public annotations:** Show only public annotations. You can see your private annotations and the public annotations of other Historian users.
 - **Show annotations for all users:** Show both public and private annotations. You can see your private annotations, as well as both the public annotations and private annotations of others.
5. Click **OK**.

For information on the **XY Scatter Plot** tab, see [Configuring Scatter Plot Properties](#).

Configure target region properties

To configure target region properties

1. On the **Chart** menu, click **Properties**. The **Trend Properties** dialog box appears.
2. Select the **Target Regions** tab.



3. In the **Opacity** box, enter the opacity with which you want the target region to appear on the trend chart.
4. In the **Excursion annunciation type** list, specify whether values that fall outside the target region should be highlighted. Select **Show with special color** to highlight parts of the trend graph that are outside the target region in a special color. To select the color, click the color box next to **Target region excursion color**. Select **None** if you do not want any special highlighting.
5. Click **OK**.

Configure a trend to use a summary tag

Summary tags consist of summarized data of tags from a Historian server. A summary tag provides an aggregation calculation (minimum, maximum, average, or sum) that is configured on a Historian server. Summary tags are of two types: analog and state.

- Analog summary tags provide summary statistics for analog tags.
- State summary tags provide the summaries of the states of the tag value of analog (integer only), discrete, and string tags.

You can select one or more summary tags from the Tag Picker and drag them to the Tag List pane. When you drag a summary tag to the Tag List, the Trend application plots the value of the aggregate calculation on the Trend chart. The aggregate calculation is performed when you configure the summary tag on the Historian server. For example, if you have configured a ReactTemp_Avg_Hourly summary tag to store the hourly averages of the Reactor temperature, the Trend application shows the hourly average value of the Reactor temperature when you drag and drop the ReactTemp_Avg_Hourly tag to the Trend chart. For more information on the Tag List pane, see [View tag definition information](#).

You can configure trend options for a summary tag. For more information, see [Configure trend options for a tag](#). For more information on retrieving summary tags, see [Configure retrieval options for a tag](#) or [Configure retrieval options](#).

Absolute or relative times

The following time modes are available for the trend chart:

- Absolute time
- Relative time

The time mode selection is saved as part of the chart definition when you save the .aaTrend file.

Absolute time

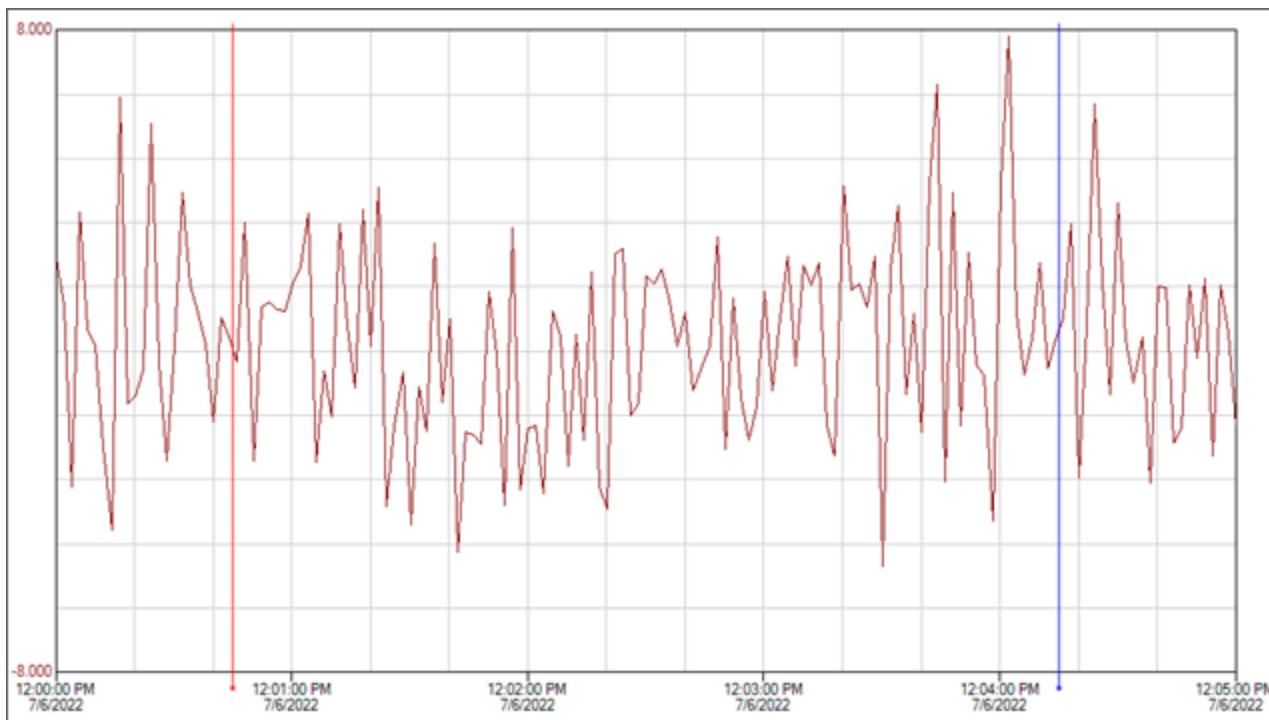
In absolute time mode, you determine the time period for the query by setting a combination of the start time, duration, and end time.

To use absolute time

On the **View** menu, in the **Time** group, select **Absolute**.

For example, the following chart shows five minutes of data in absolute time.

The query for the data starts at 12:00:00 and ends at 12:05:00, and the time axis values reflect these times.



In absolute time mode:

- The times shown for the time axis cursors, if enabled, are absolute times.
- The times shown in the Time Bar are absolute times.
- The time offset for each tag can be configured in the Tag List. For more information, see [Use time offsets to compare data in absolute time](#).

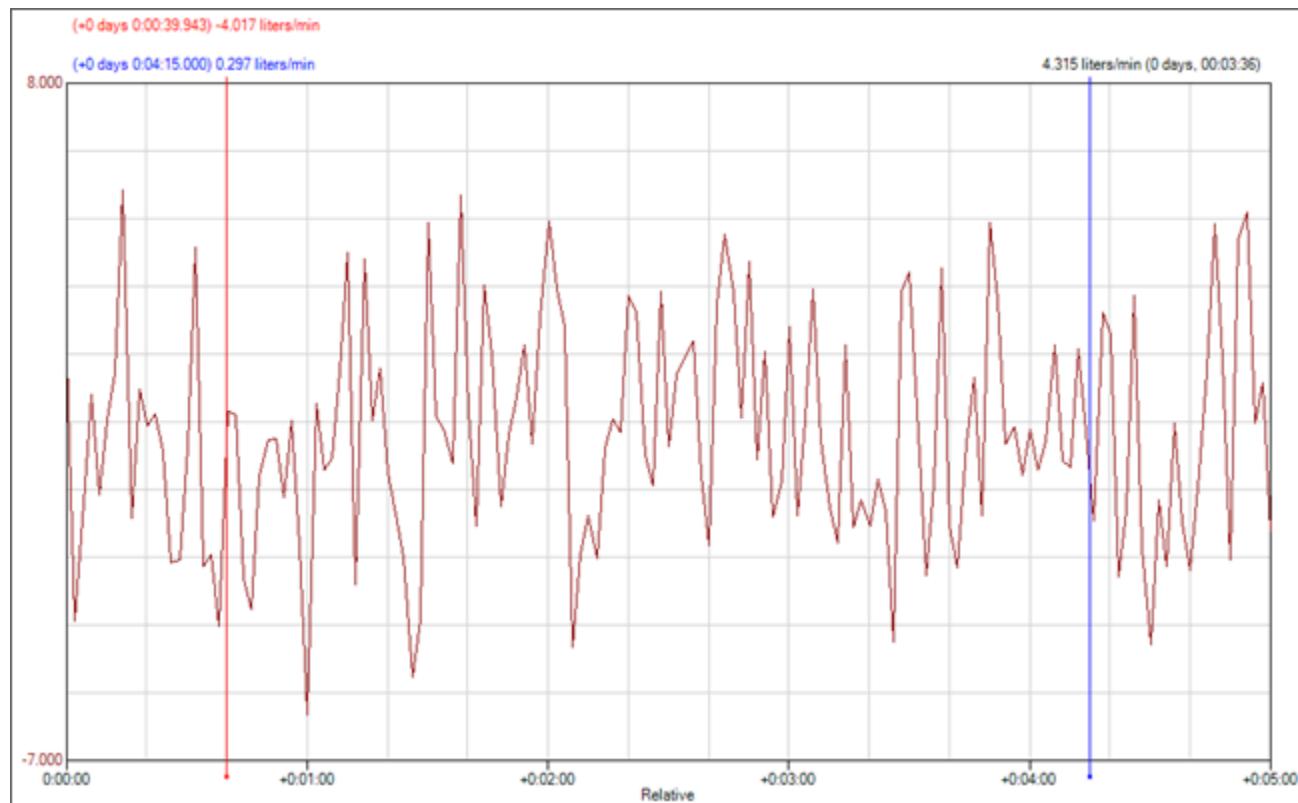
Relative time

For the relative time mode, a base value (such as 0:00:00.0) is used for the start time of the chart, and the end time is calculated based on the time span for the query. Switching to relative mode does not change the data shown in the chart nor the actual start and end time of the trend query. Only the time axis is updated.

To use relative time

On the **View** menu, in the **Time** group, select **Relative**.

For example, the following chart shows five minutes of data in relative mode. The query for the data starts at 12:00:00 and ends at 12:05:00, but the time axis shows a start time of 0:00:00 and an end time of +0:05:00.



In relative time mode:

- The times shown for the time axis cursors, if enabled, are relative times. For details on the display format for times, see [Time offset formats](#).
- The time fields in the Time Bar are relative times. The first field is the base time (for example, 0:00:00.000), and the second field is the timespan of the query. The third field is the end time of the query (the base time plus the timespan). To offset the time for the query, adjust the base time (for example, 1:00:00.000 to offset it by +1 hour).
The first field (the base time) defaults to 0:00:00 when you switch to relative mode.
- The Tag List shows the actual start time for the tag data in the Start Time column.
- The **Tag Properties** dialog box shows the start time option. For more information, see [Configure trend options for a tag](#).

Switch between absolute and relative time: example

When you change the time mode, the Time Bar and individual tag time settings convert between absolute times and relative offsets.

The following table summarizes the states for the Time Bar and three tag offsets/dates for some example data. Tag1 is the currently selected tag. In this example, the actions performed for the different steps are:

1. At exactly 2005-07-04 10:00, "2 hours" is selected from the Time Bar while the **Update to Current Time** option is enabled. The chart is in absolute mode.
2. The chart is switched to relative time mode.
3. The start and end values are changed in the Time Bar.
4. The chart is switched back to absolute time mode.

Step:	1	2	3	4
Mode:	Absolute	Relative	Relative	Absolute
Time Bar Start:	2005-07-04 8:00	0:00	-0:15	2005-07-04 7:45
Time Bar End:	2005-07-04 10:00	2:00	0:45	2005-07-04 8:45
Tag1:	- 24:00	2005-07-03 8:00	2005-07-03 8:00	- 24:00
Tag2:	0:00	2005-07-04 8:00	2005-07-04 8:00	0:00
Tag3:	+3:30	2005-07-04 11:30	2005-07-04 11:30	+3:30
Time Axis Start:	2005-07-03 8:00	0:00	-0:15	2005-07-03 7:45
Time Axis End:	2005-07-03 10:00	2:00	0:45	2005-07-03 8:45

Time offset formats

The application displays time offsets in the following formats:

[±][d]hh:mm[:ss[.fff]]

-OR-

[±]HH:mm[:ss[.fff]]

-OR-

[±]d[.FF]

Items in square brackets ([and]) are optional. Colons and periods (: and .) are literal characters.

The notation variables are as follows.

Item	Description
±	Minus sign indicating a negative time. Positive time is assumed.
d	Days
hh	Hours, ranging from 0 to 23.
HH	Hours of 24 or greater.
mm	Minutes, ranging from 0 to 59.
ss	Seconds, ranging from 0 to 59.
fff	Fractional seconds, from 1 to 7 decimal digits.
FF	Fractional days.

Time offsets are shown in the application in either a short or long form:

Short form:	HH:mm:ss.ff or d hh:mm:ss.fff
Long form:	d <label> hh:mm:ss.fff

In the short form:

- "d" is omitted for offsets less than or equal to 48 hours
- ".fff" is omitted for offsets greater than 60 seconds
- ":ss.fff" is omitted for offsets greater than 24 hours

Thus, for periods of less than 60 seconds, the short form is never longer than 11 characters. For the short form, the hours in HH format (rollover at 48 instead of 24) are shown only if days are not to be displayed anywhere on the time axis.

The "<label>" is the localized word for "days" or "day." The period (.) and colon (:) are replaced with the appropriate characters from the regional settings.

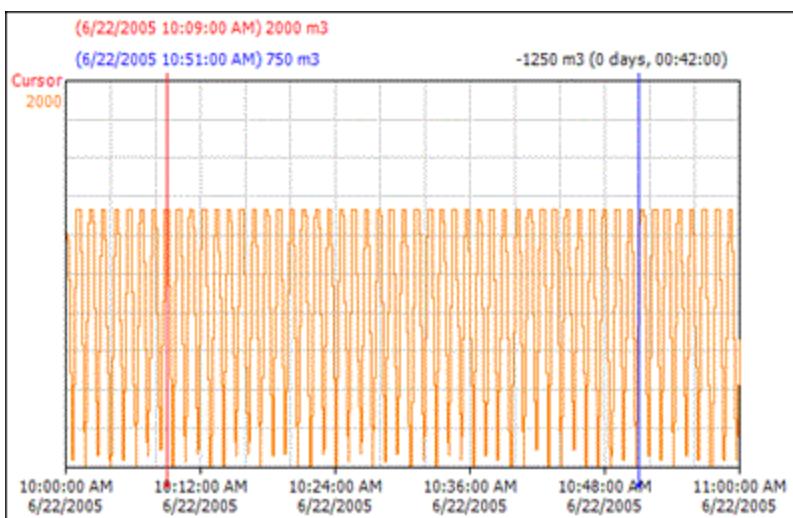
Use time offsets to compare data in absolute time

In absolute time mode, you can use a time offset to compare the same data from different time periods. For example, you may want to compare data from a shift at 10:00 a.m. to data from a shift at 11:00 a.m. The time offset feature enables you to adjust the time period for one of the shifts so that the data appears as if it occurred during the same time period as the other shift. Using a time offset enables you to easily see the differences between the data on the trend chart.

To use a time offset

1. Create a trend for batch of data that you want to use as the basis for comparison.

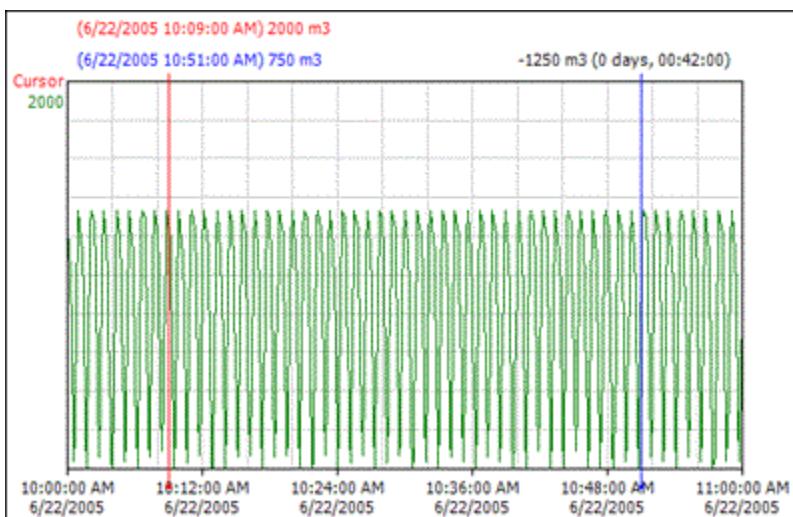
In this example, the chart is configured to show data for the ReactLevel tag between 6/22/2005 10:00:00 AM and 6/22/2005 11:00:00 AM.



2. Add the same tag again to the trend chart for the same time period.

In this example, the ReactLevel tag was added again to the chart.

3. Because the data is identical, you only see single trend curve in the chart.



4. Determine the time span for the data you want to compare with the base batch of data.
5. To specify the time offset for the data to compare, double-click on the tag in the Tag List. The **Tag Properties** dialog box appears.
6. In the **Time offset** box, configure the amount of time that the data shown in the chart is to be offset from the actual query time. For more information on the format, see [Time offset formats](#).

In this example, this data is to be compared with the base batch that occurred an hour before, so the time offset is set to one hour.

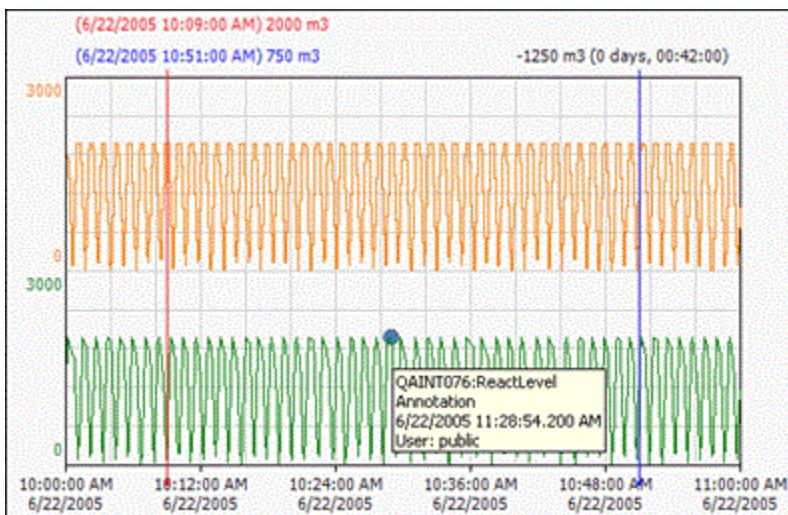
7. Click **OK**.

In this example, the data for the ReactLevel tag between 6/22/2005 11:00:00 AM and 6/22/2005 12:00:00 AM includes an annotation made around 11:30:00 AM.



8. Stack the traces so that you can see both sets of data separately and then select the first tag that you added to the chart.

In this example, the trend curve for the later set of data (shown in green) appears on the chart, even though the time axis reflects the time of the base batch of data (shown in orange).



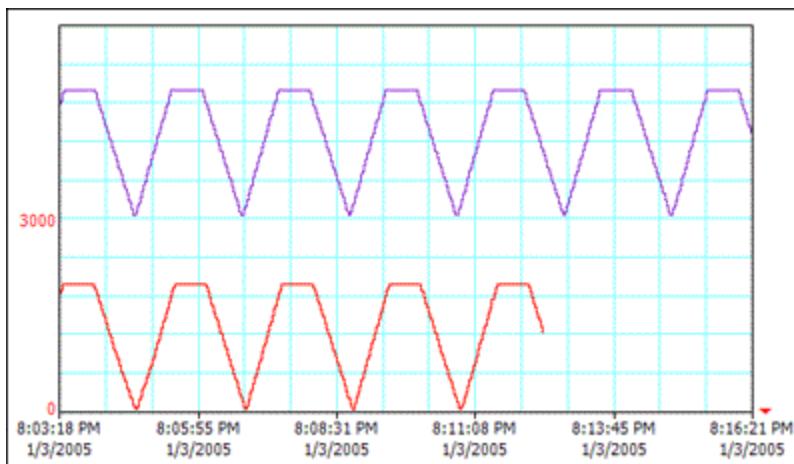
9. To view the chart in relative mode, on the **View** menu, click **Relative Time**. The time axis now shows the time span for the base batch starting at 0:00:00 instead of the actual time.



You can also use the offset to compare a trend curve against another curve either forward or backward in time. To do this, set the time offset of the primary batch of data so that the start time is the same as the start time for the batch of data you want to compare.

In the following example, the time offset for the complete batch is set to a value of -01.00.44. The complete batch appears as the top curve in the chart.

The incomplete curve at the bottom of the chart is plotted in live mode next to the complete curve at the top.



Replicated tags

You can replicate tag information in a Historian from one Historian to another. This allows you to replicate tag data from one or more Historians (known as tier-1 historians) to one or more other historians (known as tier-2 historians). You can replicate tag data to the same server as the tier-1 historian.

You can replicate tag data directly using simple replication, where the tag information is replicated directly to the tier-2 historian. For simple replication, every value for a tag is copied. You can also set up summary tags that receive a summarized version of the tag data.

You can drill down from a source tag to its replicated tag or drill up from a replicated tag to its source tag. You can add a source tag with its replicated tag or a replicated tag with its source tag in the active trend chart. You can also replace a source tag with its replicated tag or a replicated tag with its source tag in the active trend

chart.

Add a source tag or replicated tag

You can select a source tag or replicated tag from the Tag Picker to add the corresponding replicated tag or source tag to the active trend chart.

To add a source tag or replicated tag

1. Select a tag in the Tag Picker.
2. If the selected tag is a source tag, do the following:
 - In the **Tags** pane, right-click the selected tag, then select **Add - replicated tag**, and then select the tag that you want to add to the trend chart. The corresponding replicated tag is added to the active trend chart.
3. If the selected tag is a replicated tag, do the following:
 - In the **Tags** pane, right-click the selected tag, and then click **Add - source tag**. The corresponding source tag is added to the active trend chart.

The **Add** command is not available if:

- Multiple tags are selected in the Tag Picker.
- A normal tag that is neither a source tag nor a replicated tag is selected in the Tag Picker.

Note: You cannot execute the **Add** command if a source tag is deleted but its replication configuration still exists in the Historian.

The replicated tags are not listed in the context menu if:

- The replicated tags are not committed in the Historian.
- The replication schedule is removed from the Historian. For example, you are connected to a Historian 10.0 server and you create a tag called 'MyTag'. 'MyTag' is replicated as a simple tag called 'MyServer.MyTag'. When you execute the **Add - replicated tag** command, the 'MyServer.MyTag' tag is shown. When you execute the **Add - source tag** command, the 'MyTag' tag is shown. At this instance, if the replication link between 'MyTag' and 'MyServer.MyTag' is removed and if you execute the **Add - replicated tag** command, the 'MyServer.MyTag' tag is not shown in the list of replicated tags.

However, if you execute the **Add - source tag** command, the 'MyTag' tag is shown as 'MyTag'. If 'MyServer.MyTag' is the only replicated tag, 'MyTag' is considered as a normal tag.

The above scenario holds true if the entire replication schedule is removed in the Historian. If only one replication is removed, the list shows the remaining replicated tags.

Find a source tag or replicated tag

You can select a source tag or replicated tag from the Tag Picker to find the corresponding replicated or source tag.

To find a source tag or replicated tag

1. Select a tag in the Tag Picker.

2. If the selected tag is a source tag, do the following:
 - In the **Tags** pane, right-click the selected tag, point to **Find - replicated tag**, and then click the tag that you want to find.
The application navigates within the Tag Picker to find the corresponding replicated tag.
3. If the selected tag is a replicated tag, do the following:
 - In the **Tags** pane, right-click the selected tag, and then click **Find - source tag**.
The application navigates within the Tag Picker to find the corresponding source tag.

The **Find** command is not available if:

- Multiple tags are selected in the Tag Picker.
- A normal tag that is neither a source tag nor a replicated tag is selected in the Tag Picker.

Note: You cannot execute the **Find** command if a source tag is deleted but its replication configuration still exists in the Historian.

The replicated tags are not listed in the context menu if:

- The replicated tags are not committed in the Historian.
- The replication schedule is removed from the Historian. For example, you are connected to an Historian 10.0 server and you create a tag called 'MyTag'. 'MyTag' is replicated as a simple tag called 'MyServer.MyTag'. When you execute the **Find - replicated tag** command, the 'MyServer.MyTag' tag is shown. When you execute the **Find - source tag** command, the 'MyTag' tag is shown. At this instance, if the replication link between 'MyTag' and 'MyServer.MyTag' is removed and if you execute the **Find - replicated tag** command, the 'MyServer.MyTag' tag is not shown in the list of replicated tags. However, if you execute the **Find - source tag** command, the 'MyTag' tag is shown as 'MyTag'. If 'MyServer.MyTag' is the only replicated tag, 'MyTag' is considered as a normal tag.

The above scenario holds true if the entire replication schedule is removed in the Historian. If only one replication is removed, the list shows the remaining replicated tags.

Replace a source tag or replicated tag

You can replace a source tag with its replicated tag or a replicated tag with its source tag in the active trend chart.

To replace a source tag or replicated tag

1. Select a tag in the Tag List.
2. If the selected tag is a source tag, do the following:
 - Right-click the selected tag, point to **Replace with replicated tag**, and then click the tag that you want to replace with.
The source tag is replaced by the selected replicated tag in the Tag List and in the active trend chart without changing the pen configuration such as the pen color, width, or retrieval mode.
3. If the selected tag is a replicated tag, do the following:
 - Right-click the selected tag and click **Replace with source tag**.
The replicated tag is replaced by the source tag in the Tag List and in the active trend chart without

changing the pen configuration such as the pen color, width, or retrieval mode.

The **Replace** command is not available if a normal tag that is neither a source tag nor a replicated tag is selected in the Tag Picker.

Note: You cannot execute the **Replace** command if a source tag is deleted but its replication configuration still exists in the Historian.

The following happens if you execute the **Replace** command on a tag configured as an x-axis tag in a XY Scatter Plot.

- If a tag is used as the x-axis tag, the **Replace with source tag** or **Replace with replicated tag** command replaces the x-axis tag reference. For example, a tag called 'MyTagY' is configured to have 'MyTagX' as the x-axis tag. Replacing 'MyTagX' with 'MyTagX2' changes the configuration of the 'MyTagY' and the x-axis tag is replaced by 'MyTagX2'.

The replicated tags are not listed in the context menu if:

- The replicated tags are not committed in the Historian.
- The replication schedule is removed from the Historian. For example, you are connected to a Historian 10.0 server and you create a tag called 'MyTag'. 'MyTag' is replicated as a simple tag called 'MyServer.MyTag'. When you execute the **Replace with replicated tag** command, the 'MyServer.MyTag' tag is shown. When you execute the **Replace with source tag** command, the 'MyTag' tag is shown. At this instance, if the replication link between 'MyTag' and 'MyServer.MyTag' is removed and if you execute the **Replace with replicated tag** command, the 'MyServer.MyTag' tag is not shown in the list of replicated tags.

However, if you execute the **Replace with source tag** command, the 'MyTag' tag is shown as 'MyTag'. If 'MyServer.MyTag' is the only replicated tag, 'MyTag' is considered as a normal tag.

The above scenario holds true if the entire replication schedule is removed in the Historian. If only one replication is removed, the list shows the remaining replicated tags.

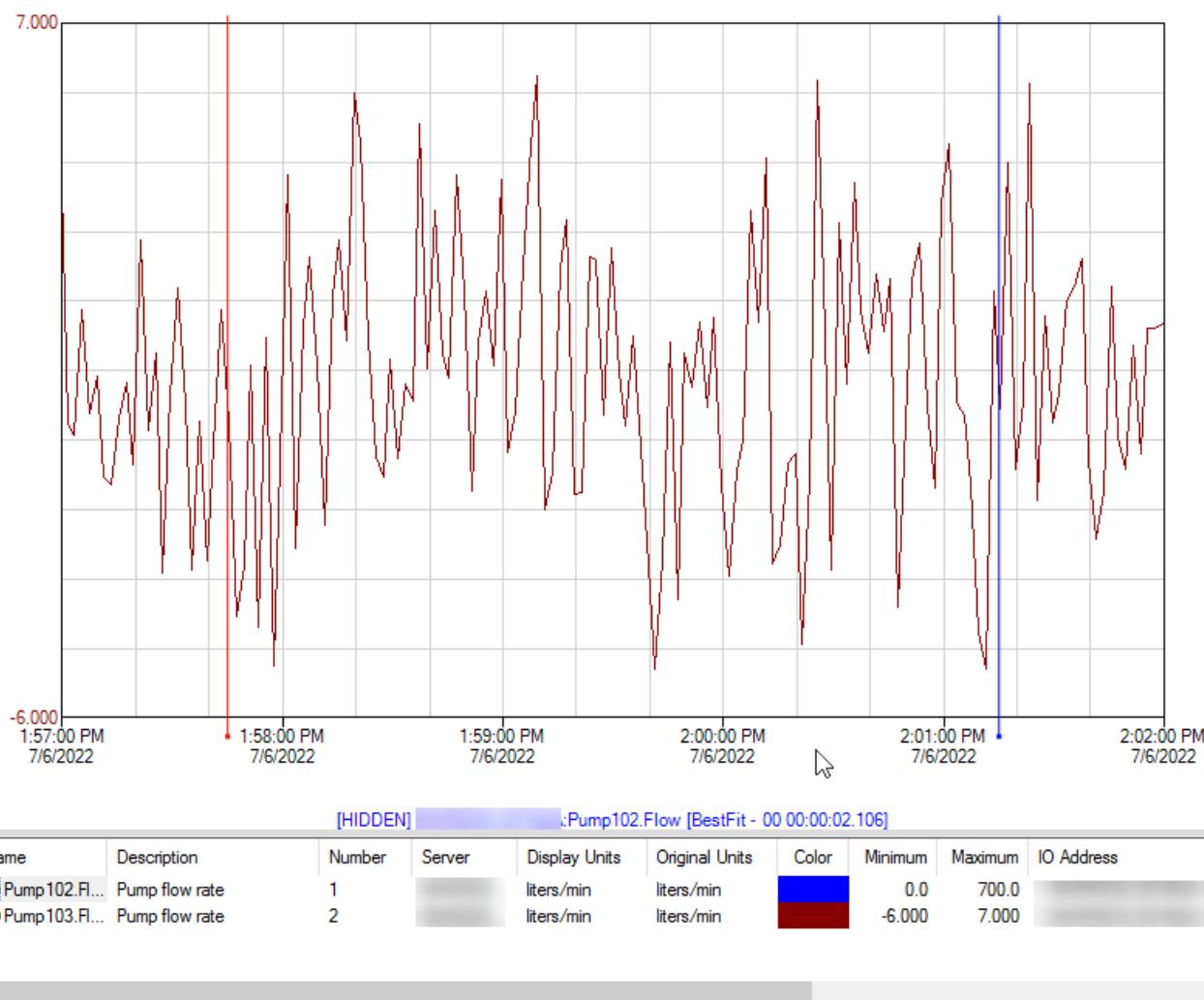
Work with runtime trend charts

This section describes typical tasks that are completed after the HistoricalTendApp has been configured and is actively showing tag values as trends.

View data in the trend chart

This section describes how to use Trend to show historical and live data for trends, as well as how to manipulate the trend display.

Information for individual tags appears in the Tag List below the chart. The name of the currently selected tag, its retrieval mode and resolution or cycle count (if applicable) appear along the bottom of the chart.



Refresh the trend chart

You can refresh the trend chart to see the most recent information.

To refresh the chart

1. Set the **Update to Current Time**  toolbar button depending on whether you want the trend's time period to be updated to the current time when refreshing the chart. If the button is not available, the time period remains the same as before the update. If the button is available, the time period is updated so that it ends with the current time.
2. Do any of the following:
 - Click the **Refresh**  toolbar button.
 - Right-click the chart, and select **Refresh**.

The trend chart is updated with current data from the database.

Delete a tag

You can delete a tag. Deleting a tag removes it from the chart.

To delete a tag from the trend

1. In the Tag List, select the tag to delete.
2. Delete the tag by doing any of the following:
 - On the **Chart** menu, click **Delete Tag**. At the prompt, click **Yes**.
 - Right-click the tag in the Tag List. In the menu that appears, click **Delete**.
 - Press the **DELETE** key .

Configure trend options for a tag

You can configure trend options for one or more tags in the Tag List window. Trend options include the appearance of the trend pen, its target region, and the retrieval mode used to retrieve data for the tag.

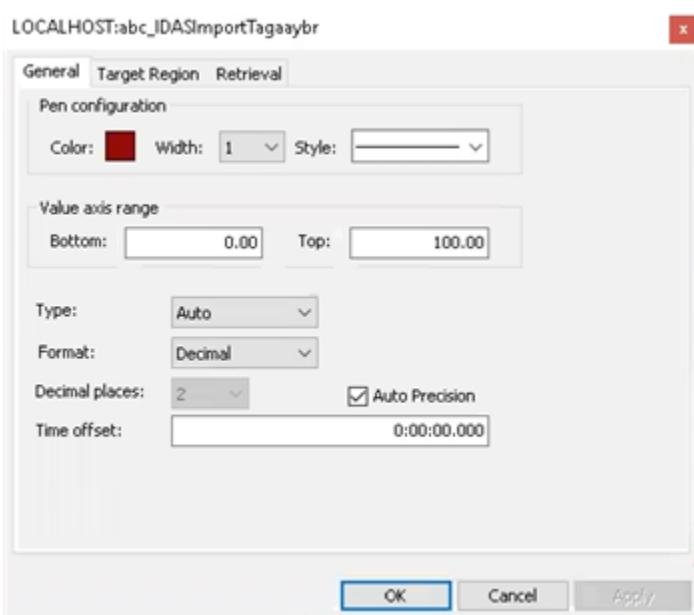
Configure display options

You can configure the pen style, value axis scale, value display options, and time offset for each tag in the chart.

To configure display options for a tag

1. In the Tag List pane, do one of the following:
 - Right-click on the name of the tag and then click **Configure**.
 - Double-click on the name of the tag.

The **<ServerName:Tagname>** dialog box appears with the **General** tab selected.



2. In the **Pen Configuration** area, configure the appearance of the curve for the selected tag.

• Color	The line color of the tag. Click the colored square to select the color from a palette or define a custom color.
• Width	The thickness of the trend curve.
• Style	The style of the trend curve. For example, a solid or dashed line.

3. In the **Value axis range** area, configure the limits for the value axis on the chart.

• Top	The maximum axis value for the tag, in engineering units.
• Bottom	The minimum axis value for the tag, in engineering units.

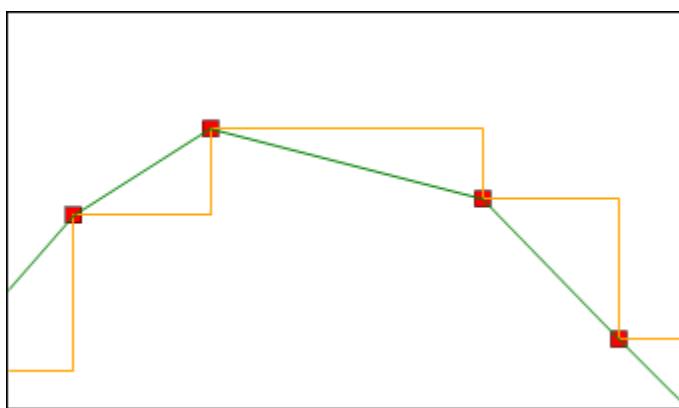
4. In the **Type** list, select the type of trend curve to draw. Options are **Auto**, **Line**, **Step line**, and **Point**.

A line curve is best suited for continuously-changing analog data. A step-line curve is best suited for discrete data and for analog data that is not continuous. By default, the line curve trend is selected for the summary tags.

When you select the **Auto** option, the curve type is determined as follows:

- For tags retrieved from a version 9.0 or a later Historian, the type is based on the tag's effective interpolation setting, which may be specified in the Trend application's retrieval settings or on the Historian. Tags that use stair-step interpolation are trended as a step line, and tags that use linear interpolation are trended as a line.
- For tags retrieved from earlier Historian versions, the curve type is based on the tag type: step line for integer tags, and line for real tags.

The following illustration shows the same data drawn using each type of curve. The line curve is shown in green, the step line curve is shown in orange, and the point curve is shown in red.



- From the **Format** list, select how the values for the tag appear, either in decimal format or scientific format.
- From the **Decimal places** list, select the number of decimal places to show for the data value of the currently selected tag. This applies only to analog tags. Valid values are 0 through 15.
- Select **Auto Precision** to have the number of decimal places set automatically based on the value range.

Note: **Auto Precision** is enabled by default. When auto precision is enabled, the **Decimal places** field is read-only.

8. Configure either the time offset or start time for the data.

For more information on time offsets, see [Absolute or relative times](#).

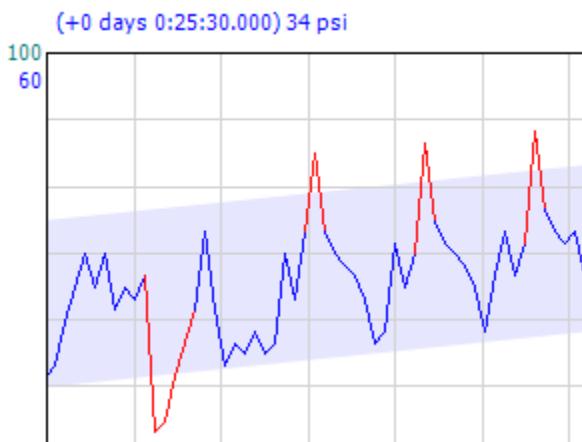
- | | |
|---------------|---|
| • Time offset | Shown only for absolute mode. The amount of time that the trend curve of the currently selected tag will be shifted from the actual time. For information on the offset notation, see Time offset formats . |
| • Start time | Shown only for relative mode. The starting time stamp for the tag data in the chart. |

9. Click **OK**.

Define a target region for a tag

For each analog, discrete, or summary tag in a trend, you can define a "target region." The target region is a highlighted area of the chart into which tag values should fall during normal operation. Values that fall outside these normal limits can be highlighted in a special color, making it easy to detect them.

The following chart shows an example of a target region (the area tinted in blue). The red spikes indicate limit excursions:



In a regular trend, you can only use target regions in relative time mode. A target region is defined by "region items," that is, pairs of high and low values at specific time offsets.

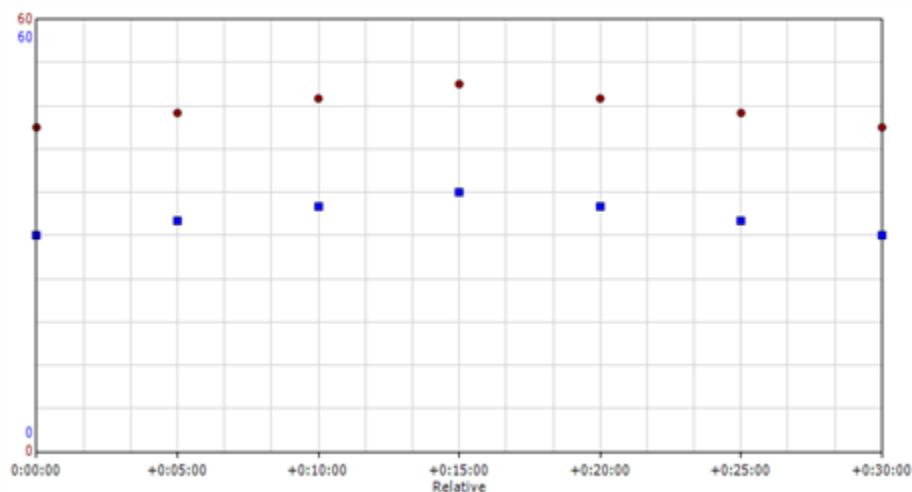
To determine the target region, a boundary is drawn that connects all of the high values, and another boundary that connects all of the low values. The area between these two boundaries constitutes the target region.

For example, assume that you define the following region items:

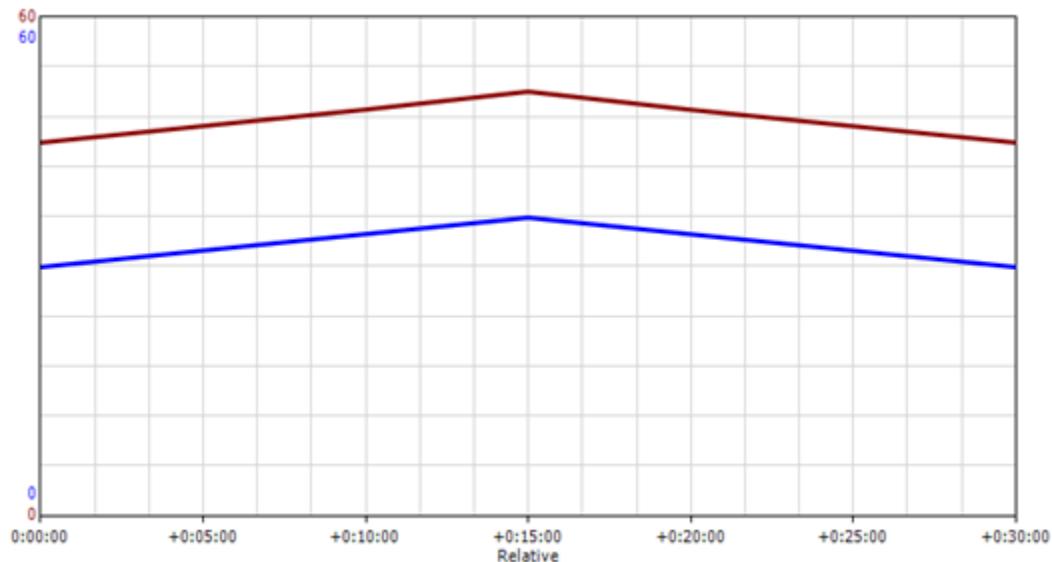
Time	Low	High
0:00:00.00	30	45
+0:05:00.00	32	47
+0:10:00.00	34	49

+0:15:00.00	36	51
+0:20:00.00	34	49
+0:25:00.00	32	47
+0:30:00.00	30	45

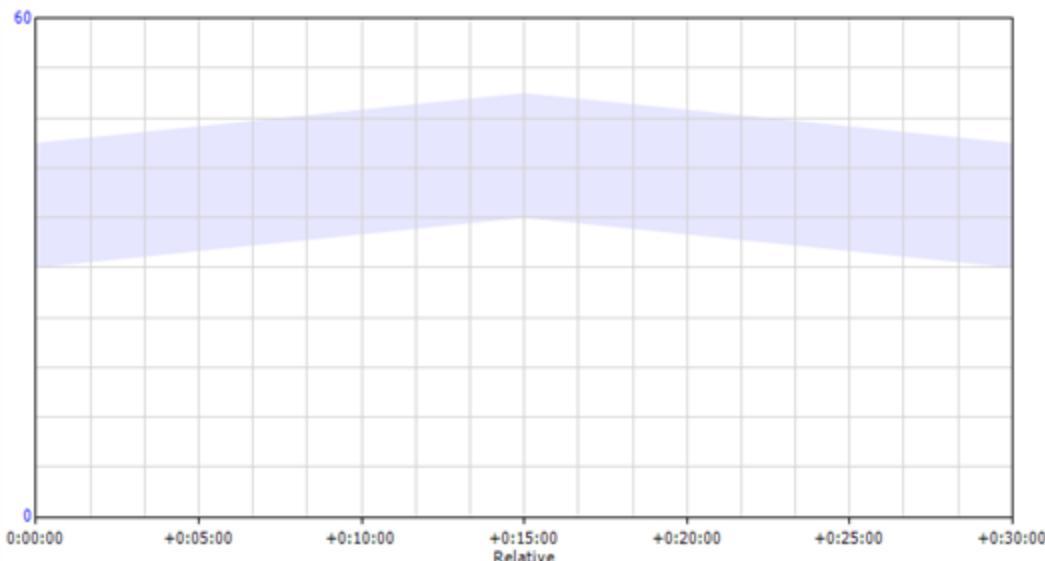
On a trend chart in relative time mode, these points look like the following:



The connecting boundaries look like the following:



The area between these boundaries constitutes the target region:



The target region has the same color as the tag's trend curve. It is only shown when the tag is currently selected in the Tag List. Also, target regions are not visible if you are using stacked traces.

You can define and save target regions separately for each tag. Target regions are saved in the trend file. If you delete a tag from the trend, its target region is deleted as well. To use the same target region for multiple tags or in different trends, either copy and paste it or create a CSV file with the target region data that you can load for each tag. For more information, see the procedures below.

The following section explains how to set up a target region for a tag. To specify whether and how values outside the target region should be highlighted, and to set the target region's opacity, see [Configure target region properties](#).

To configure a target region for a trend tag

1. In the Tag List pane, do one of the following:
 - Right-click on the name of the tag and then click **Configure**.
 - Double-click on the name of the tag.The <ServerName:Tagname> dialog box appears with the **General** tab selected.
2. Click the **Target Regions** tab.
3. Edit region items by using any of the following procedures:
 - Edit region items manually
 - Load region items from a CSV file
 - Paste region items in CSV format from another application
 - Paste region items from another tag
 - Enable or disable a tag's target region
4. Click **OK**. The target region appears in the trend chart, spanning the time period that you defined using the region items' time offsets.

To edit region items manually

- Do any of the following:

- To add an item, click **Add**. A new item appears in the list. Double-click each cell to edit it.
- To delete an item, right-click it and click **Cut**.
- To delete all items, click **Clear All**.
- To move an item up or down in the list, select it and then click the up or down arrows.

To load region items from a CSV file

1. Click **Load file**. A standard **Open** dialog box appears.
2. Select the file you want and click **Open**. The list is populated with the entries from the CSV file.

Note the following format requirements for the CSV file:

- Each row must contain a region item composed of three values: the time offset, the low value, and the high value. The format of the time offset is the same as for time offsets in relative time mode. For more information, see [Time offset formats](#).
- The Trend application tries to determine the CSV delimiter and the decimal and time separators automatically. If one of the values contains a delimiter or separator character, that value must be enclosed in double quotation marks.

To paste region items in CSV format from another application

1. In the other application, open the file that contains the region items in CSV format. Copy the CSV data to the clipboard.
2. In the Trend application, right-click the list of region items, and then click **Paste**. The list is populated with the pasted entries.

The format of the copied data must follow the same conventions as content in CSV files. For more information, see the previous section.

To paste region items from another tag

1. In the Tag List pane, double-click the name of the tag whose region items you want to copy. The **<ServerName:Tagname>** dialog box appears with the **General** tab selected.
2. Click the **Target Region** tab.
3. In the list of region items, select the item(s) you want to copy. To select multiple items, hold down SHIFT and/or CTRL while clicking.
4. Right-click the selected items and click **Copy**.
5. Click **OK** to close the dialog box. Repeat steps 1 and 2 for the tag where you want to paste the region items.
6. Right-click the list of region items and click **Paste**. The list is populated with the pasted entries.

To enable or disable a tag's target region

1. In the Tag List pane, double-click the name of the tag. The **<ServerName:Tagname>** dialog box appears with the **General** tab selected.
2. Click the **Target Region** tab. Select or clear the **Show target region** check box to enable or disable the tag's target region.

Note: Regardless of this setting, the target region for a tag is only shown when that tag is currently selected in the Tag List.

3. Click **OK**.

Configure retrieval options for a tag

You can configure retrieval options separately for each tag in a trend. Tags can either use the retrieval style specified in the trend options, a different predefined retrieval style, or custom retrieval settings.

Most of the retrieval settings that you configure here only apply if you are retrieving data from a Historian with a version of 9.0 or later.

If you are using an earlier Historian version, see [Configure other options](#) and [Work with retrieval styles](#) for details.

To configure retrieval options for a tag

1. In the Tag List pane, do one of the following:

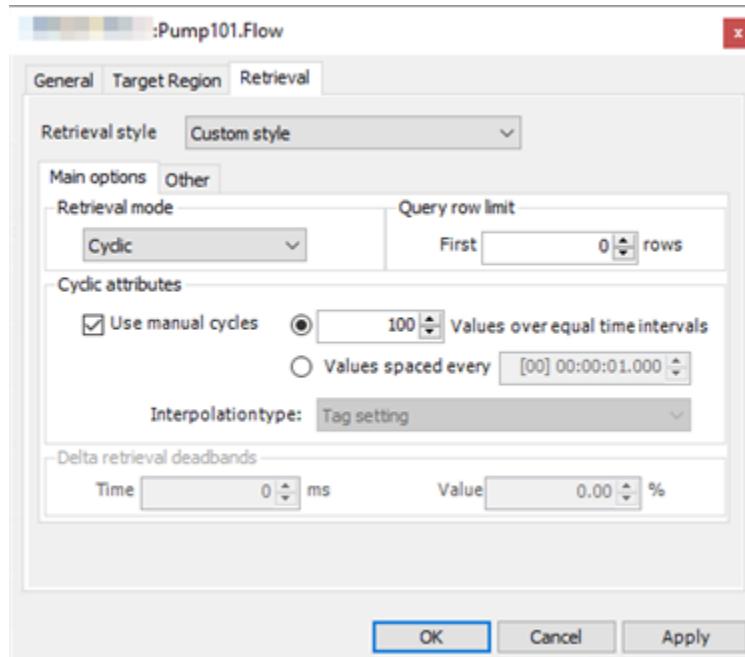
- Right-click on the name of the tag and then click **Configure**.
- Double-click on the name of the tag.

The <ServerName:Tagname> dialog box appears with the **General** tab selected.

2. Click the **Retrieval** tab.

3. Do one of the following:

- To have the tag use the same retrieval settings as specified in the trend options, click **Style selected at option level** in the **Retrieval style** list. This is the default setting when you add a tag to a trend.
- To use a predefined retrieval style, click its name in the **Retrieval style** list. For more information on retrieval styles, see [Work with retrieval styles](#).
- To use custom retrieval settings, click **Custom style** in the **Retrieval style** list.



4. Specify any additional settings required.

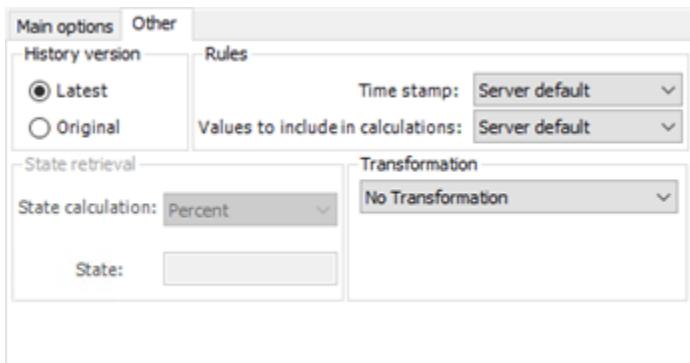
- If you are using custom retrieval settings, select a retrieval mode and specify all the settings that are

relevant to it.

- If you are using one of the predefined styles, you can edit all settings that are not covered by the style definition. For information on which settings are covered by style definitions, see [Work with retrieval styles](#). Because a style definition can contain multiple sets of retrieval settings with different retrieval modes, some of the settings available for editing here may turn out to be irrelevant for the retrieval mode that actually gets used for a given query. However, because there is no way to know in advance which retrieval mode will be used, the settings are still available for editing.

For more information on the various retrieval options, see [Understanding retrieval options](#).

5. Specify any additional settings required on the **Other** tab.



- In the **History version** area, click **Latest** or **Original** to overwrite the values of a stored tag. For more information on History version, see [History Version \(wwVersion\)](#).
- In the **Rules** area, do the following:
 - In the **Timestamp** list, click the required timestamp value. For more information on the Time stamp rule, see [Timestamp Rule \(wwTimestampRule\)](#).
 - In the **Values to include in calculations** list, click the data values that you want to include in the result. You can include the following quality rules:
 - Good and uncertain quality:** To include data values with uncertain quality in calculations.
 - Good quality:** To exclude data values with uncertain quality from calculations.
 - Optimistic quality:** To use the optimistic quality when the data values are missing. This option can be configured at either the tag level or the application level.
 - Server default:** To use the default quality rule specified at the Historian level.For more information on each option, see [Quality Rule \(wwQualityRule\)](#).
- In the **State retrieval** area, do the following:
 - In the **State calculation** list, click the state calculation.
 - In the **State** box, specify the value of the state. For example, you can specify either an open or close state for the SteamValve tag.

Note: The state calculation settings are applicable only to ValueState and RoundTrip retrieval modes.

For more information on State calculation, see [State Calculation \(wwStateCalc\)](#).
- In the **Transformation** list, click the transformations to be applied to the result. You can include the following transformations:
 - No Transformation:** If the query does not specify the filter element or if the value is not overridden for the filter element.
 - Remove outliers:** To remove outliers from a set of analog points.

- c. **Convert analog values to discrete:** To convert value streams for any analog tag to discrete value streams.
- d. **Snap to base value:** To force values in a well-defined range around one or more base values to "snap to" that base value. For more information on Transformation, see [Analog Value Filtering \(wwFilter\)](#).

Scroll through tags in a trend

You can change which tag is currently selected by using toolbar buttons. When using single tag mode, this allows you to "scroll" through tags without having the Tag List visible.

To change the currently selected tag

- From the **Chart** menu, in the **Tag** group, click either **Next** or **Previous** to scroll to the tag that you want.

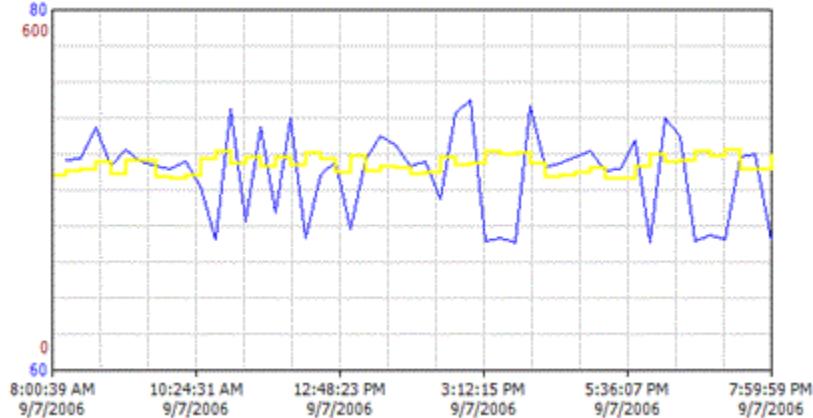
Highlight a tag

You can select and highlight a tag in the chart. To remove the highlighting, follow the same procedure so that no check mark or highlighting appears.

To highlight a tag

1. Select the tag that you want in the Tag List.
2. From the **Chart** menu, in the **Tag** group, select **Highlight**.

The tag is highlighted in the chart.



Show a single tag in the trend

When you initially create a Tag List for a trend, all the tags are included in the display. Setting the trend to single tag mode allows you to exclude all tags but one from being displayed in the trend chart, without removing them from the Tag List.

To display a single tag in the trend

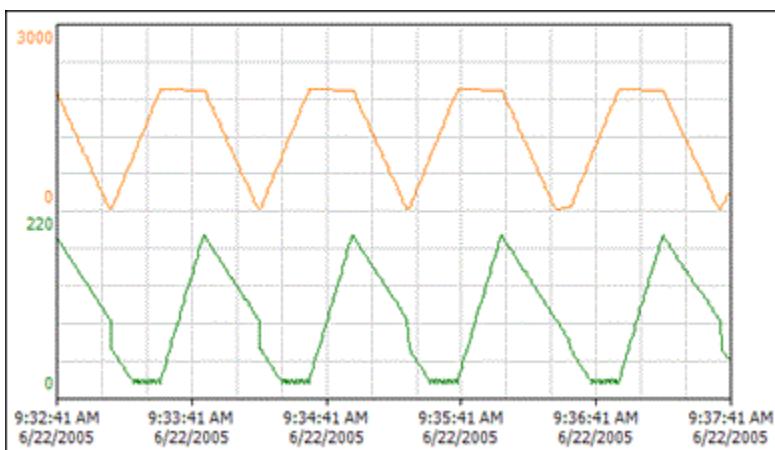
1. From the **Chart** menu, in the **Layout** group, select **Single** so that it is highlighted.
To again view multiple tags in the chart, deselect **Single** so that it is not highlighted.

Stack traces

You can view individual trends, or "traces," for multiple tags in the chart by stacking them in the display.

To stack tags in the trend

1. From the **Chart** menu, in the Layout group, select **Stacked**.



Show live data

A trend can be configured to show live data. Live data is data that is retrieved continuously in real time for a fixed duration that is relative to the current time (for example, the last hour).

When retrieving live data, the Trend application retrieves data incrementally with every update. For example, if you set the update rate to ten seconds, then every ten seconds, the Trend application retrieves data for the last ten seconds and updates the chart with that data. Additionally, it periodically retrieves data for the entire chart time span to refresh the entire chart. You can specify both the update rate and the refresh interval for the entire chart. For more information, see [Configuring General Properties](#).

If the connection to the Historian is lost while retrieving live data, any data that was retrieved up to that point is still shown on the chart until the next full refresh occurs. If the Historian is still unavailable at that time, the old data is cleared from the trend chart.

Notes: When retrieving live data, the time stamp rule for data retrieval is forced to "End." For more information on this setting, see [Timestamp Rule \(wwTimestampRule\)](#).

While the chart is displaying live data, if the trended data includes points outside of the visible plot area, any affected tags are automatically scaled to fit.

To display "live" data

1. Click the **Update to Current Time** toolbar button so that it is highlighted.
2. In the Duration list of the Time toolbar, click a duration or type one manually.
3. Click the **Live Mode** toolbar button.

To display "static" data

- Click the **Stop Live Mode** toolbar button.

Show historical data in replay mode

When you "replay" historical data, the data is continuously plotted on the chart, starting with the start date. By default, the "replay" mode uses real-time speed. For example, if you set the chart to update every second, the the start time advances one second with each update.

You can accelerate or slow down the playback by specifying a "playback speed." For example, if you select a playback speed of 2 x (that is, twice the normal speed) and set the chart to update every second, the start time advances two seconds with each update, as compared to one second for a playback speed of 1 x (normal speed) or half a second for a playback speed of 1/2 x (half the normal speed).

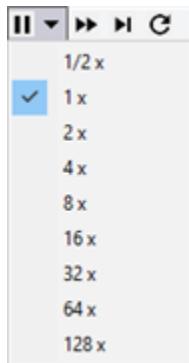
To "replay" historical data

1. Configure a query with a time period whose end time is earlier than the current time.
2. Make sure the **Update to Current Time**  toolbar button is *not* highlighted. If it is highlighted, click it so that it isn't highlighted anymore.
3. Click the **Live Mode**  toolbar button.

The trend curve is dynamically drawn on the chart. The **Replay Mode**  icon appears at the top center of the chart to indicate replay mode.

To "replay" historical data at a slower or faster speed

1. Configure a query with a time period whose end time is earlier than the current time.
2. Make sure the **Update to Current Time**  toolbar button is *not* highlighted. If it is highlighted, click it so that it isn't highlighted anymore.
3. Click the downward arrow next to the Live Mode toolbar button. A list of playback speeds appears.



4. Click the playback speed you want to use.
5. Click the **Live Mode**  toolbar button.

The trend curve is dynamically drawn on the chart at the specified speed. The **Replay Mode**  icon and the playback speed appear at the top center of the chart to indicate replay mode. To change the speed while replay mode is active, repeat steps 1 and 2.

Note: When you replay historical data at an accelerated speed, eventually the time period "catches up" with the current time. When that happens, the speed is automatically reset to normal, and the trend effectively goes into live mode. For more information, see [Showing Live Data](#).

Configure predictive data retrieval

You can define a predictive data retrieval option in the Historian Client. Predictive retrieval means the Historian Client will interpolate future data trend based on the existing data received.

For example, suppose you were using the Historian Client to view a trend for a tank level at your plant. With the data already stored, you can see the tank level values since operations started this morning, but now you would like to know what the value is likely to be in an hour.

Note: Predictive retrieval is supported for the best-fit, full, and delta retrieval modes. If this option is applied to other retrieval modes, it is ignored.

By using the Pan Right (">>") toolbar button, you can advance the trend and, with the Historian Client's new predictive data retrieval feature, you can see a prediction for that future value based on current data. Historian Client interpolates the trend based on the last value was received.

Note: When you save the trend to a CSV file, you can also see predictive data in an Excel Workbook. See [Save trend data to a .CSV file](#) for more information.

Using the Transformation Option at application level

1. From the **Tools** menu, click **Options**.
2. In the **Options** dialog box, click the **Other** tab.
3. From the **Transformation** dropdown, choose **Extrapolate using linear regression**.

Using the Transformation Option at tag level

1. From the **Trend Tag List** grid, double-click a tag.
The **Tag Properties** for that tag display.
2. From the **Retrieval** tab, click the **Other** tab.
3. From the **Transformation** dropdown, choose **Extrapolate using linear regression**.

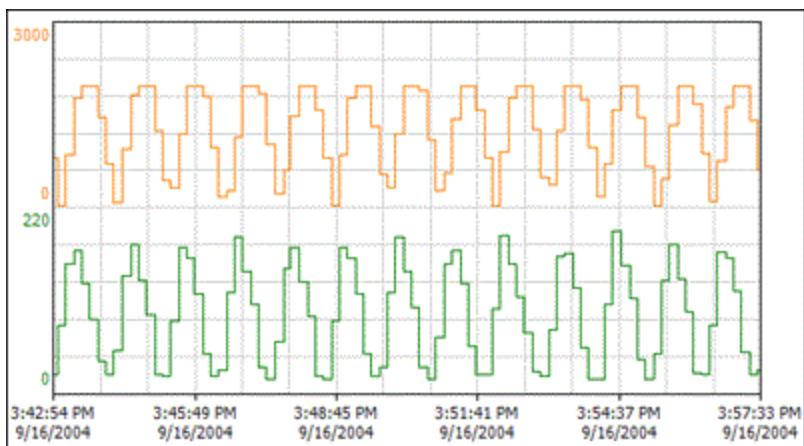
Tag scale

The scale is the minimum and maximum range of values for the tag. Each tag has its own scale, which is usually quite different from other tags in the chart. Scales for tags on the chart are always displayed along the value axis.

Only discrete, analog, and summary tags can be scaled; event and string tags cannot be scaled. For discrete tags, the message associated with the 1 value is used as the maximum scale value, and the message associated with the 0 value is used as the minimum scale value.

In the following chart, two tags are trended in stacked mode. The scale for the ReactLevel tag is from 0 to 3000. The scale for the other tag, ReactTemp, is from 0 to 220.

The minimum and maximum values of each scale appear on the value axis.



The initial scale of a tag is determined by its Min/Max EU settings in the Historian. To adjust tag scales, you have two options:

- Edit the value axis range individually for each tag. For more information, see [Configure trend options for a tag](#).
- Use the scaling commands to adjust the scale of single tag or all tags. For more information, see [Scale tags up or down](#) and subsequent sections.

You can also change the way in which scale values appear on the value axis. The following sections describe the available options.

Show no scales on the value axis

You can configure the chart to show no chart label, X and Y axes scales and cursor information. This makes the entire chart area margins smaller and you get more area to plot and view the trend chart.

To show no scales on the value axis

- From the **View** menu, in the **Scale** group, select **None**.

The following chart is configured to show no chart label, X and Y axes scales and cursor information. Stacked mode is applied.

Show a single scale on the value axis

You can configure the chart to show single value scale along the value axis.

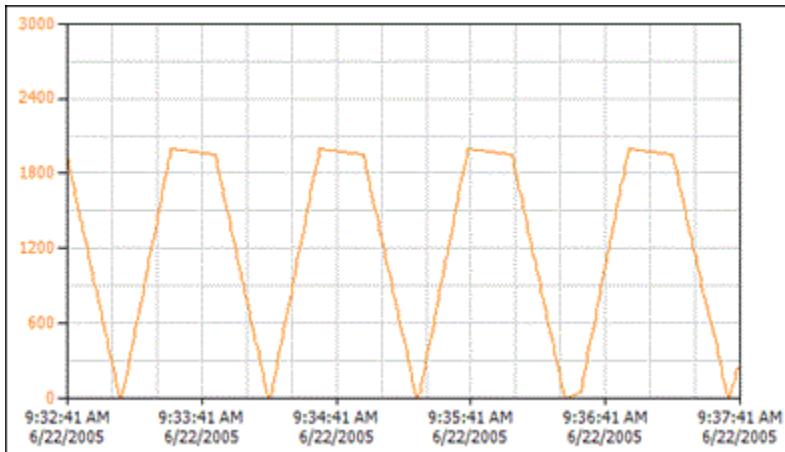
If the chart includes multiple tags, the scale of the currently selected tag in the chart is shown along the value axis. The scale label color matches the pen color of the selected tag. As you scroll through tags in the chart, the value axis always shows the scale of the selected tag. The labels along the value axis are shown even if the current tag is hidden.

To show single scale

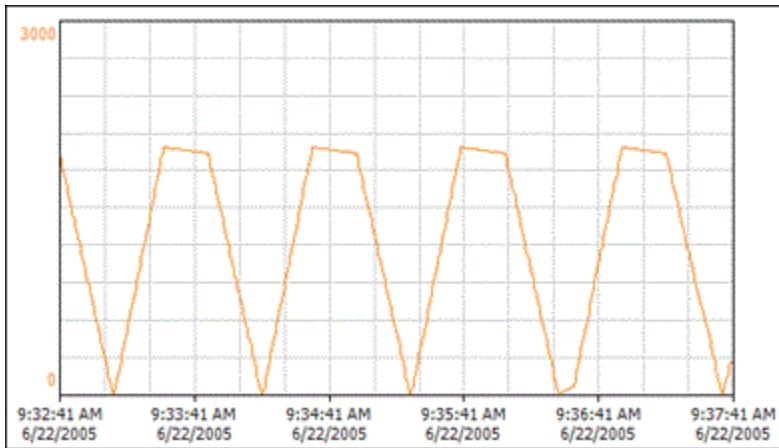
- From the **View** menu, in the **Scale** group, select **Single**.

How a chart looks when single scale is applied depends on the number of tags in the chart and whether the chart is in stacked mode.

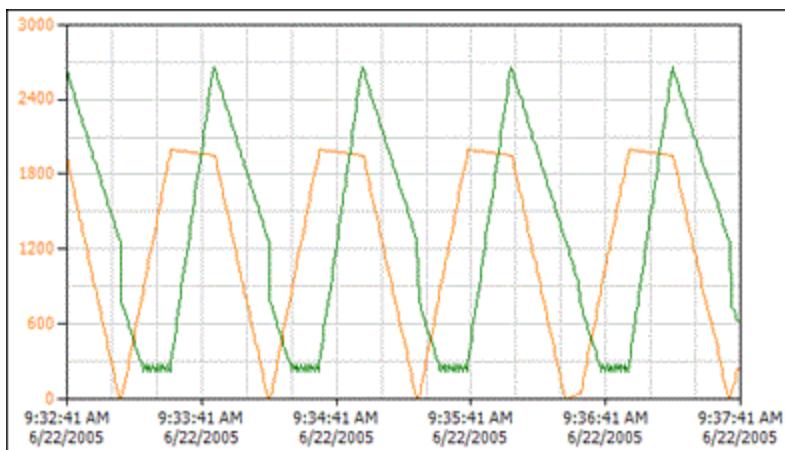
The following chart includes single tag and is configured for single scale. Stacked mode is not applied.



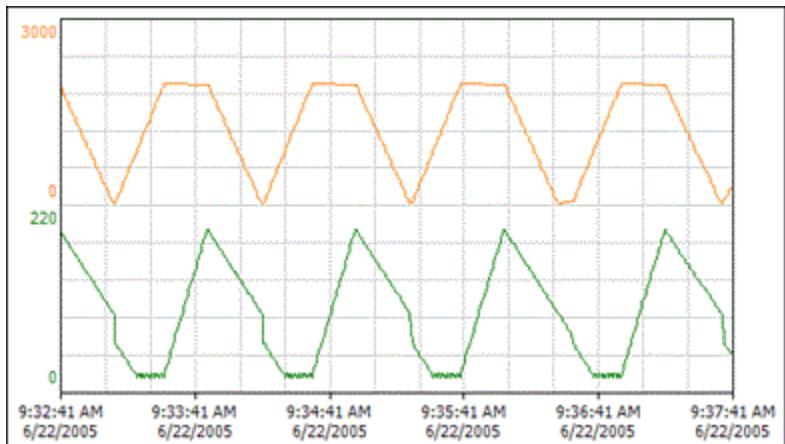
The following chart includes single tag and is configured for single scale. Stacked mode is applied. Only the minimum and maximum values are shown.



The following chart includes multiple tags and is configured for single scale. Stacked mode is not applied.



The following chart includes multiple tags and is configured for single scale. Stacked mode is applied. The minimum and maximum values for each tag are shown for the corresponding trend curve.



Show multiple scales on the value axis

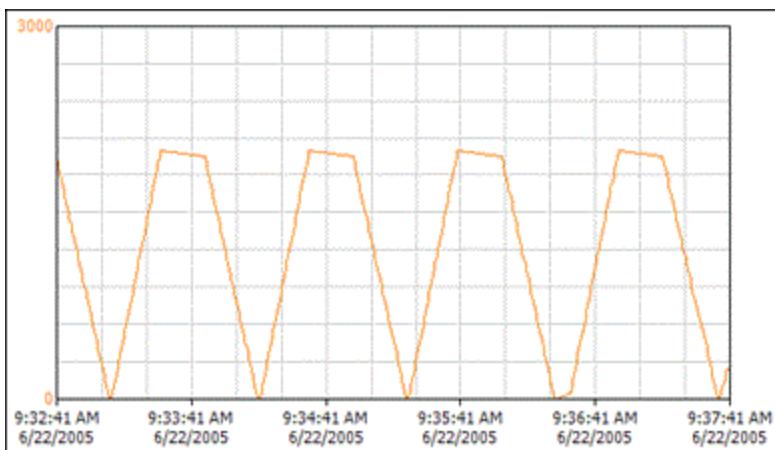
You can configure the chart to show multiple value scales. For multiple scales, only the minimum and maximum values are shown on the value axis. The scale label colors match the pen colors of the corresponding tags. The values of hidden tags are not shown.

To show multiple scales

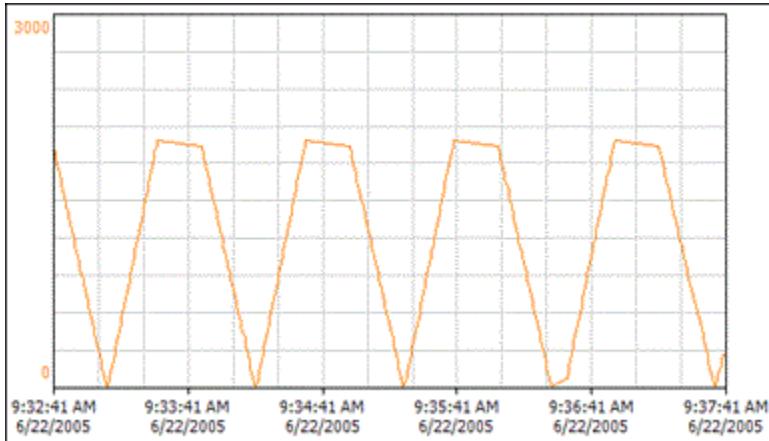
- From the **View** menu, in the **Scale** group, select **Multiple**.

How a chart looks when multiple scales are applied depends on the number of tags in the chart and whether the chart is in stacked mode. When stacked mode is applied, there is no difference between using single scale or multiple scales.

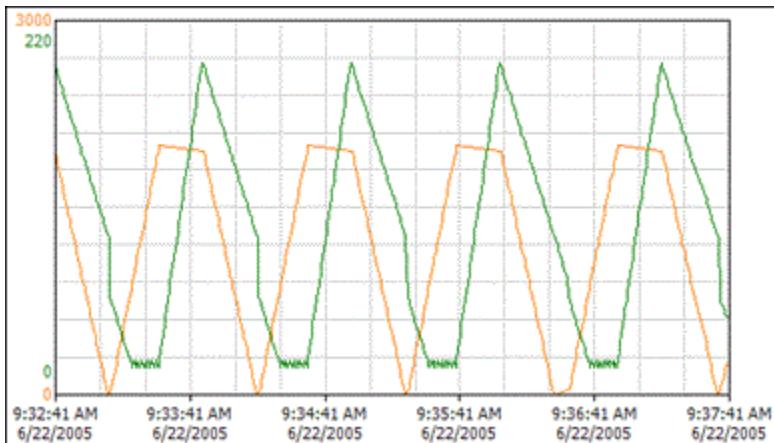
The following chart includes single tag and is configured to use multiple scales. Stacked mode is not applied.



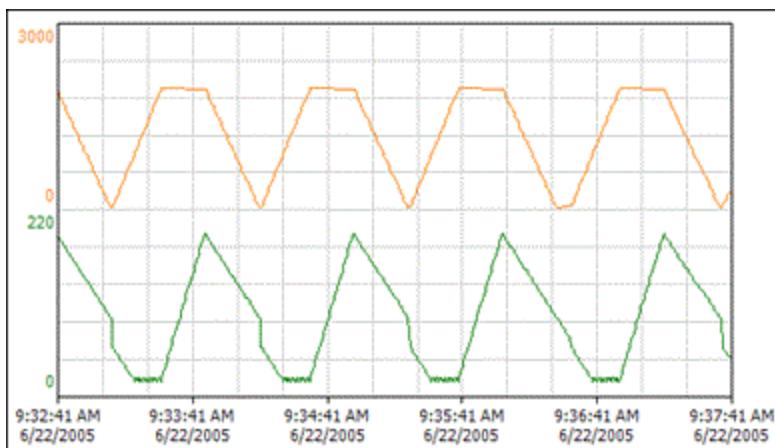
The following chart includes single tag and is configured to use multiple scales. Stacked mode is applied.



The following chart includes multiple tags and is configured to use multiple scales. Stacked mode is not applied. The top and bottom labels show the scale for the first tag in the Tag List that is included in the chart. For the second tag in the Tag List, the scale labels are shown as second from the top and second from the bottom. As you add tags to the chart, the addition of scale labels continues to progress inward toward the middle of the chart. If there is not enough space on the chart to show all of the scale labels, then the innermost values are not shown.



The following chart includes multiple tags and is configured to use multiple scales. Stacked mode is applied.



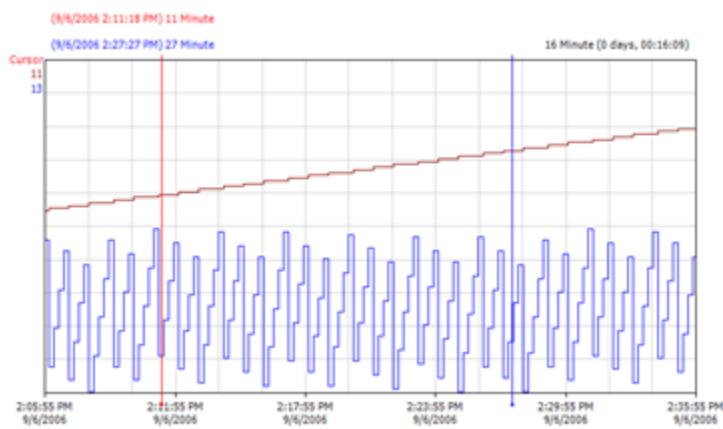
Show cursor values on the value axis

You can configure the chart so that the value axis shows the value of each tag at the position of the first X axis cursor. The axis label colors match the pen colors of the corresponding tags. The values of hidden tags are not shown.

To show cursor values on the value axis

- From the **View** menu, in the **Data** group, select **Cursor Values**.

The following chart is configured to show cursor values on the value axis. Stacked mode is applied.



Scale tags up or down

You can scale a single tag or all of the tags in a trend up or down. If you scale a tag down, the range of values increases by one third. For example, if the scale is 10 to 70, it becomes 0 to 80. If you scale a tag up, the range of values decreases by one fourth. For example, if the scale is 0 to 80, it becomes 10 to 70.

To scale up a single tag

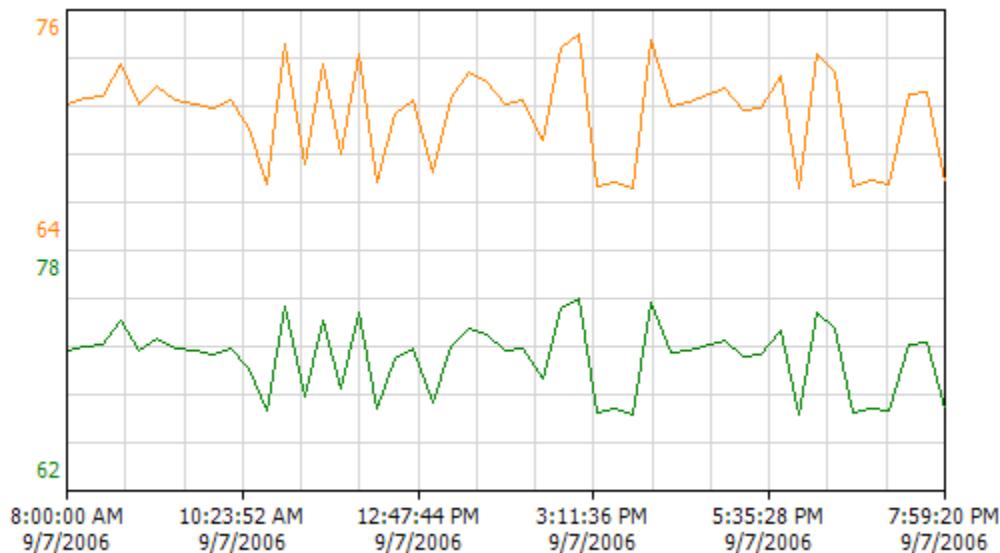
- Select a tag from the tag list.

- From the **Chart** menu, in the **Single Tag Scale** group, click **Scale Up**.

To scale all tags up

- From the **Chart** menu, in the **All Tags Scale** group, click **Scale Up**.

The following example shows a single tag scaled up:



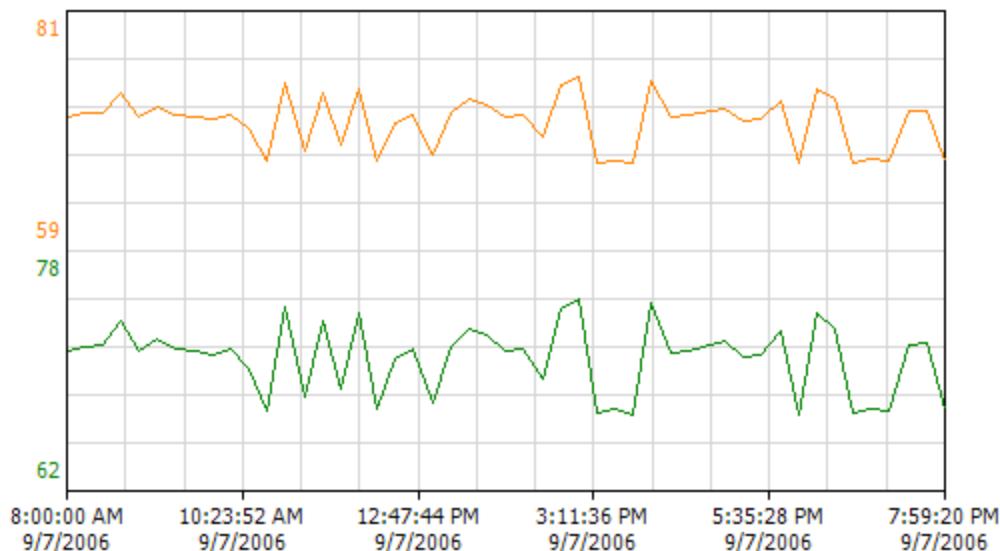
To scale down a single tag

- Select a tag from the tag list.
- From the **Chart** menu, in the **Single Tag Scale** group, click **Scale Down**.

To scale all tags down

- From the **Chart** menu, in the **All Tags Scale** group, click **Scale Down**.

The following example shows a single tag scaled down:



Scale tags automatically

When a tag is automatically scaled, the value axis range is automatically adjusted to reflect the actual data currently being displayed for the trend. For example, if the default value axis range is 0 to 3000, and the data ranged from 1827 to 2059, the scale might be automatically adjusted to a range of 1800 to 2100.

The adjusted scale does not exactly match the actual minimum and maximum data values for the chart. The calculation rounds the values so as to make the chart easier to read. Also, a percentage allocation is added to the final values so that the adjusted scale fits within the boundaries of the trend chart. Therefore, the adjusted scale is a round number slightly above the actual data values.

You can automatically scale a single tag or all of the tags in a trend. For information on resetting scales back to the original default, see [Return tags to their original scale](#).

Tag scaling is performed automatically for a trend while it is retrieving live data. If the trended data includes points outside of the scale region, any affected tags are scaled automatically.

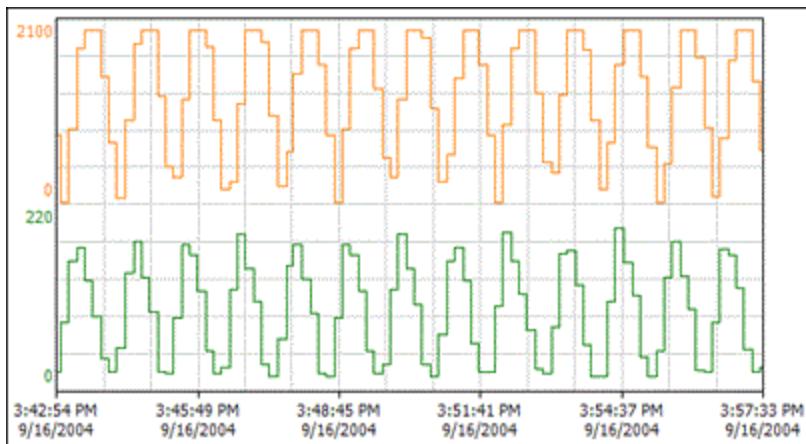
To automatically scale a single tag

1. Select a tag from the tag list.
2. From the **Chart** menu, in the Single Tag Scale group, click **Auto**.

To automatically scale all tags

- From the **Chart** menu, in the All Tags Scale group, click **Auto**.

The following example shows a tag automatically scaled:



Return tags to their original scale

You can return the value axis scale for a single tag or all of the tags in a trend to the original scale.

To return a single tag to original scale

1. Select a tag from the tag list.
2. From the **Chart** menu, in the **Single Tag Scale** group, click **Original**.

To return all tags to original scale

- From the **Chart** menu, in the **All Tags Scale** group, click **Original**.

Move tags up or down in the chart

You can move a single tag or all of the tags in a trend up or down in the trend chart. The scale is adjusted to reflect the move.

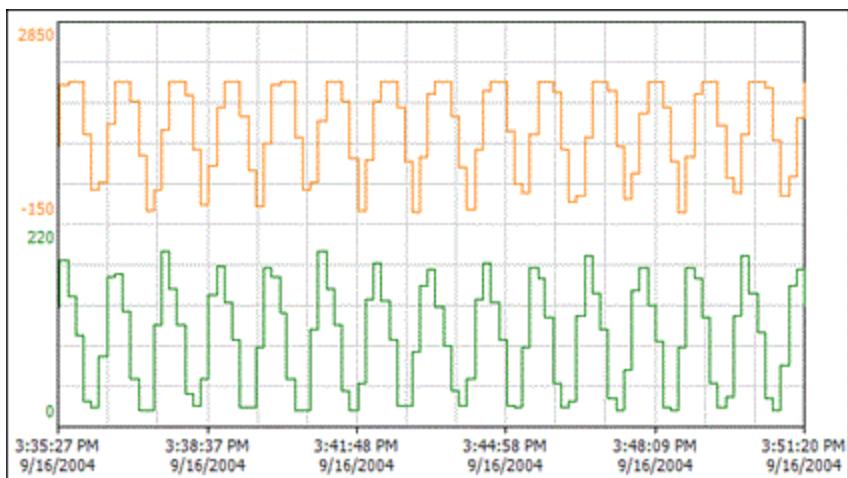
To move a single tag up

1. Select a tag from the tag list.
2. From the **Chart** menu, in the Single Tag Scale group, click **Move Up**.

To move all tags up

- From the **Chart** menu, in the **All Tags Scale** group, click **Move Up**.

The following example shows a single tag moved up in the trend chart:

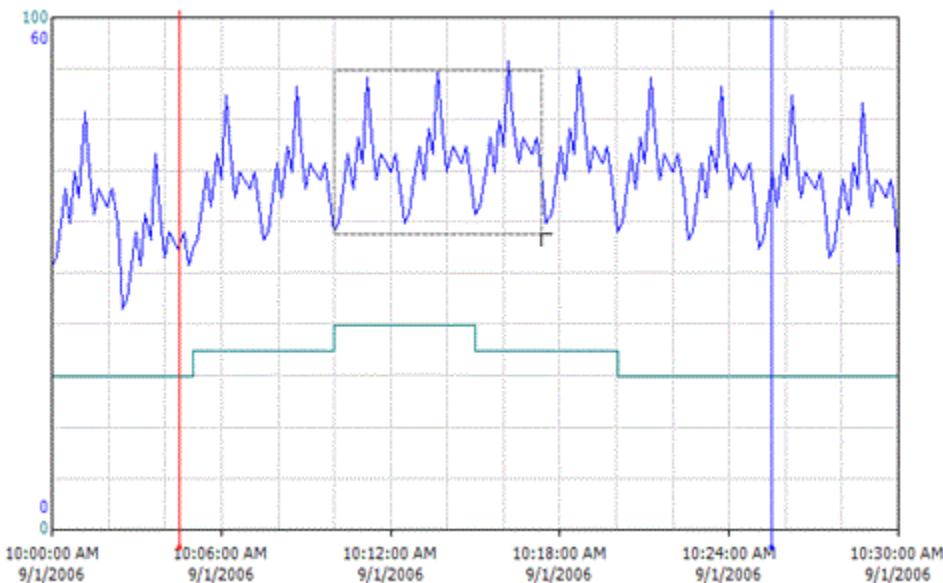


Rubber band scaling

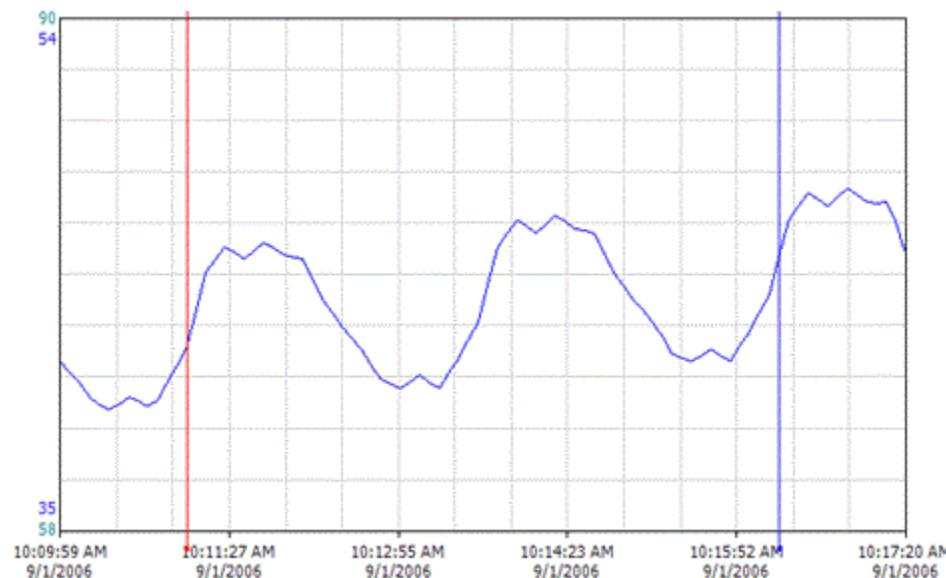
Rubber band scaling allows you to "lasso" an area of the trend chart with the mouse cursor to automatically adjust the time and value axis scales based on the area that you lassoed. If you are using stacked traces, rubber band scaling is limited to the time axis.

To use rubber band scaling

1. From the **Chart** menu, in the **Zoom** group, click **Selection** so that it is highlighted.
2. Unless you are using stacked traces, rubber band scaling affects both the time and the value axes. Time axis scaling always applies to all tags in the chart. Value axis scaling can apply to all tags or the currently selected tag only. If you want value axis scaling to apply to all tags, from the **Chart** menu, in the **Zoom** group, click **Select All** so that it is highlighted.
If you are using stacked traces, rubber band scaling affects the time axis only, and this setting has no effect.
3. Drag a box around the area you want to use for the new scale.



The trend chart is automatically redrawn using the new zooming values that you selected with the mouse.



Pan in the trend chart

By default, the chart is panned to the left or right by the time span percentage set for the chart. This time span applies to both left and right panning and is a percentage of existing data coverage on the chart. The default time span is 75 percent; that is, when you pan right or left, the chart pans by three quarters of the total time span. For example, if the time axis for the chart spans one hour, the chart is panned to the left or right by 45 minutes.

For more information on configuring the panning scale, see [Configure axis properties](#).

To pan left

- Click the **Pan Left**  toolbar button.

To pan right

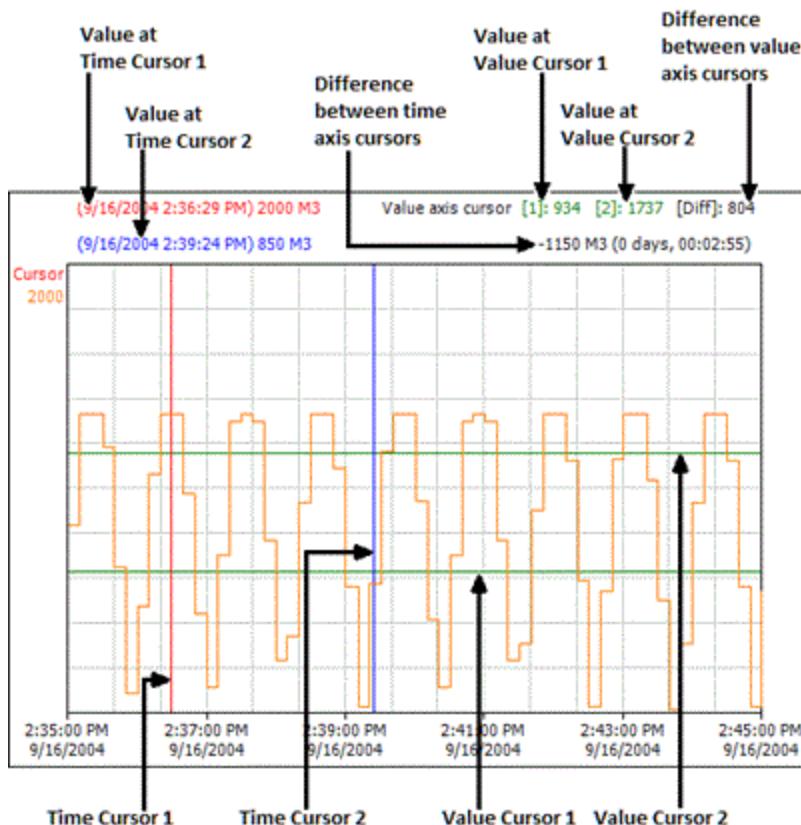
- Click the **Pan Right**  toolbar button.

If the time scale is set into the future, then white space appears.

During a pan, the time picker changes to reflect the currently displayed selection.

Axis cursors

Each trend chart has two value cursors and two time cursors. These cursors pinpoint tag values in the chart. The values shown for the axis cursors are updated continuously as the cursors are moved or as the trend curve moves in live mode.



You can show or hide the value and time cursors, as well as the values at the cursors. You can also show or hide the value cursor difference.

To configure the line and font colors for the cursors and cursor value displays, see [Configure axis properties](#).

Move a cursor

To move a cursor

1. Select the cursor with your mouse.
2. Drag the cursor to the spot on the chart.

As you move the cursor in the trend chart, the value for the tag where the cursor and the tag curve meet appears.

Show or hide the axis cursors

To show the time axis cursors

- From the **View** menu, in the **Data** group, select **Time Axis**.

To show the value axis cursors

- From the **View** menu, in the **Data** group, select **Value Axis**.

To hide the cursors, follow the same procedure so that no highlighting appears.

Show or hide the cursor difference

To show the cursor difference

- From the **View** menu, in the **Data** group, select **Cursor Difference**.

To hide the cursor difference, follow the same procedure so that no highlighting appears.

Zoom in or out

When you use zooming in the trend chart, the zoom value depends on whether you are using time axis cursors. If you are not using time axis cursors, zooming is based on the total value for the time axis. The trend chart is zoomed in or out based on the current percentage set for the zooming scale. All zooms are positioned along the middle line of the trend chart.

If you are using time axis cursors, zooming in sets the time period to the period between the cursors. Zooming out works as described above.

For information on setting the zooming percentage, see [Configure axis properties](#).

To zoom in

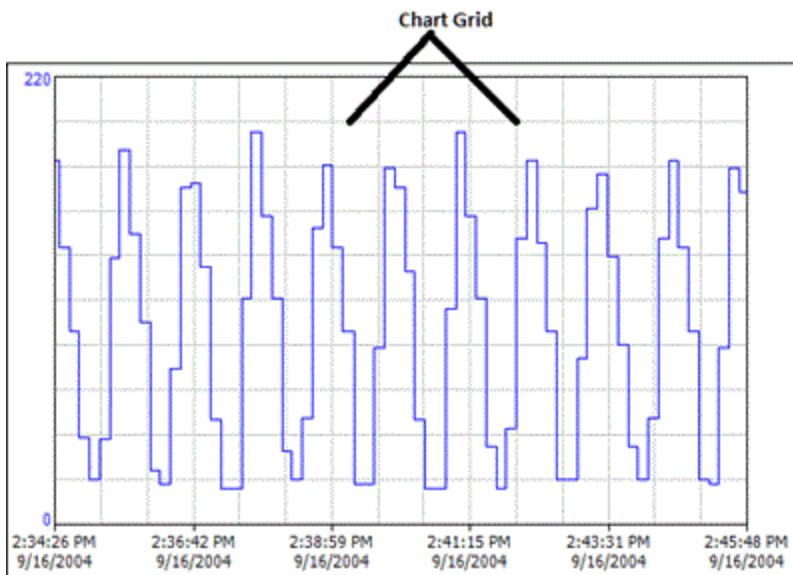
- From the **Chart** menu, in the **Zoom** group, click **In**.

To zoom out

- From the **Chart** menu, in the **Zoom** group, click **Out**.

Show or hide the chart grid

You can show/hide the horizontal and vertical chart lines.



To show the horizontal lines

- From the **View** menu, in the **Grid** group, select **Horizontal**.

To show the vertical lines

- From the **View** menu, in the **Grid** group, select **Vertical**.

To hide the lines, deselect both **Horizontal** and **Vertical**.

View tag definition information

Use the Tag List to view information for the tags that you have added to the trend chart.

By default, the Minimum Raw, Maximum Raw, Precision, Format, and Date Created columns are not visible in the Trend application's Tag List. To show the columns, drag the right column separator of the Time Offset column heading to the right. Repeat this step until all columns are visible.

To view tag information

- In the Tag List pane, scroll to the name of the tag for which you want to view definition information.

Tag Name	Description	Number	Server	Display Units	Original Units	Color	Minimum	Maximum	IO Address	Time Offset	Source Tag	Source Server	Value at X1	Value at X2
<input checked="" type="checkbox"/> @SysTimeHour	System Time : Hours	1		Hour	Hour		0.000	23.000		0:00:00 000			15.000	15.000
<input checked="" type="checkbox"/> @SysTimeMin	System Time : Minutes	2		Minute	Minute		0.00	59.00		0:00:00 000			15.00	18.00
<input checked="" type="checkbox"/> @SysTimeSec	System Time : Seconds	3		Second	Second		0.00	59.00		0:00:00 000			17.00	47.00

The grid shows the following information:

Name	Description
Tag Name	The name of the tag within the Historian. If the data values are coming from the application server, the attribute reference is shown as the tag name. For application server attributes, you can also choose to

	show the hierarchical name along with the attribute reference.
Description	The description for the tag.
Number	The serial number of the tag.
Server	The name of the server that contains the tag.
Display Units	The displayed unit of measure of the tag value. Examples: seconds, psi, and lbs. By default, this is the same as the "Original Units", but when licensed and configured, you can select different display units.
Original Units	The configured unit of measure of the tag value. Examples: seconds, psi, and lbs.
Color	The line color of the tag.
Minimum	The minimum value of the raw acquired value.
Maximum	The maximum value of the raw acquired value.
IO Address	The complete I/O address for the tag, including I/O Server name, application, topic, and item name.
Time Offset	The amount of time that the trend curve of the currently selected tag will be shifted from the actual time.
Source Tag	The name of the source tag that provides the source data for the replicated tag.
Source Server	The name of the server that contains the source tag.
Value at X1	The value that is displayed at Timer axis Cursor 1.
Value at X2	The value that is displayed at Timer axis Cursor 2.

- Select or clear the check box to show or hide the tag in the trend chart. This allows you to hide a tag without removing it from the list of tags.

View the hierarchical name in a trend

A hierarchical name represents the fully qualified name of a contained object within Application Server. For more information, see Integration with Wonderware Application Server.

To view hierarchical names in a trend

- Do one of the following:
 - On the **View** menu, from the **Canvas** group select **Hierarchical Name**.
 - Right-click in the Tag Picker and click **Use hierarchical name**.

The Trend application shows the hierarchical name instead of the tag name. For example, the Tag List, **Data Log** dialog box, and the Trend chart area show hierarchical names.

View trend data in a table format

You can view a table of all data points used in a chart. This data log can be either in a "narrow" or "wide" format. In both cases, the log only shows values for tags that aren't hidden.

View the data log in a narrow format

To view the data log

1. On the **View** menu, click **Logs**, and then select **Narrow**. The **Data log** dialog box appears.

Time	Tag Name	Server	Value	Quality
5/20/2022 7:53:01.022 AM	Pump101.Flow	[redacted]	604.3	Good
5/20/2022 7:53:02.928 AM	Pump101.Flow	[redacted]	599.9	Good
5/20/2022 7:53:04.928 AM	Pump101.Flow	[redacted]	597.6	Good
5/20/2022 7:53:06.928 AM	Pump101.Flow	[redacted]	599.3	Good
5/20/2022 7:53:08.928 AM	Pump101.Flow	[redacted]	600.9	Good
5/20/2022 7:53:10.928 AM	Pump101.Flow	[redacted]	600.8	Good
5/20/2022 7:53:12.928 AM	Pump101.Flow	[redacted]	600.3	Good
5/20/2022 7:53:14.928 AM	Pump101.Flow	[redacted]	595.8	Good

Data appears for the following columns:

• Time	The time stamp for the returned value. For delta retrieval, this is the time at which the value was acquired by the Historian. For cyclic retrieval, this is the specific time requested or calculated (using a SQL function).
• Tag Name	The name of the tag within the Historian server. The attribute reference is displayed as a tag name for data values from application server. For application server attributes, you can also choose to show the hierarchical name along with the attribute reference.
• Server	The server from which data is being retrieved.
• Value	The value of the tag at the time stamp.

- Quality

The basic data quality indicator associated with the data value.

2. To include only the data that is between the time axis cursors on the chart, on the **Options** menu, click **Data From Between Cursors**.
3. To include all of the data on the chart, on the **Options** menu, click **Data From Between Graph Start/End**.
4. To show actual values for discrete tags (for example, 1 or 0), on the **Options** menu, click **Show Actual Values For Discretes**. When retrieving data for discrete tags in ValueState mode, you must select this option to see correct time-in-state information.
5. To show messages for discrete tags (for example, ON or OFF), on the **Options** menu, click **Show Messages For Discretes**.
6. You can copy and paste data to the Windows clipboard by right-clicking in the data and selecting the appropriate option from the menu that appears.
7. To save the data as a .csv file, on the **File** menu, click **Save As**.
8. To set up a printout of the data, on the **File** menu, click **Page Setup**. Setting up the page works like in any other Windows application.
9. To preview a printout of the data, on the **File** menu, click **Print Preview**. Using the preview window works like in any other Windows application.
10. To print the data, on the **File** menu, click **Print**. Specifying printing options works like in any other Windows application.
11. To exit the dialog box, on the **File** menu, click **Exit**. Or, click the **Close** button.

View the data log in a wide format

To view the data log

1. On the **View** menu, click **Logs**, and then select **Wide**. The **Data log** dialog box appears.

Time	Pump101.Flow	Pump101.Pressure
5/20/2022 8:38:43.393 AM	598.2	1024.0
5/20/2022 8:38:44.928 AM	597.6	1029.2
5/20/2022 8:38:46.928 AM	600.1	1023.5
5/20/2022 8:38:48.928 AM	601.0	1029.3
5/20/2022 8:38:50.928 AM	600.7	1025.2
5/20/2022 8:38:52.928 AM	597.3	1024.1
5/20/2022 8:38:54.928 AM	596.0	1028.5
5/20/2022 8:38:56.928 AM	597.5	1034.4

Data appears for the following columns:

• Time	The time stamp for the returned value. For delta retrieval, this is the time at which the value was acquired by the Historian. For cyclic retrieval, this is the specific time requested or calculated (using a SQL function).
• <Tag Name>	The name of the tag within the Historian server. If the data values are coming from Application Server, the attribute reference is shown as the tag name. For Application Server attributes, you can also choose to show the hierarchical name along with the attribute reference.

2. To include only the data that is between the time axis cursors on the chart, on the **Options** menu, click **Data From Between Cursors**.
3. To include all of the data on the chart, on the **Options** menu, click **Data From Between Graph Start/End**.
4. To show actual values for discrete tags (for example, 1 or 0), on the **Options** menu, click **Show Actual Values For Discretes**. When retrieving data for discrete tags in ValueState mode, you must select this option to see correct time-in-state information.
5. To show messages for discrete tags (for example, ON or OFF), on the **Options** menu, click **Show Messages For Discretes**.
6. You can copy and paste data to the Windows clipboard by right-clicking in the data and selecting the appropriate option from the menu that appears.
7. To save the data as a .csv file, on the **File** menu, click **Save As**.
8. To set up a printout of the data, on the **File** menu, click **Page Setup**. Setting up the page works like in any other Windows application.
9. To preview a printout of the data, on the **File** menu, click **Print Preview**. Using the preview window works like in any other Windows application.
10. To print the data, on the **File** menu, click **Print**. Specifying printing options works like in any other Windows application.
11. To exit the dialog box, on the **File** menu, click **Exit**.

View statistics

You can view statistics for the data that is retrieved and displayed for a trend. Display statistics include range, average, minimum, maximum, sum, standard deviation, and query properties. Examples of query properties are query time range, start time, end time, and number of rows returned. To display data statistics

1. On the **View** menu, click **Statistics**. The **Statistics** dialog box appears.

Type	Tag Name	Server	Samples	Minimum	Time at Minimum
Analog	Pump101.Flow		151	593.6	5/20/2022 8:39:58.928
Analog	Pump101.Pressure		151	1021.7	5/20/2022 8:39:04.928

2 rows

Statistics appear in columns as follows.

• Type	The type of tag.
• Tag Name	The name of the tag within the Historian server. If the data values are coming from Application Server, the attribute reference is shown as the tag name. For Application Server attributes, you can also choose to show the hierarchical name along with the attribute reference.
• Server	The server that contains the tag.
• Samples	The number of samples in the trend.
• Minimum	Minimum value for the data that is plotted in the chart.
• Time at Minimum	The time stamp of the minimum value.
• Maximum	Maximum value for the data that is plotted in the chart.
• Time at Maximum	The time stamp of the maximum value.
• Average	Average value for the data.
• Standard Deviation	Standard deviation for the data.
• Range	Value range for the data.
• Timespan	The total amount of time that is spanned by the data.
• From	The starting date for the data.
• To	The ending date of the data.

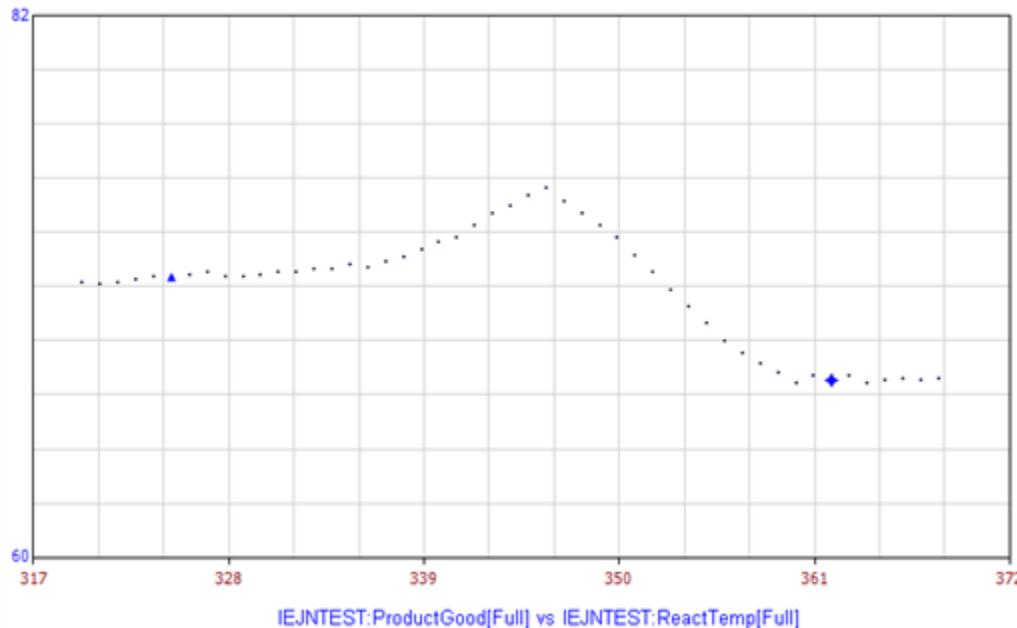
2. To include only the data that is between the time axis cursors on the chart, on the **Options** menu, click **Data From Between Cursors**.
3. To include all of the data on the chart, on the **Options** menu, click **Data From Between Graph Start/End**.

4. To show actual values for discrete tags (for example, 1 or 0), on the **Options** menu, click **Show Actual Values For Discretes**.
5. To show messages for discrete tags (for example, ON or OFF), on the **Options** menu, click **Show Messages For Discretes**.
6. You can copy and paste data to the Windows clipboard by right-clicking in the log and selecting the appropriate option from the menu that appears.
7. To save the data as a .csv file, on the **File** menu, click **Save As**.
8. To set up a printout of the data, on the **File** menu, click **Page Setup**. Setting up the page works like in any other Windows application.
9. To preview a printout of the data, on the **File** menu, click **Print Preview**. Using the preview window works like in any other Windows application.
10. To print the data, on the **File** menu, click **Print**. Specifying printing options works like in any other Windows application.
11. To exit the dialog box, on the **File** menu, click **Exit**.

Work with scatter plots

In addition to regular trends, you can display data in XY scatter plots. While a regular trend shows the variation of a tag's value over time, a scatter plot shows the variation of a tag's value over the variation of another tag's value. This allows you to see correlations between the two tags.

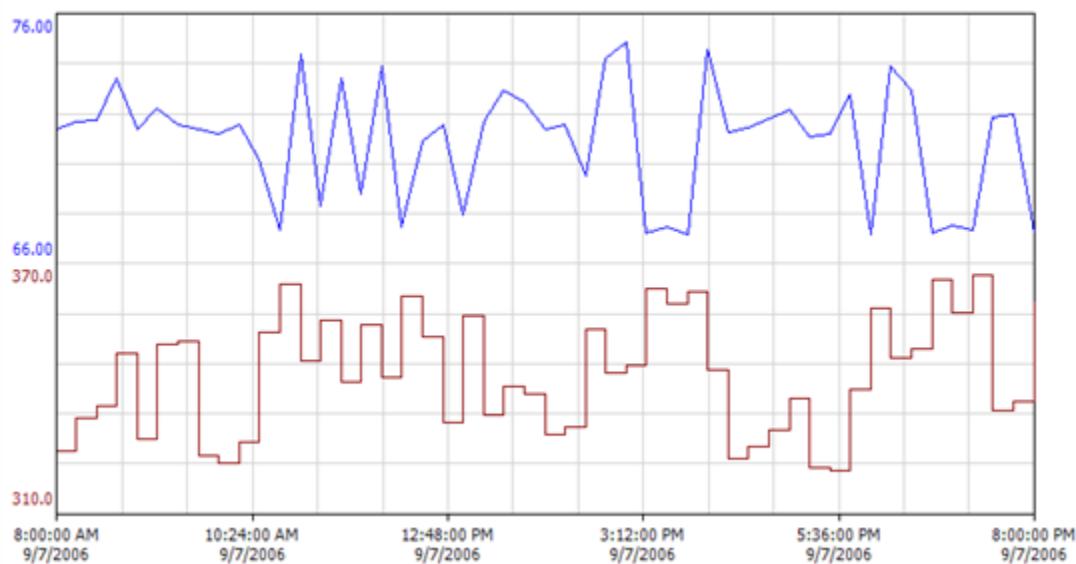
For example, you could show how product yield varies depending on the reactor temperature in a manufacturing process, and use this information to determine the optimum temperature:



In this example, the X axis represents the reactor temperature as historized by the ReactTemp tag (the "X axis tag"). The Y axis represents the product yield as historized by the ProductGood tag (the "Y axis tag"). For each available data sample of either tag during the chosen time period, a corresponding value for the other tag is matched or interpolated and plotted on the chart.

For more information, see [How Are Value Pairs Matched?](#).

Plotted over time, the two tags look like this:



Compared to this type of display, the scatter plot shows the correlation much more clearly.

The following sections show you how to configure a scatter plot and manipulate the display. Many of these features work in a regular trend. Therefore, these sections mainly explain the specific differences when working with scatter plots.

View data in a scatter plot

Scatter plots show value pairs. As in a geometric coordinate system, every data point in scatter plot must have an X value that determines its horizontal position as well as a Y value that determines the vertical position. On a regular trend, there is no such thing as an "X value" that corresponds to the Y value of a tag; instead, the horizontal position of a tag's value on the chart is determined by the value's time stamp.

On a scatter plot, however, both the X and the Y values must be supplied as tag data. Therefore, you must assign an X axis tag to every tag that you want to view in the scatter plot.

Tags without a corresponding X axis tag are visible in the Tag List, but not in the chart. For more information on how X and Y values are matched, see [How are value pairs matched?](#)

To configure a scatter plot

1. Click the **New Chart**  toolbar button. A new trend chart appears.
2. On the **Chart** menu, in the **Layout** group, select **Scatter**.
The chart switches into scatter plot mode.
3. Add tags to the chart by double-clicking them in the Tag Picker or dragging them onto the Tag List. For more information on the Tag Picker, see [Tag Picker](#).

You must add all tags that you want to use as X or Y axis tags. Note the following:

- One tag can serve as the X axis tag for multiple other tags.
- If you want to view the same tag against different X axis tags, add it to the Tag List multiple times.
- While you can add string or event tags, they serve no purpose in a scatter plot. Therefore, these tags are automatically marked as hidden.

4. Specify a time period for the chart using the time toolbar.
5. Assign an X axis tag to every tag that you want to view in the scatter plot:
 - a. Double-click the tagname in the Tag List. The <ServerName:TagName> dialog box appears with the **General** tab selected.
 - b. In the **X axis** Tag List, click the name of the tag that you want to use as the X axis tag for this tag. To remove an existing X axis tag association, click the blank entry instead.
 - c. Configure other tag options as required. For more information, see [Configuring Trend Options for a Tag](#).
 - d. Click **OK**.

Data for the X/Y tag pairs is retrieved for the specified time period and plotted in the chart. The oldest value pair appears as a triangle-shaped point, and the latest value pair as a diamond-shaped point.

Tags that do not have an X axis tag assigned to them are shown in italics at the end of the Tag List.

To quickly assign an X axis tag to a tag

If you do not need to configure any other tag settings, you can use the following steps to quickly assign an X axis tag to a tag that you want to display in a scatter plot (the Y axis tag).

1. Add the Y axis tag to the chart.
2. With the Y axis tag selected in the Tag List, drag the X axis tag from the Tag Picker onto the X axis of the chart.

Alternatively, use these steps:

1. Add the X and Y axis tags to the chart.
2. In the Tag List, click the X axis tag's name and drag it onto the **X Axis Tag** column of the Y axis tag.

Scale tags in a scatter plot

Scaling tags in a scatter plot works much like scaling tags in a regular trend. For more information, see [Scaling Tags](#). Note the following:

- Scaling a tag affects the display of all tags that use it as their X axis tag. The display of all other tags remains unchanged. If you want to plot multiple tags against the same X axis tag, but with different X axis scales, you must add the X axis tag to the chart multiple times and assign each tag a different instance of the X axis tag. You can then scale the various instances of the X axis tag individually.
- Rubber band scaling always affects all tags in the chart. It applies to both X axis and Y axis tags. It is not possible to use rubber band scaling for single tag. Rubber band scaling does not affect the chart's time period.
- The scale of the X axis changes as you select different tags in the Tag List. It reflects the scale of the X axis tag associated with the selected tag, or the scale of the tag itself if it does not have an X axis tag. The "multiple scales" option has no effect on the X axis.
- It is not possible to use cursor values as axis labels in a scatter plot.
- Stacking traces is not possible in a scatter plot.

Configuring axes in a scatter plot

Configuring the axes of a scatter plot works much like configuring the axes of a regular trend. For more information, see [Configure axis properties](#). Note the following:

- On the **Axes** tab of the **Trend Properties** dialog box, the **X time axis** area only applies to regular trends. For scatter plots, use the **X value axis** area instead.

How are value pairs matched?

To plot a data point, the scatter plot must determine which Y value belongs to a given value of the X axis tag and vice versa. This is easy if there are data samples available with the same time stamp for both the X axis tag and the Y axis tag. If there is a sample available for one tag (Tag 1) at time T, but not for the other tag (Tag 2), the missing value is calculated based on the following rules:

- If Tag 2 uses a curve type of "Point" or "Step Line", then the data point uses the latest sample of Tag 2 that is earlier than T.
- If Tag 2 uses a curve type of "Line", then the data point uses the result of a linear interpolation between the two samples of Tag 2 that surround T.

For example, assume you have the following samples available for two tags. Tag 1 uses a trace type of "Step Line." Tag 2 uses a trace type of "Line." A dash indicates that there is no sample at that point in time.

Time	Value of Tag 1	Value of Tag 2
t ₁	x _{t1}	y _{t1}
t ₂	—	y _{t2}
t ₃	x _{t3}	—
t ₄	x _{t4}	y _{t4}

According to the rules above, the missing value of Tag 1 at t₂ is assumed to be x_{t1}. The missing value of Tag 2 at t₃ is calculated using a linear interpolation between y_{t2} and y_{t4}.

If either tag has a NULL sample at a given point in time, the data point is considered "empty," which may result in a gap in the curve.

Quality calculation for data points

In the chart display, data points of uncertain, bad, or unknown quality are visually highlighted with special indicators. The overall quality of a data point in a scatter plot depends on the quality of the two tag values of which it is composed. The following table shows the overall quality that results from each possible combination of tag qualities, assuming that both tag values aren't NULL. The top row contains the quality of the first tag, the left column contains the quality of the other tag.

	Unknown	Uncertain	Bad
Good	Unknown	Uncertain	Bad
Unknown	Unknown	Uncertain	Bad
Uncertain	Uncertain	Uncertain	Bad
Bad	Bad	Bad	Bad

For example, if one tag has good quality and the other tag has bad quality, the data point is highlighted with the indicator for bad quality.

Pan and zoom in a scatter plot

Panning and zooming affect the time period used in a chart. For a scatter plot, this means that panning or zooming moves, enlarges or reduces the time period for which data is retrieved. This may result in more or fewer data points being available for display. Depending on the nature of the data, this may or may not change the visual appearance of the chart—unlike in a regular chart, where panning or zooming inevitably changes the display.

To reflect this, the panning commands in the **Chart** menu are called **Pan Earlier** and **Pan Later** in a scatter plot, as opposed to **Pan Left** and **Pan Right** in a regular trend. However, they still work the same way. The zooming options are identical. For more information, see [Pan in the trend chart](#) and [Zooming](#).

Define a target region for a scatter plot

You can configure a target region for each tag displayed in a scatter plot as you configure in a regular trend. For an overview of what a target region does, see [Defining a Target Region for a Tag](#).

Configuring a target region for a scatter plot tag is very similar to configuring one for a regular trend. The main difference is that the target region isn't defined by high and low boundaries at certain points in time, but by a series of X/Y value pairs. The target region is determined by connecting the X/Y points in the order they are given. For some examples, see [Examples for Target Regions in Scatter Plots](#).

To configure a target region for a scatter plot tag

- Follow the procedure given under [To configure a target region for a trend tag](#). The only difference is that when importing a CSV file or pasting clipboard data, each row must contain a region item that is composed of two items instead of three. The first item is the X value, the second item is the Y value.

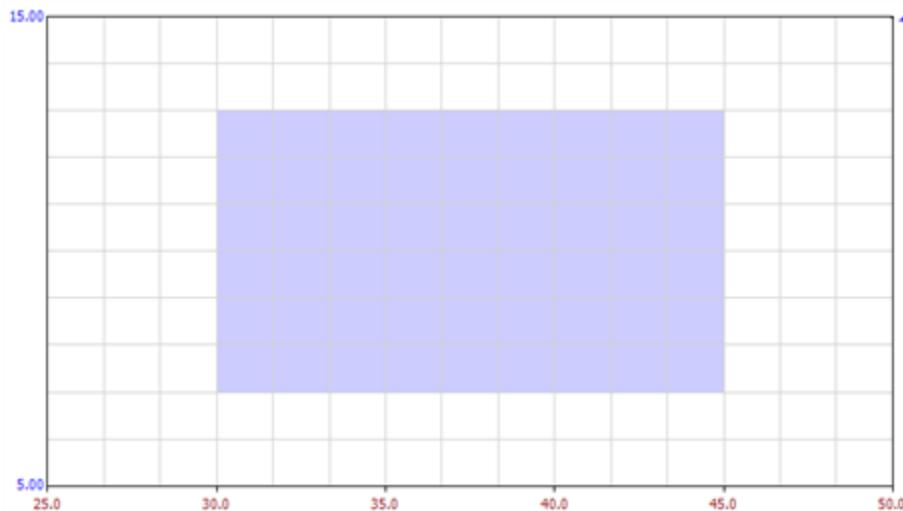
Examples of target regions in scatter plots

When defining a scatter plot target region, listing the same X/Y points in different order can result in very different target regions. For example, assume that you define the following X/Y points:

X	Y
30	7

30		13
45		13
45		7

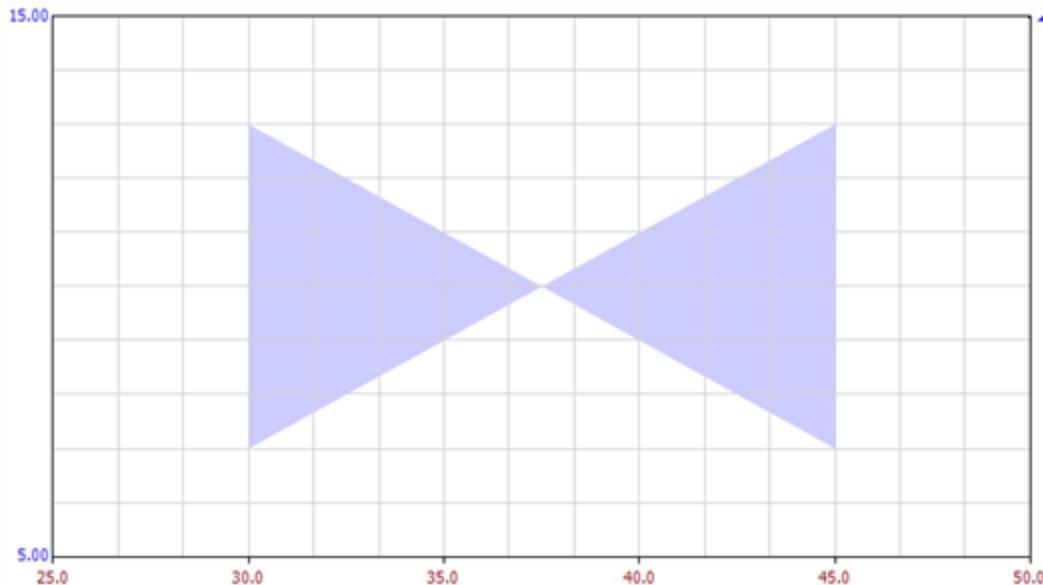
The resulting target region looks like this:



Because the points are connected in the order they are defined, reordering the points results in a different target region. Assume that you reorder the same points like this:

X	Y
30	7
30	13
45	7
45	13

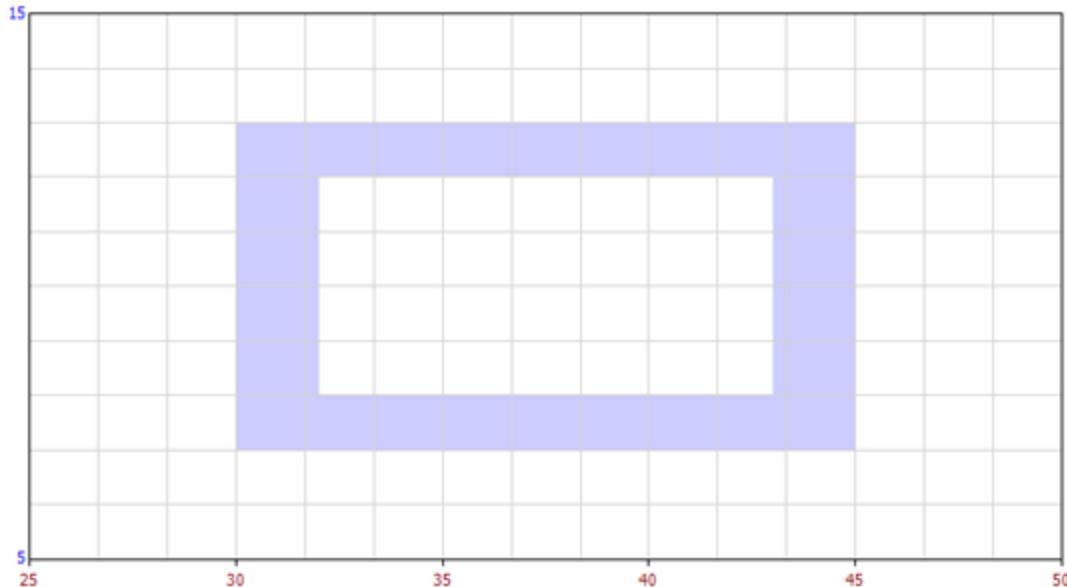
The resulting target region looks like this:



You can also create target regions with a "hole" in the middle. For example, use the following points:

X	Y
30	7
30	13
45	13
45	7
30	7
32	8
32	12
43	12
43	8
32	8

The resulting target region looks like this:



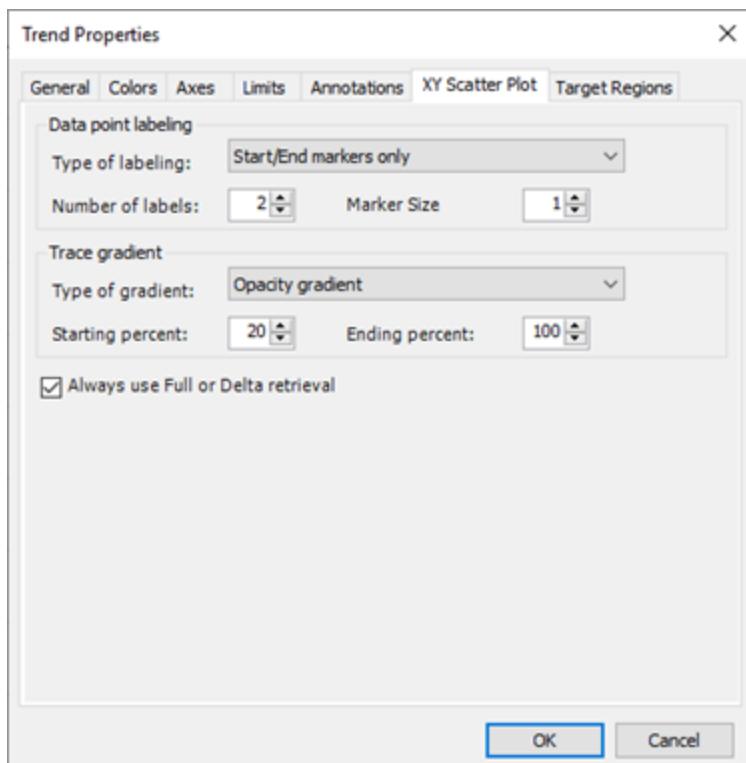
Configure scatter plot properties

You can set up a scatter plot to show time labels on the data points and use an opacity gradient for line traces to indicate the sequence of data points in time.

Also, you can override the tags' data retrieval settings so that full or delta mode retrieval is always used in scatter plots.

To configure scatter plot properties:

1. On the **Chart** menu, click **Properties**. The **Trend Properties** dialog box appears.
2. Click the **XY Scatter Plot** tab.



3. In the **Data point labeling** area, configure the following options:

• Type of labeling	Selected Option	Result
	Start/End markers only	Start/end markers are displayed, with no time labels on the data points
	Start/End markers with time labels on current tag	Start/end markers are displayed, along with time labels on the data points
	Start marker only	Start markers are displayed, but end markers are not displayed
	End marker only	End markers are displayed, but start markers are not displayed
	No marker and time labels	No markers or time labels are displayed
• Number of labels	The number of time labels that appear on the chart. The valid range is from 2 to 15, with a default value of 6. The labels are spaced evenly over the time period between the earliest and the latest data point in the chart. (Therefore, they may not be spaced evenly in distance.)	
• Marker Size	The size to display the start/end markers in the chart. The valid range increases in size from 1 to 5, with a default value of 1.	

4. In the **Trace gradient** area, configure the following options:

• Type of gradient	Select Opacity gradient if you want the opacity of the line trace in a scatter plot to change based on the time stamp of the data points that it connects. For example, the trace could be fully opaque at the most recent data point and almost transparent at the oldest data point. Select None if you want the trace to have the same opacity at all data points.
• Starting percent	The opacity of the trace at the oldest data point. 0 means transparent, 100 means fully opaque.
• Ending percent	The opacity of the trace at the most recent data point.

5. Select the **Always use Full or Delta retrieval** check box if you always want to use Full or Delta mode retrieval for all tags in a scatter plot regardless of the retrieval settings that are configured at the application or tag level. Full retrieval is used when retrieving data from a Historian with a version of 9.0 or higher. Delta retrieval is used for earlier server versions. We recommend to keep this option enabled unless the nature of your data makes full retrieval impractical.
6. Click **OK**.

Other scatter plot considerations

Also note the following when working with scatter plots:

- **Retrieval:** If a tag is neither associated with an X axis tag nor acting as an X axis tag itself, no data is retrieved for it. Therefore, the data logs do not show any data for such tags.
- **Cursors:** You can work with cursors similar to regular trend. However, the cursor commands in the **View** menu are called **X Value Axis Cursors** and **Y Value Axis Cursors** instead of **Time Axis Cursors** and **Value Axis Cursors**.
- **Curve type:** If a tag has a curve type of "Line" or "Step Line," its data points are connected in chronological order. Depending on the nature of the data, changing the curve type to "Point" may result in a clearer display.
- **Switching between chart types:** Trend options that aren't applicable to a scatter plot are disabled and/or ignored when you switch the chart type from regular trend to scatter plot. However, their values are generally retained so that they are still available if you switch the chart type back to regular trend.

Annotations

You can use Trend to make an annotation for a tag at any point in time. An annotation is a user comment about a tag. For example, you might want to save a comment about a very high spike in a trend. You can create an annotation for the value of the tag at the spike. All annotations are saved to the database and can be retrieved again at a later time.

You can create a private annotation (that only you can see) or a public annotation (which is viewable by all trend users). Private annotations are only available to the users who created them and have suitable access.

For each annotation, an annotation mark (solid circle) is added to the trend. This annotation mark can be viewed

on the trend if the trend properties are set to allow it.

When you make an annotation, the following information is stored in the Annotation table in the Runtime database of the Historian:

- Name of the tag for which the annotation is associated.
- The date/time of the annotation. The time of the annotation is based on the position of where it was created on the time axis.
- The value of the tag at the time of the annotation.
- The annotation text.

Note: You cannot use the Annotations functionality when connected to a Managed Historian.

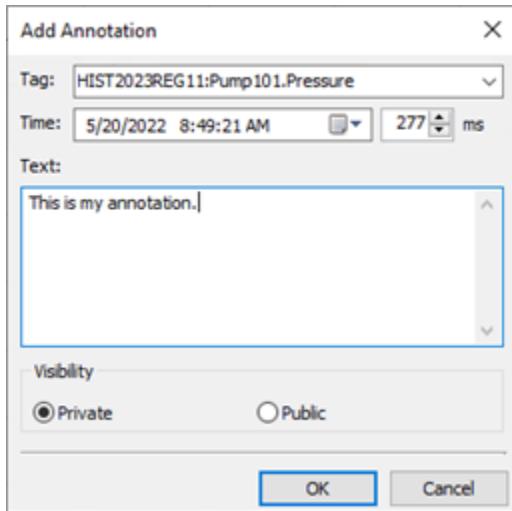
Add an annotation

Annotations are inserted in the chart at the location where the mouse button is clicked and are associated with the selected tag's value where the mouse button is clicked.

To add an annotation

1. Select the tag for which you want to add an annotation. You can do this by selecting the tag in the Tag List pane.
2. Do one of the following:
 - On the **Chart** menu, click **Add Annotation**.
 - Right-click in the chart. In the shortcut menu that appears, click **Add Annotation**.

The **Add Annotation** dialog box appears.



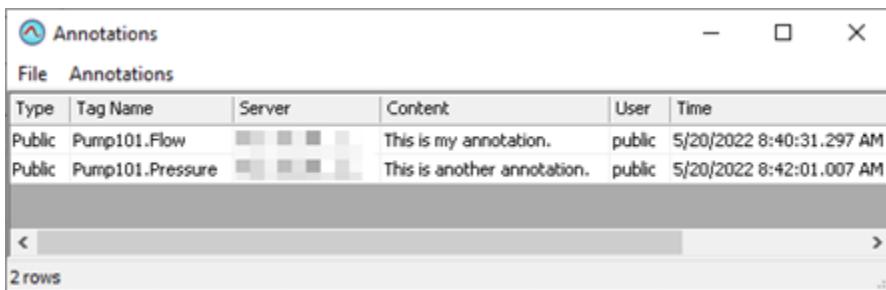
3. In the **Tag** list, click the name of the tag for which you want to add the annotation.
4. In the **Time** list, click the time stamp of the tag value for which you want to add the annotation.
5. In the **Text** window, type in your comment.
6. In the **Visibility** area, specify if you want the annotation to be visible to others. Click **Private** to have annotations only visible to you. Click **Public** to have annotations visible to anyone who is able to log on to the database.
7. Click **OK**.

An annotation marker (dot) appears on the chart at the point on the chart where the annotation was made. If you hover with the mouse on the marker, the details for the annotation will appear on the chart.

View the annotation list

To view a list of annotations

1. On the **View** menu, click **Annotations**. The **Annotations** dialog box appears.



Type	Tag Name	Server	Content	User	Time
Public	Pump101.Flow	[server icons]	This is my annotation.	public	5/20/2022 8:40:31.297 AM
Public	Pump101.Pressure	[server icons]	This is another annotation.	public	5/20/2022 8:42:01.007 AM

The table shows the following information:

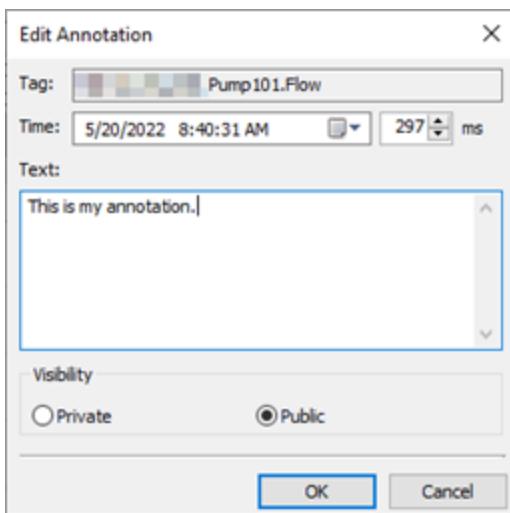
• Type	Specifies whether the annotation is public.
• Tag Name	The name of the tag within the Historian server. If the data values are coming from application server, the attribute reference is shown as the tag name. For application server attributes, you can also choose to show the hierarchical name along with the attribute reference.
• Server	The name of the server that stores the tag values.
• Content	The annotation text.
• User	The name of the database user. This is the user that created the annotation.
• Time	The timestamp of the tag value for which the user has made an annotation.
• Created On	The date that the annotation was created.

2. To sort the table according to the information in a particular column, click the column heading. Click again to reverse the sorting order.

Edit an annotation

To edit an annotation

1. On the **View** menu, click **Annotations**. The **Annotations** dialog box appears.
2. Select the annotation in the list.
3. On the **Annotations** menu, click **Edit**. The **Edit Annotation** dialog box appears.



4. Edit the annotation.
5. Click **OK**.

Delete an annotation

Deleting an annotation removes the annotation from the trend.

To delete an annotation

1. On the **View** menu, click **Annotations**. The **Annotations** dialog box appears.
2. Select the annotation in the list.
3. On the **Annotations** menu, click **Delete**. Confirm the deletion.
4. Click **OK**.

Save the annotations list as a .CSV file

To save the list of annotations as a .csv (text) file

1. On the **View** menu, click **Annotations**. The **Annotations** dialog box appears.
2. On the **File** menu, click **Save As**. The standard Windows **Save As** dialog box appears.
3. In the **File name** box, type a name for the .csv file.
4. Browse to the location in which to save the file.
5. Click **Save**.

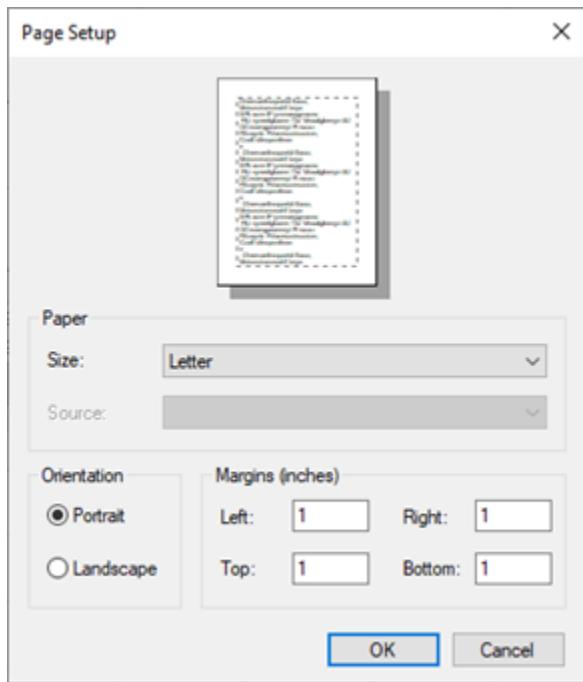
The .csv file contains the same information that appears in the **Annotations** dialog box.

Print annotations

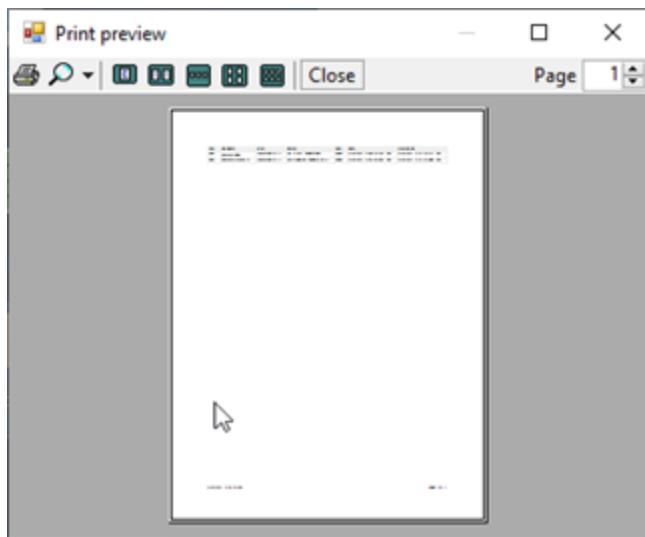
To print the list of annotations

1. On the **View** menu, click **Annotations**. The **Annotations** dialog box appears.

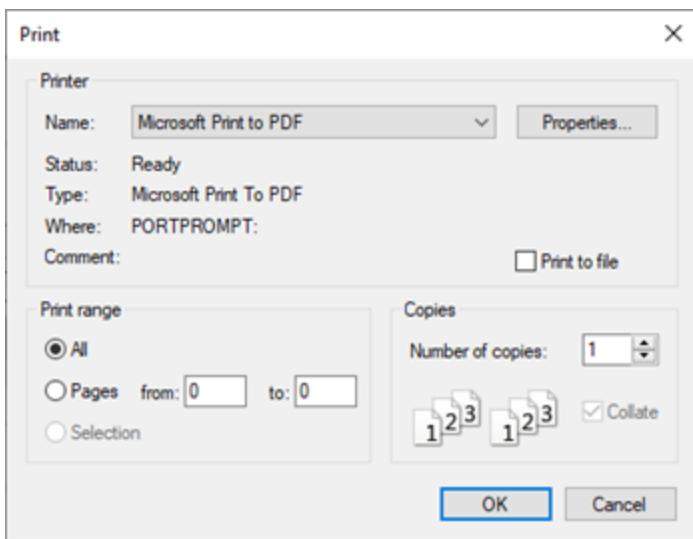
2. To configure the printing options, on the **File** menu, click **Page Setup**. The **Page Setup** dialog box appears.



3. Configure the setup options and then click **OK**.
4. To preview the printout, on the **File** menu, click **Print Preview**. The **Print preview** dialog box appears.



5. Verify the preview and then click **Close**.
6. To print the annotations, on the **File** menu, click **Print**. The **Print** dialog box appears.



7. Configure the printing options and then click **OK**.

Work with trend files

This section describes how to create, open, and save trend files. A trend file contains all of the configuration data required to trend one or more tags, such as the tags, time axes, colors, zoom level, and so on.

Any mix of analog, discrete, event, or summary tags is allowed. However, the mix of summary tags with other tags is dependant on the retrieval mode. For example, if the summary tags are supported by Cyclic retrieval mode and that retrieval mode is set, then we can mix the summary tags with the other tags for trending.

There is no pre-set limit to the number of tags that you can have in a trend; however, keep in mind performance limitations for data retrieval for your computer system.

Create a trend

Creating a new trend chart resets all trend settings to the default values.

To create a new trend

- On the **File** menu, click **New**.

To configure the new trend, see [Configure a trend](#). By default, the new trend is in standard trend mode. To create an XY scatter plot, see [View data in a scatter plot](#).

Configure default settings for a trend file

You can configure default settings for a trend in a "Default.aaTrend" file and use it as a template for your new trend documents.

The trend properties such as border thickness, border style, background color, Tag List pane height and columns, and pens and their properties are saved in the Default.aaTrend file.

When you start the Trend application or Trend control for the first time, the Default.aaTrend file is created in:

C:\Users\<User>\AppData\Local\Wonderware\ActiveFactory\Trend

To configure the default settings

1. Create a new trend file. For more information, see [Create a new trend](#).
2. Configure the default settings such as the background color, pen color, and trend chart views.
3. Save the file as Default.aaTrend in the default location.
4. Restart the Trend application.

The default settings are applied to the new trend documents. To restore the default trend settings, delete the Default.aaTrend file and create a new trend document.

Open an existing trend

To open an existing trend

1. On the **File** menu, select **Open**.
2. Select **Trend** to open a single trend file.
The standard Windows **Open** dialog box appears.
3. Browse to and select the trend file to open. All trend files have the .aaTrend extension.
4. Click **Open**. The trend appears in the chart.

Note: You can also double-click the .aaTrend file in the Windows Explorer to start up Trend.

To open a recent trend

- From the **File** menu, locate the file in **Recent Files**, and then double-click the name of the file to open.

Save a trend

To save a trend

1. From the **File** menu, select **Save**.



2. Click the **Save** button .

If you are saving the trend for the first time, the standard Windows **Save As** dialog box appears. Otherwise, the trend is saved to disk using the existing file name.

3. In the **Save As** dialog box, type a name for the trend. All trend files have the .aaTrend extension.

Note: The trend files with .aaTrend extension are not encrypted.

4. Click **OK**.

To save a trend as another name

1. From the **File** menu, select **Save**.
2. Click **Save As**.

The standard Windows **Save As** dialog box appears.

3. In the **Save As** dialog box, type a name for the trend. All trend files have the .aaTrend extension.

Note: The trend files with .aaTrend extension are not encrypted.

1. Click **OK**.

Close a trend

To close a trend

Do one of the following:

- On the **File** menu, click **Close**.
- Click the trend window's close button .

Undo or redo actions

The trend application supports 64 levels of undo/redo. You can undo/redo all of the actions for scaling and moving, as well as for adding and removing tags in the trend.

To undo an action

- Click the **Undo**  toolbar button

To redo an action

- Click the **Redo**  toolbar button

Output trend data

You can output trend data to a printer, a .csv file, or to an image file, such as .bmp, jpeg, and .gif. You can also copy and paste the trend graph and associated Tag List to the Windows Clipboard.

Note: The capability to output a trend to an image file is available only in System Platform 2020 R2 SP1 and later.

Print trend data

Before you print a chart, you can specify print options and preview the printout. Use the following commands:

- To set up the print page: From the **Home** menu, in the **Print** group, click **Setup**.
- To preview the print output: From the **Home** menu, in the **Print** group, click **Preview**.
- To print the chart: From the **Home** menu, in the **Print** group, click **Print**.

The available options for these commands work like in any other Windows application.

Note: When you print a chart, only the data that is currently displayed in the application appears in the printout. For example, if part of the Tag List is not displayed in the application, then that portion does not appear in the printout.

Print trend sets

A trend set is a saved grouping of trend files. You can specify a common trend duration (for example, the last 24 hours) to apply to all of the files in the set.

This allows you to easily print information for the same duration from multiple trend files at the same time.

Print a trend set

To print a trend set

- From the **File** menu, in the **Print** group, click **Print**.

If you double-click a trend set in Windows Explorer, the trend set opens in the Historian Client Trend application, the associated trends are printed, and then the application closes automatically.

You can also print a trend set from a command prompt by executing the trend set filename, including the extension:

```
aatrend /s <fully qualified filename>
```

To have the Trend application automatically close after the trend set is printed, omit the /s parameter.

Save trend data to a .CSV file

When you save trend data, all data is exported to comma separated values (.csv) format. The .csv file includes all time stamps and values for the tags on the current trend chart at the time of the save.

To save trend data

- From the **File** menu, select **Save**.
- Select **Save data**.
The standard Windows **Save As** dialog box appears.
- Browse to the location in which you want to save the file.
- In the **File name** box, type a name for the trend data file.
- Click **Save**.

You can view the data in any spreadsheet application that can open .csv files, such as Microsoft Excel.

If you do not want the data values to be blank unless the value is NULL, use the Cyclic retrieval style and add the following entry into the win.ini file:

```
[HistClient]
```

```
KeepAllTrendPoints=1
```

Save the trend chart as an image file

You can save a trend chart as a .bmp, .gif, .jpeg, .svg, or .png image file.

Note: The capability to output a trend to an image file is available only in System Platform 2020 R2 SP1 and later.

To save the trend chart

1. From the **File** menu, select **Save**.
2. Click **Save Image**.
The standard Windows **Save As** dialog box appears.
3. Browse to the location in which you want to save the file.
4. In the **File name** box, type a name for the trend image file.
5. In the **Save as Type** box, select an image type.
6. Click **Save**.

E-mail a trend file

To e-mail a trend file, you must have a valid SMTP server and account configured and an e-mail application correctly installed and configured with a suitable e-mail account.

Before you e-mail a trend, make sure to save the trend file on your computer.

To e-mail a trend file

1. From the **Home** menu, in the **Export** group, select **Send To**. The e-mail program starts up and a new message appears.

Note: By default, the trend file that you want to send does not have a *.aaTrend file extension.
2. Remove the attachment and then browse to the location and attach the trend file that includes the trend *.aaTrend file as an extension.
3. Use the e-mail program to send the trend file.

Copy a trend chart to the Windows clipboard

When you copy a trend chart, only the data that is currently displayed in the application is copied. For example, if part of the associated Tag List is not displayed in the application, then that portion is not copied.

To copy a trend chart

1. Click in the trend chart so that it has focus.
2. From the **Home** menu, in the **Clipboard** group, click **Copy**.
3. Open the target application (for example, Microsoft Word).
4. Paste the chart.

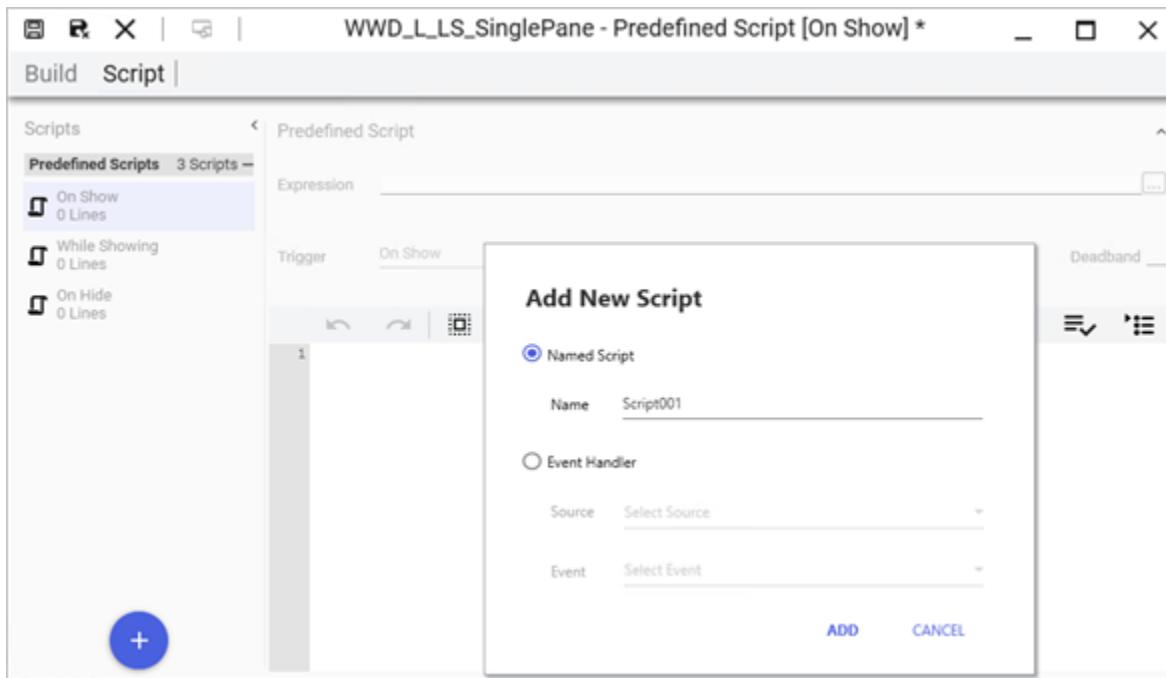
The trend chart is pasted as a graphic in the target application.

Historical trend pen scripts

The HistoricalTrendApp can show the underlying .NET Winforms and Windows Presentation Foundation (WPF) control properties, methods, and events in addition to the native properties and methods included with the app.

The underlying control properties appear on the Layout editor **Properties** grid after placing the HistoricalTrendApp onto a layout pane. Because of the large number of underlying control properties, use the Layout editor's search function to find a control property.

HistoricalTrendApp methods, properties, and events can be used by the Layout script editor to create named and event handler scripts.



For more information about writing scripts with the Layout editor, see [Build layout scripts](#).

Write an HistoricalTrendApp named script

You can write named and event handler scripts from the Layout editor after the HistoricalTrendApp has been placed onto a layout pane. The Layout script editor supports the MyContent namespace to write named scripts specifically for the HistoricalTrendApp when it is content placed on a layout pane.

The Layout script editor includes the AutoComplete feature, which shows a positional drop-down list of namespaces, content types, and the methods or properties associated with HistoricalTrendApp content.

Write an HistoricalTrendApp named script

1. Open the layout and show the items listed in the **Toolbox** tab.

2. Select the HistoricalTrendApp from the **Toolbox** list to show its preview thumbnail.

The **Toolbox** pathway to the HistoricalTrendApp is:

_Default Content > 4. Apps > AVEVA OMI Apps > HistoricalTrendApp

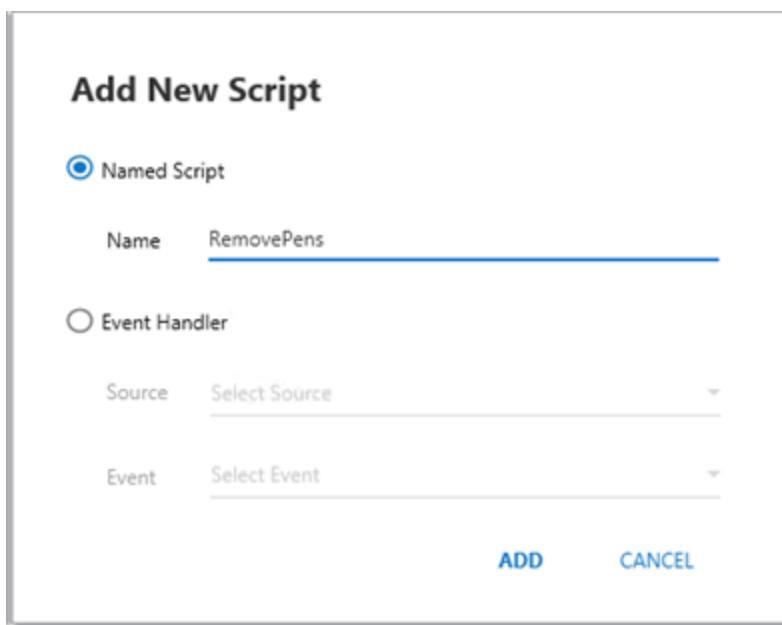
3. Drag and drop the preview thumbnail onto a layout pane.

4. Select **Script** from the menu bar of the Layout editor to open the script editor.

5. Create a named script.

- a. Select the red plus circle to show the **Add NewScript** dialog box.

- b. Select **Named Script**, assign a name to the script, and select **Add** to close the dialog.



The new script appears beneath the **Named Scripts** area at the left of the editor and the window focus is set to the script canvas to begin writing script code.

6. Enter your script code.

A named script intended for a HistoricalTrendApp placed on a layout pane uses the MyContent namespace and its content name is assigned by its **Name** property in the Properties grid of the Layout editor. Place a period immediately after the item you selected from the AutoComplete list. AutoComplete shows the next positional list of items based on your previous selections in the script statement.

AutoComplete provides a list of properties and methods that can be included in the script based on the `MyContent.NameProperty` prefix of the script statement.

- For .NET methods and properties, refer to .NET documentation on the Microsoft web site.
- For descriptions of legacy HistoricalTrendApp methods and properties, refer to the *ArchestrA Trend Client User Guide* that is delivered with System Platform 2020. The document is a PDF file located on the computer hosting the System Platform IDE.

`C:\Program Files (x86)\ArchestrA\Framework\Docs\1033\aaTrendClient.pdf`

7. Validate your script code using the script validation tool shown at the right of the script editor menu bar.



8. After making any script corrections, save and exit from the Layout script editor.

Write an event handler script

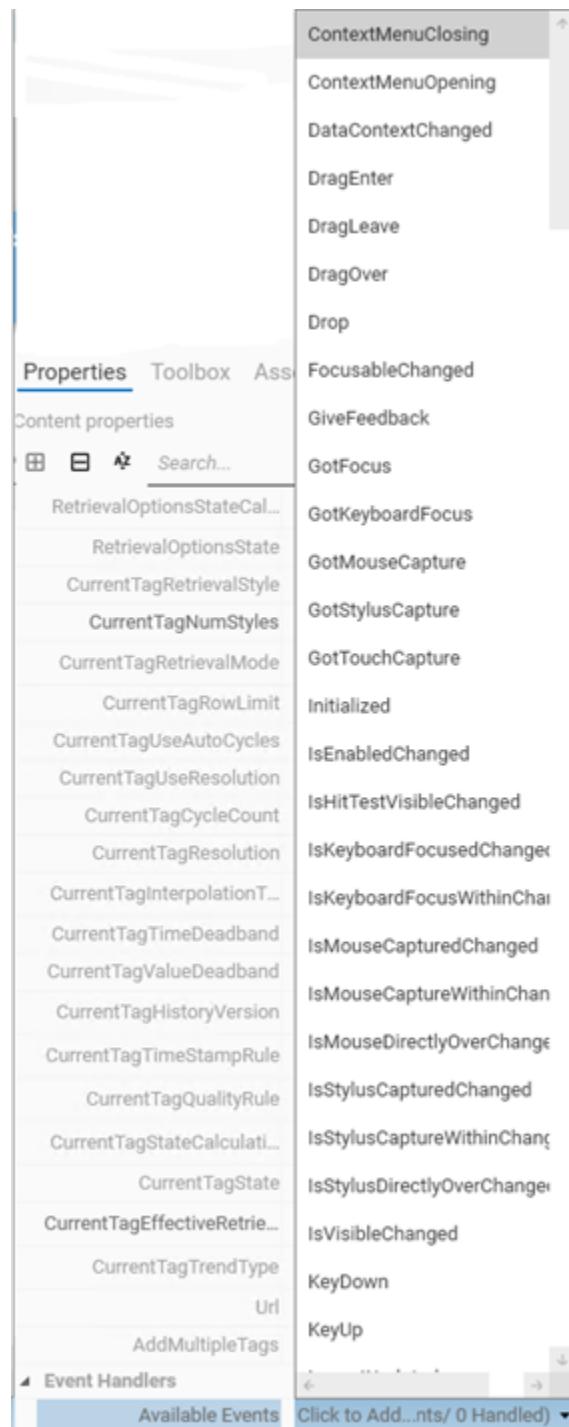
You can write event handler scripts by two methods after placing the HistoricalTrendApp onto a layout pane.

- Assigning values to the **Available Events** property on a selected control's **Properties** grid
You select an event from a list of all available events of the control to create an event handler.
- Add a new event handler script from the Layout script editor's **Add New Script** dialog.

The following topics describe the workflows to create an event handler script by these two methods.

Write an event handler script from the properties grid

The underlying control of the HistoricalTrendApp includes a set of public properties, methods, and events that can be viewed from the Layout editor's **Properties** grid. Near the bottom of the Properties grid is the **Available Events** property, which shows a list of HistoricalTrendApp events after being selected.



After selecting a public HistoricalTrendApp event from the list, the Layout script editor opens to code the event handler script.

Write an event handler script using Available Events property of the HistoricalTrendApp

1. Open the Layout editor and show the items listed in the **Toolbox** tab.
2. Select the HistoricalTrendApp from the **Toolbox** list to show its preview thumbnail.

The **Toolbox** pathway to the HistoricalTrendApp is:

_Default Content > 4. Apps > AVEVA OMI Apps > HistoricalTrendApp

The preview thumbnail of the HistoricalTrendApp appears beneath the list of **Toolbox** items.

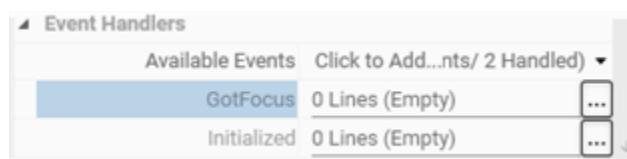
3. Drag and drop the preview thumbnail of the HistoricalTrendApp onto a layout pane.
 4. Select the pane where the HistoricalTrendApp has been placed and select the **Properties** grid.
- HistoricalTrendApp properties appear in the grid. HistoricalTrendApp events appear in a list of the **Available Events** property.
5. Select the data entry field of the **Available Events** property to show a list of events.

Events are listed in alphabetic order.

6. Select one or more events from the list.

A selected handled event name appears in bold text to indicate that it is selected.

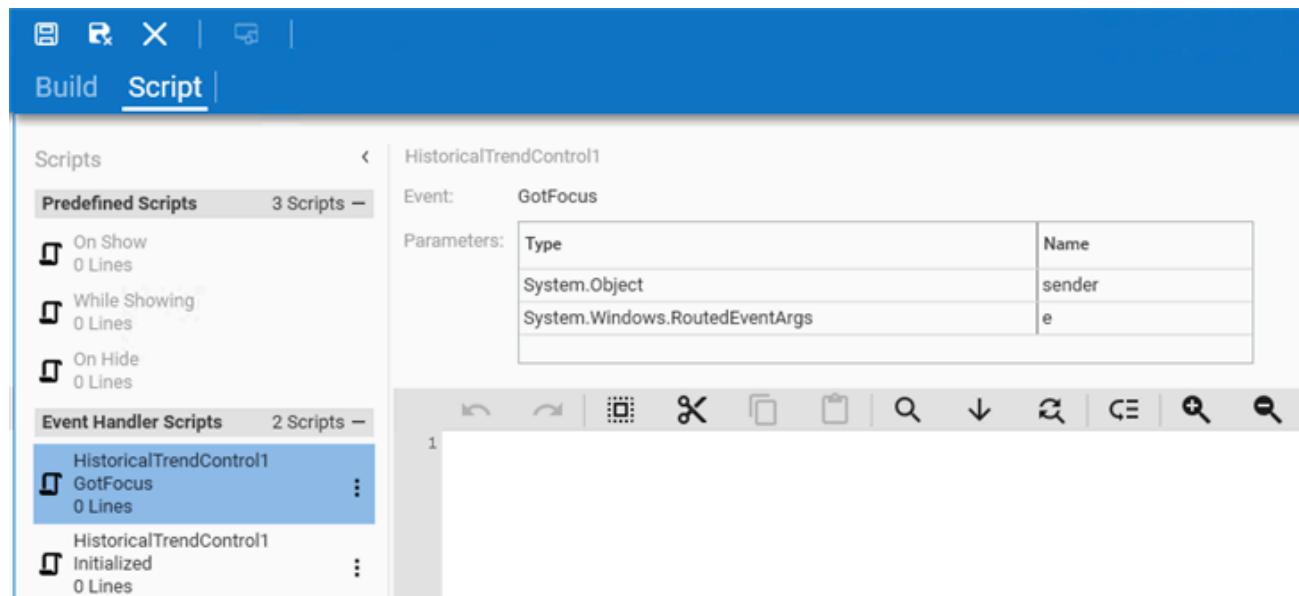
The **Available Events** property updates to show each selected handled event with a data entry field.



All handled events without any script lines are listed in alphabetic order.

7. Select the button at the right of the selected event's data entry field.

The Layout script editor opens with the selected handled events listed beneath the **Event Handler Scripts** column.



The **Event** field shows the selected event script from the list.

8. Enter script code for each event handler.

The number of lines in each event handler script are shown in the Available Events property of the Layout editor.

9. Validate your script code by selecting the validation button at the right on the menu bar above the script edit area.



10. Save your scripts and exit from the Layout script editor.

Write an event handler script from the Layout script editor

Event handler scripts can be created with the **Add New Script** dialog box of the Layout script editor. Handled events can be selected for the **Event** list of the **Add New Script** dialog box.

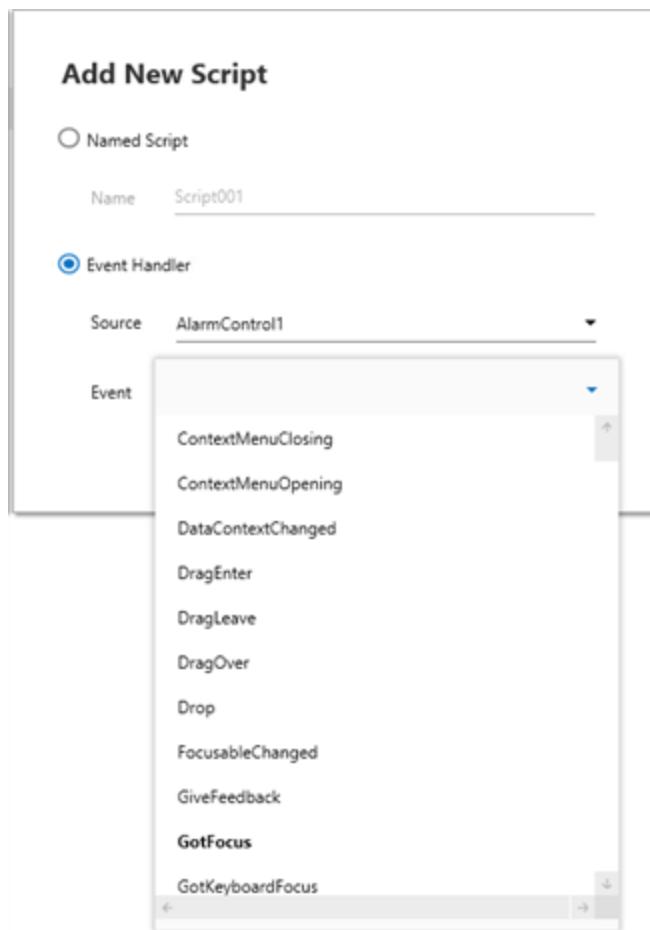
Write an HistoricalTrendApp event handler script from the control Properties page

1. Open the layout and show the items listed in the **Toolbox** tab.
2. Select the HistoricalTrendApp from the **Toolbox** list to show its preview thumbnail.

The **Toolbox** pathway to the HistoricalTrendApp is:

_Default Content > 4. Apps > AVEVA OMI Apps > HistoricalTrendApp

3. Drag and drop the preview thumbnail onto a layout pane.
4. Select **Script** from the menu bar of the Layout editor to open the script editor.
5. Create an event handler script.
 - a. Within the Layout script editor, select the red plus circle to show the **Add NewScript** dialog box.
 - b. Select **Event Handler**.
 - c. Select the name of the AlarmControl placed on the layout pane from the **Source** list.
 - d. Select the data field of **Event** to show a list of event handlers.



- e. Select one or more event handlers from the list.
- f. Select **Add** to close the **Add New Script** dialog box and return to the Layout script editor.
The **Event Handler Scripts** area shows the selected event handlers in alphabetic order.
6. Select an event handler from the list and enter script code for each event handler.
The number of lines in each event handler script are shown in the **Available Events** property of the Layout editor.
7. Validate your script code by selecting the validation button at the right on the menu bar above the script edit area.
8. Save your scripts and exit from the Layout script editor.
The new event handler scripts appears beneath the **Event Handler Scripts** area at the left of the editor and the window focus is set to the script edit area to begin writing script code.

Data retrieval options

You can use a variety of retrieval modes and options to suit different reporting needs and applications.

Understanding retrieval modes

Different retrieval modes allow you to access the data stored in an AVEVA Historian in different ways. For example, if you retrieve data for a long time period, you might want to retrieve only a few hundred evenly spaced data points to minimize response time. For a shorter time period, you might want to retrieve all values that are stored on the server to get more accurate results.

An AVEVA Historian with a version earlier than 9.0 supports two retrieval modes:

- [Cyclic Retrieval](#)
- [Delta Retrieval](#)

An AVEVA Historian with a version of 9.0 or higher supports various additional modes:

- [Full Retrieval](#)
- [Interpolated Retrieval](#)
- "Best fit" retrieval
- [Average Retrieval](#)
- [Minimum Retrieval](#)
- [Maximum Retrieval](#)
- [Integral Retrieval](#)
- [Slope Retrieval](#)
- [Counter Retrieval](#)
- [ValueState Retrieval](#)

An AVEVA Historian with a version of 10.0 or higher supports the following additional mode:

- [RoundTrip Retrieval](#)

An AVEVA Historian version 11.6.14101 or higher supports the following additional mode:

- [Predictive Retrieval](#)

An AVEVA Historian with a version of 17.3.100 or higher supports the following additional mode:

- [Bounding value retrieval](#)

Cyclic retrieval

Cyclic retrieval is the retrieval of stored data for the given time period based on a specified cyclic retrieval resolution, regardless of whether or not the value of the tag(s) has changed. It works with all types of tags. Cyclic retrieval produces a virtual rowset, which may or may not correspond to the actual data rows stored on the Historian.

In cyclic retrieval, one row is returned for each "cycle boundary." You specify the number of cycles either directly or by means of a time resolution, that is, the spacing of cycle boundaries in time. If you specify a number of cycles, the Historian returns that number of rows, evenly spaced in time over the requested period. The cyclic resolution is calculated by dividing the requested time period by the number of cycle boundaries. If you specify a

resolution, the number of cycles is calculated by dividing the time period by the resolution.

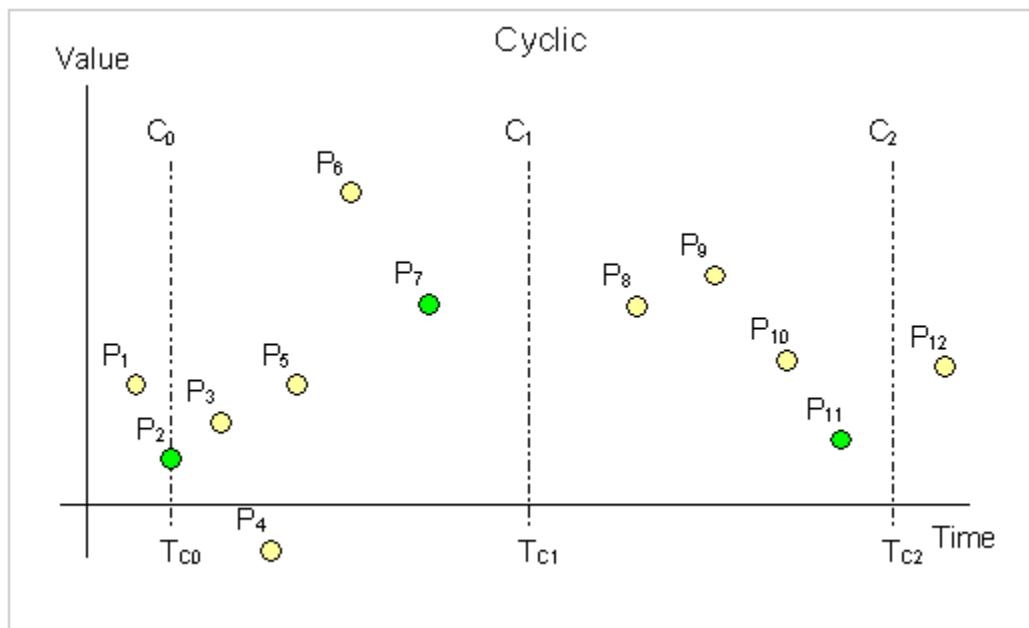
If no data value is actually stored at a cycle boundary, the last value before the boundary is returned.

The default retrieval mode is cyclic for retrieval from analog tables, including analog and state summary tables.

Cyclic retrieval is fast and therefore consumes little server resources. However, it may not correctly reflect the stored data because important process values (gaps, spikes, etc.) might fall between cycle boundaries. For an alternative, see "[Best fit](#)" retrieval.

Cyclic retrieval - how it works

The following illustration shows how values are returned for cyclic retrieval:



Data is retrieved in cyclic mode with a start time of T_{C0} and an end time of T_{C2} . The resolution has been set in such a way that the Historian returns data for three cycle boundaries at T_{C0} , T_{C1} , and T_{C2} . Each dot in the graphic represents an actual data point stored on the Historian. From these points, the following are returned:

- At T_{C0} : P_2 , because it falls right on the cycle boundary
- At T_{C1} : P_7 , because it is the last point before the cycle boundary
- At T_{C2} : P_{11} , for the same reason

Cyclic retrieval - supported value parameters

You can use various parameters to adjust which values are returned in cyclic retrieval mode. For more information, see the following sections:

- [Cycle Count \(X Values over Equal Time Intervals\) \(wwCycleCount\)](#).
- [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#).
- [History Version \(wwVersion\)](#).

- [Timestamp Rule \(wwTimestampRule\)](#) (only on Historian 9.0 and above).

Cyclic retrieval - query example

To use the cyclic retrieval mode, set the following parameter in your query.

```
wwRetrievalMode = 'Cyclic'
```

Cyclic retrieval - initial values

No special handling is done for initial values. The initial value will behave like a normal cycle boundary at the start time. For information on initial values, see [Delta Retrieval - Initial Values](#).

Cyclic retrieval - handling NULL values

No special handling is done for NULL values. They are returned just like any other value.

Delta retrieval

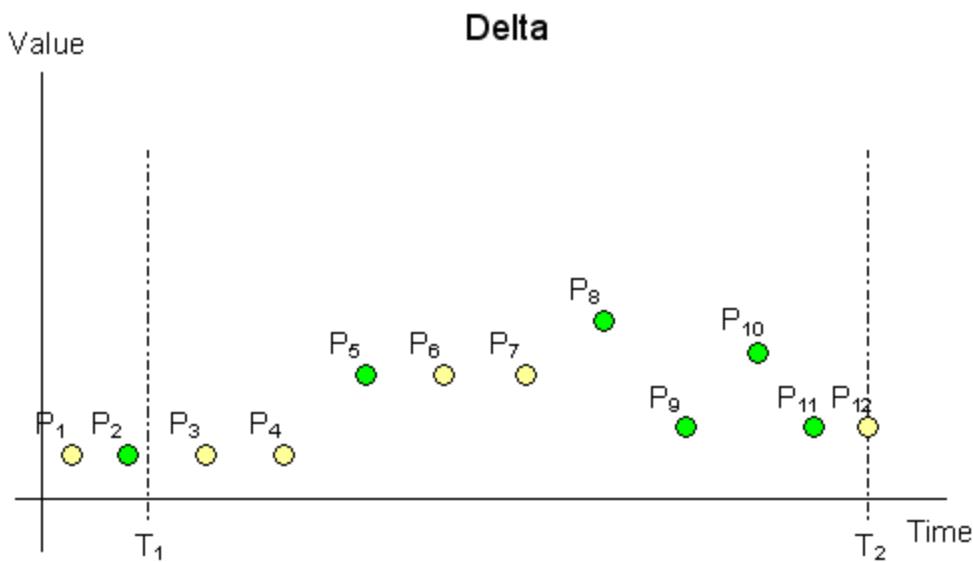
Delta retrieval, or retrieval based on exception, is the retrieval of only the changed values for a tag(s) for the given time interval. That is, duplicate values are not returned. It works with all types of tags.

Delta retrieval always produces a rowset comprised of only rows that are actually stored on the Historian; that is, a delta query returns all of the physical rows in history for the specified tags, over the specified period, minus any duplicate values. If there is no actual data point at the start time, the last data point before the start time is returned.

Delta retrieval is the default mode for discrete and string tables and from the History table.

Delta retrieval - how it works

The following illustration shows how values are returned for delta retrieval:



Data is retrieved in delta mode with a start time of T_1 and an end time of T_2 . Each dot in the graphic represents an actual data point stored on the Historian. From these points, the following are returned:

- P_2 , because there is no actual data point at T_1
- P_5, P_8, P_9, P_{10} , and P_{11} , because they represent changed values during the time period

For delta retrieval for replicated summary tags on a tier-2 historian, if a point with doubtful quality is returned as the result of a value selection from an input summary point with a contained gap, the same point can be returned again with good quality if the same value is selected again from the next input summary point that has good quality.

Delta retrieval - supported value parameters

You can use various parameters to adjust which values are returned in delta retrieval mode. For more information, see the following sections:

- [Time Deadband \(wwTimeDeadband\)](#)
- [Value Deadband \(wwValueDeadband\)](#)
- [History Version \(wwVersion\)](#)

Delta retrieval - query example

To use the delta retrieval mode, set the following parameter in your query.

```
wwRetrievalMode = 'Delta'
```

Both leading and trailing edge detection for discrete tags

If Both is specified as the parameter in the edge detection extension, all rows satisfying both the leading and trailing conditions are returned.

The following queries select values of "SysPulse" and "MyPulse" that meet an edge detection of Both for a value criterion of 1 (On) from the History and WideHistory tables between 12:59 and 1:04 a.m. on December 8, 2001.

Delta retrieval - initial values

Initial values are special values that can be returned from queries that lie exactly on the query start time, even if there is not a data point that specifically matches the specified start time. If there is not a value exactly on the query start time, the last point before the start time will be returned with its DateTime set to the query start time and its Quality set to 133. If no value exists at or prior to the query start time, a NULL value will be returned at start time with QualityDetail set to 65536, OPCQuality set to 0, and Quality set to 1.

Querying the start time in exclusive form with the > operator indicates that a value should not be returned for the query start time if one does not exist. Querying the start time in inclusive form with the >= operator indicates that an initial value should be returned.

For example, the following exclusive query statement does not return an initial value for 2009-01-01 02:00:00.

```
DateTime > '2009-01-01 02:00:00'
```

However, the following inclusive query statement does return an initial value for 2009-01-01 02:00:00.

```
DateTime >= '2009-01-01 02:00:00'
```

No special final value is returned.

Delta retrieval - handling NULL values

The initial NULL value after a non-NUL is always returned. Multiple NULL values are suppressed. The first non-NUL after a NULL is always returned even if it is the same as the previous non-NUL value.

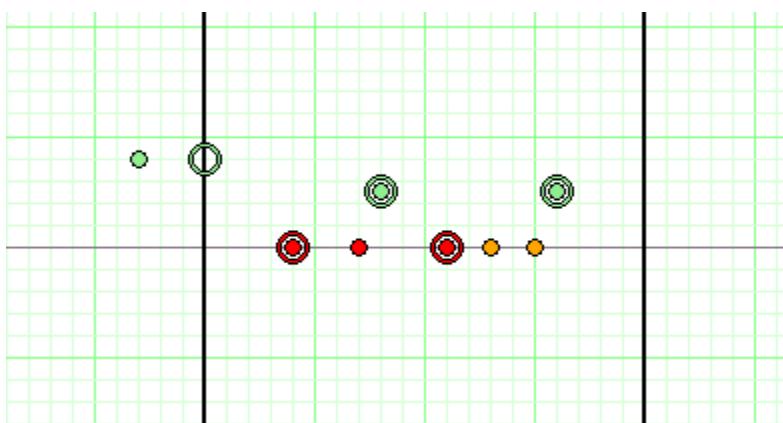
```
SELECT TagName, DateTime, Value, QualityDetail
  FROM History
 WHERE TagName = 'A001'
   AND DateTime >= '2009-09-12 00:20'
   AND DateTime <= '2009-09-12 00:40'
   AND wwRetrievalMode = 'Delta'
```

This query can be run against the following sample data:

TagName	DateTime	Value	QualityDetail
A001	2009-09-12 00:17	0.8	192
A001	2009-09-12 00:24	0.0	249

A001	2009-09-12 00:27	0.0	249
A001	2009-09-12 00:28	0.5	192
A001	2009-09-12 00:31	0.0	249
A001	2009-09-12 00:33	0.0	24
A001	2009-09-12 00:35	0.0	24
A001	2009-09-12 00:36	0.5	192

The following is a graphical representation of the data:



The results are:

TagName	Datetime	Value	QualityDetail
A001	2009-09-12 00:20	0.8	192
A001	2009-09-12 00:24	NULL	249
A001	2009-09-12 00:28	0.5	192
A001	2009-09-12 00:31	NULL	249
A001	2009-09-12 00:36	0.5	192

The sample data points and the results are mapped on the following chart. Only the data falling between the time start and end marks at 00:20 and 00:40 (shown on the chart as dark vertical lines) are returned by the query.

Because there is no value that matches the start time, an initial value at 00:20 is returned in the results based on the value of the preceding data point at 00:16. Because there is no change in the value at 00:27 from the value at 00:24, the data point appears on the chart but does not appear in the results. Similarly, the two 0.0 values at 00:33 and 00:35 are also excluded from the results. However, the non-NULL value at 00:36 is returned, even though it is the same as the value at 00:28, because it represents a delta from the preceding (NULL) value at 00:35.

Full retrieval

In full retrieval mode, all stored data points are returned, regardless of whether a value or quality has changed since the last value. This mode allows the same value and quality pair (or NULL value) to be returned consecutively with their actual timestamps. It works with all types of tags.

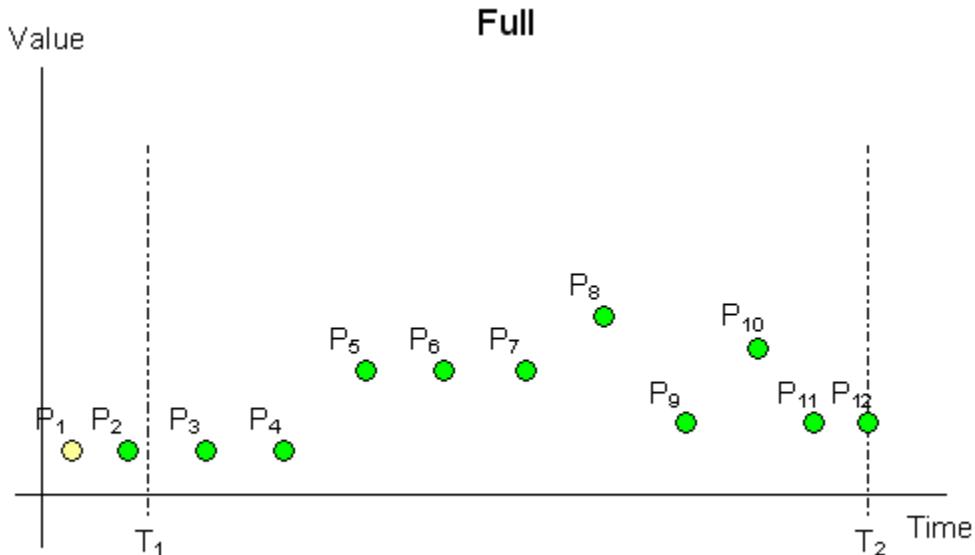
By using full retrieval in conjunction with storage without filtering (that is, no delta or cyclic storage mode is applied at the Historian), you can retrieve all values that originated from the plant floor data source or from another application.

Full retrieval best represents the process measurements recorded by the Historian. However, it creates a higher load for the server, the network and the client system because a very large number of records may be returned for longer time periods.

For full retrieval for replicated summary tags on a tier-2 historian, if a point with doubtful quality is returned as the result of a value selection from an input summary point with a contained gap, the same point can be returned again with good quality if the same value is selected again from the next input summary point that has good quality.

Full retrieval - how it works

The following illustration shows how values are returned for full retrieval:



Data is retrieved in full mode with a start time of T_1 and an end time of T_2 . Each dot in the graphic represents an actual data point stored on the Historian. From these points, the following are returned:

- P_2 , because there is no actual data point at T_1
- P_3 through P_{12} , because they represent stored data points during the time period

Full retrieval - supported value parameters

You can use various parameters to adjust which values are returned in full retrieval mode. For more information, see the following sections:

- [History Version \(wwVersion\)](#)

Full retrieval - query example

Query 1

```
SELECT DateTime, TagName, Value
  FROM History
 WHERE TagName IN ('SysTimeSec','SysTimeMin')
   AND DateTime >= '2001-12-09 11:35'
   AND DateTime <= '2001-12-09 11:36'
   AND wwRetrievalMode = 'Full'
```

Full retrieval - initial values

Full retrieval mode handles initial values the same way as delta mode. For more information on initial values, see [Delta Retrieval - Initial Values](#).

Interpolated retrieval

Interpolated retrieval works like cyclic retrieval, except that interpolated values are returned if there is no actual data point stored at the cycle boundary.

This retrieval mode is useful if you want to retrieve cyclic data for slow-changing tags. For a trend, interpolated retrieval results in a smoother curve instead of a "stair-stepped" curve. This mode is also useful if you have a slow-changing tag and a fast-changing tag and want to retrieve data for both. Finally, some advanced applications require more evenly spaced values than would be returned if interpolation was not applied.

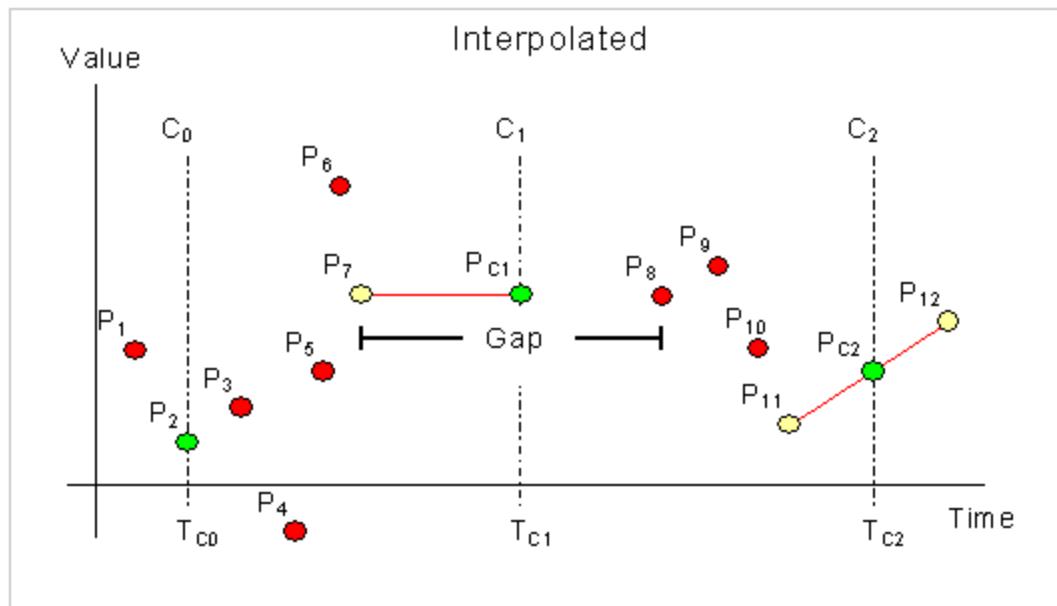
By default, interpolated retrieval uses the interpolation setting specified for the tag in the Historian. This means that if a tag is set to use stair-step interpolation, interpolated retrieval gives the same results as cyclic retrieval.

Interpolation is only applied to analog tags. If you retrieve data for other types of tags, stair-step interpolation is used, and the results are the same as for cyclic retrieval.

Interpolated retrieval is a bit slower than cyclic retrieval. It shares the limitations of cyclic retrieval in that it may not accurately represent the stored process data.

Interpolated retrieval - how it works

The following illustration shows how the values for an analog tag that is configured for linear interpolation are returned when using interpolated retrieval.



Data is retrieved in interpolated mode with a start time of T_{C0} and an end time of T_{C2} . The resolution has been set in such a way that the Historian returns data for three cycle boundaries at T_{C0} , T_{C1} , and T_{C2} . P_1 to P_{12} represent actual data points stored on the Historian. Of these points, eleven represent normal analog values, and one, P_7 , represents a NULL value due to an I/O Server disconnect, which causes a gap in the data between P_7 and P_8 .

The green points (P_2 , P_{c1} , P_{c2}) are returned. The yellow points (P_7 , P_{11} , P_{12}) are used to interpolate the returned value for each cycle. The red points are considered, but not used in calculating the points to return.

Because P_2 is located exactly at the query start time, it is returned at that time without the need for any interpolation. At the following cycle boundary, point P_{c1} is returned, which is the NULL value represented by P_7 shifted forward to time T_{C1} . At the last cycle boundary, point P_{c2} is returned, which has been interpolated using points P_{11} and P_{12} .

Interpolated retrieval - supported value parameters

You can use various parameters to adjust which values are returned in interpolated retrieval mode. For more information, see the following sections:

- Cycle Count (X Values over Equal Time Intervals) ([wwCycleCount](#))
- Resolution (Values Spaced Every X ms) ([wwResolution](#))
- History Version ([wwVersion](#))
- Interpolation Type ([wwInterpolationType](#))
- Timestamp Rule ([wwTimestampRule](#))
- Quality Rule ([wwQualityRule](#))

Interpolated retrieval - query example

To use the interpolated mode, set the following parameter in your query.

```
wwRetrievalMode = 'Interpolated'
```

Interpolated retrieval - initial and final values

A value is returned at the start time and end time of the query using interpolation of the surrounding points.

Interpolated retrieval - handling NULL values

When a NULL value precedes a cycle boundary, that NULL will be returned at the cycle boundary.

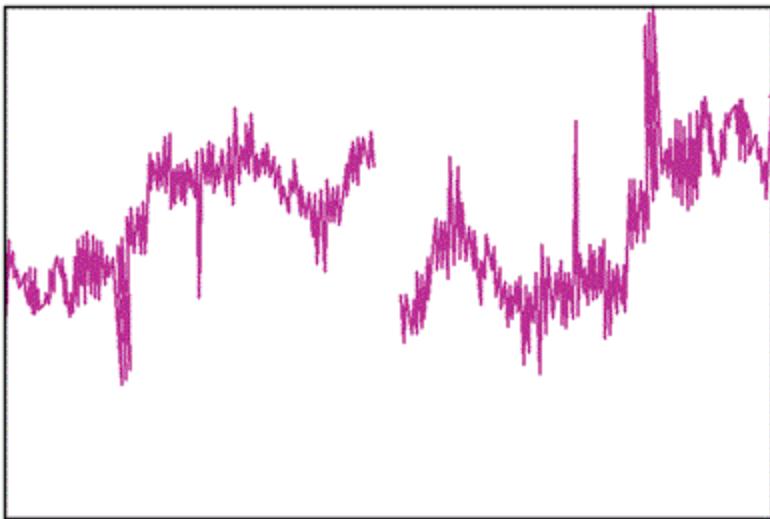
If a valid value precedes a cycle boundary, but is followed by a NULL value after the cycle boundary, no interpolation will be used and wwInterpolationType will be set to STAIRSTEP for that value.

"Best fit" retrieval

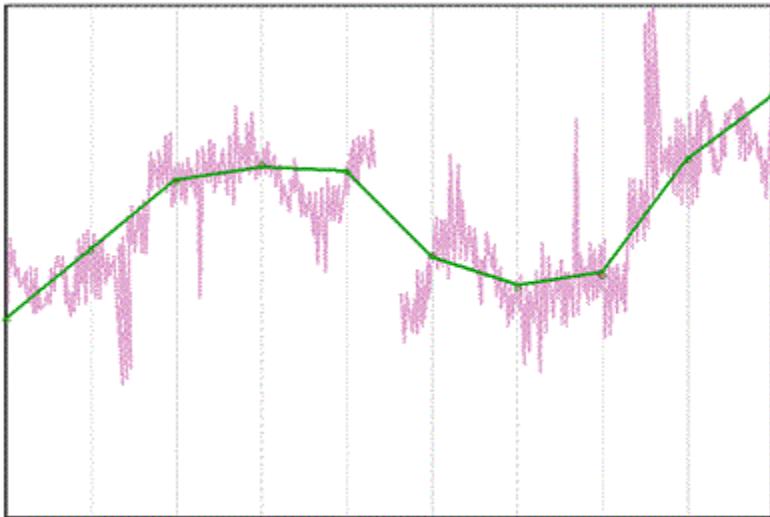
For the "best fit" retrieval mode, the total time for the query is divided into even sub-periods, and then up to five values are returned for each sub-period:

- First value in the period
- Last value in the period
- Minimum value in the period, with its actual time
- Maximum value in the period, with its actual time
- The first "exception" in the period (non-Good quality)

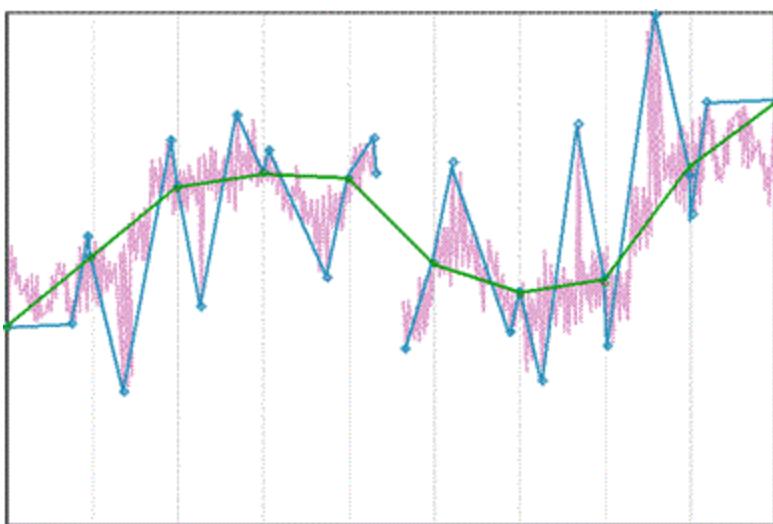
"Best fit" retrieval allows for a compromise between delta retrieval and cyclic retrieval. For example, delta retrieval can accurately represent a process over a long period of time, as shown in the following trend. However, to achieve this representation, a large number of data values must be returned.



If cyclic retrieval is used to retrieve the data, the retrieval is much more efficient, because fewer values are returned. However, the representation is not as accurate, as the following trend shows.



"Best fit" retrieval allows for faster retrieval, as typically achieved by using cyclic retrieval, plus the better representation typically achieved by using delta retrieval. This is shown in the following trend.



For example, for one week of five-second data, 120,960 values would be returned for delta retrieval, versus around 300 values for best-fit retrieval.

Best-fit retrieval uses retrieval cycles, but it is not a true cyclic mode. Apart from the initial value, it only returns actual delta points. For example, if one point is both the first value and the minimum value in a cycle, it is returned only one time. In a cycle where a tag has no points, nothing is returned.

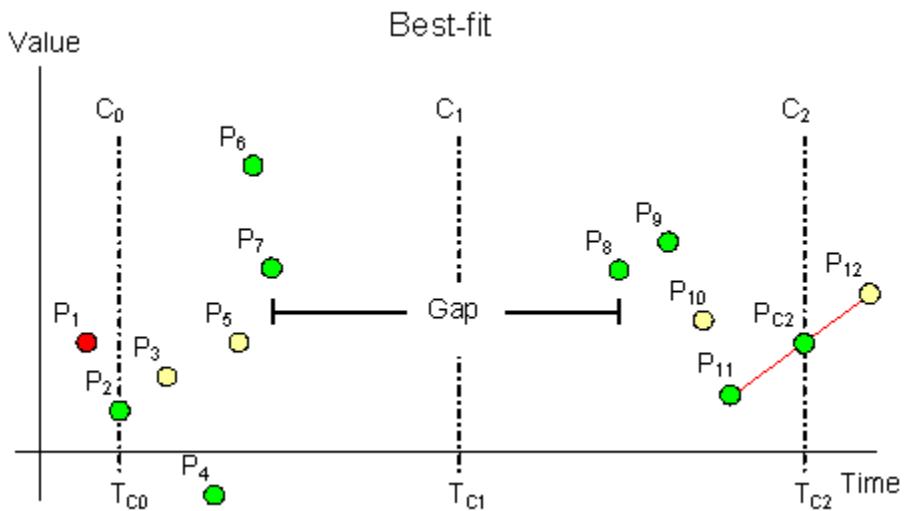
As in cyclic retrieval, the number of cycles is based on the specified resolution or cycle count. However, the number of values returned is likely to be more than one per cycle.

All points are returned in chronological order. If multiple points are to be returned for a particular timestamp, then those points are returned in the order in which the corresponding tags were specified in the query.

The best-fit algorithm is only applied to analog and analog summary tags. For all other tags, delta results are returned.

Best fit retrieval - how it works

The following illustration shows how the best-fit algorithm selects points for an analog tag.



Data is retrieved in best-fit mode with a start time of T_{C0} and an end time of T_{C2} . The resolution has been set in such a way that the Historian returns data for two complete cycles starting at T_{C0} and T_{C1} and an incomplete cycle starting at T_{C2} . P_1 to P_{12} represent actual data points stored on the Historian. Of these points, eleven represent normal analog values, and one, P_7 , represents a NULL value due to an I/O Server disconnect, which causes a gap in the data between P_7 and P_8 .

Because P_2 is located exactly at the start time, no initial value needs to be interpolated at the start time. Therefore, point P_1 is not considered at all. All other points are considered, but only the points indicated by green markers on the graph are returned.

From the first cycle, four points are returned:

- P_2 as the initial value of the query, as well as the first value in the cycle
- P_4 as the minimum value in the cycle
- P_6 as both the maximum value and the last value in the cycle
- P_7 as the first (and only) occurring exception in the cycle

From the second cycle, three points are returned:

- P_8 as the first value in the cycle
- P_9 as the maximum value in the cycle
- P_{11} as both the minimum value and the last value in the cycle
- As no exception occurs in the second cycle, none is returned.

Because the tag does not have a point exactly at the query end time, where an incomplete third cycle starts, the end value P_{C2} is interpolated between P_{11} and P_{12} , assuming that linear interpolation is used.

Best fit retrieval - supported value parameters

You can use various parameters to adjust which values are returned in best-fit retrieval mode. For more information, see the following sections:

- [Cycle Count \(X Values over Equal Time Intervals\) \(wwCycleCount\)](#)
- [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#)
- [History Version \(wwVersion\)](#)
- [Interpolation Type \(wwInterpolationType\)](#)
- [Quality Rule \(wwQualityRule\)](#)

Best fit retrieval - query example

To use the best fit retrieval mode, set the following parameter in your query.

```
wwRetrievalMode = 'BestFit'
```

Best fit retrieval - initial and final values

A point will be returned at the query start time and the query end time for each tag queried, if a point exists for that tag at or after the end time of the query. The values of the initial and final points will be determined by interpolating the points preceding and following the query start or query end time.

Standard interpolation rules will be used to return the initial and final values. For more information, see [Interpolated Retrieval](#).

Best fit retrieval - handling NULL values

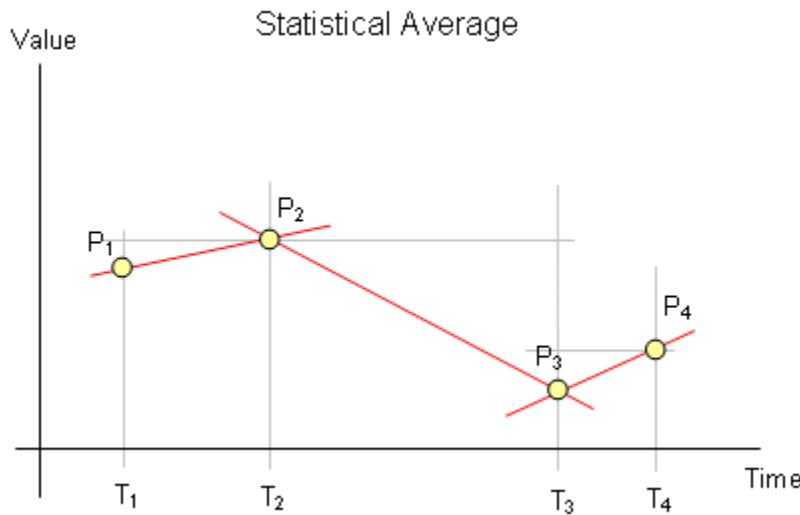
When any of the four good points are returned from a cycle that contains gaps or from an incomplete cycle with the query end time located inside of the calculation cycle the quality detail of each of the non-null points returned is modified to alert the user to this fact. This is done by performing a logical OR operation of the value 4096, which means partial cycle, onto the existing quality detail. (This is the delta point equivalent to the use of PercentGood for cyclic.)

Average retrieval

For the time-weighted average (in short: "average") retrieval mode, a time-weighted average algorithm is used to calculate the value to be returned for each retrieval cycle.

For a statistical average, the actual data values are used to calculate the average. The average is the sum of the data values divided by the number of data values. For the following data values, the statistical average is computed as:

$$(P_1 + P_2 + P_3 + P_4) / 4 = \text{Average}$$

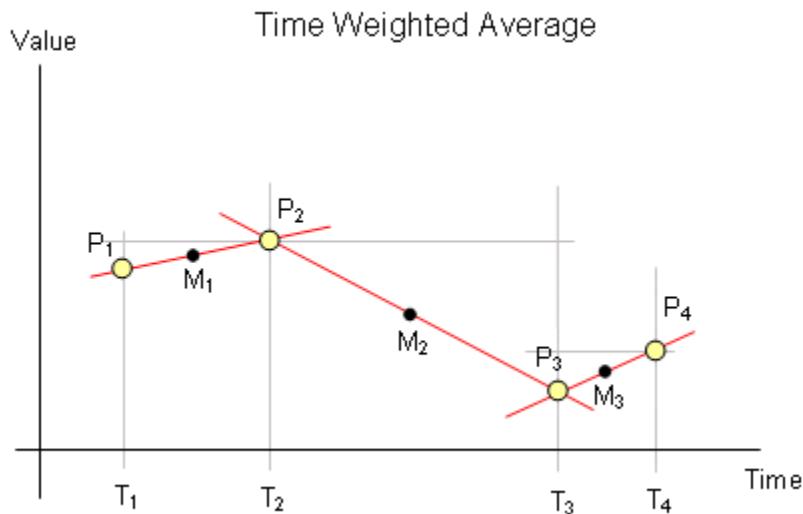


For a time-weighted average, values are multiplied by the time difference between the points to determine the time-weighted value. Therefore, the longer a tag had a particular value, the more weight that value holds in the overall average. The overall average is determined by adding all of the time-weighted values and then dividing that number by the total amount of time.

Which values are weighted depends on the interpolation setting of the tag. For a tag that uses linear interpolation, the midpoints between values are weighted. For a tag that uses stair-step interpolation, the earlier of two values is weighted.

For the following data values of a tag that uses linear interpolation, the time-weighted average is computed as:

$$((P_1 + P_2) / 2) \times (T_2 - T_1)) + (((P_2 + P_3) / 2) \times (T_3 - T_2)) + (((P_3 + P_4) / 2) \times (T_4 - T_3)) / (T_4 - T_1) = \text{Average}$$



If the same tag uses stair-step interpolation, the time-weighted average is:

$$((P_1 \times (T_2 - T_1)) + (P_2 \times (T_3 - T_2)) + (P_3 \times (T_4 - T_3))) / (T_4 - T_1) = \text{Average}$$

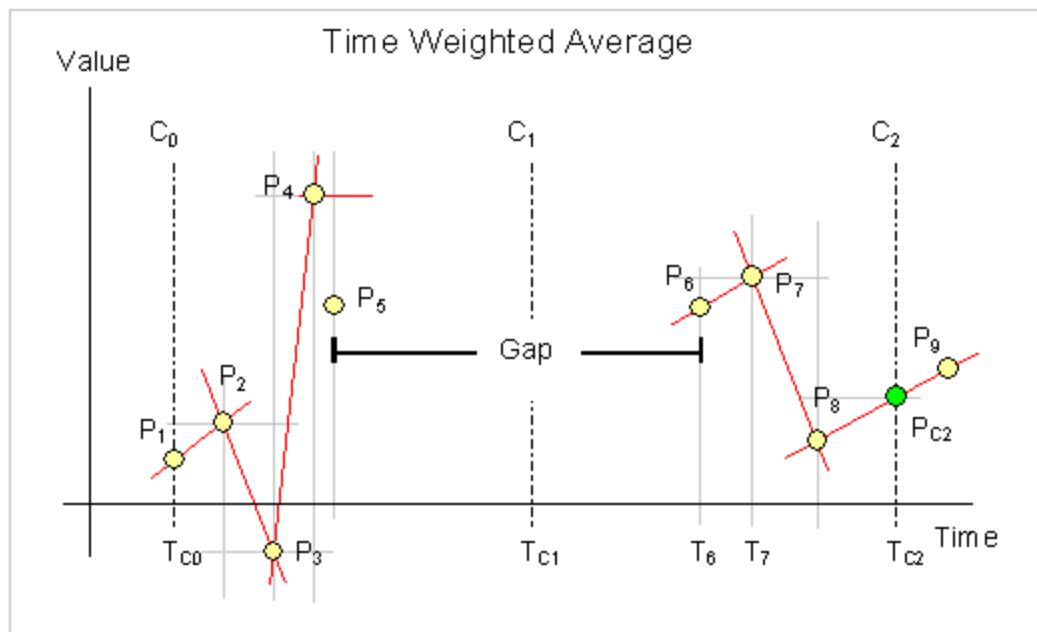
The SQL Server AVG aggregate is a simple statistical average. Using the average retrieval mode with a cycle count of 1 returns data much faster than the AVG aggregate, and usually more accurately due to the time weighting. The event subsystem also returns a simple statistical average.

Average retrieval returns one row for each tag in the query for each cycle. The number of cycles is based on the specified resolution or cycle count.

The time-weighted average algorithm is only applied to analog and analog summary tags. If you use average retrieval with other tags, the results are the same as when using regular cyclic retrieval.

Average retrieval - how it works

The following illustration shows how the time-weighted average is calculated for an analog tag that uses linear interpolation.



Data is retrieved in average mode with a start time of T_{C0} and an end time of T_{C2}. The resolution has been set in such a way that the Historian returns data for two complete cycles starting at T_{C0} and T_{C1} and an incomplete cycle starting at T_{C2}. P₁ to P₉ represent actual data points stored on the Historian. Of these points, eight represent normal analog values, and one, P₅, represents a NULL due to an I/O Server disconnect, which causes a gap in the data between P₅ and P₆. Assume that the query calls for timestamping at the end of the cycle.

Results are calculated as follows:

- The "initial value" returned at the query start time (T_{C0}) is the time-weighted average of the points in the last cycle preceding T_{C0}.
- The value returned at T_{C1} is the time-weighted average of the points in the cycle starting at T_{C0}.
- The value returned at the query end time (T_{C2}) is the time-weighted average of the points in the cycle starting at T_{C1}.

To understand how the time-weighted average is calculated, observe the last cycle as an example. First, the area under the curve must be calculated. This curve is indicated by the red line through P₆, P₇, P₈ and P_{C2}, where P_{C2} represents the interpolated value at time T_{C2} using points P₈ and P₉. The data gap caused by the I/O Server disconnect does not contribute anything to this area. If a quality rule of "good" has been specified, then points with doubtful quality will not contribute anything to the area, either.

To understand how the area is calculated, consider points P₆ and P₇. The area contribution between these two points is calculated by multiplying the arithmetic average of value P₆ and value P₇ by the time difference between the two points. The formula is:

$$((P_6 + P_7) / 2) \times (T_7 - T_6)$$

When the area for the whole cycle has been calculated, the time-weighted average is calculated by dividing that area by the cycle time, less any periods within the cycle that did not contribute anything to the area calculation. The result is returned at the cycle end time.

If you take a closer look at points P₄ and P₅ in the example, you can see that the red line through point P₄ is parallel to the x-axis. This is because P₅ represents a NULL, which cannot be used to calculate an arithmetic average. Instead, the value P₄ is used for the whole time period between points P₄ and P₅.

The area calculation is signed. If the arithmetic average between two points is negative, then the contribution to the area is negative.

Average retrieval - supported value parameters

You can use various parameters to adjust which values are returned in average retrieval mode. For more information, see the following sections:

- [Cycle Count \(X Values over Equal Time Intervals\) \(wwCycleCount\)](#)
- [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#)
- [History Version \(wwVersion\)](#)
- [Interpolation Type \(wwInterpolationType\)](#)
- [Timestamp Rule \(wwTimestampRule\)](#)
- [Quality Rule \(wwQualityRule\)](#)

Average retrieval - query examples

To use the average mode, set the following parameter in your query.

```
wwRetrievalMode = 'Average'
```

Query 1

The time-weighted average is computed for each of five 1-minute long cycles.

Note that the wwTimeStampRule parameter is set to "Start" in the query. This means that the value stamped at 11:18:00.000 represents the average for the interval 11:18 to 11:19, the value stamped at 11:19:00.000 represents the average for the interval 11:19 to 11:20, and so on. If no timestamp rule is specified in the query, then the default setting in the TimeStampRule system parameter is used.

In the first cycle there are no points, so a NULL is returned. In the second cycle value points are found covering 77.72 percent of the time as returned in PercentGood. This means that the returned average is calculated based on 77.72 percent of the cycle time. Because the same OPCQuality is not found for all the points in the cycle, OPCQuality is set to Doubtful. In the remaining three cycles, only good points occur, all with an OPCQuality of 192.

Because no quality rule is specified in the query using the wwQualityRule parameter, the query uses the default as specified by the QualityRule system parameter. If a quality rule of Extended is specified, any point stored with doubtful OPCQuality will be used to calculate the average, and the point time will therefore be included in the calculation of PercentGood.

```
SELECT DateTime, TagName, CONVERT(DECIMAL(10, 2), Value) AS Value, OPCQuality, PercentGood  
FROM History
```

```
WHERE TagName = 'ReactTemp'  
AND DateTime >= '2005-04-11 11:18:00'  
AND DateTime < '2005-04-11 11:23:00'  
AND wwRetrievalMode = 'Average'  
AND wwCycleCount = 5
```

```
AND wwTimeStampRule = 'Start'
```

The results are:

	Date	TagName	Value	OPCQuality	PercentGood
(cycle 1)	2005-04-11 11:18:00.000	ReactTemp	NULL	0	0.0
(cycle 2)	2005-04-11 11:19:00.000	ReactTemp	70.00	64	77.72
(cycle 3)	2005-04-11 11:20:00.000	ReactTemp	153.99	192	100.0
(cycle 4)	2005-04-11 11:21:00.000	ReactTemp	34.31	192	100.0
(cycle 5)	2005-04-11 11:22:00.000	ReactTemp	134.75	192	100.0

Query 2

This query demonstrates the use of the average retrieval mode in a wide query. Time-weighted average values are returned for the analog tags ReactTemp and ReactLevel, while regular cyclic points are returned for the discrete tag, WaterValve.

```
SELECT * FROM OpenQuery(INSQL,
    'SELECT DateTime, ReactTemp, ReactLevel, WaterValve FROM WideHistory
     WHERE DateTime >= "2004-06-07 08:00"
       AND DateTime < "2004-06-07 08:05"
       AND wwRetrievalMode = "Average"
       AND wwCycleCount = 5
    ')
```

The results are:

Date	ReactTemp	ReactLevel	WaterValve
2004-06-07 08:00:00.000	47.71621	1676.69716	1
2004-06-07 08:01:00.000	157.28076	1370.88097	0
2004-06-07 08:02:00.000	41.33734	797.67296	1
2004-06-07 08:03:00.000	122.99525	1921.66771	0
2004-06-07 08:04:00.000	105.28866	606.40488	1

Average retrieval - initial and final values

If `wwTimeStampRule = END`, the initial value is calculated by performing an average calculation on the cycle leading up to the query start time. No special handling is done for the final value.

If `wwTimeStampRule = START`, the final value is calculated by performing an average calculation on the cycle following the query end time. No special handling is done for the initial value.

Average retrieval - handling NULL values

Gaps introduced by NULL values are not included in the average calculations. The average only considers the time ranges with good values. `TimeGood` indicates the total time per cycle that the tags value was good.

Minimum retrieval

The minimum value retrieval mode returns the minimum value from the actual data values within a retrieval cycle. If there are no actual data points stored on the Historian for a given cycle, nothing is returned. NULL is returned if the cycle contains one or more NULL values.

As in cyclic retrieval, the number of cycles is based on the specified resolution or cycle count. However, minimum retrieval is not a true cyclic mode. Apart from the initial value, all points returned are delta points.

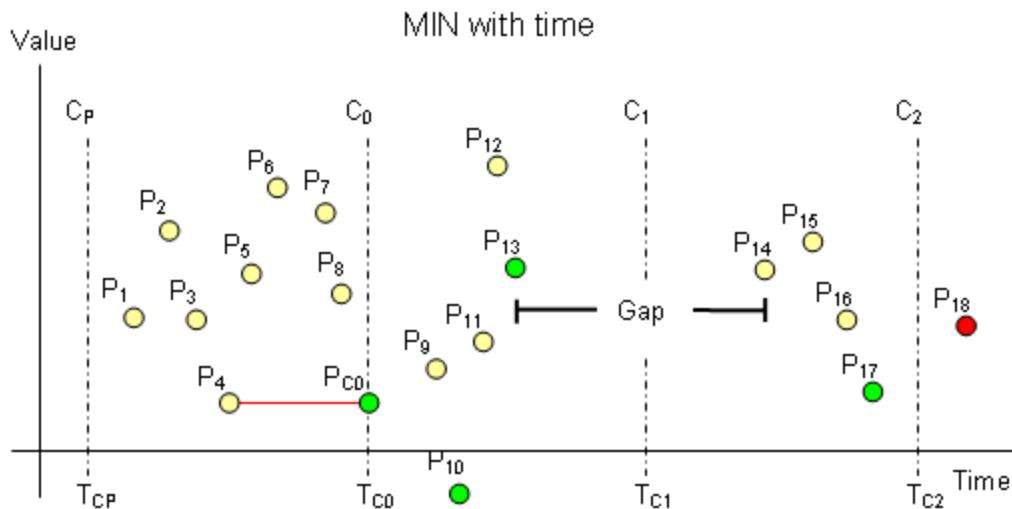
Minimum retrieval only works with analog tags. For all other tags, normal delta results are returned.

All returned values are in chronological order. If multiple points are to be returned for a particular timestamp, they are returned in the order in which the tags were specified in the query. If the minimum value occurs several times in a cycle, the minimum value with the earliest timestamp is returned.

Using the minimum retrieval mode with a cycle count of 1 returns the same results as the SQL Server MIN aggregate; however, the data is returned much faster.

Minimum retrieval - how it works

The following illustration shows how the minimum value is selected for an analog tag.



This example has a start time of T_{C0} and an end time of T_{C2} . The resolution has been set in such a way that the Historian returns data for two complete cycles starting at T_{C0} and T_{C1} , a "phantom" cycle starting at T_{CP} , and an incomplete cycle starting at T_{C2} . The phantom cycle has the same duration as the first cycle in the query period, extending back in time from the query start time.

For the queried tag, a total of 18 points are found throughout the cycles, represented by the markers P_1 through P_{18} . Of these points, 17 represent normal analog values. The point P_{13} represents a NULL due to an I/O Server disconnect, which causes a gap in the data between P_{13} and P_{14} .

The minimum value for the "phantom" cycle starting at T_{CP} is returned as the initial value at T_{C0} . Point P_{18} is not considered at all because it is outside of the query time frame. All other points are considered, but only the points indicated by green markers on the graph are returned (P_{10} , P_{13} , and P_{17}).

In total, four points are returned:

- P_4 as the minimum value of the "phantom" cycle and the initial point
- P_{10} as the minimum value in the first cycle
- P_{13} as the first and only exception occurring in the first cycle
- P_{17} as the minimum value in the second cycle

No points are returned for the incomplete third cycle starting at the query end time, because the tag does not have a point exactly at that time.

If the minimum value of the first cycle is located exactly at the query start time, both this value and the minimum value of the phantom cycle are returned.

Minimum retrieval - supported value parameters

You can use various parameters to adjust which values are returned in minimum retrieval mode. For more information, see the following sections:

- [Cycle Count \(X Values over Equal Time Intervals\) \(wwCycleCount\)](#)
- [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#)

- History Version (wwVersion)
- Quality Rule (wwQualityRule)

Minimum retrieval - query examples

To use the minimum mode, set the following parameter in your query:

```
wwRetrievalMode = 'Min'
```

or

```
wwRetrievalMode = 'Minimum'
```

Minimum retrieval - initial and final values

For analog tags, the minimum value of the tag in the cycle leading up to the query start time is returned with its timestamp changed to the query start time. If there is no point exactly at the "phantom" cycle start time, the point leading up to the phantom cycle is also considered for the minimum calculation.(No adjustments are made to the quality of the initial point even though the timestamp may have been altered.) Apart from the initial value, all points returned are delta points. (For more information on initial values, see [Delta Retrieval - Initial Values.](#))

If a point occurs exactly on the query end time, that point will be returned with the partial cycle bit, 4096, set in quality detail. If there is more than one such point, only the first point will be returned.

Minimum retrieval - handling NULL values and incomplete cycles

The first NULL value in a cycle is returned.

When a minimum value is returned from a cycle that contains gaps (including a gap extended from the previous cycle) or from an incomplete cycle with the query end time located inside of the calculation cycle, the point's quality detail is modified to flag this. This is done by performing a logical OR operation of the value 4096, which indicates a partial cycle, onto the existing quality detail.

As an example of how minimum retrieval mode handles NULLs, consider the following query:

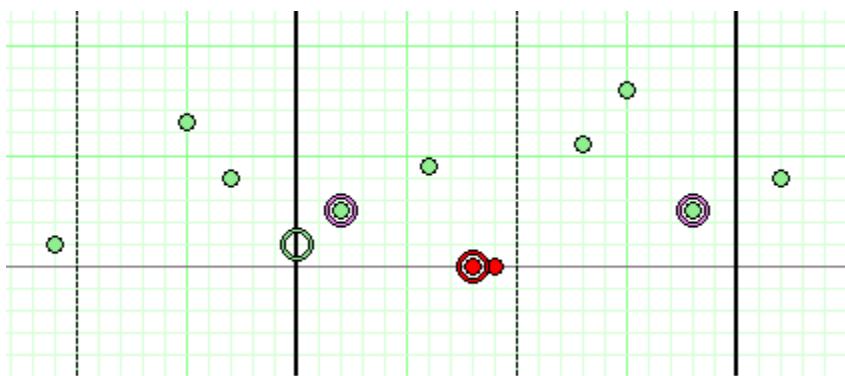
```
SELECT TagName, DateTime, Value, QualityDetail
  FROM History
 WHERE TagName = 'A001'
   AND DateTime >= '2009-09-12 00:20'
   AND DateTime <= '2009-09-12 00:40'
   AND wwResolution = 10000
   AND wwRetrievalMode = 'Minimum'
```

This query can be run against the following sample data:

Tagname	DateTime	Value	QualityDetail
A001	2009-09-12 00:09	0.2	192

A001	2009-09-12 00:15	1.3	192
A001	2009-09-12 00:17	0.8	192
A001	2009-09-12 00:22	0.5	192
A001	2009-09-12 00:26	0.9	192
A001	2009-09-12 00:28	0.0	249
A001	2009-09-12 00:29	0.0	249
A001	2009-09-12 00:33	1.1	192
A001	2009-09-12 00:35	1.6	192
A001	2009-09-12 00:38	0.5	192
A001	2009-09-12 00:42	0.8	192

The following is a graphical representation of the data:



The results are:

Tagname	Datetime	Value	QualityDetail
A001	2009-09-12 00:20	0.2	192
A001	2009-09-12 00:22	0.5	4288
A001	2009-09-12 00:28	NULL	249
A001	2009-09-12 00:38	0.5	4288

The sample data points and the results are mapped on the following chart. Only the data falling between the time start and end marks at 00:20 and 00:40 (shown on the chart as dark vertical lines) are returned by the query. The resolution is set at 10,000 milliseconds.

Because there is no value that matches the start time, an initial value at 00:20 is returned based on the minimum value of the preceding cycle, which is the data point at 00:09. In the two subsequent cycles, the minimum values are at 00:22 and 00:38. The quality for these two values is set to 4288 (4096 + 192). The remaining data points are excluded because they are not minimums. In addition, the first NULL at 00:28 is included, but the second NULL (at 00:29) is not.

Maximum retrieval

The maximum value retrieval mode returns the maximum value from the actual data values within a retrieval cycle. If there are no actual data points stored on the Historian for a given cycle, nothing is returned. NULL is returned if the cycle contains one or more NULL values.

As in cyclic retrieval, the number of cycles is based on the specified resolution or cycle count. However, maximum retrieval is not a true cyclic mode. Apart from the initial value, all points returned are delta points.

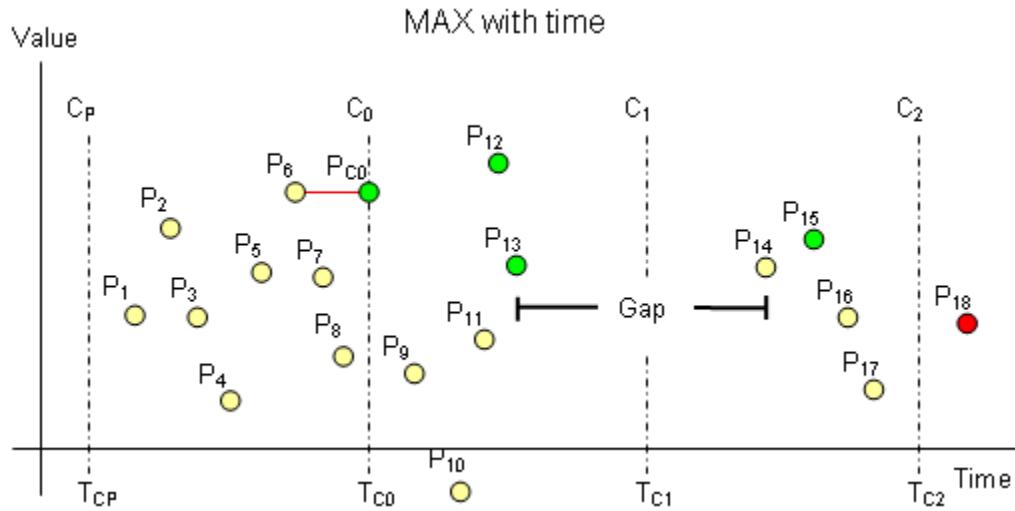
Maximum retrieval only works with analog tags. For all other tags, normal delta results are returned.

All returned values are in chronological order. If multiple points are to be returned for a particular timestamp, they are returned in the order in which the tags were specified in the query. If the maximum value occurs several times in a cycle, the maximum value with the earliest timestamp is returned.

Using the maximum retrieval mode with a cycle count of 1 returns the same results as the SQL Server MAX aggregate; however, the data is returned much faster.

Maximum retrieval - how it works

The following illustration shows how the maximum value is selected for an analog tag.



This example has a start time of T_{C0} and an end time of T_{C2} . The resolution has been set in such a way that the Historian returns data for two complete cycles starting at T_{C0} and T_{C1} , a "phantom" cycle starting at T_{CP} , and an incomplete cycle starting at T_{C2} . The phantom cycle has the same duration as the first cycle in the query period, extending back in time from the query start time.

For the queried tag, a total of 18 points are found throughout the cycles, represented by the markers P_1 through P_{18} . Of these points, 17 represent normal analog values. The point P_{13} represents a NULL due to an I/O Server disconnect, which causes a gap in the data between P_{13} and P_{14} .

The maximum value for the "phantom" cycle starting at T_{CP} is returned as the initial value at T_{C0} . Point P_{18} is not considered at all because it is outside of the query time frame. All other points are considered, but only the points indicated by green markers on the graph are returned (P_{12} , P_{13} , and P_{15}).

In total, four points are returned:

- P_6 as the maximum value of the "phantom" cycle and the initial point

- P₁₂ as the maximum value in the first cycle
- P₁₃ as the first and only exception occurring in the first cycle
- P₁₅ as the maximum value in the second cycle

No points are returned for the incomplete third cycle starting at the query end time, because the tag does not have a point exactly at that time.

If the maximum value of the first cycle is located exactly at the query start time, this value and the maximum value of the phantom cycle are returned.

Maximum retrieval - supported value parameters

You can use various parameters to adjust which values are returned in maximum retrieval mode. For more information, see the following sections:

- Cycle Count (X Values over Equal Time Intervals) ([wwCycleCount](#))
- Resolution (Values Spaced Every X ms) ([wwResolution](#))
- History Version ([wwVersion](#))
- Quality Rule ([wwQualityRule](#))

Maximum retrieval - query examples

To use the maximum mode, set the following parameter in your query:

```
wwRetrievalMode = 'Max'
```

or

```
wwRetrievalMode = 'Maximum'
```

Maximum retrieval - initial and final values

For analog tags, the maximum value of the tag in the cycle leading up to the query start time is returned with its timestamp changed to the query start time. If there is no point exactly at the phantom cycle start time, the point leading up to the phantom cycle is also considered for the maximum calculation. No adjustments are made to the quality of the initial point even though the timestamp may have been altered. Apart from the initial value, all points returned are delta points.

If a point occurs exactly on the query end time, that point is returned with the partial cycle bit, 4096, set in quality detail. If there is more than one such point, only the first point is returned.

Maximum retrieval - handling NULL values and incomplete cycles

The first NULL value in a cycle is returned.

When a maximum value is returned from a cycle that contains gaps (including a gap extended from the previous cycle) or from an incomplete cycle with the query end time located inside of the calculation cycle, the point's quality detail is modified to flag this. This is done by performing a logical OR operation of the value 4096, which indicates a partial cycle, onto the existing quality detail.

As an example of how maximum retrieval mode handles NULLs, consider the following query:

```
SELECT TagName, DateTime, Value, QualityDetail
  FROM History
 WHERE TagName = 'A001'
   AND DateTime >= '2009-09-12 00:20'
   AND DateTime <= '2009-09-12 00:40'
   AND wwResolution = 10000
   AND wwRetrievalMode = 'Maximum'
```

If you run this query against the following sample data:

Tagname	DateTime	Value	QualityDetail
A001	2009-09-12 00:09	0.2	192
A001	2009-09-12 00:15	1.3	192
A001	2009-09-12 00:17	0.8	192
A001	2009-09-12 00:22	0.5	192
A001	2009-09-12 00:26	0.9	192
A001	2009-09-12 00:28	0.0	249
A001	2009-09-12 00:29	0.0	249
A001	2009-09-12 00:33	1.1	192
A001	2009-09-12 00:35	1.6	192
A001	2009-09-12 00:38	0.5	192
A001	2009-09-12 00:42	0.8	192

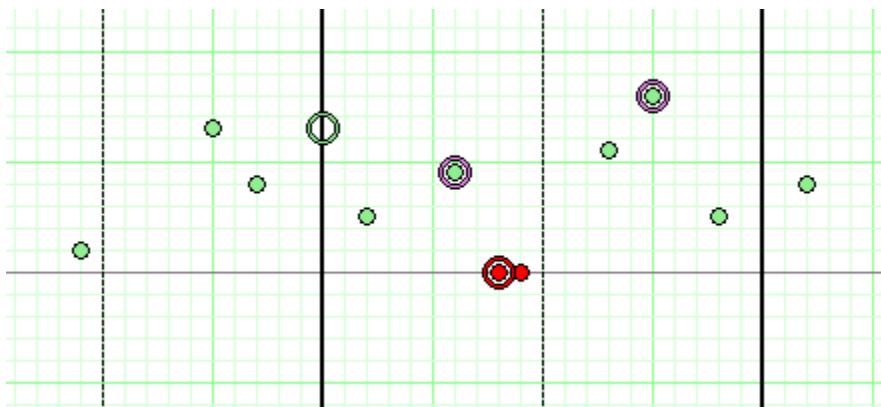
The results are:

Tagname	DateTime	Value	QualityDetail
A001	2009-09-12 00:20	1.3	192
A001	2009-09-12 00:26	0.9	4288

A001	2009-09-12 00:28	NULL	249
A001	2009-09-12 00:35	1.6	4288

The sample data points and the results are mapped on the following chart. Only the data falling between the time start and end marks at 00:20 and 00:40 (shown on the chart as dark vertical lines) are returned by the query. The resolution is set at 10,000 milliseconds.

Because there is no value that matches the start time, an initial value at 00:20 is returned based on the maximum value of the preceding cycle, which is the data point at 00:15. In the two subsequent cycles, the maximum values are at 00:26 and 00:35. The quality for these two values is set to 4288 (4096 + 192). The remaining data points are excluded because they are not maximums. In addition, the first NULL at 00:28 is included, but the second NULL (at 00:29) is not.



Integral retrieval

Integral retrieval calculates the values at retrieval cycle boundaries by integrating the graph described by the points stored for the tag. Therefore, it works much like average retrieval, but it additionally applies a scaling factor. This retrieval mode is useful for calculating volume for a particular tag. For example, if one of your tags represents product flow in gallons per second, integral retrieval allows you to retrieve the total product flow in gallons during a certain time period.

Integral retrieval is a true cyclic mode. It returns one row for each tag in the query for each cycle. The number of cycles is based on the specified resolution or cycle count.

Integral retrieval only works with analog tags. For all other tags, normal cyclic results are returned.

Integral retrieval - how it works

Calculating values for a cycle in integral retrieval is a two-step process:

- First, the Historian calculates the area under the graph created by the data points. This works the same as in average retrieval. For more information, see [Average Retrieval](#).
- After this area has been found, it is scaled using the value of the IntegralDivisor column in the EngineeringUnit table. This divisor expresses the conversion factor from the actual rate to one of units per second.

For example, if the time-weighted average for a tag during a 1-minute cycle is 3.5 liters per second, integral retrieval returns a value of 210 for that cycle (3.5 liters per second multiplied by 60 seconds).

Integral retrieval - supported value parameters

You can use various parameters to adjust which values are returned in integral retrieval mode. For more information, see the following sections:

- [Cycle Count \(X Values over Equal Time Intervals\) \(wwCycleCount\)](#)
- [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#)
- [History Version \(wwVersion\)](#)
- [Interpolation Type \(wwInterpolationType\)](#)
- [Timestamp Rule \(wwTimestampRule\)](#)
- [Quality Rule \(wwQualityRule\)](#)

Integral retrieval - query example

To use the integral retrieval mode, set the following parameter in your query.

```
wwRetrievalMode = 'Integral'
```

Integral retrieval - initial and final values

If `wwTimeStampRule = END`, the initial value is calculated by performing an integral calculation on the cycle leading up to the query start time. No special handling is done for the final value.

If `wwTimeStampRule = START`, the final value is calculated by performing an integral calculation on the cycle following the query end time. No special handling is done for the initial value.

Integral retrieval - handling NULL values

Gaps introduced by NULL values are not included in the integral calculations. The average only considers the time ranges with good values. `TimeGood` indicates the total time per cycle that the tags value was good.

Slope retrieval

Slope retrieval returns the slope of a line drawn through a given point and the point immediately before it, thus expressing the rate at which values change.

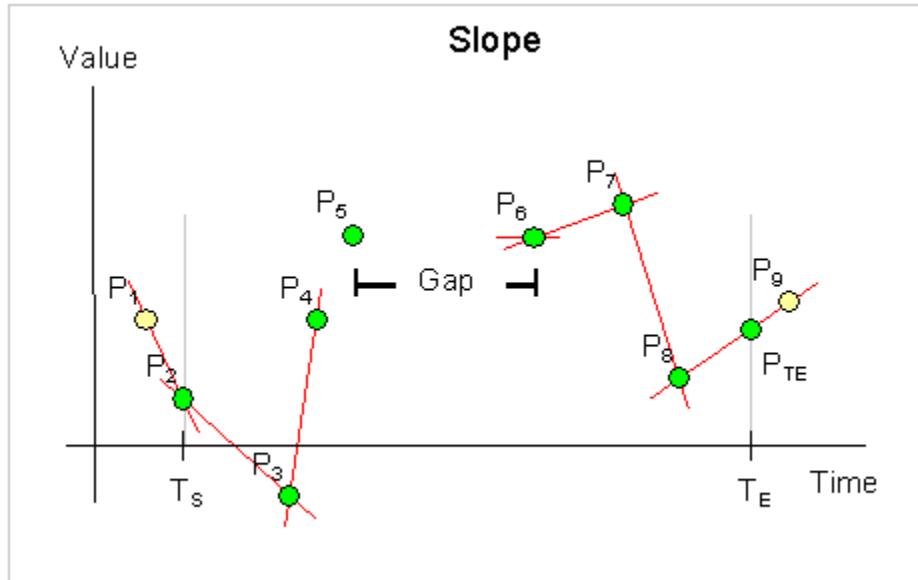
This retrieval mode is useful for detecting if a tag is changing at too great a rate. For example, you might have a temperature that should steadily rise and fall by a small amount, and a sharp increase or decrease could point to a potential problem.

The slope retrieval mode can be considered a delta mode. Apart from the initial value and a value at the query

end time, all returned points are calculated delta points returned with the timestamp of an actual delta point. Slope retrieval only works with analog tags. For all other tags, normal delta results are returned. All returned values are in chronological order. If multiple points are to be returned for a particular timestamp, they are returned in the order in which the tags were specified in the query.

Slope retrieval - how it works

The following illustration shows how the slope is calculated for an analog tag.



This example has a start time of T_S and an end time of T_E .

For the queried tag, a total of nine points are found, represented by the markers P_1 through P_9 . Of these points, eight represent normal analog values. The point P_5 represents a NULL due to an I/O Server disconnect, which causes a gap in the data between P_5 and P_6 .

For every point in the time period, slope retrieval returns the slope of the line going through that point and the point immediately before it. For two points P_1 and P_2 occurring at times T_1 and T_2 , the slope formula is as follows:

$$(P_2 - P_1) / (T_2 - T_1)$$

The difference between T_1 and T_2 is measured in seconds. Therefore, the returned value represents the change in Engineering Units per second.

In this example, point P_2 is located at the query start time, and because there is a prior value (P_1), the slope of the line through both points is calculated and returned at time T_S . Similarly, slopes are calculated to be returned at times T_3 , T_4 , T_7 , and T_8 . The slope is also calculated for the line through P_8 and P_9 , but that value is returned as point P_{TE} at the query end time.

For point P_6 , there is no prior point with which to perform a slope calculation. Instead, the slope of the flat line going through the point (that is, the value 0) is calculated. At the time of P_5 , NULL is returned.

The quality detail and OPC quality returned with a slope point is always directly inherited from the point that also provides the time stamp. In this example, this means that point P_2 provides the qualities for the slope point returned at the query start time, T_S .

Slope retrieval - supported value parameters

You can use various parameters to adjust which values are returned in slope retrieval mode. For more information, see the following sections:

- [History Version \(wwVersion\)](#)
- [Quality Rule \(wwQualityRule\)](#)

Slope retrieval - query example

To use the slope retrieval mode, set the following parameter in your query.

```
wwRetrievalMode = 'Slope'
```

Slope retrieval - initial and final values

An initial value is always generated. If a point is stored exactly at the query start time, the slope is returned as the slope between that point and the previous point. Otherwise, the slope is calculated using the slope of the points before and after the query start time.

A final value is always generated. If a point is stored exactly at the query end time, the slope is returned as the slope between that point and the previous point. Otherwise, the slope is calculated using the slope of the points before and after the query end time.

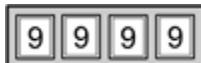
Slope retrieval - handling NULL values

The first NULL following a non-NUL value is returned. Subsequent NULL values are not. If a point is preceded by a NULL, the slope for that point will be zero.

Counter retrieval

Counter retrieval allows you to accurately retrieve the delta change of a tag's value over a period of time even for tags that are reset upon reaching a "rollover value." The rollover value is defined in the Historian for each tag.

This retrieval mode is useful for determining how much of an item was produced during a particular time period. For example, you might have an integer counter that keeps track of how many cartons were produced. The counter has an indicator like this:



The next value after the highest value that can be physically shown by the counter is called the rollover value. In this example, the rollover value is 10,000. When the counter reaches the 9,999th value, the counter rolls back to 0. Therefore, a counter value of 9,900 at one time and a value of 100 at a later time means that you have produced 200 units during that period, even though the counter value has dropped by 9,800 (9,900 minus 100). Counter retrieval allows you to handle this situation and receive the correct value. For each cycle, the counter

retrieval mode shows the increase in that counter during the cycle, including rollovers.

Counter retrieval also works with floating point counters, which is useful for flow meter data. Similar to the carton counter, some flow meters "roll over" after a certain amount of flow accumulates. For both examples, the need is to convert the accumulating measure to a "delta change" value over a given period.

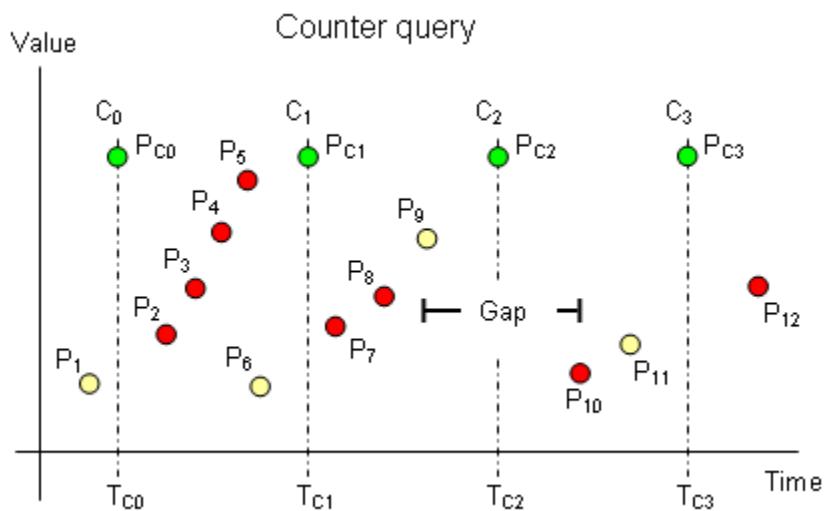
Counter retrieval is a true cyclic mode. It returns one row for each tag in the query for each cycle. The number of cycles is based on the specified resolution or cycle count.

The counter algorithm is only applied to analog tags and to discrete tags. For integer analog tags, the result will be an integer returned as a float data type. For a real analog tag, the rollover value and the result may be real values and can include fractional values. If a query contains tags of other types, then no rows are returned for those tags. For discrete tags, the rollover value is assumed to be 2.

The rules used to determine the OPCQuality returned with a counter value are the same as for a time weighted average query. For more information, see [Quality Rule \(wwQualityRule\)](#). When a rollover has occurred in the calculation cycle, a special quality detail of 212 is returned in all non-NULL cases.

Counter retrieval - how it works

The following illustration shows how the counter algorithm determines the count for an analog tag.



This example has a start time of T_{C0} and an end time of T_{C3} . The resolution has been set in such a way that the Historian returns data for three complete cycles starting at T_{C0} , T_{C1} , and T_{C2} , and an incomplete cycle starting at T_{C3} .

For the queried tag, a total of twelve points are found throughout the cycles represented by the markers P_1 through P_{12} . Of these points, eleven represent normal analog values. The point P_9 represents a NULL due to an I/O Server disconnect, which causes a gap in the data between P_9 and P_{10} . Point P_{12} is not considered because it is outside of the query time frame.

All points are considered in the counter calculation, but only the yellow ones are actually used to determine which values to return to the client. The returned points are P_{C0} , P_{C1} , P_{C2} and P_{C3} , shown in green at the top to indicate that there is no simple relationship between them and any of the actual points.

All cycle values are calculated as the delta change between the cycle time in question and the previous cycle time, taking into account the number of rollovers between the two points in time. The counter algorithm

assumes that a rollover occurred if the current value is lower than the previous value. The initial value at the query start time (P_{C1}) is calculated the same way, only based on a phantom cycle before the query start time.

For example, the formula to calculate P_{C1} is as follows:

$$P_{C1} = n * V_R + P_6 - P_1$$

where:

n = the number of rollovers that have occurred during the cycle

V_R = the rollover value for the tag

If either n or V_R are equal to zero, P_{C1} is simply the difference between the values P_1 and P_6 .

In the case of cycle C_2 , there is no value at the cycle time, so the NULL value represented by point P_9 is returned.

In the case of cycle C_3 , a NULL is again returned, because there is no counter value at the previous cycle boundary to use in the calculation. There must be a full cycle of values in order for the counter to be calculated.

If a gap is fully contained inside a cycle, and if points occur within the cycle on both sides of the gap, then a counter value is returned, even though it may occasionally be erroneous. Zero or one rollovers are assumed, even though the counter might have rolled over multiple times.

Calculations for a manually reset counter

If you have a counter that you typically reset manually before it rolls over, you must set the rollover value for the tag to 0 so that the count is simply how much change occurred since the manual reset.

For example, assume that you have the following data values for five consecutive cycle boundaries, and that the value 0 occurs as the first value within the last cycle:

100, 110, 117, 123, 3

If you set the rollover value to 0, the counter retrieval mode assumes that the 0 following the value 123 represents a manual reset, and returns a value of 3 for the last cycle, which is assumed to be the count after the manual reset. The value 0 itself does not contribute 1 to the counter value in this case.

If the rollover value is instead set to 200, then the counter retrieval mode assumes that the value 0 represents a normal rollover, and a count of 80 is calculated and returned ($200 - 123 + 3$). In this case, the value 0 contributes 1 to the counter value, and that is the change from the value 199 to the value 200.

Counter retrieval - supported value parameters

You can use various parameters to adjust which values are returned in integral retrieval mode. For more information, see the following sections:

- [Cycle Count \(X Values over Equal Time Intervals\) \(wwCycleCount\)](#)
- [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#)
- [History Version \(wwVersion\)](#)
- [Timestamp Rule \(wwTimestampRule\)](#)
- [Quality Rule \(wwQualityRule\)](#)

Counter retrieval - initial and final values

An initial value is returned using the period leading up to the query start time.

A data point that has a cycle time is used to generate the counter value for its preceding cycle. A NULL point with cycle time will cause the preceding cycle to end in a gap and the following cycle to start with a gap.

Counter retrieval - handling NULL values

If `wwQualityRule` is configured as `OPTIMISTIC`, NULL data points will not be used in calculation. 0.0 will be used as the starting base value for the query unless the query data starts with a NULL. If the query starts with a NULL, the value change for the cycle is calculated from the first actual value in the cycle, rather than 0.

Otherwise, if any points considered in a cycle have `UNCERTAIN` quality, the result for that row will also have `UNCERTAIN` quality. Any cycle that starts or ends in a gap will have a quality detail of 65536.

The quality detail of `DOUBTFUL` will be used with the counter result for the cycles, if a NULL point is considered for the cycle and the counter result is not NULL.

Counter retrieval - handling illegal values

If the configured rollover value is larger than 0.0, then the data points whose values are greater than or equal to the rollover value causes the counter value for the cycle to be set to 0.0, with `qdIO_FILTEREDPOINT` applied to the quality detail.

Similarly, if any data point with a value less than 0.0 is found in a cycle, the counter value for the cycle is set to 0.0, with `qdIO_FILTEREDPOINT` applied to the quality detail.

Counter retrieval - query example

To use the counter mode, set the following parameter in your query.

```
wwRetrievalMode = 'Counter'
```

In the following example, the rollover value for the `SysTimeSec` system tag is set to 0. In a two-minute time span, the `SysTimeSec` tag increments from 0 to 59 two times. The following query returns the total count within the two-minute time span. The QualityDetail of 212 indicates that a counter rollover occurred during the query time range.

```
select DateTime, TagName, Value, Quality, QualityDetail as QD from History
  where TagName = 'systimesec'
    and DateTime >= '2009-08-13 1:00'
    and DateTime < '2009-08-13 1:02'
    and wwRetrievalMode = 'counter'
    and wwCycleCount = 1
```

The results are:

Date/Time	Tag Name	Value	Quality	QD
2009-08-13 01:00:00.0000000	SysTimeSec	120	0	212

ValueState retrieval

ValueState retrieval returns information on how long a tag has been in a particular value state during each retrieval cycle. That is, a time-in-state calculation is applied to the tag value.

This retrieval mode is useful for determining how long a machine has been running or stopped, how much time a process spent in a particular state, how long a valve has been open or closed, and so on. For example, you might have a steam valve that releases steam into a reactor, and you want to know the average amount of time the valve was in the open position during the last hour. ValueState retrieval can return the shortest, longest, average, or total time a tag spent in a state, or the time spent in a state as a percentage of the total cycle length.

When you use ValueState retrieval for a tag in a trend chart, you must specify a single value state for which to retrieve information. ValueState retrieval then returns one value for each cycle—for example, the total amount of time that the valve was in the "open" state during each 1-hour cycle. This information is suitable for trend display.

If you *don't* specify a state, ValueState retrieval returns one row of information for each value that the tag was in during each cycle. For example, this would return not only the time a valve was in the "open" state, but also the time it was in the "closed" state. This information is not suitable for meaningful display in a regular trend. You can, however, retrieve this type of information in a query and view it as a table.

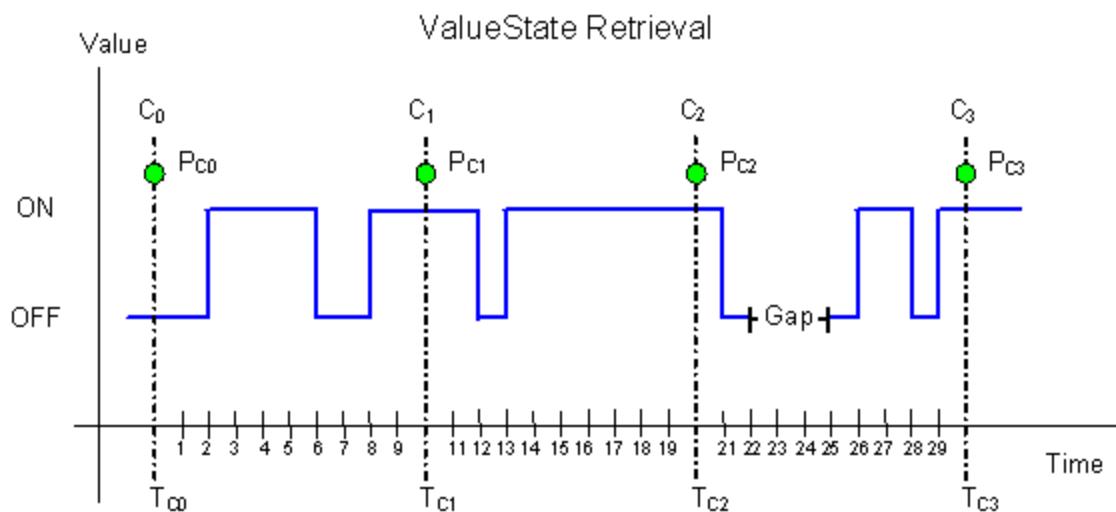
ValueState retrieval works with integer, discrete, string, and state summary tags. For other types of tags, no rows are returned. NULL values are treated like any other distinct state.

The values returned at the query start time are the result of applying the algorithm to a "phantom" cycle preceding the query range. It is assumed that the tag value at the start of the cycle is located at that point in time.

To specify the type of calculation, set the `wwStateCalc` parameter in the query. For more information, see [State Calculation \(wwStateCalc\)](#).

ValueState retrieval - how it works

The following illustration shows how ValueState retrieval returns values for a discrete tag.



This example has a start time of T_{c0} and an end time of T_{c3}. The resolution has been set in such a way that the Historian returns data for three complete cycles starting at T_{c0}, T_{c1}, and T_{c2}, and an incomplete cycle starting at T_{c3}. Time is measured seconds.

A gap in the data occurs in the third cycle due to an I/O Server disconnect.

The state calculation is based on each cycle, and the values returned at the query start time are not regular initial values, but are the resulting values that occur after applying the algorithm to the last cycle preceding the query range. The returned points are P_{c0}, P_{c1}, P_{c2} and P_{c3}, shown in green at the top to indicate that there is no simple relationship between the calculated values and any of the actual points.

Assume the query is set so that the total time (wwStateCalc = 'Total') in the two states are returned. The timestamping is set to use the cycle end time.

- For T_{c0}, the query returns two rows (one for the "on" state and one for the "off" state), calculated as a result of the "phantom" cycle that precedes the query start time. The values have a timestamp of the query start time.
- For T_{c1}, one row is returned for the "on" state. The "on" state occurred twice during the cycle--one time for four seconds and again for two seconds before the cycle boundary, and the total time returned is six seconds. The state was "off" twice during the cycle for a total time of four seconds, and one row is returned with that value.
- For T_{c2}, one row is returned for the "on" state, and one row is returned for the "off" state. The "on" state occurred for a total of nine seconds between the cycle boundaries, and the "off" state occurred for a total of one second.
- For T_{c3}, one row is returned for the "on" state, and one row is returned for the "off" state. The "on" state occurred for a total of four seconds between the cycle boundaries, and the "off" state occurred for a total of three seconds. An additional row is returned for the NULL state occurring as a result of the I/O Server disconnect.

Using the same data, if you queried the **total contained** time for the states, the following is returned:

- For T_{c0}, the query returns two values (one for the "on" state and one for the "off" state), calculated as a result of the "phantom" cycle the precedes the query start time. The value has a timestamp of the query start time.

- For T_{C1}, one row is returned for the "on" state, and one row is returned for the "off" state. The "on" state occurred one time for four seconds within the cycle. The two seconds of "on" time that crosses the cycle boundary does not contribute to the total time. The state was "off" one time during the cycle for two seconds completely within the cycle boundary.
- For T_{C2}, the state was not "on" for any contained time between the cycle. Both occurrences of the "on" state cross over a cycle boundary, so no rows are returned for this state. One row is returned for the "off" state. The state was "off" one time during the cycle for one seconds completely within the cycle boundary.
- For T_{C3}, one row is returned for the "on" state, and one row is returned for the "off" state. The state was "on" for a single contained time of two seconds between the cycle boundaries. The state was "off" three times during the cycle for three seconds completely within the cycle boundary. An additional row is returned for the NULL state occurring as a result of the I/O Server disconnect. The state was NULL for a total of three seconds. The I/O Server disconnect that "disrupted" the off state is treated as its own state, thereby changing what would have been a single "off" state instance of five seconds into two instances of the "off" state for one second each.

ValueState retrieval - supported value parameters

You can use various parameters to adjust which values are returned in ValueState retrieval mode. For more information, see the following sections:

- [Cycle Count \(X Values over Equal Time Intervals\) \(wwCycleCount\)](#)
- [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#)
- [History Version \(wwVersion\)](#)
- [Timestamp Rule \(wwTimestampRule\)](#)
- [Quality Rule \(wwQualityRule\)](#)
- [State Calculation \(wwStateCalc\)](#)

ValueState retrieval - query examples

To use theValueState retrieval mode, set the following parameter in your query.

```
wwRetrievalMode = 'ValueState'
```

To specify the type of aggregation, set the wwStateCalc parameter in the query, such as:

```
wwStateCalc = 'Total'
```

Be sure that you use the "<=" operator for ending date/time.

ValueState retrieval - initial and final values

The values returned at the query start time are the result of applying the algorithm to the last cycle preceding the query range.

ValueState retrieval - handling NULL values

NULLs are considered a state and are reported along with the other states.

RoundTrip retrieval

RoundTrip retrieval is very similar to ValueState retrieval in that it performs calculations on state occurrences in the within a cycle period you specify. However, ValueState retrieval uses the time spent in a certain state for the calculation, and RoundTrip retrieval uses the time between consecutive leading edges of the same state for its calculations.

You can use the RoundTrip retrieval mode for increasing the efficiency of a process. For example, if a process produces one item per cycle, then you would want to minimize the time lapse between two consecutive cycles.

The RoundTrip mode returns a rows for each state in any given cycle. RoundTrip retrieval only works with integer analog tags, discrete tags, and string tags. If real analog tags are specified in the query, then no rows are returned for these tags. RoundTrip retrieval is not applied to state summary or analog summary tags. NULL values are treated as any other distinct value and are used to analyze the round trip for disturbances.

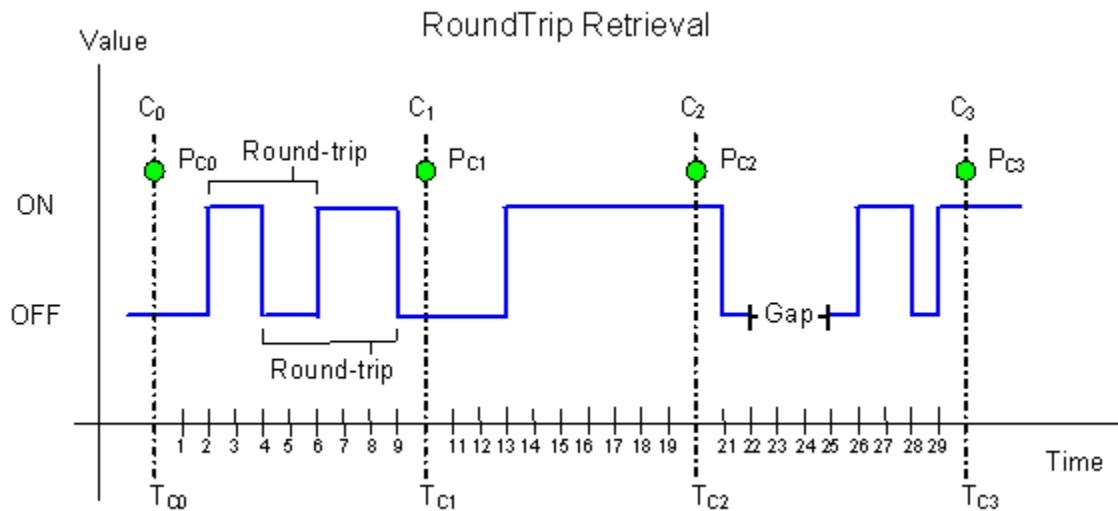
RoundTrip retrieval is supported for the History and StateWideHistory tables.

Any point on the boundary of the end cycle will be considered to the next cycle. The point on the boundary of the end query range is not counted in the calculation except that it is used to indicate that the previous state is a contained state.

If no roundtrip state is found within the cycle for a supported tag, a NULL StateTime value is returned. If there is no valid point prior to the phantom cycle, a NULL state is returned for the phantom cycle.

RoundTrip retrieval - how it works

The following illustration shows how RoundTrip retrieval returns values for a discrete tag.



This example has a start time of T_{C0} and an end time of T_{C3} . The resolution has been set in such a way that the Historian returns data for three complete cycles starting at T_{C0} , T_{C1} , and T_{C2} , and an incomplete cycle starting at

Tc3. Time is measured seconds.

A gap in the data occurs in the third cycle due to an I/O Server disconnect.

The state calculation is based on each cycle, and the values returned at the query start time are not regular initial values, but are the resulting values that occur after applying the algorithm to the last cycle preceding the query range. The returned points are P_{c0}, P_{c1}, P_{c2} and P_{c3}, shown in green at the top to indicate that there is no simple relationship between the calculated values and any of the actual points.

Assume the query is set so that the total contained time in the two states are returned. The timestamping is set to use the cycle end time. The RoundTrip retrieval mode returns values for states that are completely contained within the cycle boundaries. The following is returned:

- For T_{c0}, the query returns two values (one for the "on" state and one for the "off" state), calculated as a result of the "phantom" cycle that precedes the query start time. The value has a timestamp of the query start time.
- For T_{c1}, one row is returned for the "on" state, and one row is returned for the "off" state. The round-trip for the "on" state occurred one time for four seconds completely within the cycle boundary. The round-trip for the "off" state occurred one time during the cycle for five seconds.
- For T_{c2}, a round-trip did not occur for either state within the cycle boundaries. One NULL row is returned for this cycle.
- For T_{c3}, one row is returned for the "on" state, and one row is returned for the "off" state. The state was "on" for a single contained time of two seconds between the cycle boundaries. The state was "off" one time during the cycle for one second completely within the cycle boundary. An additional row is returned for the NULL state occurring as a result of the I/O Server disconnect.
- For T_{c3}, one row is returned for the "on" state, and one row is returned for the "off" state. The state was "on" for a single contained time of three seconds between the cycle boundaries. One row is returned for the "off" state for a total contained time of seven seconds. (The first round-trip for the "off" state includes the I/O Server disconnect for a length of four seconds. The second round-trip has a length of three seconds.) An additional row is returned for the NULL state occurring as a result of the I/O Server disconnect, and the returned value is NULL because there is no round-trip during the cycle for it. The I/O Server disconnect that "disrupted" the off state is treated as its own state, thereby changing what would have been a single "off" state instance of five seconds into two instances of the "off" state for one second each.

RoundTrip retrieval - supported value parameters

You can use various parameters to adjust the values that must be returned in the RoundTrip retrieval mode. For more information, see the following sections:

- [Timestamp Rule \(wwTimestampRule\)](#)
- [Quality Rule \(wwQualityRule\)](#)
- [State Calculation \(wwStateCalc\)](#)

RoundTrip retrieval - query examples

To use the RoundTrip retrieval mode, set the following parameter in your query:

```
wwRetrievalMode = 'RoundTrip'
```

The following queries compare the results between ValueState retrieval and RoundTrip retrieval.

This first ValueState retrieval query returns the average amount of time that the 'Reactor1OutletValve' tag is in "on" state and the average amount of time it is in the "off" state for a single cycle. Any state changes that occur across the cycle boundaries are not included.

```
SELECT DateTime, vValue, StateTime
  FROM History
 WHERE TagName IN ('Reactor1OutletValve')
   AND DateTime >= '2009-09-16 12:35:00'
   AND DateTime <= '2009-09-16 12:55:00'
   AND wwRetrievalMode = 'ValueState'
   AND wwStateCalc = 'AvgContained'
   AND wwCycleCount = 1
```

The results are:

DateTime	vValue	StateTime
2009-09-16 12:35:00.0000000	0	215878
2009-09-16 12:35:00.0000000	1	61729
2009-09-16 12:55:00.0000000	1	62827.5
2009-09-16 12:55:00.0000000	0	212856

The first two rows are for the "phantom" cycle leading up to the query start time and have a timestamp of the query start time.

The second two rows show the average amount of time for each state and have a timestamp of the query end time (the default).

Compare these results to same basic query that instead uses RoundTrip retrieval:

```
SELECT DateTime, vValue, StateTime
  FROM History
 WHERE TagName IN ('Reactor1OutletValve')
   AND DateTime >= '2009-09-16 12:35:00'
   AND DateTime <= '2009-09-16 12:55:00'
   AND wwRetrievalMode = 'RoundTrip'
   AND wwStateCalc = 'AvgContained'
   AND wwCycleCount = 1
```

DateTime	vValue	StateTime
2009-09-16 12:35:00.0000000	1	277607
2009-09-16 12:35:00.0000000	0	278580
2009-09-16 12:55:00.0000000	0	275683.5
2009-09-16 12:55:00.0000000	1	273845

RoundTrip retrieval - initial and final values

The values returned at the query start time are the result of applying the algorithm to the last cycle preceding the query range.

RoundTrip retrieval - handling NULL values

Like in the ValueState retrieval mode, the NULL state is treated as a valid distinct value. This allows you to analyze round trips for disturbances.

Predictive retrieval (wwFilter)

AVEVA Historian supports predictive retrieval. Beginning with AVEVA Historian 2014 R2 Patch 01, the historian can return predictive data based on a "simple linear regression" (SLR) algorithm. More capabilities will be added in future releases.

With AVEVA Historian, you can create a query based on data you have stored to predict additional values in a trend. Historian returns predictive data based on a "simple linear regression" (SLR) algorithm.

For example, based on your currently stored values, you could use the predictive retrieval feature to help predict if a certain production target will be met by the end of the shift. Or, if the Historian loses communication with the data source, you could use predictive retrieval to determine whether and when a tank is likely to become empty.

You can predict:

- Values in between other values.
- Values that extend beyond stored values.

For example, suppose you already captured data for a tag with timestamps up to 3 p.m. on a certain day, but not for the rest of the shift, which ran until 5 p.m., because of a power cut. With predictive retrieval, you can view the interpolated results based between 3 p.m. and 5 p.m. These results are based on the data you received through 3 p.m.

The following is an example of a query that retrieves stored values and reports both those values and additional predictive data:

```
SELECT DateTime, Value, wwFilter
FROM History
WHERE TagName = 'Tag1'
    AND DateTime >= '2014-01-01 0:00:00.000'
    and DateTime < '2014-01-01 1:00:00.000'
    and wwFilter = 'SLR()'
```

In this example, "SLR" stands for "simple linear regression," the algorithm used by AVEVA Historian to analyze currently stored values and predict other values within the detected trend.

Bounding value retrieval

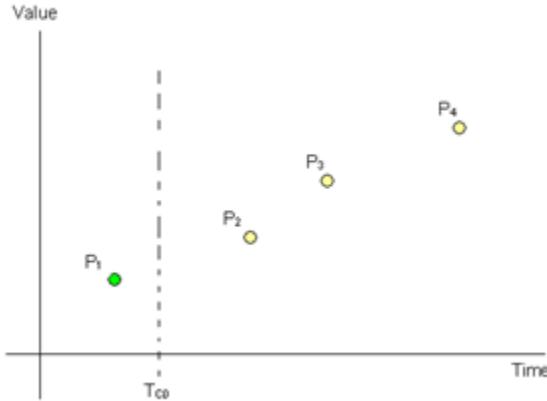
The bounding value retrieval mode returns either the start bound point or the end bound point for a requested

point in time. For a start bound point, Historian retrieves the first value on or before the requested date/time. For an end bound point, Historian retrieves the first value after the requested date/time.

If no time is specified, Historian returns the bounding point at the current time.

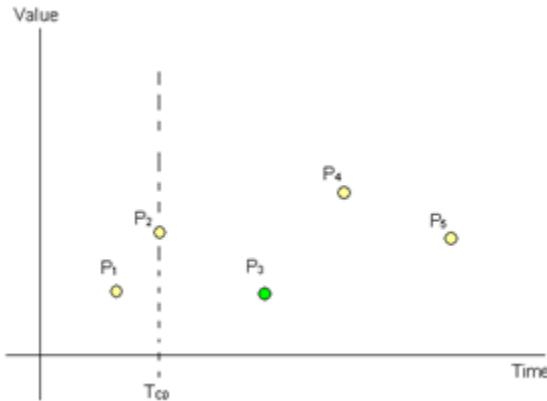
Bounding value retrieval - how it works

The following illustration shows how bounding value retrieval returns a start bound point:



In this case, Historian retrieves the first point on or before the datetime requested in the query. The line (T_{c0}) is the timestamp for which the start bound point is requested. P_1 is returned because that is the start or first point for the query date time.

Historian can also use bounding value retrieval to return an end bound point, as in the following illustration:



In this case, Historian returns first point after the datetime requested in the query. T_{c0} is the timestamp for which the end bound point is requested and P_3 is returned as the ending bound point because this is the first point after the query date time.

Bounding value retrieval - query examples

You can use the bounding value retrieval mode to return a start bound point or an end bound point for a specified date and time. If no time is specified, Historian returns the bounding point at the current time.

To return a start bound point, set the following parameter in your query.

```
wwRetrievalMode = 'StartBound'
```

To return an end bound point, set the following parameter in your query.

```
wwRetrievalMode = 'EndBound'
```

Example 1 - Retrieve start bound point

```
select DateTime,TagName,Value
where TagName 'Plant2.R31.BatchNum'
and wwRetrievalMode = 'StartBound'
and DateTime >= '2019-04-24 12:00:00'
```

The results are:

DateTime	TagName	Value
2019-04-24 11:53:08.5430000	Plant2.R31.BatchNum	912

Example 2 - Retrieve end bound point

```
select DateTime,TagName,Value
where TagName in 'Plant2.R31.BatchNum'
and wwRetrievalMode = 'EndBound'
and DateTime >= '2019-04-24 12:00:00'
```

The results are:

DateTime	TagName	Value
2019-04-24 14:11:13.3840000	Plant2.R31.BatchNum	926

Understanding retrieval options

In all retrieval modes, you can adjust which values the Historian returns by specifying retrieval options. The retrieval options are implemented as special parameters that you set as part of the retrieval query. This section explains each of these options. For an overview of which options apply to which retrieval modes, see [Which Options Apply to Which Retrieval Modes?](#).

Which options apply to which retrieval modes?

The following table shows which retrieval options apply to which modes. If you specify an option in a mode where it isn't applicable, the option is ignored.

	Cycle Count (X Values over Equal Time Intervals) (wwCycleCount)	Resolution (Values Space d Every X ms) (wwResolution)	Time Dead band (wwTimeDeadband)	Value Dead band (wwValueDeadband)	Histor y Version (wwVersion)	Interpolation Type (wwInterpolationType)	Times tamp Rule (wwTimestampRule)	Quali ty Rule (wwQualityRule)	State Calcul ation (wwStateCalc)	Analo g Value Filteri ng (wwFiltering)	Edge detec tion for event s (wwEdgeDetection)	Predi ctive retrie val (wwFilter)*	Expres sions (wwExpression)
Cyclic Retrieval	Y	Y			Y		Y*			Y	Y		Y
Delta Retrieval			Y	Y	Y					Y	Y	Y	Y
Full Retrieval					Y					Y	Y	Y	Y
Interpolated Retrieval	Y	Y			Y	Y	Y	Y		Y	Y		
"Best fit" retrieval	Y	Y			Y	Y		Y		Y	Y	Y	Y
Average Retrieval	Y	Y			Y	Y	Y	Y		Y	Y		
Minimum Retrieval	Y	Y			Y			Y		Y	Y		
Maxi	Y	Y			Y			Y		Y	Y		

Mum Retrieval													
Integral Retrieval	Y	Y			Y	Y	Y	Y		Y	Y		
Slope Retrieval					Y			Y		Y	Y		
Counter Retrieval	Y	Y			Y		Y	Y		Y			
Value State Retrieval	Y	Y			Y		Y	Y	Y	Y			
RoundTrip Retrieval	Y	Y			Y		Y	Y	Y	Y			
Bounding value retrieval	Y	Y											

* - only on AVEVA Historian 9.0 and later

** - only AVEVA Historian 2014 R2 P01 and later

Using retrieval options in a Transact-SQL statement

You can retrieve data in the Historian extension tables using normal Transact-SQL code, as well as the specialized SQL time domain extensions provided by the Historian. The Historian extensions provide an easy way to query time-based data from the history tables. They also provide additional functionality not supported by Transact-SQL.

Note: The wwParameters extension is reserved for future use. The wwRowCount parameter is still supported, but is deprecated in favor of wwCycleCount.

The extensions are implemented as "virtual" columns in the extension tables. When you query an extension table, you can specify values for these column parameters to manipulate the data that will be returned. You will need to specify any real-time extension parameters each time that you execute the query.

For example, you could specify a value for the wwResolution column in the query so that a resolution is applied

to the returned data set:

```
SELECT DateTime, Value
  FROM History
 WHERE TagName = 'SysTimeSec'
   AND DateTime >= '2001-12-02 10:00:00'
   AND DateTime <= '2001-12-02 10:02:00'
   AND Value >= 50
   AND wwResolution = 10
   AND wwRetrievalMode = 'cyclic'
```

Although the Microsoft SQL Server may be configured to be case-sensitive, the values for the virtual columns in the extension tables are always case-insensitive.

Note: You cannot use the IN clause or OR clause to specify more than one condition for a time domain extension. For example, "wwVersion IN ('original', 'latest')¹" and "wwRetrievalMode = 'Delta' OR wwVersion = 'latest'" are not supported.

For general information on creating SQL queries, see your Microsoft SQL Server documentation.

Cycle count (X values over equal time intervals) (wwCycleCount)

In retrieval modes that use cycles, the cycle count determines the number of cycles for which data is retrieved. The number of actual return values is not always identical with this cycle count. In "truly cyclic" modes (Cyclic, Interpolated, Average, and Integral), a single data point is returned for every cycle boundary. However, in other cycle-based modes (Best Fit, Minimum, Maximum, Counter, ValueState, and RoundTrip), multiple or no data points may be returned for a cycle, depending on the nature of the data.

The length of each cycle (the "resolution" of the returned values) is calculated as follows:

$$D_C = D_Q / (n - 1)$$

Where D_C is the length of the cycle, D_Q is the duration of the query, and n is the cycle count.

Instead of specifying a cycle count, you can specify the resolution. In that case, the cycle count is calculated based on the resolution and the query duration. For more information, see [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#).

This option is relevant in the following retrieval modes:

- [Cyclic Retrieval](#)
- [Interpolated Retrieval](#)
- ["Best fit" retrieval](#)
- [Average Retrieval](#)
- [Minimum Retrieval](#)
- [Maximum Retrieval](#)
- [Integral Retrieval](#)
- [Counter Retrieval](#)
- [ValueState Retrieval](#)
- [RoundTrip Retrieval](#)

The application of the cycle count also depends on whether you are querying a wide table. If you are querying the History table, the cycle count determines the number of rows returned per tag. For example, a query that applies a cycle count of 20 to two tags will return 40 rows of data (20 rows for each tag). If you are querying a

WideHistory table, the cycle count specifies the total number of rows to be returned, regardless of how many tags were queried. For example, a query that applies a cycle count of 20 to two tags returns 20 rows of data.

Values chosen:

- If wwResolution and wwCycleCount are not specified, then a default of 100 cycles are chosen.
- If wwResolution and wwCycleCount are set to 0, then a default of 100000 cycles are chosen.
- If wwResolution and wwCycleCount are both set, then wwCycleCount is ignored.
- If wwCycleCount is specified and is less than 0, then a default of 100 cycles are chosen.
- For ValueState retrieval, if the start time of the cycle is excluded, no states are returned for the first cycle.
- For ValueState retrieval, if the end time of the cycle is excluded, no states are returned for the last cycle.

For more information, see [Understanding retrieval options](#).

Resolution (values spaced every X ms) (wwResolution)

In retrieval modes that use cycles, the resolution is the sampling interval for retrieving data, that is, the length of each cycle.

The number of cycles, therefore, depends on the time period and the resolution:

$$\text{number of cycles} = \text{time period} / \text{resolution}$$

The number of actual return values is not always identical with this cycle count. In "truly cyclic" modes (Cyclic, Interpolated, Average, and Integral), a single data point is returned for every cycle boundary. However, in other cycle-based modes (Best Fit, Minimum, Maximum, Counter, and ValueState), multiple or no data points may be returned for a cycle, depending on the nature of the data.

Note: The rowset is guaranteed to contain one row for each tag in the normalized query at every resolution interval, regardless of whether a physical row exists in history at that particular instance in time. The value contained in the row is the last known physical value in history, at that instant, for the relevant tag.

Instead of specifying a resolution, you can specify the cycle count directly. In that case, the resolution is calculated based on the cycle count and the query duration. For more information, see [Cycle Count \(X Values over Equal Time Intervals\) \(wwCycleCount\)](#).

This option is relevant in the following retrieval modes:

- [Cyclic Retrieval](#)
- [Interpolated Retrieval](#)
- ["Best fit" retrieval](#)
- [Average Retrieval](#)
- [Minimum Retrieval](#)
- [Maximum Retrieval](#)
- [Integral Retrieval](#)
- [Counter Retrieval](#)
- [ValueState Retrieval](#)
- [RoundTrip Retrieval](#)

For delta retrieval, you can achieve similar results by using a time deadband. For more information, see [Time](#)

Deadband (wwTimeDeadband).

Resolution - query examples

The following query returns rows that are spaced at 2 sec (2000 msec) intervals over the requested time period. Data is retrieved cyclically.

```
SELECT DateTime, TagName, Value
  FROM History
 WHERE TagName IN ('SysTimeMin','SysTimeSec')
   AND DateTime >= '2001-12-09 11:35'
   AND DateTime <= '2001-12-09 11:36'
   AND wwRetrievalMode = 'Cyclic'
   AND wwResolution = 2000
```

The results are:

DateTime	TagName	Value
2001-12-09 11:35:00.000	SysTimeMin	35
2001-12-09 11:35:00.000	SysTimeSec	0
2001-12-09 11:35:02.000	SysTimeMin	35
2001-12-09 11:35:02.000	SysTimeSec	2
2001-12-09 11:35:04.000	SysTimeMin	35
2001-12-09 11:35:04.000	SysTimeSec	4
2001-12-09 11:35:06.000	SysTimeMin	35
.		
.		
.		
2001-12-09 11:35:54.000	SysTimeMin	35
2001-12-09 11:35:54.000	SysTimeSec	54
2001-12-09 11:35:56.000	SysTimeMin	35
2001-12-09 11:35:56.000	SysTimeSec	56
2001-12-09 11:35:58.000	SysTimeMin	35
2001-12-09 11:35:58.000	SysTimeSec	58
2001-12-09 11:36:00.000	SysTimeMin	36
2001-12-09 11:36:00.000	SysTimeSec	0

About "phantom" cycles

The phantom cycle is the name given to the cycle that leads up to the query start time. It is used to calculate which initial value to return at the query start time for all retrieval modes. Some retrieval modes use the phantom cycle to simply find the last known value prior to the query start time, whereas other retrieval modes use the entire cycle to calculate aggregates. The different uses of the phantom cycle can be seen in the following table.

Simple use of phantom cycle	Cycles not defined, but similar simple use of time before query start time	Phantom cycle used to calculate aggregates
Cyclic	Delta	Min
Interpolated	Full	Max
Best Fit	Slope	Average
		Integral
		Counter
		ValueState
		RoundTrip

It's common to expect a single aggregate row returned for a particular time interval. You can accomplish this in several ways.

The following example is querying for hourly averages. It returns a single row time stamped at the query start time. If the query included the query end point by including an equal sign for it, the query would also have returned an additional row at the query end time.

```
SELECT DateTime, Value, Quality, QualityDetail, OPCQuality
  FROM History
 WHERE TagName IN ('SysTimeSec')
   AND DateTime >= '2009-10-16 08:00:00'
   AND DateTime < '2009-10-16 09:00:00'
   AND wwRetrievalMode = 'Avg'
   AND wwResolution = 3600000
```

The results are:

DateTime	Value	Quality	QualityDetail	OPCQuality
2009-10-16 08:00:00.0000000	29.5	0	192	192

What may be confusing in this example is the calculation of the average in the returned row for the phantom cycle leading up to the query start time. The query specifies a positive one hour time interval between the query start time and the query end time. You may therefore expect that the calculated and returned average should be for the specified interval.

However, the time difference between start and end time in the above query is actually not required because the resolution has been provided explicitly (wwResolution = 3600000). If the query specified an end time equal to

the specified start time and if it included the equal sign for the end time, the query would still return the same single row of data.

```
SELECT DateTime, Value, Quality, QualityDetail as QD, OPCQuality
  FROM History
 WHERE TagName IN ('SysTimeSec')
   AND DateTime >= '2007-12-11 08:00:00'
   AND DateTime <= '2007-12-11 09:00:00'
   AND wwRetrievalMode = 'Avg'
   AND wwCycleCount = 1
```

The results are:

DateTime	Value	Quality	QD	OPCQuality
2009-10-16 08:00:00.000000	29.5	0	192	192

This second example also asks for hourly averages and it also returns only a single row of data stamped at the query start time. This query, however, must specify a time difference between the start and end time, because the resolution is not explicitly defined in the query.

As in the preceding query, the specified interval and cycle count of 1 may look like the returned row has been calculated for the specified interval, but the returned row is once again for the phantom cycle leading up to the start time.

The StartDateTime makes it easier to see which time interval was used to calculate the returned aggregate. This column returns the time stamp of the beginning of the cycle used for the aggregate calculation. The time stamp is always returned in accordance with the specified time zone and always has the same offset as the time stamp returned in the DateTime column, even when the two time stamps are on different sides of a DST change.

Assuming results are timestamped at the end of the cycle (as is done by default when wwTimeStampRule is set to END), the initial rows in the examples above would return a DateTime equal to '2009-10-16 08:00:00', and the StartDateTime column would return '2009-10-16 07:00:00' making it easy to interpret the result.

If instead the query were to ask for results time stamped at the beginning of the cycle with wwTimeStampRule set to START, the initial rows in the same examples would still return a DateTime equal to '2009-10-16 08:00:00', but the time stamp has now been shifted in accordance with the time stamp request. The result is therefore calculated for the specified time interval between 8 a.m. and 9 a.m. In this example, the new StartDateTime column would return the same time stamp as the DateTime column, '2009-10-16 08:00:00', again making it easier to interpret the result.

For retrieval modes for which cycles are defined, the StartDateTime column returns the cycle start time. These modes are:

- Cyclic
- Interpolated
- BestFit
- Min
- Max
- Average
- Integral
- Counter
- ValueState

- RoundTrip

In the remaining retrieval modes, the `StartTime` column returns the same time stamp as the `DateTime` column.

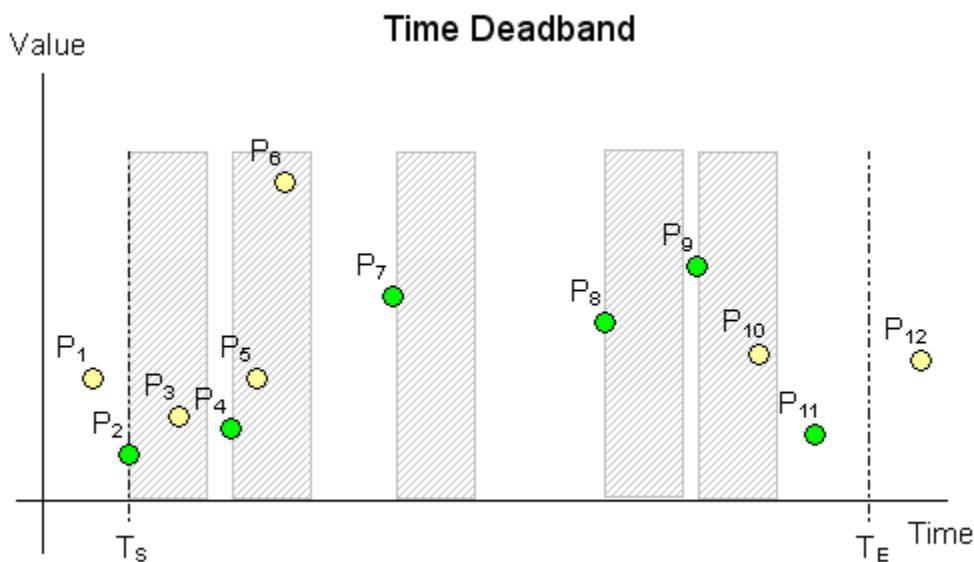
Time deadband (`wwTimeDeadband`)

A time deadband for retrieval controls the time resolution of data returned in delta mode. Any value changes that occur within the time deadband are not returned.

Time deadbands can be applied to analog, discrete, and string tags.

The deadband "base value" is reset each time that a value is returned, so that the last value returned acts as the basis for the deadband.

The following illustration shows an example of applying a time deadband:



Data is retrieved for the time period starting with T_S and ending with T_E . All points in the graphic represent data values stored on the Historian. The grey areas represent the time deadband, which starts anew with every returned value. Only the green points ($P_2, P_4, P_7, P_8, P_{11}$) are returned. The other points are not returned because they fall within a deadband.

Time deadband - query examples

To apply a time deadband, set the `wwTimeDeadband` parameter in your query.

The following queries return data values for the analog tag 'SysTimeSec'.

Value deadband (`wwValueDeadband`)

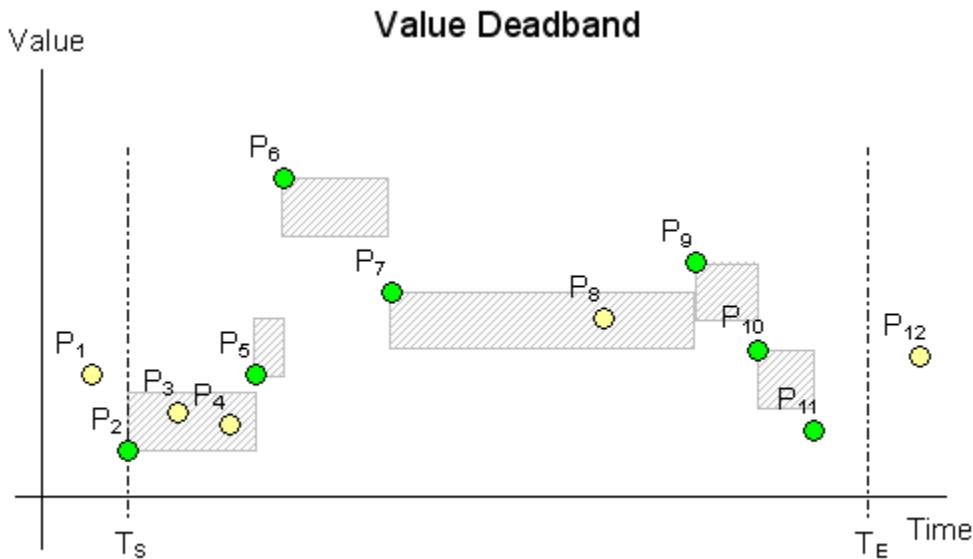
A value deadband for retrieval controls the value resolution of data returned in delta mode. Any data values that change less than the specified deadband are not returned. The deadband is a percentage of a tag's full scale in

engineering units.

The deadband "base value" is reset each time that a value is returned, so that the last value returned acts as the basis for the deadband.

Changes in quality will force a value to be returned even if the value deadband has not been met.

The following illustration shows an example of applying a value deadband:



Data is retrieved for the time period starting with T_S and ending with T_E . All points in the graphic represent data values stored on the Historian. The grey areas represent the value deadband, which starts anew with every returned value. Only the green points ($P_2, P_5, P_6, P_7, P_9, P_{10}, P_{11}$) are returned. The other points are not returned because they fall within a deadband.

Value deadband - query examples

The following queries return data values for the analog tag 'SysTimeSec'. The minimum engineering unit for 'SysTimeSec' is 0, and the maximum engineering unit is 59.

History version (wwVersion)

The Historian allows you to overwrite a stored tag value with later versions of the value. The original version of the value is still maintained, so that effectively, multiple versions of the tag value exist at the same point in time.

When retrieving data, you can specify whether to retrieve the originally stored version or the latest version that is available. To do this, set the history version option to "Original" for the original version or "Latest" for the latest available version. If you do not specify the version, the latest version is returned.

To distinguish between a latest value and an original value, the Historian returns a special QualityDetail value of 202 for a latest point with good quality.

This option is relevant in all retrieval modes.

History version - query example

For example:

```
SELECT TagName, DateTime, Value, wwVersion
  FROM History
 WHERE TagName IN ('SysTimeHour', 'SysTimeMin')
   AND DateTime >= '2001-12-20 0:00'
   AND DateTime <= '2001-12-20 0:05'
   AND wwRetrievalMode = 'Delta'
   AND wwVersion = 'Original'
```

The results are:

TagName	DateTime	Value	wwVersion
SysTimeMin	2001-12-20 00:00:00.000	0	ORIGINAL
SysTimeHour	2001-12-20 00:00:00.000	0	ORIGINAL
SysTimeMin	2001-12-20 00:01:00.000	1	ORIGINAL
SysTimeMin	2001-12-20 00:02:00.000	2	ORIGINAL
SysTimeMin	2001-12-20 00:03:00.000	3	ORIGINAL
SysTimeMin	2001-12-20 00:04:00.000	4	ORIGINAL
SysTimeMin	2001-12-20 00:05:00.000	5	ORIGINAL

When retrieving the latest version, the `wwVersion` parameter always returns with a value of LATEST for all values, even though many of the values may actually be the original values that came from the I/O Server. To distinguish between an actual latest value and an original value, a special `QualityDetail` of 202 is returned for a good, latest point.

For example:

```
SELECT DateTime, Value, Quality, QualityDetail, OPCQuality, wwVersion FROM History
 WHERE TagName IN ('PV')
   AND DateTime >= '2005-04-17 11:35:00'
   AND DateTime <= '2005-04-17 11:36:00'
   AND wwRetrievalMode = 'Delta'
   AND wwVersion = 'Latest'
```

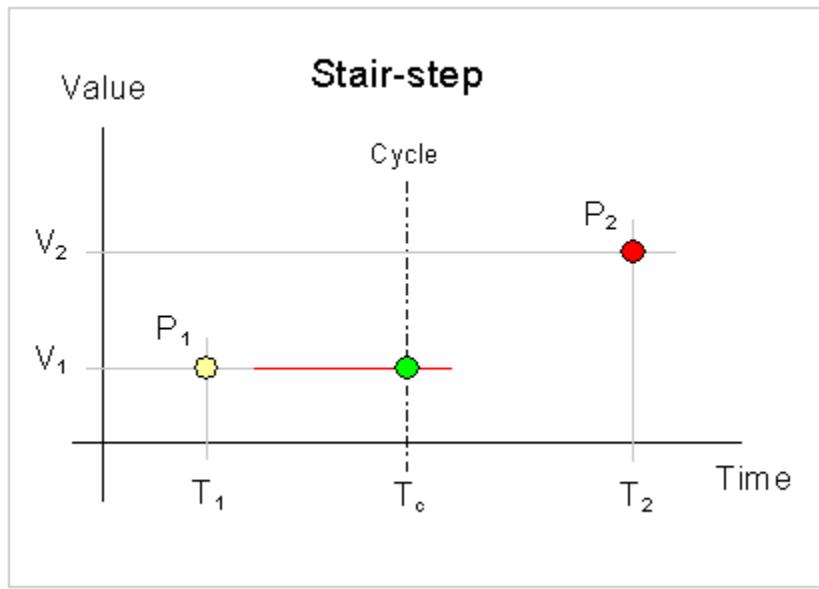
The results are:

Date/Time	Value	Quality	QualityDetail	OPCQuality	wwVersion
2005-04-17 11:35:00.000	12.5	0	192	192	LATEST
2005-04-17 11:35:15.000	17.3	0	192	192	LATEST
2005-04-17 11:35:30.000	34.0	0	202	192	LATEST
2005-04-17 11:35:45.000	43.1	0	192	192	LATEST
2005-04-17 11:36:00.000	51.2	0	192	192	LATEST

Interpolation type (wwInterpolationType)

For various retrieval modes, you can control how analog tag values at cycle boundaries are calculated if there is no actual value stored at that point in time. The options are as follows:

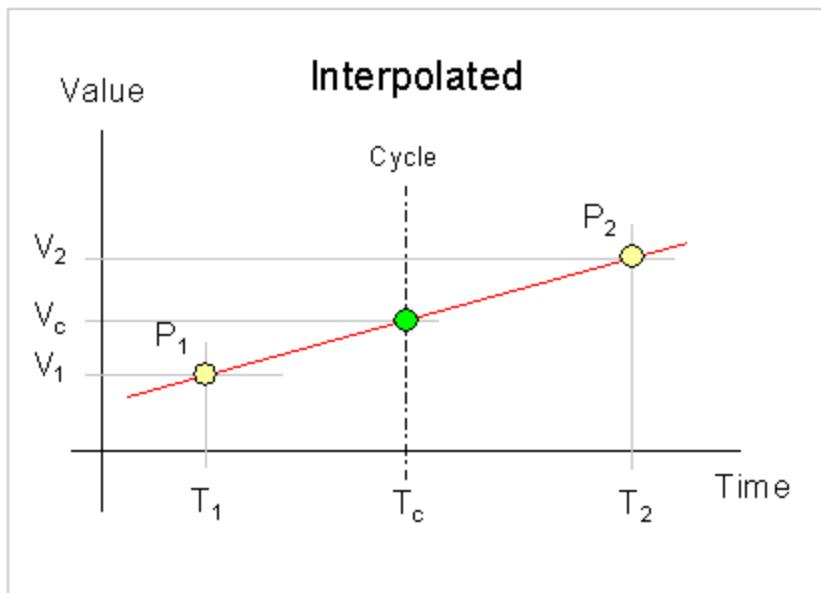
- **Stairstep:** No interpolation occurs. The value at the cycle boundary is assumed to be the same value as the last stored value before the cycle boundary. The last known point is returned with the given cycle time. If no valid value can be found, a NULL is returned.



- **Linear:** The Historian calculates a new value at the cycle boundary by interpolating between the last stored value before the boundary and the first stored value after the boundary. If either of these values is NULL, it returns the last stored value before the boundary.

Expressed as a formula, V_c is calculated as:

$$V_c = V_1 + ((V_2 - V_1) * ((T_c - T_1) / (T_2 - T_1)))$$



The type of data that you want to retrieve usually determines the interpolation type to use. For example, if you have a thermocouple, the temperature change is linear, so it's best to use linear interpolation. If you have a tag that contains discrete measurements, such as a set point, then you probably want to use stair-stepped values. In general, it is recommended that you use linear interpolation as the general setting, and use stair-stepped values for the exceptions.

This option is relevant in the following retrieval modes:

- [Interpolated Retrieval](#)
- "Best fit" retrieval
- [Average Retrieval](#)
- [Integral Retrieval](#)

The quality of an interpolated point is determined by the `wwQualityRule` setting. For more information, see [Quality Rule \(wwQualityRule\)](#).

The interpolation type can be set on three levels:

- The Historian system-wide setting. The system-wide setting must be either stair-step or interpolated. This setting is configured using the Historian Configuration Editor.
- The individual analog tag setting. You can configure an individual analog tag to use the system-wide setting or either stair-stepped values or linear interpolation. The individual tag setting will override the system-wide setting. This setting is configured using the Historian Configuration Editor.
- The setting for the `wwInterpolationType` parameter in the query. This setting overrides any other setting for all tags in the query.

The `wwInterpolationType` parameter is dynamically used both for input for the query, when you need to override the individual tag settings, and for output for each individual row to show whether a particular row value was calculated using linear interpolation (returned as "LINEAR") or if it is a stair-stepped value (returned as "STAIRSTEP").

To force a query to always use linear interpolation whenever applicable, specify the following in the query:

```
AND wwInterpolationType = 'Linear'
```

To force a query to always return stair-stepped values, specify the following in the query:

```
AND wwInterpolationType = 'StairStep'
```

Timestamp rule (wwTimeStampRule)

For various cycle-based retrieval modes, you can control whether the returned values are timestamped at the beginning or at the end of each cycle.

To force a query to timestamp results at the start of a cycle, specify the following in the query:

```
AND wwTimeStampRule = 'Start'
```

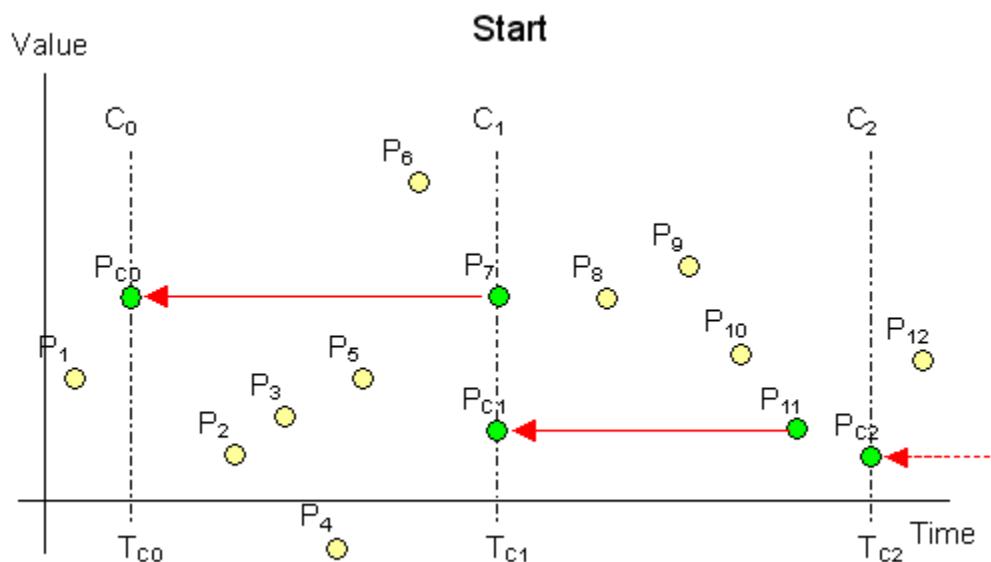
To force a query to timestamp results at the end of a cycle, specify the following in the query:

```
AND wwTimeStampRule = 'End'
```

If you include the `wwTimeStampRule` column in your `SELECT` statement, it will show which timestamp rule has been applied for the individual row, if applicable.

The options are as follows:

- **Start:** The value for a given cycle is stamped with the cycle start time. For example, in the following illustration of a cyclic query, the following values are returned at the cycle boundaries:
 - At T_{C0} : P_7 , because it falls on the cycle boundary. In cyclic mode, if there is a value right on the cycle boundary, it is considered to belong to the cycle before the boundary. In this case, this is the cycle starting at T_{C0} and ending at T_{C1} , and because the Start timestamp rule is used, the value is timestamped at T_{C0} .
 - At T_{C1} : P_{11} , because it is the last value in the cycle starting at T_{C1} and ending at T_{C2}
 - At T_{C2} : The last value in the "phantom" cycle starting at T_{C2}



- **End:** The value for a given cycle is stamped with the cycle end time. For example, in the following illustration of a cyclic query, the following values are returned at the cycle boundaries:

- At T_{C0} : P_1 , because it is the last value in the "phantom" cycle ending at T_{C0} . Because the End timestamp rule is used, the value is timestamped at T_{C0} .
- At T_{C1} : P_7 , because it falls on the cycle boundary. In cyclic mode, if there is a value right on the cycle boundary, it is considered to belong to the cycle before the boundary. In this case, this is the cycle starting at T_{C0} and ending at T_{C1} , and because the End timestamp rule is used, the value is timestamped at T_{C1} .
- At T_{C2} : P_{11} , because it is the last value in the cycle ending at T_{C2} .
- **Server default:** Either Start or End is used, depending on the system parameter setting on the Historian.

This option is relevant in the following retrieval modes:

- [Cyclic Retrieval](#) (only for Historian 9.0 and later)
- [Interpolated Retrieval](#)
- [Average Retrieval](#)
- [Integral Retrieval](#)
- [Counter Retrieval](#)
- [ValueState Retrieval](#)
- [RoundTrip Retrieval](#)

Time zone (wwTimeZone)

For Historian version 8.0 and later, all history data is stored in Coordinated Universal Time (UTC). The `wwTimeZone` extension allows you to specify the time zone to be used for the timestamps of the returned data values. The retrieval subsystem will convert the timestamps to local time in the specified time zone.

The `wwTimeZone` extension may be assigned any of the values stored in the `TimeZone` column of the `TimeZone` table in the Runtime database. In addition to specifying the name of the timezone in the `wwTimeZone` parameter, you can also specify the `TimeZoneID` (as a string). For example, on a typical US English system, specifying "`wwTimeZone = 'Mountain Standard Time'`" and "`wwTimeZone = '64'`" yields the same result.

The `TimeZone` table is repopulated at every system startup from Microsoft operating system registry entries, and will therefore reflect the time zones available from the server operating system, including any new or custom time zones which might be added by operating system service packs or installed software.

The retrieval subsystem will automatically correct for daylight savings time in the requested time zone. When computing daylight savings and time zone parameters, the settings of the server operating system are used. The retrieval sub-system does not provide any means for using client-side settings.

If `wwTimeZone` is not specified, the time zone for retrieval defaults to the time zone of the Historian computer.

For example:

```
SELECT TagName, DateTime, Value, wwTimeZone
  FROM History
 WHERE TagName IN ('SysTimeHour', 'SysTimeMin')
   AND DateTime >= '2001-12-20 0:00'
   AND DateTime <= '2001-12-20 0:05'
   AND wwRetrievalMode = 'Delta'
   AND wwTimeZone = 'W. Europe Standard Time'
```

The results are:

TagName	Datetime	Value	wwTimeZone
SysTimeMin	2001-12-20 00:00:00.000	0	W. Europe Standard Time
SysTimeHour	2001-12-20 00:00:00.000	15	W. Europe Standard Time
SysTimeMin	2001-12-20 00:01:00.000	1	W. Europe Standard Time
SysTimeMin	2001-12-20 00:02:00.000	2	W. Europe Standard Time
SysTimeMin	2001-12-20 00:03:00.000	3	W. Europe Standard Time
SysTimeMin	2001-12-20 00:04:00.000	4	W. Europe Standard Time
SysTimeMin	2001-12-20 00:05:00.000	5	W. Europe Standard Time

If you are using date/time functions and the wwTimeZone parameter, you will need to use the **faaTZgetdate()** function.

Quality rule (wwQualityRule)

For various retrieval modes, you can explicitly exclude values with uncertain quality from data retrieval in modes that calculate return values.

Where applicable, the quality rule can be used to specify whether values with certain characteristics are explicitly excluded from consideration by data retrieval. This parameter will override the setting of the QualityRule system parameter. Valid values are GOOD, EXTENDED, or OPTIMISTIC.

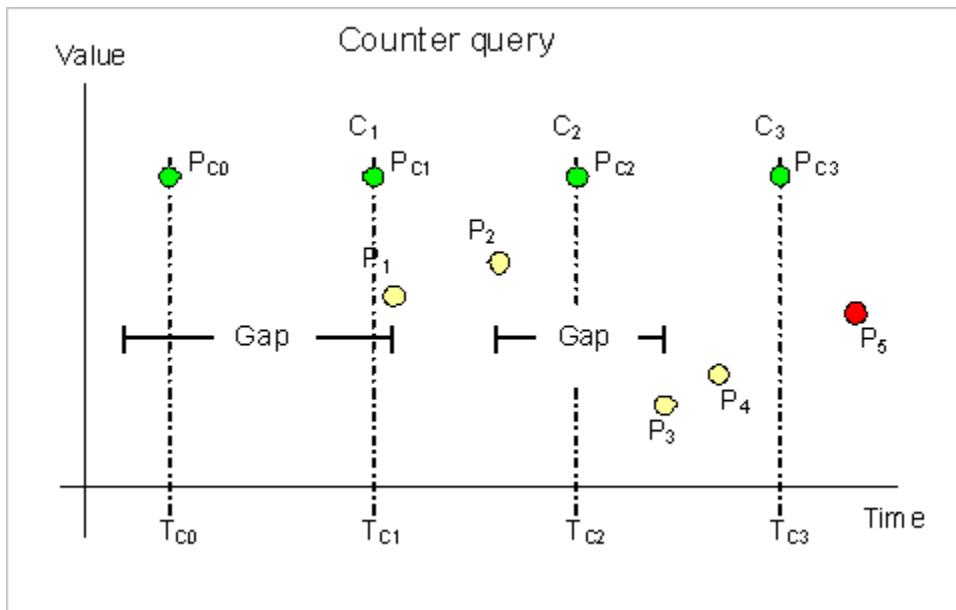
- A quality rule of GOOD means that data values with uncertain (64) OPC quality are not used in the retrieval calculations and are ignored. Values with bad QualityDetail indicate gaps in the data.
- A quality rule of EXTENDED means that data values with both good and uncertain OPC quality are used in the retrieval calculations. Values with bad QualityDetail indicate gaps in the data.
- A quality rule of OPTIMISTIC means that calculations that include some good and some NULL values do not cause the overall calculations to return NULL.

You can apply a quality rule to all retrieval modes.

The OPTIMISTIC setting for the quality rule lets you retrieve information that is possibly incomplete but may nevertheless provide better results in the counter and integral retrieval modes where the calculation cycle contains data gaps. This setting calculates using the last known good value prior to the gap (if possible). The logic for determining the quality of the points returned remains unchanged in both retrieval modes. The integral retrieval mode is an exception to this where the integral is scaled up to cover gaps. For more information, see [Integral Retrieval](#).

The following figure shows a counter retrieval situation in which three of the four shown cycle boundaries are

located in data gaps. Without using OPTIMISTIC, counter queries would return a NULL at all cycle boundaries because the mode needs valid good values at each end of the cycle calculate a precise difference.



If the query were to specify OPTIMISTIC, the counter mode will always return rows with numeric counter values and good quality. These rows may or may not be precise. The PercentGood column of the row returns the percentage of time in each cycle in which retrieval was able to find values stored with good quality, so if the PercentGood is anything less than 100, then the returned row may be incorrect. Quality is returned as uncertain if percent good is not 100 percent.

Now look at the counter values that are returned using OPTIMISTIC quality in the preceding illustration. The query skips the value to be returned at the first cycle boundary, because there is not enough information about the cycle prior to that boundary. At the second cycle boundary, the value 0 will be returned, because there was a gap in the data for the entire first cycle. In the second cycle, there are two points, P1 and P2. The query uses P2 as the end value of the cycle and infers a start value of the cycle from P1. At the third cycle boundary, Tc2, the query returns P2 – P1. Similarly, at the last cycle boundary, the query returns P4 – P3.

For the integral retrieval mode, the query does not summarize data for gaps because there is no way to know which value to use for the summarization. However, if the query specifies OPTIMISTIC quality, the query uses the last known good value for the summarization in the gap. As described for the counter retrieval example, the PercentGood column also expresses the quality of the calculated value in integral retrieval, so if the PercentGood is anything less than 100, then the returned row may be incorrect.

Quality rule - query examples

To force a query to exclude points with doubtful OPC quality, specify the following in the query:

```
AND wwQualityRule = 'Good'
```

To force a query to use points with both good and doubtful OPC quality, specify the following in the query:

```
AND wwQualityRule = 'Extended'
```

If you include the wwQualityRule column in a SELECT statement, it will show which quality rule was used for the individual row, if applicable.

You can combine OPC qualities in a query. For example, if you combine a mixture of good OPC qualities (such as

192 to 219), a good OPC quality (192) will be returned as a combined result.

```
SELECT TagName, DateTime, Value, QualityDetail, OPCQuality, wwRetrievalMode
  FROM History
    WHERE TagName = 'I0R5'
      AND DateTime >= '2009-09-12 00:20'
      AND DateTime <= '2009-09-12 00:40'
      AND wwResolution = 10000
      AND wwRetrievalMode = 'Avg'
```

If you run this query against the following sample data:

Tagname	DateTime	Resolution	QualityDetail
I0R5	2009-09-12 00:07	2	193
I0R5	2009-09-12 00:14	3	195
I0R5	2009-09-12 00:22	0	196
I0R5	2009-09-12 00:25	1	199
I0R5	2009-09-12 00:27	0	200
I0R5	2009-09-12 00:29	2	207
I0R5	2009-09-12 00:33	3	215
I0R5	2009-09-12 00:36	0	216
I0R5	2009-09-12 00:39	1	219

The results are:

Tagname	DateTime	Value	QualityDetail	OPCQuality	wwRetrievalMode
I0R5	2009-09-12 00:20	2.6	192	192	AVERAGE
I0R5	2009-09-12 00:30	1.0	192	192	AVERAGE
I0R5	2009-09-12 00:40	1.6	192	192	AVERAGE

Similarly, if you combine a mixture of doubtful OPC qualities, a doubtful OPC quality (64) will be returned as the combined OPC quality.

```
SELECT TagName, DateTime, Value, QualityDetail, OPCQuality, wwRetrievalMode
  FROM History
    WHERE TagName = 'I0R5'
      AND DateTime >= '2009-09-12 00:20'
      AND DateTime <= '2009-09-12 00:40'
      AND wwResolution = 10000
      AND wwRetrievalMode = 'Integral'
```

If you run this query against the following sample data:

Tagname	Datetime	Resolution	QualityDetail
I0R5	2009-09-12 00:07	2	65
I0R5	2009-09-12 00:14	3	68
I0R5	2009-09-12 00:22	0	71
I0R5	2009-09-12 00:25	1	74
I0R5	2009-09-12 00:27	0	79
I0R5	2009-09-12 00:29	2	80
I0R5	2009-09-12 00:33	3	88
I0R5	2009-09-12 00:36	0	92
I0R5	2009-09-12 00:39	1	64

The results are:

Tagname	Datetime	Value	QualityDetail	OPCQuality	wwRetrievalMode
I0R5	00:20	26.0	64	64	INTEGRAL
I0R5	00:30	10.0	64	64	INTEGRAL
I0R5	00:40	16.0	64	64	INTEGRAL

When you combine the same OPC quality then that OPC quality will be returned. However, when there is no good point in a cycle for cyclic modes such as Integral, Average, Counter, or AnalogSummary, the returned NULL value will have an OPC quality of 0 and a Quality Detail of 65536, regardless of combined qualities.

```
SELECT TagName, StartDateTime, EndDateTime, OPCQuality, PercentGood, wwRetrievalMode, first
FROM AnalogSummaryHistory
WHERE TagName = 'F0R5'
AND StartDateTime >= '2009-09-12 00:20'
AND EndDateTime <= '2009-09-12 00:40'
AND wwResolution = 10000
AND wwRetrievalMode = 'Cyclic'
```

If you run this query against the following sample data:

Tagname	Datetime	Resolution	QualityDetail
F0R5	2009-09-12 00:07	1.6	78
F0R5	2009-09-12 00:14	3.1	78
F0R5	2009-09-12 00:22	0.2	78
F0R5	2009-09-12 00:25	0.8	78

F0R5	2009-09-12 00:27	0.4	78
F0R5	2009-09-12 00:29	2.2	78
F0R5	2009-09-12 00:33	3.3	78
F0R5	2009-09-12 00:36	0.3	78
F0R5	2009-09-12 00:39	1.2	78

The results are:

TagName	StartTime	EndTime	OPCQuality	PercentGood	wwRetrievalMode	first
F0R5	2009-09-12 00:20	2009-09-12 00:30	78	100	CYCLIC	0.200
F0R5	2009-09-12 00:30	2009-09-12 00:40	78	100	CYCLIC	3.300

```
SELECT TagName, DateTime, Value, QualityDetail, OPCQuality, wwRetrievalMode
FROM History
WHERE TagName = 'F0R5'
AND DateTime >= '2009-09-12 00:20'
AND DateTime <= '2009-09-12 00:40'
AND wwResolution = 10000
AND wwRetrievalMode = 'Avg'
```

If you run this query against the following sample data:

TagName	DateTime	Resolution	QualityDetail
F0R5	2009-09-12 00:07	1.6	15
F0R5	2009-09-12 00:14	3.1	15
F0R5	2009-09-12 00:22	0.2	15
F0R5	2009-09-12 00:25	0.8	15
F0R5	2009-09-12 00:27	0.4	15
F0R5	2009-09-12 00:29	2.2	15
F0R5	2009-09-12 00:33	3.3	15
F0R5	2009-09-12 00:36	0.3	15
F0R5	2009-09-12 00:39	1.2	15

The results are:

Tagname	DateTime	Value	QualityDetail	OPCQuality	wwRetrievalMode
F0R5	2009-09-12 00:20	NULL	65536	0	AVERAGE
F0R5	2009-09-12 00:30	NULL	65536	0	AVERAGE
F0R5	2009-09-12 00:40	NULL	65536	0	AVERAGE

When you combine a mixture of good, bad, and uncertain OPC qualities, a doubtful OPC quality (64) will be returned as a combined result.

```
SELECT TagName, DateTime, Value, QualityDetail, OPCQuality, wwRetrievalMode
FROM History
WHERE TagName = 'F0R5'
AND DateTime >= '2009-09-12 00:20'
AND DateTime <= '2009-09-12 00:40'
AND wwResolution = 10000
AND wwRetrievalMode = 'Avg'
AND wwQualityRule = 'Optimistic'
```

If you run this query against the following sample data:

Tagname	DateTime	Resolution	QualityDetail
F0R5	2009-09-12 00:07	1.6	15
F0R5	2009-09-12 00:14	3.1	69
F0R5	2009-09-12 00:22	0.2	78
F0R5	2009-09-12 00:25	0.8	200
F0R5	2009-09-12 00:27	0.4	15
F0R5	2009-09-12 00:29	2.2	92
F0R5	2009-09-12 00:33	3.3	88
F0R5	2009-09-12 00:36	0.3	199
F0R5	2009-09-12 00:39	1.2	196

The results are:

Tagname	DateTime	Value	QualityDetail	OPCQuality	wwRetrievalMode
F0R5	2009-09-12 00:20	2.012	64	64	AVERAGE
F0R5	2009-09-12	0.820	64	64	AVERAGE

		00:30			
F0R5	2009-09-12 00:40	1.751	64	64	AVERAGE

For RoundTrip, StateSummary, and ValueState modes, the OPC qualities are only combined with the same state in a cycle. If the state only occurs once in a cycle, then the qualities of that state will be returned. The returned NULL state will always have an OPC quality of 0 and Quality Detail of 65536. The same qualities are returned for a state that has no roundtrip in RoundTrip mode.

```
SELECT TagName, DateTime, Value, QualityDetail, OPCQuality, StateTime
  FROM History
 WHERE TagName = 'I001'
   AND DateTime >= '2009-09-12 00:20'
   AND DateTime <= '2009-09-12 00:40'
   AND wwResolution = 10000
   AND wwRetrievalMode = 'RoundTrip'
   AND wwStateCalc = 'MaxContained'
```

If you run this query against the following sample data:

Tagname	DateTime	Resolution	QualityDetail
I001	2009-09-12 00:12	1	90
I001	2009-09-12 00:15	2	65
I001	2009-09-12 00:22	1	85
I001	2009-09-12 00:23	2	75
I001	2009-09-12 00:26	1	75
I001	2009-09-12 00:29	2	70

The results are:

Tagname	DateTime	Value	QualityDetail	OPC-Quality	StateTime
I001	2009-09-12 00:20	NULL	65536	0	NULL
I001	2009-09-12 00:20	1.0	90	90	NULL
I001	2009-09-12 00:20	2.0	65	65	NULL
I001	2009-09-12 00:20	1.0	64	64	4000
I001	2009-09-12 00:20	2.0	64	64	6000

The returned Quality Detail is the same as OPC quality unless there is special flag for certain indication for

example when there is indication for role over in counter mode.

```
SELECT TagName, DateTime, Value, QualityDetail, OPCQuality
  FROM History
 WHERE TagName = 'I0R5'
   AND DateTime >= '2009-09-12 00:20'
   AND DateTime <= '2009-09-12 00:40'
   AND wwResolution = 10000
   AND wwRetrievalMode = 'Avg'
```

If you run this query against the following sample data:

Tagname	DateTime	Resolution	QualityDetail
I0R5	2009-09-12 00:07	2	218
I0R5	2009-09-12 00:14	3	218
I0R5	2009-09-12 00:22	0	218
I0R5	2009-09-12 00:25	1	218
I0R5	2009-09-12 00:27	0	218
I0R5	2009-09-12 00:29	2	218
I0R5	2009-09-12 00:33	3	218
I0R5	2009-09-12 00:36	0	218
I0R5	2009-09-12 00:39	1	218

The results are:

Tagname	DateTime	Value	QualityDetail	OPCQuality
I0R5	2009-09-12 00:20	2.6	218	218
I0R5	2009-09-12 00:30	1.0	218	218
I0R5	2009-09-12 00:40	1.6	218	218

For Interpolated mode only the returned row with Linear wwInterpolationType will have combined qualities.

```
SELECT TagName, DateTime, Value, QualityDetail, OPCQuality, wwRetrievalMode,
wwInterpolationType
  FROM History
 WHERE TagName = 'I0R5'
   AND DateTime >= '2009-09-12 00:20'
   AND DateTime <= '2009-09-12 00:40'
   AND wwResolution = 10000
   AND wwRetrievalMode = 'Interpolated'
   AND wwInterpolationType = 'Linear'
```

If you run this query against the following sample data:

Tagname	Datetime	Resolution	QualityDetail
I0R5	2009-09-12 00:07	2	193
I0R5	2009-09-12 00:14	3	195
I0R5	2009-09-12 00:22	0	196
I0R5	2009-09-12 00:25	1	199
I0R5	2009-09-12 00:27	0	200
I0R5	2009-09-12 00:29	2	207
I0R5	2009-09-12 00:33	3	215
I0R5	2009-09-12 00:36	0	216
I0R5	2009-09-12 00:39	1	219

The results are:

Tagname	Datetime	Value	QualityDetail	OPCQuality
I0R5	2009-09-12 00:20	0.8	192	192
I0R5	2009-09-12 00:30	2.3	192	192
I0R5	2009-09-12 00:40	1.0	192	219

Note: Cyclic, Full, Delta, Maximum, Minimum, and BestFit do not have combined qualities; therefore, the rules are not applied to these modes.

State calculation (wwStateCalc)

The state calculation setting applies to ValueState and RoundTrip retrieval.

For ValueState retrieval, you can choose the type of state calculation (aggregation) to be performed on the data:

- **Minimum:** The shortest amount of time that the tag has been in each unique state.
- **Maximum:** The longest amount of time that the tag has been in each unique state.
- **Average:** The average amount of time that the tag has been in each unique state.
- **Total:** The total amount of time that the tag has been in each unique state.
- **Percent:** The total percentage of time that the tag has been in each unique state.
- **MinContained:** The shortest amount of time each tag has been in each unique state for each cycle, disregarding the occurrences that are not fully contained with the calculation cycle.
- **MaxContained:** The longest amount of time that the tag has been in each unique state for each cycle, disregarding the occurrences that are not fully contained with the calculation cycle.
- **AvgContained:** The average amount of time that the tag has been in each unique state for each cycle, disregarding the occurrences that are not fully contained with the calculation cycle.

- **TotalContained:** The total amount of time that the tag has been in each unique state for each cycle, disregarding the occurrences that are not fully contained with the calculation cycle.
- **PercentContained:** The percentage of time that the tag has been in each unique state for each cycle, disregarding the occurrences that are not fully contained with the calculation cycle.

All results except Percent are in milliseconds. Percent is a percentage typically between 0.0 and 100.0. The percentage can be higher than 100 in certain circumstances.

The nature of the data and how you set the cycle count determines whether you should use a "contained" version of the aggregation. The calculations apply to each unique value state that the tag was in during each retrieval cycle for the query. The total and percent calculations are always exact, but the minimum, maximum, and average calculations are subject to "arbitrary" cycle boundaries that do not coincide with the value changes. Therefore, non-intuitive results may be returned. This is most apparent for slowly-changing tags queried over long cycles.

For example, a string tag that assumes only two distinct values changing every 10 minutes is queried with a cycle time of two hours. Going into a cycle, the value (state) at the cycle boundary is recorded. If the value then changes a short while into the cycle, the state found at the cycle start time will most likely end up being the minimum value. Likewise, the state at the cycle end time is cut short at the cycle end time. The two cut-off occurrences in turn skew the average calculation.

For RoundTrip retrieval, you can only specify the following types of state calculations (aggregations) to be performed on the data. The calculations are for each unique state within each retrieval cycle for the query.

- **MinContained.** The shortest time span between consecutive leading edges of any state that occurs multiple times within the cycle, while disregarding state occurrences that are not fully contained inside of the cycle.
- **MaxContained.** The longest time span between consecutive leading edges of any state that occurs multiple times within the cycle, while disregarding state occurrences that are not fully contained inside of the cycle.
- **AvgContained.** The average time span between consecutive leading edges of any state that occurs multiple times within the cycle, while disregarding state occurrences that are not fully contained inside of the cycle. (This is the default.)
- **TotalContained.** The total time span between consecutive leading edges of any state that occurs multiple times within the cycle while disregarding state occurrences that are not fully contained inside of the cycle.
- **PercentContained.** The percentage of the cycle time spent in time span between consecutive leading edges for a state that occurs multiple times within the cycle while disregarding value occurrences that are not fully contained inside of the cycle.

Analog value filtering (wwFilter)

You can use the following analog filters for all retrieval modes:

- Statistical removal of outliers
- Analog to discrete conversion
- Zero around a base value

These filters are applied in a query to retrieve data from the History table, WideHistory table, or StateWideHistory table. These filter only apply to analog tags. All other types of tags, including analog summary tags, are not supported.

You need to specify a filter name in the virtual column wwFilter, with or without an override, to the set of

parameters that are defined for the specified filter. The filters are specified as C-like functions: parentheses are always required, even when you choose not to override the default parameters by passing no parameters.

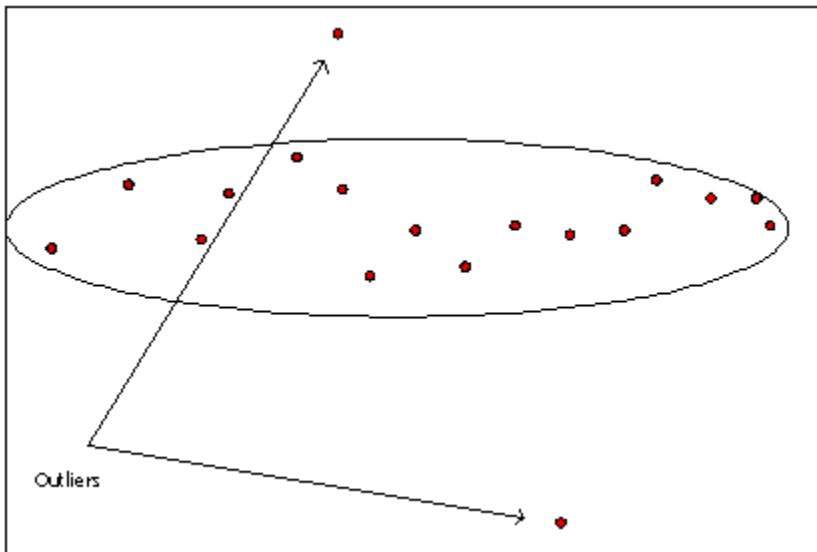
The default value for the `wwFilter` column is ‘NoFilter’. If the query does not specify the `wwFilter` element at all, or if its default value is not overridden, then no filter is applied.

When you use the analog filters in a query that uses `wwQualityRule`, `wwQualityRule` is applied first and `wwFilter` is applied later. You can only use one filter per query.

Statistically removing outliers (SigmaLimit)

This analog filter removes outliers from a set of analog points based on the assumption that the distribution of point values in the set is a normal distribution.

The following illustration shows an example of outliers.



You can filter outliers by specifying a filter called ‘`SigmaLimit()`’. This filter has one parameter defined for specifying the value of n . This parameter is of type double. If the parameter is omitted, then a default parameter of 2.0 is used.

When this filter is specified in any retrieval mode, a time weighted mean, \bar{x} (μ), and time weighted standard deviation, σ (σ), are found for each analog tag for the entire query range including phantom cycles if any, and points falling outside of the range $[\bar{x} - n\sigma, \bar{x} + n\sigma]$ are removed from the point set before the points are processed further. In other words, the value will be filtered out if $\text{value} > \bar{x} + n\sigma$ or $\text{value} < \bar{x} - n\sigma$.

Time weighted standard deviation is calculated as:

```
Math.Sqrt( (integralOfSquares - 2 * timeWeightedAverage * integral + totalTime * timeWeightedAverage * timeWeightedAverage)/totalTime);
```

This is the single pass equivalent to the formula:

$$s_{\text{weighted}}^2 = \frac{1}{N} \sum_{i=1}^{N_w} (X_i - \bar{x})^2$$

Ranges where the value is NULL are excluded from these calculations.

A cyclic query example using a 'SigmaLimit()' filter without specifying the n value would look like this:

```
SELECT DateTime, Value, wwFilter
  FROM History
 WHERE TagName = ('TankLevel')
   AND DateTime >= '2008-01-15 15:00:00'
   AND DateTime <= '2008-01-15 17:00:00'
   AND wwRetrievalMode = 'Cyclic'
   AND wwFilter = 'SigmaLimit()'
```

Not specifying the n-value as done here is the same as specifying 'SigmaLimit(2)'. The result set might look like this:

DateTime	Value	wwFilter
2008-01-15 15:00:00.000	34.56	SigmaLimit()
2008-01-15 16:00:00.000	78.90	SigmaLimit()
2008-01-15 17:00:00.000	12.34	SigmaLimit()

If the first value would be filtered out by the SigmaLimit filter, the value will be replaced with the time weighted mean.

Converting analog values to discrete values (ToDiscrete)

The analog to discrete conversion filter allows you to convert value streams for any analog tag in the query tag list into discrete value streams. The filter can be used with all the retrieval modes.

To convert analog values to discrete values, specify the ToDiscrete() filter in the wwFilter column. This filter has two parameters:

Parameter	Valid Values	Default Value
CutoffValue	any double value	0.0
Operator	>, >=, or <=	>

The following are supported syntaxes.

- ToDiscrete()
- ToDiscrete(2)
- ToDiscrete(2, >=)

The following are unsupported syntaxes.

- ToDiscrete(2,)
- ToDiscrete(, >=)
- ToDiscrete(>=)

The cutoff value holds the value that signifies the boundary between values that are to be interpreted as OFF and values that are to be interpreted as ON.

The operator parameter controls the value range relative to the cutoff value to convert to the ON value and vice versa.

NULLs encountered in the analog value stream are copied unchanged to the discrete value stream. The quality of each discrete point is copied from the analog point that causes its production. However, the quality detail for values modified with this filter will have the QualityDetail flag 0x2000 (value changed by filter) set. For example, consider the following ValueState query:

```
SELECT DateTime, vValue, StateTime, wwFilter
  FROM History
 WHERE TagName IN ('SysTimeSec')
   AND DateTime >= '2008-01-15 15:00:00'
   AND DateTime <= '2008-01-15 17:00:00'
   AND wwRetrievalMode = 'ValueState'
   AND wwStateCalc = 'MinContained'
   AND wwResolution = 7200000
   AND wwFilter = 'ToDiscrete(29, >)'
```

Here the operator is specified as `>`, so values greater than but not including 29 are internally converted to ON, whereas values from 0 to 29 are converted to OFF. This query could return the following rows:

DateTime	vValue	StateTime	wwFilter
2008-01-15 15:00:00.000	0	30000	ToDiscrete(29, >)
2008-01-15 15:00:00.000	1	30000	ToDiscrete(29, >)
2008-01-15 17:00:00.000	0	30000	ToDiscrete(29, >)
2008-01-15 17:00:00.000	1	30000	ToDiscrete(29, >)

The values returned in the StateTime column show that the shortest amount of time that SysTimeSec had values equivalent to either ON or OFF and remained in that state was 30 seconds. A similar RoundTrip query would look like this:

```
SELECT DateTime, vValue, StateTime, wwFilter
  FROM History
 WHERE TagName IN ('SysTimeSec')
   AND DateTime >= '2008-01-15 15:00:00'
   AND DateTime <= '2008-01-15 17:00:00'
   AND wwRetrievalMode = 'RoundTrip'
   AND wwStateCalc = 'MaxContained'
   AND wwResolution = 7200000
   AND wwFilter = 'ToDiscrete(29, <=)'
```

Here the operator is specified as `<=`, so the resulting conversion is exactly opposite to that performed in the previous query. Now values smaller than or equal to 29 are internally converted to ON, whereas values from 30 to 59 are converted to OFF. This query could return the following rows:

Date/Time	vValue	State/Time	wwFilter
2008-01-15 15:00:00.000	0	60000	ToDiscrete(29, <=)
2008-01-15 15:00:00.000	1	60000	ToDiscrete(29, <=)
2008-01-15 17:00:00.000	0	60000	ToDiscrete(29, <=)
2008-01-15 17:00:00.000	1	60000	ToDiscrete(29, <=)

The values returned in the State/Time column now show that the longest amount of time found between roundtrips for both the OFF and the ON state within the 2-hour cycles was 60 seconds.

Using the ToDiscrete() filter is similar to using edge detection for event tags. Edge detection returns the actual value with a timestamp in history for when a value matched a certain criteria. The ToDiscrete() filter returns either a 1 or 0 to show that the criteria threshold was crossed. The ToDiscrete() filter is more flexible, however, in the following ways:

- You can use it with delta and full retrieval.
- You can combine it with "time-in-state" calculations to determine how long a value is above a certain threshold or the duration between threshold times.

Use the ToDiscrete() filter if you are mostly interested in when something occurred, and not necessarily the exact value of the event.

For more information on edge detection, see [Edge Detection for Events \(wwEdgeDetection\)](#).

"Zeroing" around a base value (SnapTo)

This analog filter lets you force values in a well-defined range around one or more base values to "snap to" that base value. For example, you can use this filter when a tank is known to be empty, but the tag that stores the tank level returns a "noisy" value close to zero.

The filter can be used with all retrieval modes, but its main benefits are in the aggregate retrieval modes: average, integral, minimum, and maximum.

To zero values around the base value, specify the SnapTo() filter in the wwFilter column of the query.

The syntax for this filter is:

```
SnapTo([tolerance[,base_value_1[, base_value_2]...]])
```

This filter has two parameters:

Parameter	Valid Values	Default Value
Tolerance	any double value	0.01
BaseValue	zero, one, or up to 100 comma-separated double values	single base value of 0.0

The following are supported syntaxes.

- SnapTo() – Same as SnapTo(0.01, 0.0)
- SnapTo(3.7) – Same as SnapTo(3.7, 0.0)
- SnapTo(3,) – Syntax Error
- SnapTo(,0) – Syntax error
- SnapTo(,) – Syntax error
- SnapTo(3, 4, -5) – Tolerance=3, Base Values 4 and -5.

When the Snap to filter is specified, point values falling inside any of the ranges [Base value – Tolerance, Base value + Tolerance] will be forced to the base value before the point goes into further retrieval processing. The result is undefined if the base value +/- tolerance exceeds the range of the double data type. The range is calculated using this expression:

```
If (x <= Base value + Tolerance AND x >= Base value - Tolerance)
x = Base value
```

where x is the value of the point then

If ranges overlap, the first matching base value will be used.

A query example from the History table looks like this:

```
SELECT DateTime, Value, wwFilter
  FROM History
 WHERE TagName = ('TankLevel')
       AND DateTime >= '2008-01-15 15:00:00'
       AND DateTime <= '2008-01-15 17:00:00'
       AND wwRetrievalMode = 'Average'
       AND wwResolution = 3600000
       AND wwFilter = 'SnapTo(0.01, 0, 1000)'
```

The following rows might be returned:

DateTime	Value	wwFilter
2008-01-15 15:00:00.000	0	SnapTo(0.01, 0, 1000)
2008-01-15 16:00:00.000	875.66	SnapTo(0.01, 0, 1000)
2008-01-15 17:00:00.000	502.3	SnapTo(0.01, 0, 1000)

When a value is snapped, the QualityDetail bit flag 0x2000 is set.

If the filter syntax is not correct, a syntax error is returned and no rows are returned.

Selecting values for analog summary tags (wwValueSelector)

For an analog summary tag, multiple summarized values can be stored in the Historian for a single summarization period. When you query analog summary data, a single value, time, and quality (VTQ) must first be extrapolated from the summarized values.

You set the value selector in the query to specify which summarized value to return. The possible values are as follows:

Value Selector Setting	Value Returned	Timestamp Returned
AUTO	The retrieval mode determines the value. See the following table for how AUTO applies to the value selection. This is the default value.	The retrieval mode determines the timestamp. See the following table for how AUTO applies to the value selection. This is the default value.
FIRST	The first value that occurs within the summary period.	The actual timestamp of the first value occurrence within the summary period.
LAST	The last value that occurs within the summary period.	The actual timestamp of the last value occurrence within the summary period.
MIN or MINIMUM	The first minimum value that occurs within the summary period.	The actual timestamp of the first minimum value occurrence within the summary period.
MAX or MAXIMUM	The first maximum value that occurs within the summary period.	The actual timestamp of the first maximum value occurrence within the summary period.
AVG or AVERAGE	A time-weighted average calculated from values within the summary period.	The summary period start time.
INTEGRAL	An integral value calculated from values within the summary period.	The summary period start time.
STDDEV or STANDARDEVIATION	A standard deviation calculated from values within the summary period.	The summary period start time.

The following table describes the value to be considered if the value selector is set to AUTO:

Retrieval Mode	Analog Summary Behavior
Cyclic	The last value within the summary period is used. The actual timestamp of the last value occurrence within the summary period is used.
Delta	The last value within the summary period is used. The actual timestamp of the last value occurrence within the summary period is used.
Full	The last value within the summary period is used. The actual timestamp of the last value occurrence within the summary period is used.
Interpolated	The retrieval mode determines the appropriate value to return. See the following table for how AUTO applies to the value selection. This is the default value.
Best Fit	The first, last, min, and max points from analog summaries are all considered as analog input points. Best fit analysis is done with these points. If the analog summary percentage good is not 100%, the cycle is considered to have a NULL.
Average	<p>The averages of analog summaries are calculated using the values from the Average column of the AnalogSummaryHistory table. Interpolation type is ignored for analog summary values, and STAIRSTEP interpolation is always used. PercentGood is calculated by considering the TimeGood of each analog summary.</p> <p>If cycle boundaries do not exactly match the summary periods of the stored analog summaries, the averages and time good are calculated by prorating the average and time good values for the portion of the time the summary period overlaps with the cycle. Quality will be set to 64 (uncertain) if cycle boundaries do not match summary periods.</p> <p>If the QualityDetail of any analog summary considered for a cycle is uncertain (64), the resulting quality is set to 64.</p>
Minimum	The first minimum value within the summary period is used. The actual timestamp of the first minimum value occurrence within the summary period is used.
Maximum	The first maximum value within the summary period is used. The actual timestamp of the first maximum value occurrence within the summary period is used.

Retrieval Mode	Analog Summary Behavior
Integral	<p>The integrals of analog summaries are calculated using the Integral column of the AnalogSummaryHistory table. Interpolation type is ignored for analog summary values, and STAIRSTEP interpolation is always used. PercentGood is calculated by considering the TimeGood of each analog summary.</p> <p>If cycle boundaries do not exactly match the summary periods of the stored analog summaries, the integrals and time good are calculated by prorating the integral and time good values for the portion of the time the summary period overlaps with the cycle. Quality is set to 64 (uncertain) if cycle boundaries do not match summary periods.</p> <p>If the QualityDetail of any analog summary considered for a cycle is uncertain (64), the resulting quality will be set to 64.</p>
Slope	<p>The last value within the summary period is used. The actual timestamp of the last value occurrence within the summary period is used.</p>
ValueState	<p>Cannot be used with analog summary data. No values are returned.</p>
Counter	<p>Cannot be used with analog summary data. No values are returned.</p>
RoundTrip	<p>Cannot be used with analog summary data. No values are returned.</p>

For an analog summary tag, if any of the data within a summary period has an OPCQuality other than Good, the OPCQuality returned will be Uncertain. This is true even for summary values that are not calculated, such as first, last, minimum, maximum, and so on. For example, if the OPCQuality for a last value is actually Good, but there was a I/O Server disconnect during the summary calculation period, the OPCQuality for the last value is returned as Uncertain. A QualityDetail of 202 is used to distinguish between the original point and the latest point.

Edge detection for events (wwEdgeDetection)

An event is the moment at which a detection criterion is met on historical tag values in the Historian. At a basic level, anything that can be determined by looking at stored data can be used as an event.

When detecting events, it is useful to pinpoint rows in a result set where the satisfaction of criteria in a WHERE clause changed from true to false, or vice versa. For example, you may want to know when the level of a tank went above 5 feet. The WHERE clause in a query for this example might be TANKLEVEL > 5. As the tank level approaches 5 feet, the criterion does not return true. Only when the level crosses the line from not satisfying the criterion to satisfying it, does the event actually occur. This imaginary "line" where the change occurs is called the *edge*.

Over a period of time, there may be many instances where the criteria cross the "edge" from being satisfied to not satisfied, and vice-versa. The values on either side of this "edge" can be detected if you configure your event tag to include this information. There are four possible options for edge detection: *none*, *leading*, *trailing*, or *both*. You will get differing results based on which option you use:

Option	Results
None	Returns all rows that successfully meet the criteria; no edge detection is implemented at the specified resolution.
Leading	Returns only rows that are the first to successfully meet the criteria (return true) after a row did not successfully meet the criteria (returned false).
Trailing	Returns only rows that are the first to fail the criteria (return false) after a row successfully met the criteria (returned true).
Both	All rows satisfying both the leading and trailing conditions are returned.

Edge detection only applies to analog and discrete value detectors. Also, edge detection is handled slightly differently based on whether you are using analog tags or discrete tags.

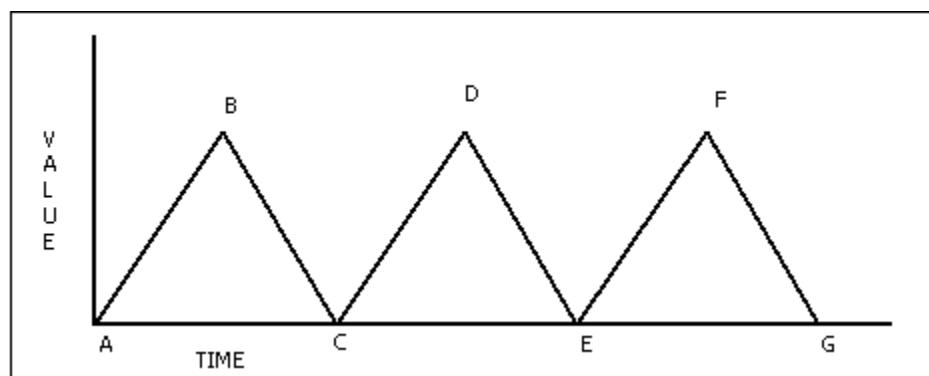
For more information, see the following topics:

- [Edge Detection for Analog Tags](#)
- [Edge Detection for Discrete Tags](#)

You can also use the `ToDiscrete()` query filter to determine when data values cross a particular threshold. For more information, see [Converting Analog Values to Discrete Values \(ToDiscrete\)](#).

Edge detection for analog tags

For example, the behavior of the WHERE clause as it processes a result set can be illustrated as:



Slopes A-B, C-D and E-F are rising edges, while slopes B-C, D-E and F-G are falling edges. The slopes are affected by the WHERE clause, which is a combination of the `wwEdgeDetection` option and the comparison operator used

against the value.

The following table describes the rows that would be returned for the various edge detection settings:

Operator	=	<	>	<=	>=
Leading	Falling and rising edges; first value that meets the criteria.	Falling edge only; first value to meet the criteria.*	Rising edge only; first value to meet the criteria.	Falling edge only; first value to meet the criteria. *	Rising edge only; first value to meet the criteria.
Trailing	Falling and rising edges; first value to fail the criteria after a value meets the criteria.	Rising edge only; equal to the first value to fail the criteria.	Falling edge only; first value to fail the criteria.*	Rising edge only; first value to fail the criteria.	Falling edge only; first value to fail the criteria.*

* If the falling edge is a vertical edge with no slope, the query will return the lowest value of that edge.

The following query selects all values of "SysTimeSec" that are greater than or equal to 50 from the History table between 10:00 and 10:02 a.m. on December 2, 2001. No edge detection is specified.

```
SELECT DateTime, Value
  FROM History
 WHERE TagName = 'SysTimeSec'
   AND DateTime >= '2001-12-02 10:00:00'
   AND DateTime <= '2001-12-02 10:02:00'
   AND wwRetrievalMode = 'Cyclic'
   AND wwResolution = 2000
   AND Value >= 50
   AND wwEdgeDetection = 'None'
```

The results are:

DateTime	Value
2001-12-02 10:00:50.000	50
2001-12-02 10:00:52.000	52
2001-12-02 10:00:54.000	54
2001-12-02 10:00:56.000	56
2001-12-02 10:00:58.000	58
2001-12-02 10:01:50.000	50
2001-12-02 10:01:52.000	52
2001-12-02 10:01:54.000	54
2001-12-02 10:01:56.000	56
2001-12-02 10:01:58.000	58

Leading edge detection for analog tags

If *Leading* is specified as the parameter in the edge detection time domain extension, the only rows in the result set are those that are the first to successfully meet the WHERE clause criteria (returned true) after a row did not successfully meet the WHERE clause criteria (returned false).

The following query selects the first values of "SysTimeSec" from the History table to meet the Value criterion between 10:00 and 10:02 a.m. on December 2, 2001.

```
SELECT DateTime, Value
  FROM History
 WHERE TagName = 'SysTimeSec'
       AND DateTime >= '2001-12-02 10:00:00'
       AND DateTime <= '2001-12-02 10:02:00'
       AND wwRetrievalMode = 'Cyclic'
       AND wwResolution = 2000
       AND Value >= 50
       AND wwEdgeDetection = 'Leading'
```

The query will return only the two values that were greater than or equal to 50 for the time period specified:

DateTime	Value
2001-12-02 10:00:50.000	50
2001-12-02 10:01:50.000	50

Compare these results with the same query using no edge detection, as shown in [Edge Detection for Analog Tags](#). Notice that even though the original query returned ten rows, the edge detection only returns the first row recorded after the event criteria returned true.

Trailing edge detection for analog tags

If *Trailing* is specified as the parameter in the edge detection extension, the only rows in the result set are those that are the first to fail the criteria in the WHERE clause (returned false) after a row successfully met the WHERE clause criteria (returned true).

The following query selects the first values of "SysTimeSec" from the History table to fail the Value criterion between 10:00 and 10:02 a.m. on December 2, 2001.

```
SELECT DateTime, Value
  FROM History
 WHERE TagName = 'SysTimeSec'
       AND DateTime >= '2001-12-02 10:00:00'
       AND DateTime <= '2001-12-02 10:02:00'
       AND wwRetrievalMode = 'Cyclic'
       AND wwResolution = 2000
       AND Value >= 50
       AND wwEdgeDetection = 'Trailing'
```

The query returns only the two values that were the first to fail the criteria in the WHERE clause for the time period specified:

DateTime	Value
2001-12-02 10:01:00.000	0
2001-12-02 10:02:00.000	0

Compare these results with the same query using no edge detection, as shown in [Edge Detection for Analog Tags](#). Notice that even though the original query returned ten recorded rows for each value, the edge detection only returns the first row recorded after the event criteria returned false.

Both leading and trailing edge detection for analog tags

If Both is specified as the parameter in the edge detection extension, all rows satisfying both the leading and trailing conditions are returned.

The following query selects values of "SysTimeSec" from the History table that meet both the Leading and Trailing criteria between 10:00 and 10:02 a.m. on December 2, 2001.

```
SELECT DateTime, Value
  FROM History
 WHERE TagName = 'SysTimeSec'
   AND DateTime >= '2001-12-02 10:00:00'
   AND DateTime <= '2001-12-02 10:02:00'
   AND wwRetrievalMode = 'Cyclic'
   AND wwResolution = 2000
   AND Value >= 50
   AND wwEdgeDetection = 'Both'
```

The results are:

DateTime	Value
2001-12-02 10:00:50.000	50
2001-12-02 10:01:00.000	0
2001-12-02 10:01:50.000	50
2001-12-02 10:02:00.000	0

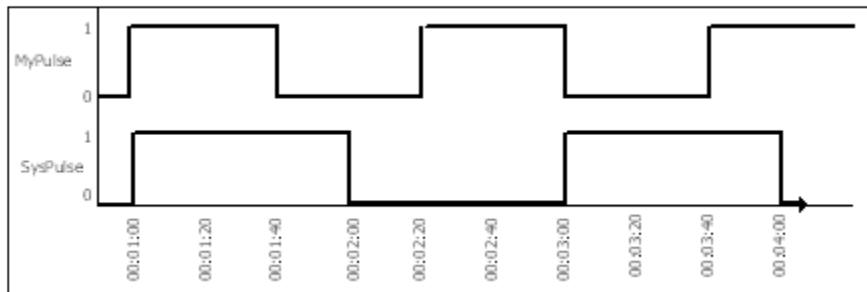
Compare these results with the same query using no edge detection, as shown in [Edge Detection for Analog Tags](#). Notice that value of the first row in the original query is returned in the result set.

Edge detection for discrete tags

Edge detection for discrete tags operates differently than for analog tags. For example, assume the following discrete tags are stored.

Tag	Description
SysPulse	Transitions between 1 and 0 every minute.
MyPulse	Transitions between 1 and 0 every 40 seconds.

A representation of the data stored in the system is as follows:



The following queries select values of "SysPulse" and "MyPulse" that have a value of 1 (On) from the History and WideHistory tables between 12:59 and 1:04 a.m. on December 8, 2001. No edge detection is specified.

Leading edge detection for discrete tags

If Leading is specified as the parameter in the edge detection time domain extension, the only rows in the result set are those that are the first to successfully meet the WHERE clause criteria (returned true) after a row did not successfully meet the WHERE clause criteria (returned false).

The following queries select values of "SysPulse" and "MyPulse" that have a value of 1 (On) from the History and WideHistory tables between 12:59 and 1:04 a.m. on December 8, 2001.

Trailing edge detection for discrete tags

If Trailing is specified as the parameter in the edge detection extension, the only rows in the result set are those that are the first to fail the criteria in the WHERE clause (returned false) after a row successfully met the WHERE clause criteria (returned true).

Retrieval styles for trend

The Trend application allows you to use "retrieval styles" that automatically switch retrieval modes for trend tags based on the trend duration and/or tag type. For example, you could configure a style that uses delta retrieval for short time periods and best-fit retrieval for longer periods. This helps you to balance response speed and accuracy. Also, retrieval styles allow you to calculate moving averages and retrieve data from the Historian's summary tables.

Working with retrieval styles

Retrieval styles cover the following retrieval options:

- Retrieval mode
- Resolution for cycle-based modes (as time duration or number of pixels per cycle)
- Data retrieval source (history or summary tables)
- Moving average calculation
- State calculation for ValueState retrieval

If you want to use additional retrieval options, such as deadbands, row limits or a quality rule, you must specify them in the Trend application at the application and/or tag level. For more information, see [Configuring Retrieval Options](#) and [Configuring Trend Options for a Tag](#).

You can use retrieval styles at the application and/or tag level. When you specify a retrieval style at the application level, that style is used for all tags that do not have a different style specified.

The Trend application comes with various predefined styles that you can use. For a description of each style, see [Using the Standard Retrieval Styles](#). Alternatively, you can define your own retrieval styles to suit your needs. For more information, see [Location and Structure of Retrieval Styles](#) and [Creating and Editing Retrieval Styles](#).

Location and structure of retrieval styles

Retrieval styles are stored at the application level in the following XML file:

C:\ProgramData\ArchestrA\ActiveFactory\Trend\RetrievalStyles.xml

After you add or edit a style in this file, it is available to all users of the Trend application on the system. You can edit the file in any text or XML editor. Note the following requirements:

- You must save the file in UTF-8 encoding.
- Names and values are case-sensitive. If any name or value is misspelled, the Trend application may hang on startup.

Structure of the retrieval styles file

Structure of the retrieval styles file

The retrieval styles file has the following structure:

```
XML header
Style collection
  Retrieval style 1
    Style names for different locales
    Duration range 1
      Retrieval type 1
      ...
      Retrieval type n
    Duration range 2
    ...
    Duration range n
  Retrieval style 2
  ...
```

```
Retrieval style n
End of style collection
```

That is:

- The file contains exactly one style collection, represented by the [styleCollection XML Element](#). For more information, see [styleCollection XML Element](#).
- The style collection contains one or more retrieval styles, represented by the [retrievalStyle XML element](#). Each of these represents a style that is available for use in the Trend application. For more information, see [retrievalStyle XML Element](#).
- Each retrieval style contains one or more duration ranges, represented by the [duration XML element](#). A duration range defines the retrieval types that are to be used for trend queries whose duration is within a specified range. Duration ranges should be arranged in descending length. For more information, see [duration XML Element](#).
- Each duration range contains one or more retrieval types, represented by the [retrieval XML element](#). The retrieval type defines the retrieval options that are to be used for tags of a certain type. For more information, see [retrieval XML Element](#).

The retrieval type that gets used for a given tag is determined as follows:

1. You select a retrieval style in the Trend application.
2. The Trend application searches that retrieval style for a duration range that covers the duration of your trend. This would be the first duration range with a time period that is shorter than the trend duration.
3. When it has found a suitable duration range, it searches that duration range for a retrieval type that suits the type of the tag.

A simple example file might look like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<styleCollection version="9.2" xmlns="urn:retrievalstyle-schema">
    <retrievalStyle server="InSQL" minVersion="8.0" maxVersion="9.0" enabled="true">
        <styleName locale="en">My style</styleName>
        <styleName locale="ja">My style in Japanese</styleName>
        <styleName locale="zh-CN">My style in Chinese</styleName>
        <styleName locale="de">My style in German</styleName>
        <styleName locale="fr">My style in French</styleName>
        <duration minSpan="P0Y0M1DT0H0M0S">
            <retrieval tagType="Discrete" source="History" retrievalMode="Delta"
                stateCalc="*" resolution="0" pixels="0" movingAverageValues="0" />
            <retrieval tagType="All" source="History" retrievalMode="Cyclic" stateCalc="*"
                resolution="0" pixels="5" movingAverageValues="0" />
        </duration>
        <duration minSpan="P0Y0M0DT0H0M0S">
            <retrieval tagType="All" source="History" retrievalMode="Delta" stateCalc="*"
                resolution="0" pixels="0" movingAverageValues="0" />
        </duration>
    </retrievalStyle>
</styleCollection>
```

In this case, the file only defines one style named My style. When querying two days of data for a discrete tag using this style, delta retrieval is used (the first retrieval element in the first duration element). For an analog tag, cyclic retrieval with one cycle for every five pixels of the trend width is used instead (the second retrieval element in the first duration element). For queries that are shorter than one day, delta retrieval is used

regardless of the tag type (the only retrieval element in the second duration element).

Creating and editing retrieval styles

To create or edit retrieval styles, you edit the retrieval styles file in a text or XML editor. Read [Location and Structure of Retrieval Styles](#) first to get an overview of how this file is structured.

The following procedure provides you the basic steps to add a new style. For details on each XML element, refer to the corresponding subsection.

To create a new style:

1. Under the `styleCollection` element, add a `retrievalStyle` element.
2. Under the `retrievalStyle` element, add a `styleName` elements for each locale in which you want to use the style. For more information, see [retrievalStyle XML Element](#).
3. Decide at which trend durations you want to switch retrieval options. Under your `retrievalStyle` element, add `duration` elements for each of these "switching points." For more information, see [duration XML Element](#).
4. For each `duration` element, add `retrieval` elements as needed to define retrieval types. For more information, see [retrieval XML Element](#).

Retrieval style XML elements

The following sections describe each of the XML elements in the retrieval styles file. For information on how they fit together, see [Location and Structure of Retrieval Styles](#).

styleCollection XML element

styleCollection XML element

The `styleCollection` element represents a collection of retrieval styles. It is the container for multiple retrieval styles represented by `retrievalStyle` elements. It has two required attributes:

- `version`: Specifies the format version of the style collection. The only valid value is 9.2.
- `xmlns`: Specifies the XML namespace to be used. Set this attribute to `urn:retrievalstyle-schema`.

The retrieval styles file can only contain single `styleCollection` element.

retrievalStyle XML element

retrievalStyle XML element

The retrievalStyle element represents single retrieval style. It is the container for two other elements:

- **styleName**: Specifies the name of the style for the locale specified by the `locale` attribute. This is the name by which you can access the style when the Trend application runs under the specified locale.

You can specify the locale as a two-character ISO language code or a four-character combination of language code and country code. If you specify a name for a two-character locale, it is used for all sub-locales that do not have a separate name defined. For example, if you specify a name for the de locale, it is used for the de-DE, de-AT and de-CH locales unless you specify separate names for those locales.

You must specify a `styleName` element for all styles that you want to use in a given locale. If a style does not have a name defined for a locale, the Trend application does not show it when running under that locale.

The only exception is when you run the Trend application under a locale for which no style names are defined at all. In that case, the styles are shown with their names for the en locale.

- **duration**: Specifies a duration range. For more information, see [duration XML Element](#).

It has three required attributes:

- **server**: Specifies the server type for which the style can be used. Always set this attribute to `InSQL`.
- **minVersion**: The minimum Historian version that the retrieval style can work with, either 8.0 or 9.0. If the Trend application is connected to a Historian whose version is lower than the version specified here, the style is not used.

Specify 9.0 if your style uses functionality that is not supported for AVEVA Historian 8.0.

- **enabled**: Specifies whether the style is active. To temporarily disable the style, set this attribute to `false`.

It has one optional attribute:

- **maxVersion**: The maximum Historian version against which the retrieval style can be used. This attribute is not currently used.

duration XML element

duration XML element

The duration element represents a duration range. It contains the retrieval types that are used when the trend period is longer than the time period it specifies.

A retrieval style can contain any number of duration elements. However, you should arrange these elements in descending length. This is because the Trend application uses the first suitable duration range that it finds, that is, the first duration range with a time period shorter than the current trend period.

For example, assume you have three duration ranges defined in the following order:

- 1 day
- 4 hours
- 0 seconds

For a query with a duration of 2 days, the Trend application uses the retrieval types defined for the "1 day" duration range because it is the first range whose time period is shorter than 2 days. Now assume the same duration ranges are ordered like this:

- 4 hours
- 1 day
- 0 seconds

In this case, the Trend application uses the retrieval types defined for the "4 hours" duration range because again, it is the first range whose time period is shorter than 2 days. The more suitable "1 day" duration range is ignored.

Note: You should always define a duration range with a time period of 0 seconds. This serves as a "catch-all" for trend periods that aren't covered by any other duration range.

The duration element has one required attribute:

- `minSpan`: Specifies the time period as a standard XML duration value, for example, P0Y0M1DT0H0M0S. The number to the left of Y represents the number of years, the number to the left of M represents the number of months, and so on (D = days, H = hours, M = minutes, S = seconds). P and T are separator characters.

It is the container for one other element:

- `retrieval`: Specifies a retrieval type. For more information, see [retrieval XML Element](#).

retrieval XML element

retrieval XML element

The `retrieval` element represents a retrieval type, that is, a set of retrieval options for a certain type of tag.

You can have multiple `retrieval` elements in a duration range. However, you should order them from the most specific to the least specific one. This is because the Trend application uses the first suitable retrieval type that it finds, that is, the first retrieval type with a matching tag type and available history source.

For example, assume that you have three retrieval types defined in the following order:

- Analog tags, Summary data
- Analog tags, History data
- All tags, History data

For an analog tag, the Trend application first tries to retrieve summary data according to the first retrieval type. If no summary data is available, it retrieves history data according to the second retrieval type. Now assume the retrieval types are ordered like this:

- Analog tags, History data
- Analog tags, Summary data
- All tags, History data

In this case, the Trend application never tries to retrieve summary data for an analog tag; it never considers the second retrieval type because it has already found a suitable retrieval type in the first one.

You should always define a retrieval type with a tag type of "All" and a history source of "History." This serves as a "catch-all" for tags that aren't covered by any other retrieval style.

The `retrieval` element has seven required attributes:

- `tagType`: Specifies the tag type for which the retrieval type should be used. Valid values are `All`, `Analog`, `Discrete`, and `String`.
- `source`: Specifies the history source from which to retrieve data. Valid values are `History` to retrieve data from history tables and `Summary` to retrieve data from summary tables. When using `Summary`, you must specify the summary frequency in the `resolution` attribute.
- `retrievalMode`: Specifies which retrieval mode to use. Valid values are `Cyclic`, `Delta`, `Full`, `Interpolated`, `BestFit`, `Average`, `Min`, `Max`, `Integral`, `Slope`, `Counter`, `ValueState`, and `RoundTrip`. Make sure that you specify a retrieval mode that is supported for the specified tag type. For example, Counter retrieval does not work with string tags. Therefore, if you try to retrieve data for a string tag in Counter mode, the Historian does not return any data.

For information on each mode, see [Understanding retrieval modes](#).

- `stateCalc`: Specifies which state calculation to use in `ValueState` and `RoundTrip` retrieval. Valid values are `Min`, `Max`, `Average`, `Total`, and `Percent`. For more information, see [State Calculation \(wwStateCalc\)](#). If you are not using `ValueState` retrieval, specify an asterisk (*).
- `resolution`: Specifies the retrieval resolution in milliseconds when retrieving history data in cycle-based retrieval modes, or the summary frequency in seconds when retrieving summary data. For more information, see [Resolution \(Values Spaced Every X ms\) \(wwResolution\)](#)
Alternatively, you can set this attribute to 0 and specify a retrieval resolution using the `pixels` attribute.
- `pixels`: Specifies the retrieval resolution for cycle-based retrieval modes as the number of pixels per cycle. The number of cycles is the width of the trend chart divided by the value of this attribute. For example, if the chart is 500 pixels wide and the `pixels` attribute is set to 5, then 100 cycles are used.
Alternatively, you can set this attribute to 0 and specify a retrieval resolution using the `resolution` attribute. If you specify non-zero values for both the `pixels` and the `resolution` attributes, `resolution` takes precedence.
- `movingAverageValues`: Specifies whether to calculate moving averages when retrieving history data. If set to 0, no moving averages are calculated. Otherwise, moving averages are calculated using the number of values specified in this attribute.

Using the standard retrieval styles

The following table describes the standard retrieval styles available in the Historian Client Trend application.

Style name	Description
BestFit-5	Best Fit retrieval with one cycle per five pixels.
BestFit-10	Best Fit retrieval with one cycle per ten pixels.
BestFit-15	Best Fit retrieval with one cycle per 15 pixels.
Cyclic (ActiveFactory 9.1)	Cyclic retrieval with one cycle per two pixels.
Integral (ad hoc)	Integral retrieval for queries longer than 15 minutes. Resolution depends on query duration. Best-fit retrieval with one cycle per ten pixels for queries shorter than 15 minutes.
Averages (From Summaries or Ad Hoc)	Tries to retrieve analog averages from summary tables. If no summary data is available, uses Average retrieval for analog tags and ValueState retrieval for discrete tags. Resolution depends on query duration.
Averages (ad hoc)	Average retrieval for analog tags and ValueState retrieval for discrete tags. Resolution depends on query duration.
Summary (InSQL 8.0)	Tries to retrieve analog averages from summary tables for queries longer than 15 minutes. If no summary data is available, uses Cyclic retrieval with one cycle per pixel for queries longer than 8 hours and Delta retrieval for shorter queries.
Counter-20	Counter retrieval with one cycle per 20 pixels.
Counter on round periods	Counter retrieval with cycles at even time periods (one minute, one hour, etc. depending on the resolution).
Time In State (percent)	ValueState retrieval with percent calculation for queries longer than one minute. Resolution depends on query duration. Full retrieval for shorter queries.
Time In State (Avg)	ValueState retrieval with average calculation for queries longer than one minute. Resolution depends on query duration. Full retrieval for shorter queries.
RoundTrip (PercentContained)	RoundTrip retrieval with percentcontained calculation for queries longer than one minute. Resolution depends on query duration. Full retrieval for shorter queries.
RoundTrip (AvgContained)	RoundTrip retrieval with averagecontained calculation for queries longer than one minute. Resolution

Style name	Description
	depends on query duration. Full retrieval for shorter queries.
MovingAverage (12-5 sec)	Moving averages for analog tags with 12 values and a resolution of five seconds. Delta retrieval for all other tags.
MovingAverage (30-1 sec)	Moving averages for analog tags with 30 values and a resolution of one second. Delta retrieval for all other tags.
MovingAverage (10-1 pixel)	Moving averages for analog tags with 10 values and a resolution of one cycle per pixel. Delta retrieval for all other tags.

Retrieval styles, application settings, and tag settings

There are various ways to set retrieval options in the Trend application: using a retrieval style vs. using custom retrieval options, using the application-wide options vs. using tag-level options. Also, there are some differences depending on which Historian version you are using. The following guidelines help you understand which retrieval settings are actually used in a given situation.

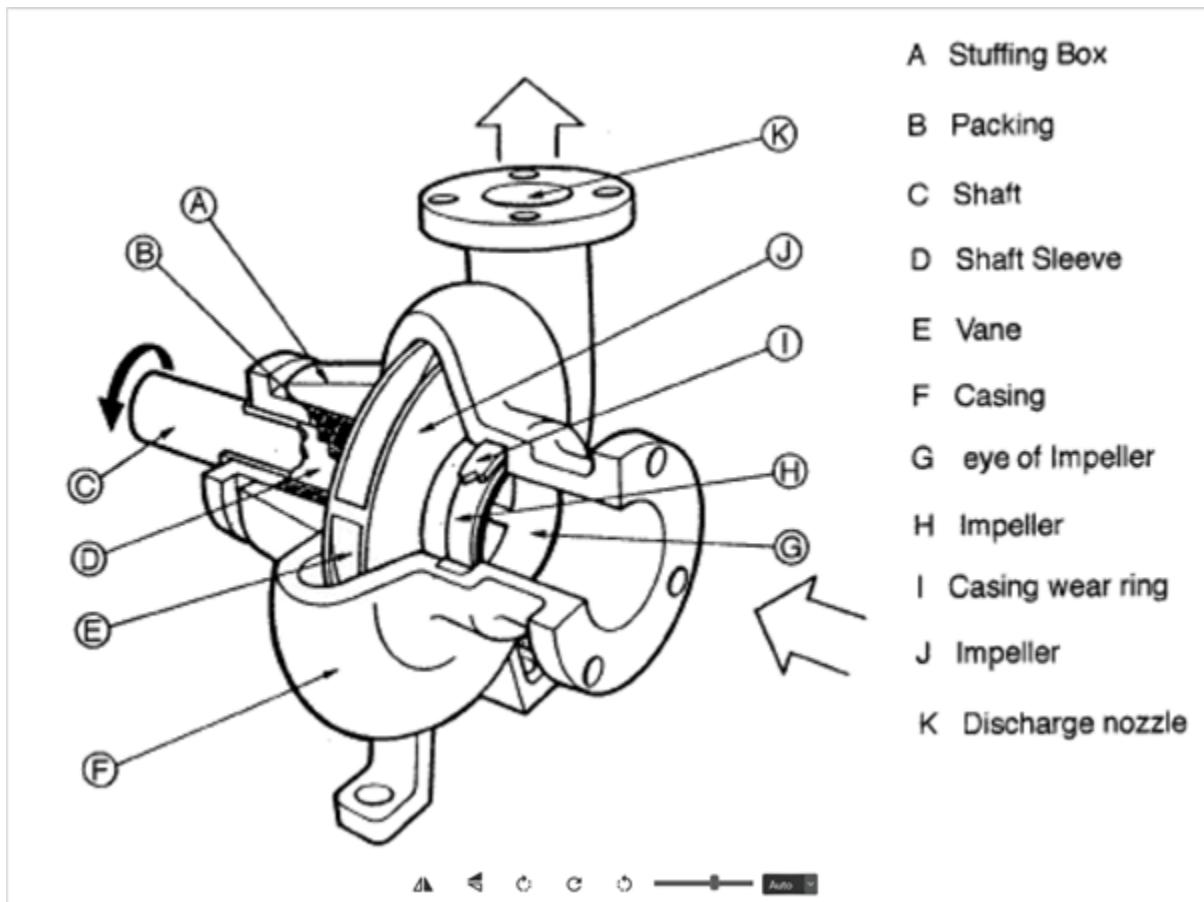
1. Settings at the tag level override settings at the application level. For the aaHistClientTrend control, this means that the properties starting with CurrentTag... override the properties starting with RetrievalOptions...
2. Because a retrieval style definition can contain multiple sets of retrieval settings with different retrieval modes, some of the settings available for editing at the application or tag level may turn out to be irrelevant for the retrieval mode that actually gets used for a given query. However, because there is no way to know in advance which retrieval mode will be used, the settings are still available for editing. The same applies to properties in the aaHistClientTrend control.
3. At the application level, you can specify additional options for retrieving data from Historians with a version earlier than 9.0. For more information, see [Configuring Other Options](#). These settings override any style that you may have selected at the application level. For example, if you have set these options to enable delta retrieval for periods below 15 minutes, but you have selected a style at the application level that specifies cyclic retrieval for all time periods, the Trend application enforces delta retrieval for all time periods below 15 minutes regardless of the settings in the style.

However, if you select the style at the tag level, then the style settings override the application options. In the above example, cyclic retrieval would be used for all time periods regardless of the application settings specifying delta retrieval.

4. If there is a conflict between a setting specified in a style and a setting specified using one of the aaHistClientTrend control's properties (for example, retrieval resolution), the style setting overrides the property setting.

ImageViewerApp

The ImageViewerApp shows an image within a pane of a ViewApp while the ViewApp is running or displayed in preview mode. Users can change the size of the image and its orientation while it is being displayed, but cannot make any permanent changes to the appearance of an image.



The ImageViewerApp supports the following image file types:

- Bitmap (bmp)
- Graphics Interchange Format (gif)
- Joint Photographic Experts Group (jpeg) or (jpg)
- Portable Network Graphics (png)
- Tagged Image File Format (tiff)

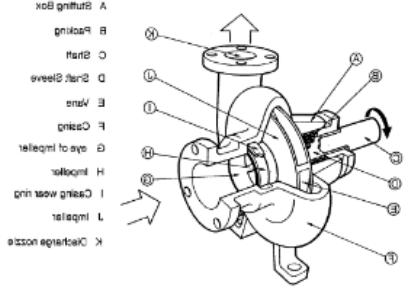
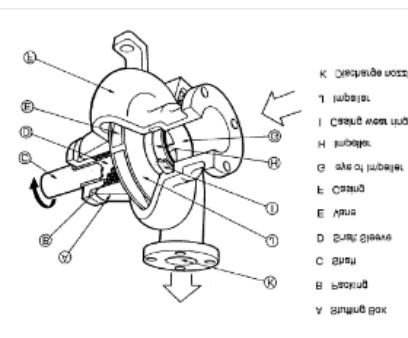
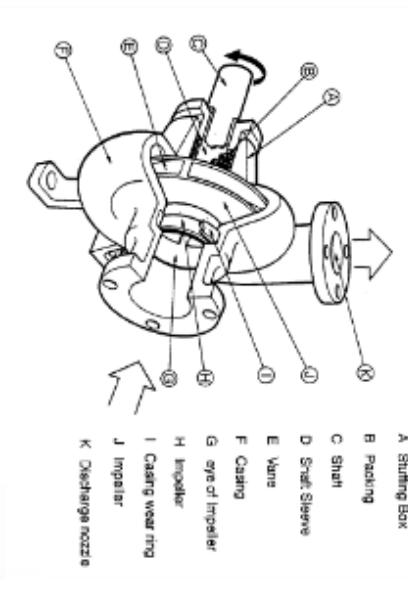
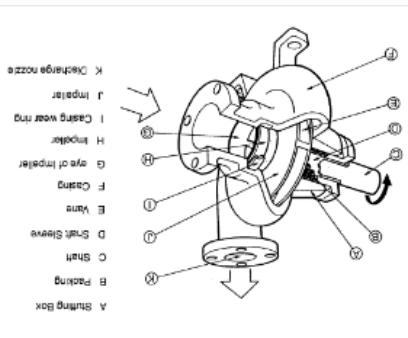
The ImageViewerApp includes a set of visual controls that appear when you change the focus to the pane that contains the image. These controls can be used to change the zoom level, or rotate or flip the image.



Change the orientation of an image

When the user selects the pane containing an image during runtime or in preview mode, a set of controls

appears beneath the image. Users can select a control to change the orientation of the image.

	Flip the image horizontally.	
	Flip the image vertically.	
	Rotate the image 90 degrees clockwise.	
	Rotate the image 180 degrees clockwise.	

Change the size of an image

During runtime or preview mode, users can change the size of a displayed image using a set of controls or by keyboard shortcuts.

Change Image Size by Image Controls

The ImageViewerApp includes controls to change the size of an image being displayed during preview mode or while a ViewApp is running. The user must set the focus of the ViewApp to the pane showing an image and before selecting a size control. Image size can be reduced to a minimum of ten percent and increased to a maximum of 800 percent of its original size.

	Move the slider left or right to decrease or increase the size of the image.
	Change image size by selecting a zoom percentage from a list. Selecting the inverted arrow at the right of the button displays a list of available zoom percentages to resize the image.

Change Image Size by Keyboard Shortcuts

The ImageViewerApp supports keyboard shortcuts to enable the user to reduce or increase the size of an image.

- Increase the size of an image by simultaneously pressing the Ctrl key and the key containing the plus sign (+).
- Reduce the size of an image by simultaneously pressing the Ctrl key and the key containing the minus sign (-).

Scroll an image

Users can scroll an image vertically or horizontally when the size of an image exceeds the boundaries of the image viewer window. Scroll bars appear in the viewer window when an image is greater than the boundaries of the viewer window. Users can move a scroll bar with their mouse. Also, if the monitor displaying the ViewApp is touch enabled, users can use a single finger vertical or horizontal swipe gesture to scroll the displayed area of an image.

Touch support

The ImageViewerApp supports single finger gestures on touch screens or portable devices.

A displayed graphic can be swiped or flicked vertically to move the window view to the top or bottom of the

graphic. A displayed graphic can be swiped or flicked horizontally to move the window view to the left or right of the graphic.

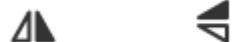
A single finger tap can be used to select the visual icons shown within the viewer window of the ImageViewerApp.

See [Optimize the ImageViewerApp for a touch device](#) for instructions to enlarge the size of the visual icons to make it easier to select them while using touch gestures.

Optimize the ImageViewerApp for a touch device

The visual icons shown within the viewer window of the ImageViewerApp can be enlarged to make them easier to select on touch screens or portable devices.

The following screen captures show the default size of the ImageViewerApp's visual icons and their size after they have been enlarged for touch. AVEVA OMI includes the OptimizeForTouch Boolean attribute that belongs to the MyViewApp.Settings namespace. The default value of OptimizeForTouch is false. When OptimizeForTouch is set to true, the size of the visual icons is enlarged.

OptimizeForTouch Attribute Value	Icon Size	Example Icons
MyViewApp.Settings.OptimizeForTouch=False	Regular Size (Non-Touch)	
MyViewApp.Settings.OptimizeForTouch=True	Optimized for Touch	

A common way to dynamically adjust the size of the visual icons in a running ViewApp is to place a graphic element like a button on a pane and then associate user input animation to toggle the state of the OptimizeForTouch attribute.

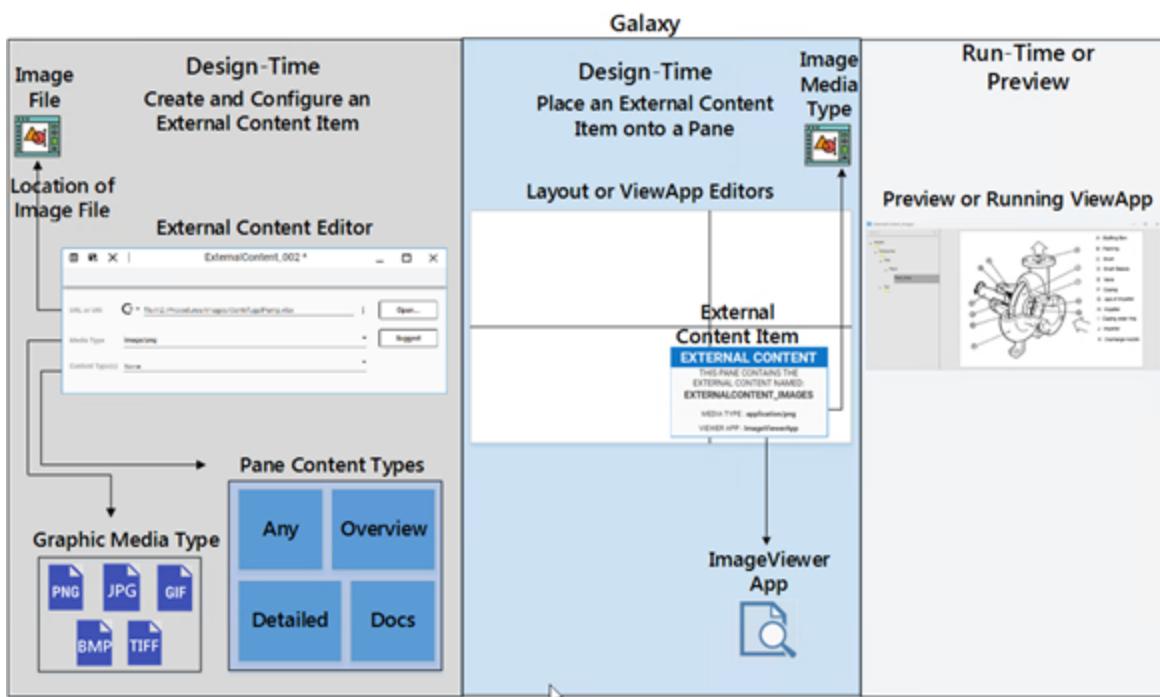
Compatibility of the ImageViewerApp

The ImageViewerApp includes compatibility with current and future versions of System Platform.

- The ImageViewerApp package must be installed on a computer running System Platform 2017 Update 3. The ImageViewerApp cannot be installed on any other version of System Platform except Update 3.
- After the ImageViewerApp is installed on a System Platform 2017 Update 3 system, the ImageViewerApp will be able to migrate to future versions of System Platform InTouch OMI when they become available.

Configure the ImageViewerApp

The following figure summarizes using an External Content item to point to a remote computer containing an image that shows the different parts of a pump. An External Content item is configured during design time to identify the location of the image file and its media type. Typically, the External Content item is associated with an asset or a user defined object. The asset or object is selected from the **Toolbox** tab and placed onto a layout pane. During runtime, the user selects an asset associated with a navigation item to show the graphic image.



The major steps to implement the ImageViewerApp to show a graphic in a ViewApp are summarized in the following list:

1. Place a graphic image on a computer accessible from your Galaxy.
2. Create an External Content item.
3. Configure the External Content item that identifies the location and media type of the graphic.
4. Associate an External Content item to a graphic or object.
5. Place an asset or graphic on a ViewApp pane associated with the External Content item.

Before you begin

Before starting the steps to configure an External Content item, ensure the following prerequisites have been met:

- If you are placing the graphic file on a remote computer in your network, ensure connectivity between the computer where the ViewApp will run and the computer where the remote graphic has been saved.

Mapping a network drive is a typical way to connect to a remote computer hosting documents that you want show in a running ViewApp. For this configuration to work successfully, it is important that the computer to which the ViewApp is launched from has the same drive letter mapped to a location having the referenced document.

The ImageViewerApp shows an error message to the user if the specified URI connection to the graphic is bad. For more information about the error message, see [ImageViewerApp error message](#).

- Save the graphic in a presentation format supported by the ImageViewerApp.

The ImageViewerApp supports GIF, JPG or JPEG, PNG, or TIFF graphic file types.

ImageViewerApp error message

An External Content item includes an **URL or URI** option to specify the location of the graphic file to be shown by the ImageViewerApp during runtime or in preview mode. The specified location of the file must be complete and accurate. If the file is moved, deleted, or renamed, the ImageViewerApp shows an error message to the user during runtime, but the ViewApp continues to run.



Create an External Content item

You create an External Content item from the Visualization folder of the System Platform IDE. Initially, the External Content item defaults to a set of values that can be changed using the External Content editor.

To create an External Content item

1. Open the IDE and select the **Visualization** tab.
2. Select a folder within the **Visualization** folder if you want to create a new External Content item at a specific location.
3. On the **Home** ribbon, in the **Create** area, select **External content**.

You can also create an External Content item by other methods:

- **Keyboard Shortcut**

Press Ctrl + Shift + C

- **Shortcut Menu**

Right-click a folder of the **Visualization** folder to show a shortcut menu. Select the **New** option, and then select **External Content**.

A new **External Content** item is created in the **Visualization** folder.



The name of the new item follows a default naming convention of appending a three-digit number to the word **ExternalContent**.

4. Rename the External Content item.

After you create an External Content item, it must be configured to specify the remote location of the content and its media type.

Configure an External Content item

Each External Content item has three properties that must be assigned values:

- URL or URI

- Media Type
- Content Type(s)

An External Content item is modified from the External Content editor. The editor can be opened by double-clicking an item from the Visualization folder. Also, an External Content item that has been placed onto a layout pane can be edited from the Layout and ViewApp editors by selecting the item from the Actions list.

To configure an External Content Item

1. Open the IDE and select the **Visualization** tab to show the External Content items available in the Galaxy.
2. Double-click an External Content item to open within the External Content editor.

For more information about each field in an External Content item, see [Details of the External Content editor](#).

3. Enter a URI string or web site URL in the **URL or URI** field where the graphic file is saved.

The value you enter is the location of the external content specified by a URI-formatted string. A media location must be specified.

You can browse for the graphic file by selecting the vertical dots icon to the right of the data entry field. A drop-down list includes a **Browse for file** option that enables you to browse your network and select a file. The **URL or URI** field updates and shows a formatted URI path to the file.

You can also enter a URL to a web site by selecting the **http://** or **https://** options from the drop-down list. The **http://** and **https://** options assist the user when entering URLs by prefixing the URLs with the selection.

See [More information about specifying a URI](#) for information about the format of a URI string.

4. Select the **Open** button to validate the URI or URL you entered.

The **Open** button launches the appropriate application on the computer associated with the graphic file type.

5. Select the type of media associated with the graphic file type from the drop-down list of the **Media Type** field.

The ImageViewerApp supports five image media types:

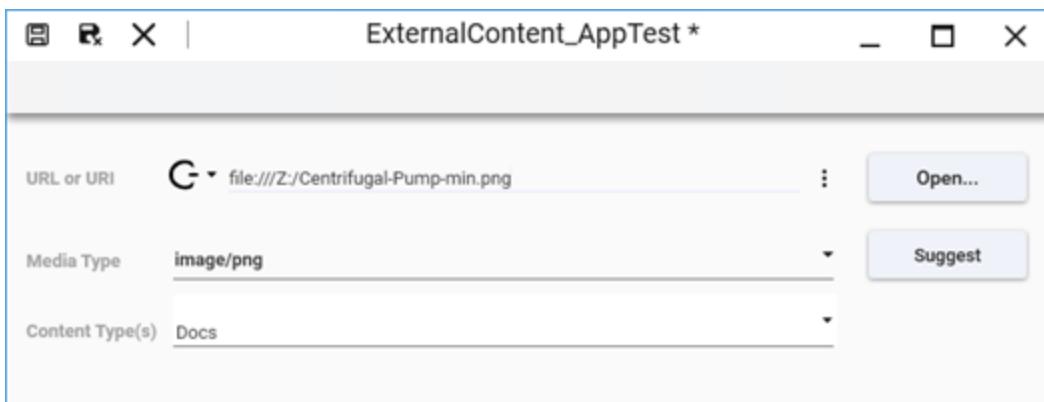
- **image/bmp**
- **image/gif**
- **image/jpeg**
- **image/png**
- **image/tiff**

See [More information about media types](#) for more information about supported media types.

6. Select the type of content from the drop-down list of the **Content Type(s)** field.

The value you enter is used by content-placement algorithms within AVEVA OMI during run time to determine the placement of content when AutoFill or ShowContent calls are invoked.

The configured External Content item for an image file should look like the following screen capture.



7. Save your changes and exit from the External Content item.

Link an External Content item to an object or template

There are three ways to integrate an External Content in a ViewApp.

- Associate an External Content item to an object template, which is described in the following procedure
- Associate an External Content item to a custom navigation item that appears in the navigation model of the ViewApp.
- Place the External Content item on a layout pane, which is the simplest way and is described in [Adding an External Content item to a ViewApp](#).

To link an External Content item to an object template

After you have created and configured an External Content item, you can link it to an object template. External Content items can be linked only to objects as opposed to graphics, which can be linked to or owned by an object. All External Content items reside in the Visualization folder. A single External Content item can be linked to multiple objects.

1. Open an object template in the **Object Editor**.
2. Select the **Attributes** tab.
3. In the **Content** pane, select the **Link Content**  button.
The Galaxy Browser opens.
4. Navigate to the folder that contains the **External Content** item to be linked.
5. Select the item, then click **OK**.

The item is added to the object and appears in the **Content** tab.

Note: After linking an External Content item, you can modify it by selecting it and then pressing the **Edit** button. However, any changes you make will apply to all objects that link to the item.

6. Save your editing changes to the object template and exit from the Object Editor.
7. Create an instance of the linked object template.
8. Create a layout which contains pane with a Content Type that matches the content type of the External Content item.
9. Add a navigation tree to the layout
10. Create a ViewApp that incorporates the layout.

11. Deploy the ViewApp.

In runtime, navigate to the object or any instance created from the template. External Content item will auto-fill the matching pane to show the external content.

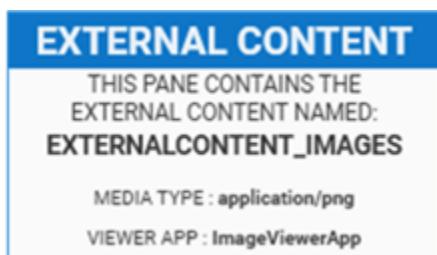
Adding an External Content item to a ViewApp

You can place an External Content item directly onto a pane from either the Layout or ViewApp editors . External Content items are listed in the **Toolbox** or **Assets** tabs of either editor.

To associate external content to a ViewApp

1. Open the System Platform IDE and select the **Visualization** tab to see the list of layouts.
2. Select a layout that you want to insert external content and open it in the Layout editor.
3. Select the **Toolbox** tab of the Layout editor to show the list of content accessible from the Galaxy.
4. Locate the External Content item you want to place in the layout you selected.
External Content thumbnails appear beneath of the list of content shown in the **Toolbox** list.
5. Select an External Content thumbnail from the **Toolbox** area, and then drag and drop it onto a pane of a layout incorporated into a ViewApp.

The External Content thumbnail appears in the pane containing the name of the External Content item, the media type of the external content, and the viewer app that will display the media during runtime. Also, the layout **Actions** area shows the name of the External Content item and the pane that it was placed in.



Editing an External Content Item After Being Placed onto a Pane

An External Content item can be modified after being placed in a layout or a ViewApp, which automatically propagates the changes

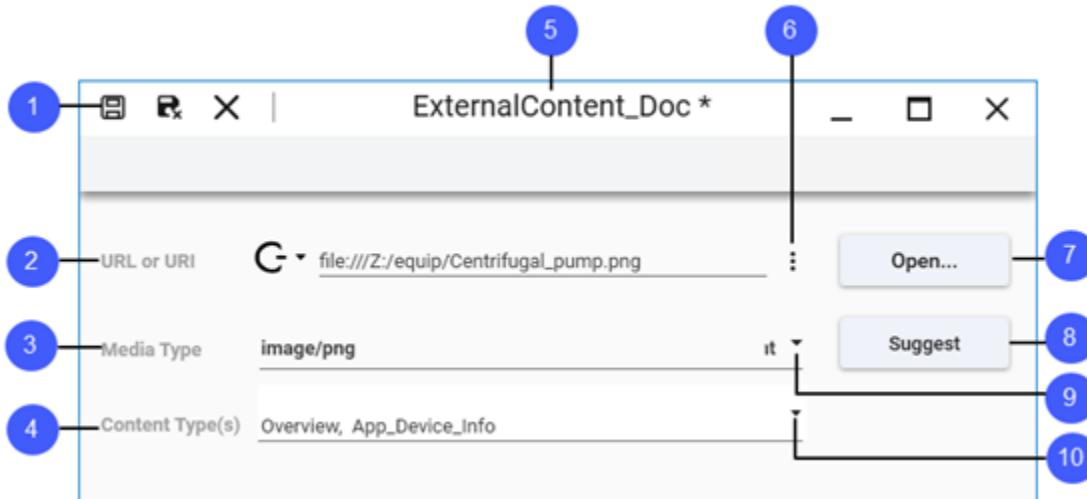
To edit an External Content item in a layout

1. If necessary, open the layout containing an External Content item.
2. Locate the item in the layout **Actions** list.
3. Select the **Options** icon shown at the right of the item in the **Actions** list.
A drop-down list shows an **Edit External Content** command.
4. Select the **Edit** command.
The External Content editor opens the item for editing.
5. Update the values assigned to the item and save your changes.
The External Content thumbnail placed in a pane updates to reflect the changes made to the item.

Details of the External Content editor

You use the External Content editor to configure an External Content item for a document. Before you configure an External Content item, you must know:

- Network location of a document
- Media type of the document
- Content type(s) of the layout pane that will host the External Content item



1	Commands to save, save and close, and close the External Content editor.
2	URL or URI that specifies the location of external media using a standard format. See Specify the URL of External Content for the format of a URI or URL. A warning message appears if the document cannot be located during runtime.
3	A media type is a two-part identifier that specifies the type of application required to process or view remote content. A media type can be entered in the field or selected from a drop-down list. See Specify media types for External Content for the format of a Media type value. A warning message appears if a media type is specified that does not have an associated viewer application.
4	Content type assigned to the external media that enables ViewApp algorithms to place content in specific panes during runtime.

5	Name of the External Content item.
6	Drop-down list with options to browse for an external content file or specify a URL using HTTP or HTTPS.
7	Validates the location of external media specified in the URL or URI field. An attempt is made to display the external media in an application assigned as the default by the operating system, not the app specified for the external content media type. A warning message appears if the content cannot be found at the location specified in the URL or URI field.
8	Suggest is optional. When selected, it automatically selects a media type based on the entry in the URL or URI field. The entered value can be changed if the suggestion does not match the expected media type. Note: Ensure the suggested media type is correct. The external media does not appear if the media type is incorrect.
9	Drop-down list of commonly used media types. Media types that have apps in the Galaxy capable of servicing them are shown in bold text.
10	Drop-down list of content types that can be used at preview\runtime to assign the external content to panes of a matching content type. Note: None is not a content type. Instead, it simply means a content type has not been assigned to the pane.

More information about media types

A media type is a standard two-part string analogous to a MIME type that identifies external content file types and their format. System Platform uses a media type to identify external content and the type of app required to display media during runtime.

A media type consists of a type and a subtype, which is further structured into a tree.

type "/" [tree "."] subtype ["+" suffix] *[";" parameter]

For more information about the format of media types, see <https://www.iana.org/assignments/media-types/media-types.xhtml>

System Platform provides several default media types that include associated apps to display media. The drop-

down list of the **Media Type** field shows the default media types in bold text to indicate a viewer app is available in the Galaxy that can service external content with these media types. Other listed media types that appear in plain text require an app to be created and imported into a Galaxy to play the specified media.

More information about specifying a URI

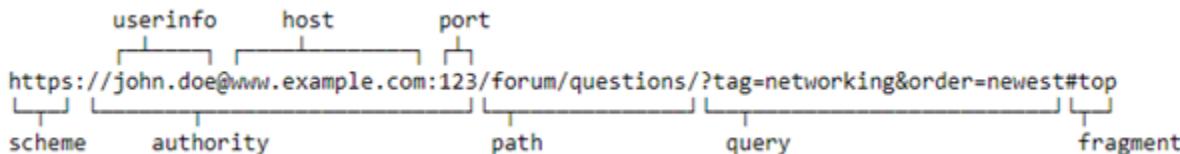
In the context of External Content, a Uniform Resource Identifier (URI) is a string that identifies the pathway to specific content that can appear in a running ViewApp.

All URIs adhere to predefined syntax rules.

- Generic URI syntax consists of a hierarchical sequence of components

```
URI = scheme:[//authority]path[?query][#fragment]
```

Example



- Each URI begins with a scheme name followed by a colon
Examples of common scheme names include http:, https:, and file:
 - An optional authority component preceded by two slashes (//)
 - Use information component consisting of a user name and optional password preceded by a colon followed by an at symbol (@)
`//username:password@`
 - Host subcomponent consisting of a registered hostname or IP address
 - A path component consisting of a sequence of path segments separated by a slash (/). A path is always defined for a URI, though the defined path may be empty
 - An optional query component preceded by a question mark (?), containing a query string of non-hierarchical data.

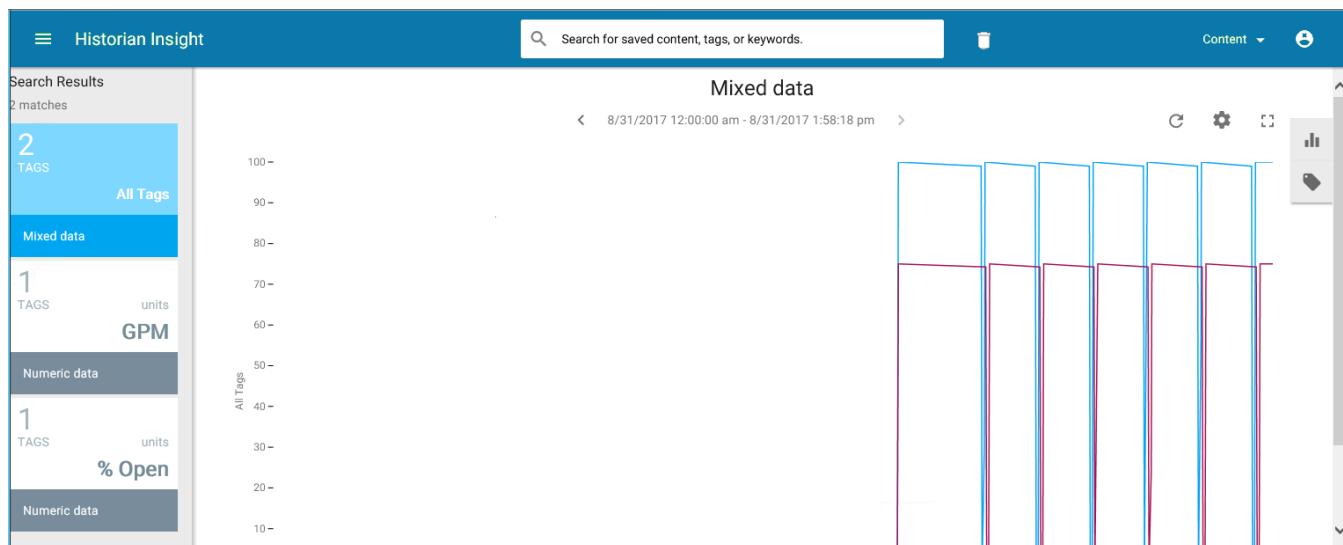
For more information about the format of a URI, see <https://tools.ietf.org/html/rfc3986>

The following examples show common URIs that identify External Content within a Galaxy

- `ftp://ftp.is.co.za/rfc/rfc1808.txt` -- ftp scheme for File Transfer Protocol services
- `file:///Z:/Media/Runtime%20Language%20Switching.mp4`
- `http://www.exampleserver.com/documents/index.html`

InSightApp

AVEVA OMI provides the InSight App to run an active InSight browser session within a pane of a running ViewApp. The InSight app provides a visualization of operational data by retrieving information from the Historian. The InSight app includes a search engine. Users enter keywords into the search box to retrieve relevant information.



Steps to configure the InSightApp

InSightApp instance level properties are evaluated only when the default values are changed and have a higher preference over chart configuration properties set with the InSight Global Editor.

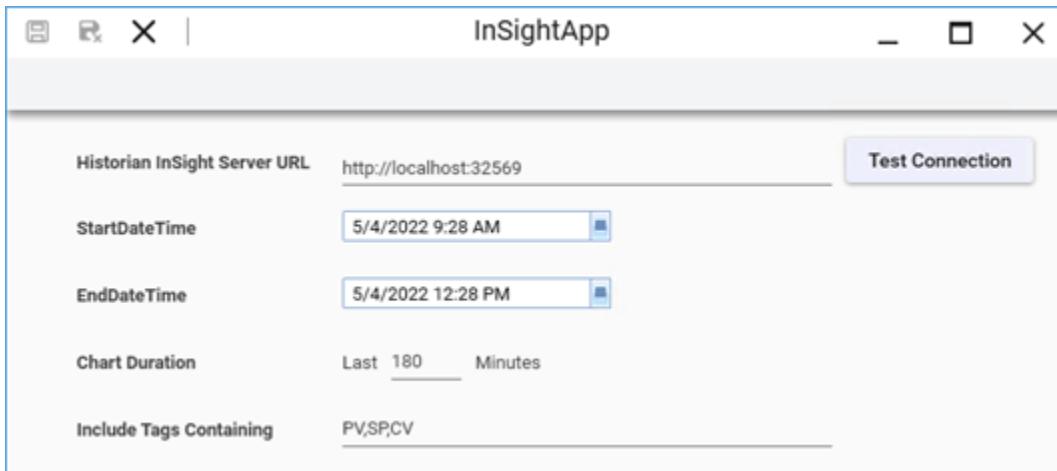
Important: You must complete all user security or authentication tasks required to access data from the Historian Client Web server. For more information about security requirements, see [InSightApp security requirements](#).

Configuring the InSightApp is a two-step process of validating the connection to the InSight server and setting app properties.

Chart configuration properties set by the Global Editor and properties set from the Layout or ViewApp editors may not be the same. Changing any property value from the InSight Global Editor does not appear at instance level during design time. Instance level properties continue to show default values at design time.

To configure the InSightApp

1. Open the System Platform IDE **Visualization** tab and locate the InSightApp in the AVEVA OMI Apps folder.
2. Double-click on the InSightApp to open the **InSightApp** dialog box.



The dialog box shows the default values assigned to the InSight Server URL, the chart duration, and tags recorded in an InSightApp chart.

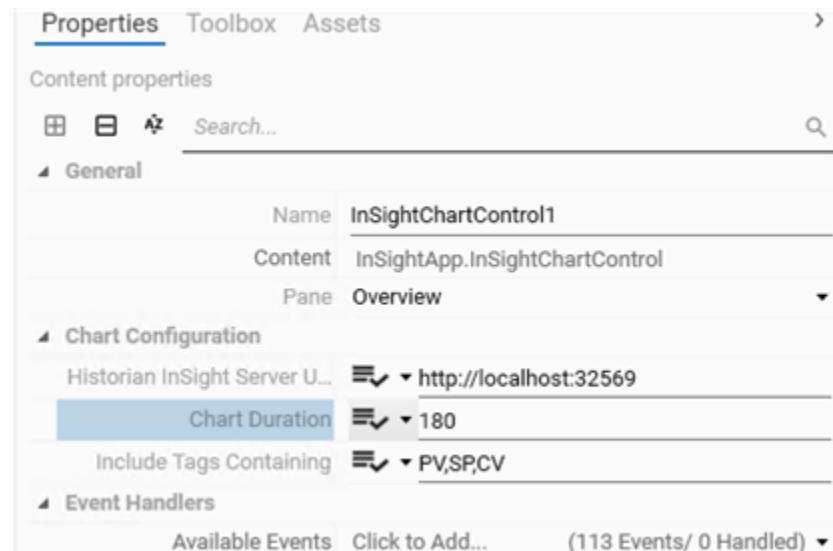
Note: These same configuration properties can be specified after the InSightApp has been placed on a layout pane.

3. Test the connection to the InSight server and update the URL if necessary.
4. Change the chart duration and the monitored tags if you do not want to accept the default values.
5. Save your changes.

To configure **InSightChartControl** properties

1. Open the Layout or ViewApp Editor and show the items listed in the **Toolbox** tab.
2. Select **InSightChartControl** from the **Toolbox** list to show its preview thumbnail.
3. Drag and drop the thumbnail on a layout pane.
4. Select the **InSightAppChartControl** thumbnail in the pane and select the **Properties** grid.

The **Properties** grid shows the **InSightChartControl** properties.



The **Content** and **Pane** fields show the assigned content type and the name of the pane in which the **InSightChartControl** is placed.

5. Enter the URL in the **Historian Insight Server URL** field.

The default Historian Client Web server URL is <http://localhost:32569>. A static or dynamic URL can be configured. The type of control property /attribute binding can be specified by selecting a value from the drop-down list.

Constant	Static binding to the specified URL. Enter a value in the data entry field.
In	Dynamic URL with read only binding by the control to a reference.
Out	Dynamic URL with write only binding by the control to a reference.

InOut	Dynamic URL with read/write binding by the control to a reference.
Reset	Reset the current property to its default value.

Connection settings to the Historians are configured on the server side. This configuration is normally done when System Platform is installed. The connection can be configured to either **favor** or **require** an encrypted connection, regardless of the URL (encrypted or unencrypted) that you configure here.

- **Favor trusted connections, but permit untrusted connections.** If this option has been configured, and if a trusted connection available, users will be prompted to select one of three options:
 - **Always use the trusted connection**
 - **Use the trusted connection this time**
 - **Continue with the untrusted connection (not recommended)**
6. Enter a chart duration time as the value of the **Chart Duration** property.
- The default chart duration is 180 minutes. Valid durations range from 1 minute to 5256000 (10 years). A static or dynamic **Chart Duration** property value can be configured and the type of app property /attribute binding can be specified with the property. Use the same switch settings shown for the **Historian Client Web Server URL** property to configure static or dynamic binding for the **Chart Duration** property.
7. Enter a comma delimited search string, which is used by Historian Client Web to show for **Include Tags Containing** property.
- The default search string is PV,SP,CV.
- The InSightApp uses this string to search for matching attributes. Attributes that match the search string appear in the InSight chart. This property cannot be empty. Valid property values can include numbers, alphabetic characters, and the following special characters \$ # _ .
- A static or dynamic **Include Tags Containing** property value can be configured and the type of app property /attribute binding can be specified with the property. Use the same switch settings shown for the **Historian Client Web Server URL** property to configure static or dynamic binding for the **Include Tags Containing** property.
8. If you want to create event handler scripts, select the **Available Events** data entry field to show a list of InsightApp events. For more information about creating event handler scripts, see [Add an Event Handler Script](#).
 9. Save your changes.

InsightApp security requirements

A logged in user must be an authorized Historian user to view a chart displayed by the InsightApp in a running ViewApp. InSightApp uses the SSO (Single Sign-On) feature.

Typically, users can connect to a Historian server from the InSightApp using an unencrypted (HTTP) connection. (Even without an encrypted connection, the user credentials exchanged during login are still encrypted.) You can also use an encrypted connection (HTTPS) for the REST API, and this requires configuring an X.509 certificate for TLS (transport layer security).

About TLS, HTTPS, and X.509 certificates

TLS allows for encrypted authentication credentials to be passed between a server and client. A certificate containing a private key is passed between the client and server to verify identification and allow access.

Using HTTPS ensures that communication between the client and server is encrypted, helping to prevent third parties from stealing or tampering with your data.

To configure the HTTPS connection to the Historian, you need an X.509 certificate. The certificate can be from a trusted authority or a self-signed certificate. During the installation and configuration of the Historian, you can import a certificate from a trusted authority if you have one, otherwise the configurator can create a self-signed certificate for you.

InSightApp supports OS Group or OS User based security. Galaxy authentication mode is not supported. Security is configured on the server-side (typically, at time of installation), and this configuration determines what happens when an InSightApp user connects using an unencrypted (HTTP) connection. There are two server-side connection options:

- **Favor trusted connections, but permit untrusted connections.** See [Favor trusted connections](#) for more information.
- **Require trusted connections.** For this option to work, the client must trust the certificate. See [Require trusted connections](#) for more information.

InSightApp runs in the context of the logged on ViewApp user. A message appears if security is not enabled or if an authorized Historian user is not logged on to the ViewApp. A message also appears if a secure connection was configured but is not available.

Favor trusted connections

Favor trusted connections, but permit untrusted connections

If this option was configured for the server, users are informed there is a trusted connection available, and they can decide how to proceed using one of three options:

You are using an **untrusted** connection to this Historian, but a trusted connection is available.

- [Always use the trusted connection](#)
[Use the trusted connection this time](#)
[Continue with the untrusted connection \(not recommended\)](#)

- **Always use the trusted connection**

If the user clicks this link, their browser will be permanently redirected to the HTTPS connection. Any future attempts to use the HTTP connection with the same browser are automatically redirected to the HTTPS connection without a prompt.

- **Use the trusted connection this time**

Clicking this link redirects the browser to the HTTPS connection, but only for this session. The next time a connection is made in a new browser session, the user is prompted to choose again.

- **Continue with the untrusted connection (not recommended)**

If the user clicks this link, the browser continues using the HTTP connection, but only for this session. The next time a connection is made in a new browser session, the user is prompted to choose again.

Require trusted connections

Require trusted connections (clients must trust this certificate)

If this option was configured for the server, and you are using a certificate from a trusted authority, users are redirected to the HTTPS connection.

If you are using an untrusted certificate, such as a self-signed certificate, the following informational message is displayed:

This Historian requires an encrypted connection, but the server is not fully configured in a way your browser will trust it. If you are an administrator, you can [learn more about this problem and how to correct it](#) and if you are not, please contact your administrator about this problem. If you accept the warning messages from your browser, you can switch to an [untrusted, but encrypted connection](#):

[Use the untrusted, encrypted connection](#)

Users can click **Use the untrusted, encrypted connection** to use the HTTPS connection.

Warning: It is important to understand the risks associated with using an untrusted self-signed certificate. The browser warnings encountered while using a self-signed certificate could also indicate that the server has been compromised or hijacked by a third party. To avoid the risk of conditioning users to ignore important security warnings, follow the steps in the next section to enable remote clients to trust the self-signed certificate.

Using a self-signed certificate

If you choose to use a self-signed certificate with the Historian, you are responsible for configuring all clients to trust that certificate. Clients that haven't trusted the certificate see a security warning in their browser.

For example, if the Historian was configured to use a self-signed certificate, users connecting with the Google Chrome browser see a warning message similar to the following:



Your connection is not private

Attackers might be trying to steal your information from [REDACTED] (for example, passwords, messages, or credit cards). [Learn more](#)

NET::ERR_CERT_AUTHORITY_INVALID

[Hide advanced](#)

[Back to safety](#)

This server could not prove that it is [REDACTED]; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

[Proceed to \[REDACTED\] \(unsafe\)](#)

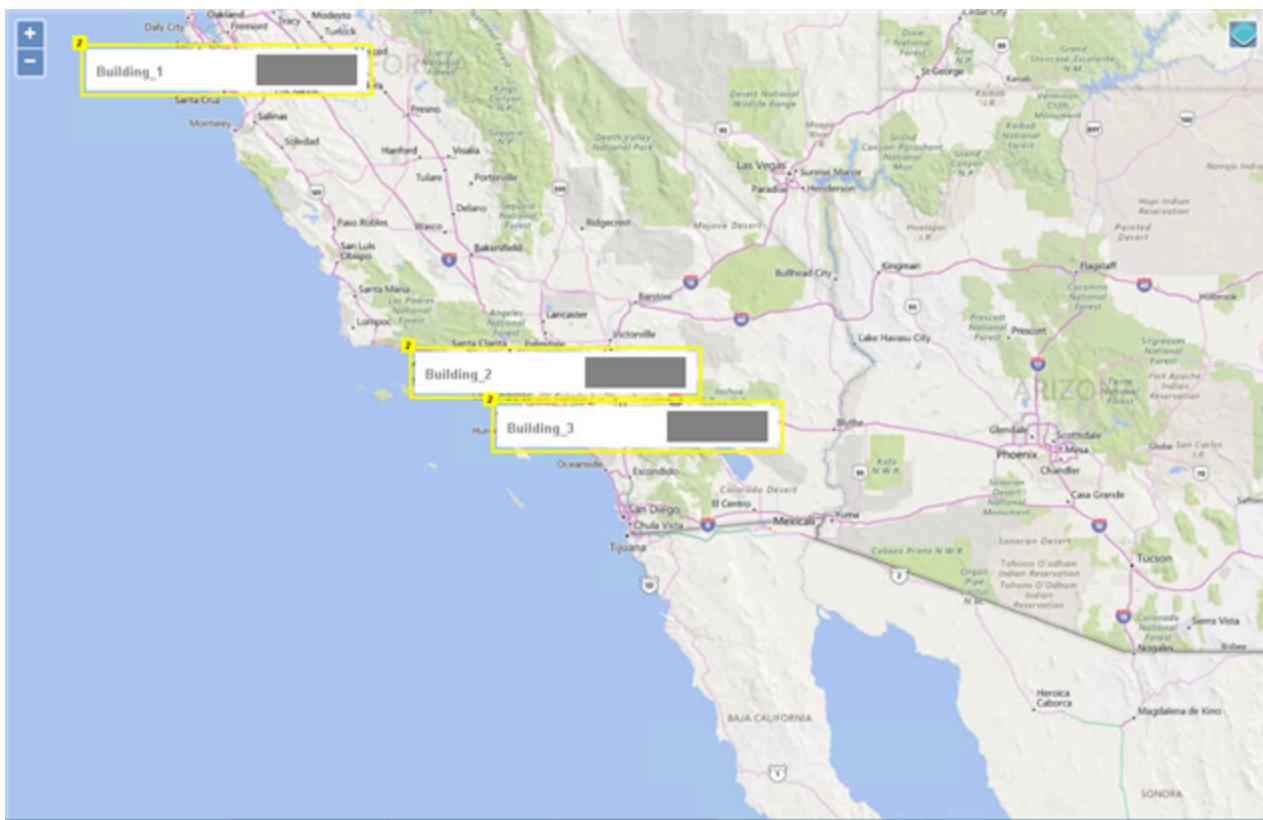
MapApp

The AVEVA OMI MapApp can display a map containing graphics within a running ViewApp. During run time, the map provides controls and touch support to enable users to pan to different areas of the map and zoom in or out to show more or less map detail. Graphics placed in a map typically represent business assets located within an area shown by the map. These graphics can include alarming to show the current state of processes at each business location.

You can add map pins to a map through an action script and change the assets displayed on a map at different zoom levels.

The following example map shows alarm indicators indicating the location of buildings and the current alarm state of each building. During runtime, users can click zoom buttons at the upper left corner of the map to zoom the view in or out. Different map data appears based on how you assigned your company's asset data to the zoom layers. Selecting the icon at the upper right corner of a map shows a list of base maps and overlays that can be selected to show different data within the current map view.

The MapApp supports touch gestures when a ViewApp runs on a touch-enabled device. Users can pan, zoom, or select items shown on a map using standard touch gestures.



Configuring the MapApp: overview

To include a map in an OMI ViewApp you must do the following:

1. Complete a set of preparatory tasks:

- Set up accounts with the supported map providers whose data you want to show in your ViewApps, and, using the map provider, define each map that you want to be able to use.
- Create user credentials to control access to the data shown by specific map vendors.
- Optionally, configure a local map server to save data from your map vendors.
- Create a set of graphics that will appear on your maps.

For more information and instructions, see [Before configuring the MapApp](#).

2. Configure global settings for the MapApp:

- Define the source maps. A source map definition includes a name for the map, the map provider, whether the map is a base map or an overlay, and additional configuration information, which varies depending on the map provider. You must define each source map that you will use in any ViewApp. When you configure the MapApp for use in a particular layout pane, you select which source map(s) to include in that pane.

The provider data can be hosted on either a local or remote map server. Most providers will require a web server, but other providers offer map tiles that can be stored locally in a shared folder.

- Create a set of named zoom layers. Each layer is a range between minimum and maximum zoom percentages.
- Place the graphics representing your company's assets onto each zoom layer at their proper geographic

locations. At runtime, each graphic will display only when the map zoom is set within the range for its zoom layer. You can assign any graphic to multiple zoom layers and to multiple locations on a zoom layer, if desired.

For more information and instructions, see [Configure global MapApp properties](#).

3. Add a MapControl to a layout pane and configure the properties for this particular map. These properties include which source map(s) to use, the initial location and zoom level, and more.

For more information and instructions, see [Configure a MapControl](#)

Note: "MapApp" refers to the OMI App that you can use to include a map in an OMI ViewApp. When you place the MapApp in a layout pane for use in a ViewApp, it creates a "MapControl".

Configuring the MapApp is a global process that you need to perform only once, though you can add or change information at any time. Each time you add the MapApp to a layout pane, it creates a new MapControl, which you must configure.

Before configuring the MapApp

Complete the following prerequisite tasks before configuring the MapApp.

- Set up accounts with the various map providers and, on each provider, define the map(s) that you want to use in the MapApp.

See [Configure global MapApp properties](#) for a list of map providers supported by the MapApp. Each map provider includes a unique set of properties that must be configured in the MapApp.

- Create user credentials if your map providers require an API key or a specific user account to access their data.

See [Create user credentials](#) for instructions to authenticate users by OS-based security credentials.

- If desired, create a local map server to store geographic data from the various supported map vendors. This is not required, but allows you to retrieve the map information at runtime from your local map server instead of from the map provider using the Internet.

See [Set up a map server](#) for an example of setting up and using an open source map server to store your map data.

Note: This example explains how to set up a map server using open source Java Runtime Environment (JRE) and another open source product called GeoServer. This example is intended to show the typical workflows to create a map server to store your map data on a local server. But, there are a wide variety of other commercial and open source products that can be used to build a map server and you are not restricted to the combination of products used in this example.

- Create a set of graphics to show the location of your assets on a map.

Design graphics large enough to be viewed easily at different map layers yet small enough not to overlap other graphics in the same layer. If you want, configure the graphic to show the current alarm state of some process variable occurring at the geographic asset.

Create user credentials

If one or more of your map providers require a user name and password to access their data, you must use the IDE to create a user credential for each map provider. When a user logs into a ViewApp, they can see a map layer only if they belong to a user group which has had a credential assigned for that provider. Layers which use a

provider that needs a credential are hidden if the user does not belong to a group which has had the credential assigned to it.

For more information about setting user credentials, see the Application Server Help or User Guide.

Important: Some map providers do not require user credentials. In that case, you do not need to create credentials to use with the MapApp.

Set up a map server

The following sample procedure describes how to install and configure a map server using the open-source Java Runtime Environment (JRE) and an open source map server called GeoServer. The procedure is intended only as an example to show the typical workflows to configure a web-based map server. You can use these products or other open source or commercial products as your local map server to save map data that will be shown by the MapApp.

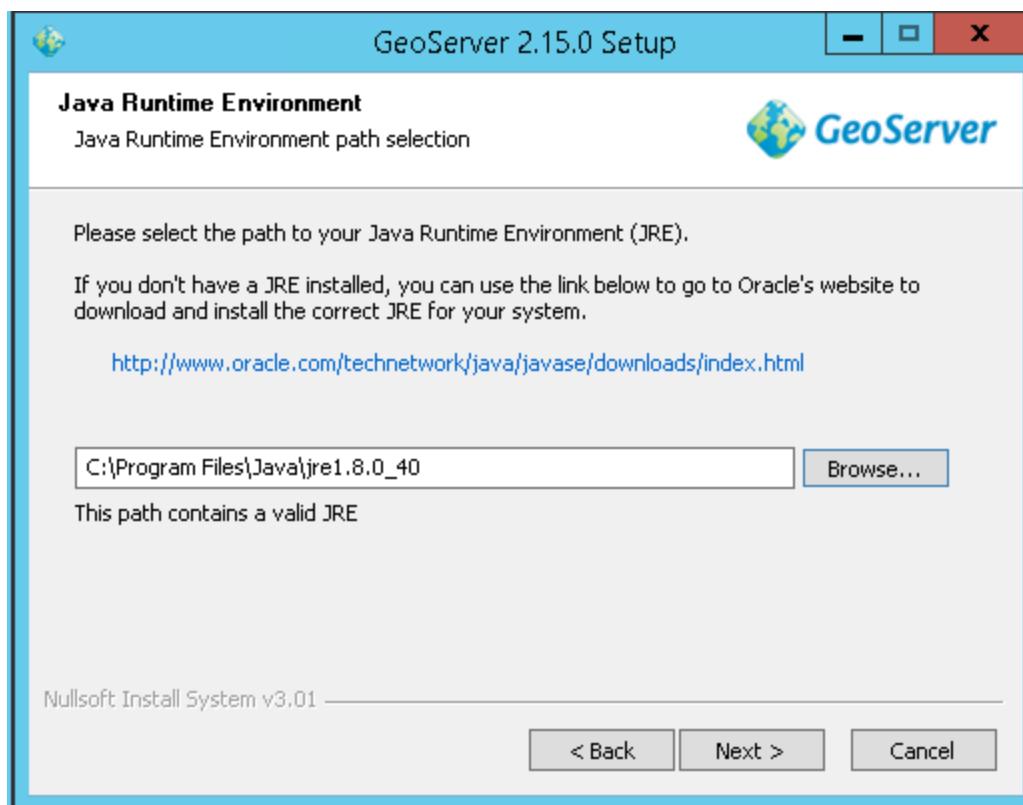
After you have set up your local map server and saved your desired source maps to it, you must configure the MapApp map sources to point to the maps' locations on the local server, not to the original providers' web sites. See [Map source properties](#) for information on defining map sources.

Note: For users to be able to view map data when connecting from another domain, you must enable Cross-Origin Resource Sharing (CORS) on the map server. CORS allows a JavaScript on one server to access resources on another server. **This can be a security risk**, so most servers disable CORS to reduce the likelihood of malicious scripts accessing resources on the server. Be sure to consider all of the security implications of enabling CORS before deciding whether to do so.

To configure a map server

This is a **sample procedure only**. You can use any available map server product that fits your needs.

1. Install a 64-bit version of Java Runtime Environment (JRE).
2. Install GeoServer and locate the folder path to the JRE.



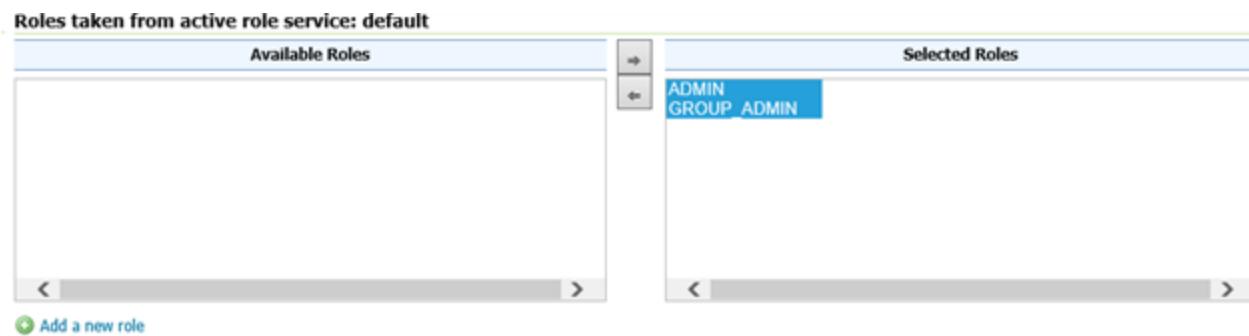
3. Accept the default data directory.
`C:\Program Files (x86)\GeoServer 2.15.0\data_dir`
4. Set the username and password of the GeoServer administrator.
The default administrator username and password are **admin** and **geoserver**.
5. Set the GeoServer web server port, which is 8080 by default.
6. Select the **Run manually** option for the type of installation.
7. Select **Install** to start installing the map server.
8. Select **Finish** to end the installation procedure.
9. Start GeoServer.
You can find the **Start GeoServer** option in the Windows Start Menu, or you can run `startup.bat` in the GeoServer folder(such as `C:\Program Files (x86)\GeoServer 2.15.0\bin`).
10. Open a web browser and log on to the GeoServer administrator page using your administrator credentials.
`http://localhost:8080/geoserver/web`

The screenshot shows the GeoServer welcome page at <http://localhost:8080/geoserver/web/?1>. The left sidebar contains links for About & Status, Data, Services, Settings, Tile Caching, Security, and Demos. The main content area displays a summary of 19 layers, 9 stores, and 7 workspaces. It also includes several warning messages about master and administrator passwords, cryptography, and software version (2.15.0). A note states that the instance is running version 2.15.0.

11. Select **Users, Groups, Roles** from the **Security** area in the left column of the administrator page.
12. Select **Add new user** from the **Users, Groups, and Roles** page.
13. Add a new user who is a member of a credential group and save your changes.

The screenshot shows the 'Add a new user' page. The left sidebar has the same structure as the welcome page. The main form is titled 'Add a new user' and asks to specify a new user name, password, properties, and associate groups/roles with the user. Fields include 'User name' (Operator1), 'Enabled' (checked), 'Password' (*****), and 'Confirm password' (*****). A 'User properties' section is visible below the form.

14. Move the available roles assigned to the credential user to the **Selected Roles** field.



15. Select **Authentication** from the Security area of the administrator web page.
16. Select default from the **Filter Chains** area of the **Authentication** web page.

Filter Chains									
Position	Name	Patterns	Check HTTP Method	HTTP methods	No Security	HTTP Session	SSL	Role Filter	Remove
1	web	/web/**,/gwc/rest/web/**,/			✓				✖
2	webLogin	/_spring_security_check/_spring_security_check/			✓				✖
3	webLogout	/_spring_security_logout/_spring_security_logout/							✖
4	rest	/rest/**						✖	✖
5	gwc	/gwc/rest/**						✖	✖
6	default	/**						✖	✖

The **Filter chain** web page opens.

17. Move anonymous to the **Available** list.



Configure global MapApp properties

The global properties for the MapApp consist of two types of information:

- Source map definitions. You must define each map that you want to acquire from a map provider for use in any ViewApp that includes a MapControl. You can define one or more maps from any or all supported map providers. Each source map is either a base map or a map overlay. (Some providers support only base maps, while others support both base maps and overlays.) During runtime, a base map displays as the opaque background and one or more overlays can display on top of it, with degrees of transparency that you set for each one.
- Zoom layers and the graphics they contain: You create zoom layers; each layer defines a zoom range (for example, between 10% and 20%). You then add graphics representing your organization's assets (or any other desired graphics) to each zoom layer at specific geographical locations. You can either enter the location or base it on the values of specified attributes. A graphic assigned to a particular zoom layer is shown only when the map zoom level is within the range for that zoom layer. Zoom layers and the graphics they contain are independent of the source maps, so the same graphics can appear at runtime in ViewApps

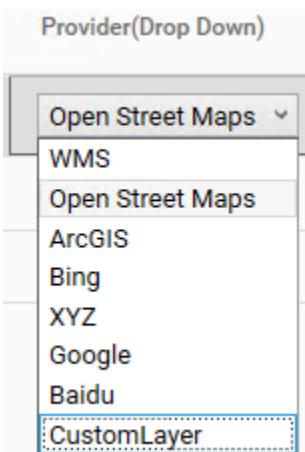
that use different source maps.

You must define these global properties before you can add MapControls to layout panes for use in ViewApps. However, you can go back and change or add to the global properties at any time. For more information about adding and configuring MapControls, see [Configure a MapControl](#).

Note: If you change the MapApp global properties, you need to redeploy any existing ViewApps that use the MapApp to make them reflect those changes.

To configure MapApp global properties

1. Open the System Platform IDE and select the **Visualization** tab.
2. Locate the MapApp in the following folder in the **Visualization** tab.
...\\Default Content\\4 Apps\\OMI Apps
3. Open the MapApp to show the **Map Settings** dialog box.
4. Define source maps:
 - a. Select the **Sources** tab.
 - b. In the first empty **Name** field, enter the name to give this source map.
 - c. In the **Provider** field, select the map provider.



The **Properties** area updates to include fields applicable to this provider. Also, the **Configuration** field shows parameters for this provider. You cannot make changes directly in the **Configuration** field; it reflects the information you enter in the **Properties** fields.

- d. Assign values for this source map in the **Properties** area. For more information about the properties of each map provider, see [Map Source Properties](#).

The map provider properties listed in the **Configuration** field update to show the properties you selected from the **Properties** field.

- e. If the map provider requires a credential, select the proper credential in the **Credential Name** field.

Note: If the map provider does not require user credentials, skip this step.

- f. Repeat steps b-e to define all of your source maps.

Note: If you want to use different source maps from the same provider, you must define each one separately.

5. Create zoom layers. Zoom layers allow you to determine what graphics are shown on a map based at a given zoom level. For more information, see [Zoom layers](#).

A 0 percent zoom level means no zoom, which is a view of the entire planet. A 100 percent zoom level is the maximum zoom supported, which shows a map view at the street level.

- a. Select the **Zoom Layers** tab.
 - b. In the first empty **Name** field, enter the name to give this zoom layer.
 - c. Enter the **Minimum Zoom** and **Maximum Zoom** percentages.
 - d. Repeat steps b-c to create as many zoom layers as you need.
 - e. Set one layer as the default zoom layer by selecting it, then right-clicking and selecting **Set as default**. When you add graphics to the zoom layers in the next step, they will be assigned to this layer unless you select a different zoom layer.
6. Add graphics to zoom layers at the desired locations.
- a. Select the **Locations** tab.
 - b. On the **Assets** or **Toolbox** tab, navigate to the desired graphic and drag it to the **Locations** grid.
 - c. In **Layer**, select a zoom layer for the graphic. The zoom layer determines when the graphic will appear at runtime. For more information, see [Zoom layers](#)
 - d. To specify the latitude and longitude where the graphic will appear on a map, select the **Properties** tab.

In the **Latitude** and **Longitude** fields, enter the coordinates where you want the graphic to appear. You can enter either static values or dynamic values with binding to a User-Defined Attribute (UDA).

To use static coordinates, enter:

- a. **Latitude:** +/- 0-90.0
- b. **Longitude:** +/- 0-180.0

To use dynamic coordinates, enter a reference to a UDA, in the format:

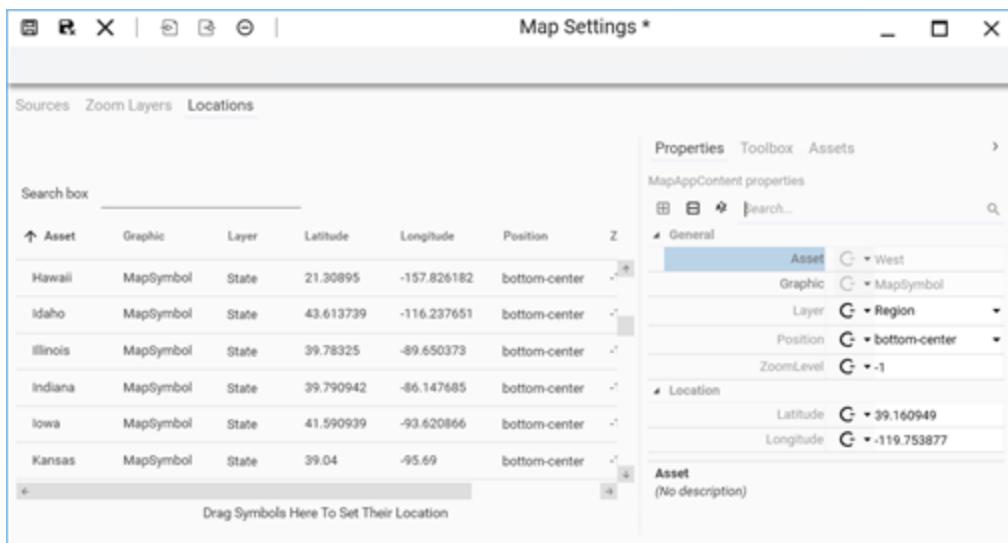
Asset_Name.Reference_Name

For example:

Static Coordinate Values	Dynamic Coordinate Values
Latitude C 47.617	Latitude C MyObject.TagName
Longitude C -95.645	Longitude C MyObject.OtherTag

The center bottom of the graphic will be placed at the specified location.

- e. Repeat steps b-d until you have added all desired graphics. You can add the same graphic to different zoom layers, and to multiple locations on a zoom layer.



Map source properties

The following tables list the properties you need to enter for each map data provider supported by the MapApp. If you use a local map server to store maps from your map provider, the URL you enter for each map source should point to the map on the local server, not to the web site for the map provider.

Important: You may need to set up accounts with some map providers to be able to access the data from their servers. These map providers require an API key or a specific account when you configure map source properties.

ArcGIS

Note: To use an ArcGIS map and include x,y (latitude,longitude) and optionally z (elevation) coordinates, use the XYZ map provider. The XYZ provider is described below.

Type	Base maps and map overlays.
Source URL	The URL for the ArcGIS REST service and the defined ArcGIS map you want to use. The format will be similar to this: <a href="https://<servername>.arcgisonline.com/ArcGIS/rest/services/Specialty/<mapname>/MapServer">https://<servername>.arcgisonline.com/ArcGIS/rest/services/Specialty/<mapname>/MapServer
Opacity	For overlays only, the opacity of the overlay. A number from 0 to 1 where 0 indicates 100% transparent and 1 indicates 100% opaque.

<p>Source Layer</p>	<p>ArcGIS maps can have multiple layers. You can choose which layers of the ArcGIS map to use in this source map.</p> <p>There are four Source Layer actions to select from. Select the action, then list the layers in the field to the right::</p> <ul style="list-style-type: none"> • show: Use only the listed layers. • hide: Use all layers except the ones you list. • include: Use the default layers plus the ones you list. • exclude: Use the default layers, but exclude the ones you list. <p>To list layers, use the layer number as defined on the ArcGIS map. Separate multiple layers with a comma. For example. 1,5 specifies layers 1 and 5.</p>
----------------------------	---

Baidu

<p>Type</p>	<p>Base maps only.</p>
<p>Source URL</p>	<p>The URL for the Baidu map that you want to use as a base layer.</p>

Bing

<p>Type</p>	<p>Base maps only.</p>
<p>API Key</p>	<p>Registered key to authenticate the MapApp to the Bing Maps server providing data.</p> <p>For information about obtaining an API key, see the Microsoft Web site. At the time of writing, this was the URL:</p> <p>https://www.microsoft.com/maps/create-a-bing-maps-key.aspx</p>
<p>Base sources</p>	<p>Base map layer types supported by Bing Maps.</p> <ul style="list-style-type: none"> • Road: Roads within a geographic area. • Satellite: Orthographic aerial and satellite imagery. • Hybrid: Orthographic aerial and satellite imagery with overlaid roads and labels. <p>You can select one or more base map layer types. During run time, users can select base sources to be</p>

	shown by the MapApp.
--	----------------------

CustomLayer

Type	Base maps and map overlays.
Zip File Name	The name of a ZIP file containing the code and associated files needed to define the custom layer. For information about creating the ZIP file, please see Create a custom layer ZIP file .

Google

Type	Base maps only.
API Key	Registered key to authenticate the MapApp to the Google Maps server providing data. For more information about obtaining an API key, see the Google maps platform web site. At the time of writing, the URL was: https://developers.google.com/maps/documentation/javascript/get-api-key

Open Street Maps

Property	Description
Type	Base maps only.
Source URL	URL of the map server providing base map tiles. At the time of writing, for a street map of the entire world, you could use: https://www.openstreetmap.org Even though map tiles provided by Open Street Maps are free, usage restrictions apply to the OSM server at https://openstreetmaps.org . If you use the OSM base map tile server, ensure your usage is in compliance with OSM license requirements.

WMS

WMS (the Web Map Service) does not refer to a specific map provider. Rather, it is an interface standard that is supported by multiple map providers, including the U. S. Geological Survey, ArcGIS, and many others.

Type	Base maps and map overlays.
Source URL	URL for the map you want to use. Different map sources use different URL formats; you can get the format from the map provider.
Source Layer	<p>When making a query to retrieve tiles from a WMS server, you must specify a layer, as map provider servers support multiple layers.</p> <p>If you are unsure about the layers available for a specific WMS server, you can query the available layers by entering the following URL in your browser:</p> <pre><Source URL> ?request=GetCapabilities&service=WMS</pre> <p>This query returns an XML file listing all of the server's capabilities. Under "Capability/Layer" you will find a list of layers. Each layer has a node called "name", this is the layer name that should be specified in this property.</p>
Custom Data	This will be sent as custom data for the WMS server.
Opacity	For overlays only, the opacity of the overlay. A number from 0 to 1 where 0 indicates 100% transparent and 1 indicates 100% opaque.

XYZ

XYZ is a function of the ArcGIS map provider to add tabular data that contains geographic locations in the form of x,y coordinates to your map. If the table also contains z-coordinates, such as elevation values, you can add tabular data as 3D content into your globe or scene. To add a table of x,y coordinates, the table must contain two fields: one for the x-coordinate and one for the y-coordinate. The values in the fields may represent any coordinate system and units, such as latitude and longitude or meters. A field for the z-coordinates that enables 3D geometry is optional.

Type	Base maps and map overlays.
Source URL	URL for your XYZ server, in a format similar to this: <code>https://<servername>/ArcGIS/rest/services/<mapname>/MapServer/tile/{z}/{y}/{x}</code>
Opacity	For overlays only, the opacity of the overlay. A number from 0 to 1 where 0 indicates 100% transparent and 1 indicates 100% opaque.

Create a custom layer ZIP file

Important: The instructions and explanations in this topic assume you have a basic knowledge of coding in

Javascript.

To use a custom layer with the MapApp, you must code the layer using JavaScript. There is a sample project that you can use for development at this site:

<https://codesandbox.io/s/zen-cloud-ieb2mm?file=/main.js>.

You can edit this sample code, preview the results as you go, then create the ZIP file and download it to your local computer whenever you want.

The sample contains these seven files:

- index.html
- index.js
- locations.json
- main.js
- map.js
- package.json
- webpack.common.js

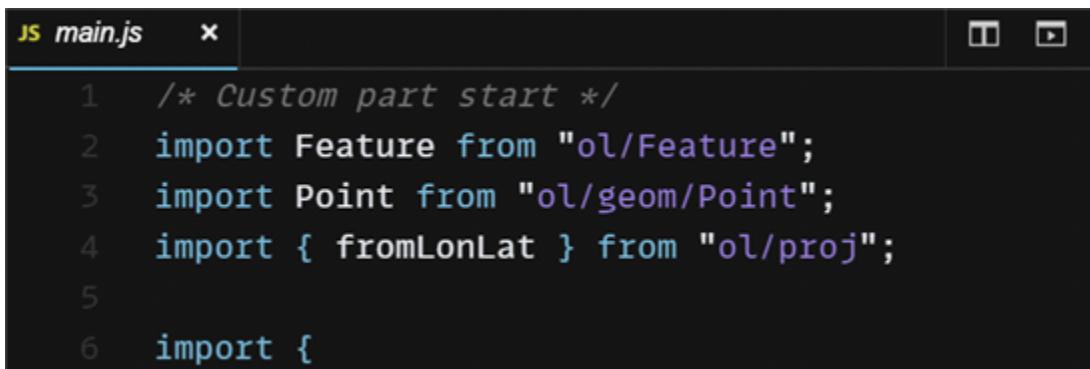
You will do your editing in the **main.js** file.

Note: You should develop the custom layer based on OpenLayer 6.15.1.

Editing main.js

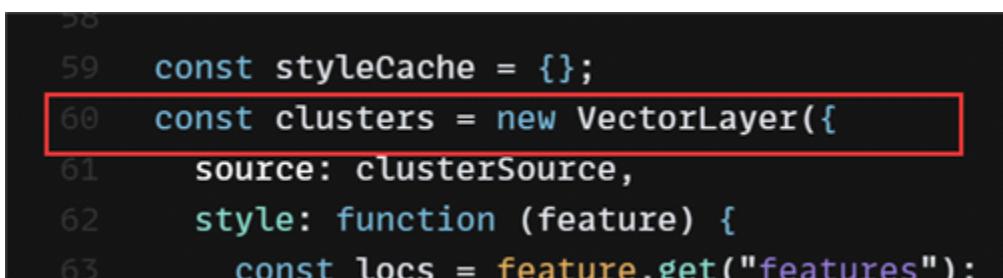
The editing changes in the **main.js** file fall into three main areas: creating the custom layer, exposing the custom layer so it can be used in a map, and associating assets with the layer.

1. The first part of the file is custom code to create the layer.



```
JS main.js  x
1  /* Custom part start */
2  import Feature from "ol/Feature";
3  import Point from "ol/geom/Point";
4  import { fromLonLat } from "ol/proj";
5
6  import {
```

Create the layer in the function `CreateCustomLayer`. Here, the layer name is `VectorLayer`:



```
58
59  const styleCache = {};
60  const clusters = new VectorLayer({
61    source: clusterSource,
62    style: function (feature) {
63      const locs = feature.get("features");
```

```
function CreateCustomLayer() {  
    const jsonData = require("./locations.json");  
    /*jsonData.forEach((loc) => {
```

2. Expose the custom layer so it can be used. The exposed function of this class is `customLayer.CreateCustomLayer`:

```
/* <important: DON'T REMOVE THIS SECTION START> Created layer  
assigned to vectorLayer.layer */  
  
var customLayer = {};  
customLayer.CreateCustomLayer = CreateCustomLayer;  
  
export default customLayer;  
/* <important: DON'T REMOVE THIS SECTION END>*/
```

Note: You cannot change the class name `customLayer` or the function name `customLayer.CreateCustomLayer`.

3. If you want users to be able to associate locations on the custom layer with assets, you can use code similar to the sample shown below.

The `registAssetChange` function takes the parameter `assetChange`. `assetChange` is a callback function that will trigger an asset change on the map. You can add any logic here and invoke the callback function to change the asset. For example, the example in the sandbox changes the specified asset when the user clicks a pin on the custom layer.

Note: You cannot change the function name `registAssetChangestomLayer`. The `assetChange` function must pass the asset name as a parameter.

```
137     customLayer.onMap = null;  
138     customLayer.registAssetChange = function (assetChange) {  
139         /* Parameter: assetChange is exposed function to trigger map  
140          * changes */  
140         customLayer.olMap.on("click", (e) => {  
141             customLayer.layer.getFeatures(e.pixel).then((clickedFeatures) => {  
142                 if (clickedFeatures.length) {  
143                     // clickedFeatures[0].assetName = "Plant_Area";  
144                     // assetChange(clickedFeatures[0].values_.features[0].  
145                     clickedFeatures[0].assetName = "UDO1";  
146                     assetChange(clickedFeatures[0].assetName);  
147                 }  
148             });  
149         });  
150     };
```

4. To have the code take further action when the asset is changed, use the `onAssetChanged` function. This function is triggered when the asset is changed--for example, when the user selects a different navigation node in the ViewApp. You can add any code to handle asset change logic based on the `assetName` parameter. In the sample below, it invokes the `view.setCenter` map function to set the map center position.

```
customLayer.onAssetChanged = function (assetName) {
    /* This function will be triggered when map asset changes.
     * user can add their own logic here, for example center
     * the map on the asset
     */
    var feature = {
        longitude: -103.795,
        latitude: 40.246
    };
    if (customLayer.olMap && customLayer.olMap.getView) {
        var view = customLayer.olMap.getView();
        if (view != null) {
            view.setCenter(
                ol.proj.transform(
                    [feature.longitude, feature.latitude],
                    "EPSG:4326",
                    "EPSG:3857"
                )
            );
        }
    }
}
```

Note: You cannot change the function name `onAssetChanged`.

If you want to set the map center position you must use the map function `view.setCenter`. However, if both a symbol and a pin are associated with the same asset, `view.setCenter` may set the map center to either one, because it may query the asset's position several times. When the navigation changes, the map needs to be zoomed to a certain level to set the pin to center.

Resource files

You can create JSON resource files to use in `main.js`. For example, the sample in the sandbox includes `locations.json`. To use a JSON file, import it with `requires(<path/filename>.json)`, as shown below, and use it directly.

```
16
17  const jsonData = require("./locations.json");
18  jsonData.forEach((loc) => {
```

You can also use other resource files, such as images, CSS files, and so on. When referring to these files, make sure the path to them is correct.

To match the MapApp path, `baseUrl` is exposed to specify the ViewApp path. All resource file paths should include `customLayer.bareUrl`, resulting in this format:

```
customLayer.baseUrl + "/<resource_file_relative_path>"
```

For example:

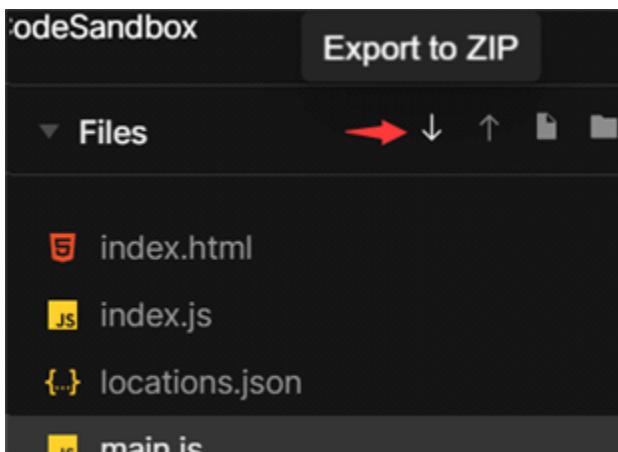
```
features[i].set(
  "myIcon",
  loc.WellSite_Shop === "SHOP"
    ? customLayer.baseUrl + "/data/building.png"
    : customLayer.baseUrl + "/data/oil.png"
);
```

Note: Other files found in the template in the sandbox are used to preview the map or compiled into the final .js file. Do not change or remove these files.

Generating the package

Once you have completed developing your custom layer, you need to create the ZIP file which can be used to define the custom layer to the MapApp. If you have used the sandbox described above, follow these steps:

1. Click the down arrow to export all of the files in the sandbox to a local file.



2. Install **node.js** if you do not already have it installed.
3. Launch a command prompt and change to the directory where you have saved the generated custom layer file.
4. Run this script to compile the custom layer with webpack:

```
npm run build_customLayer
```

```
C:\Users\wwuser\Desktop\ngfr61>npm run build_customLayer
> build_customLayer
> webpack --config webpack.common.js

asset CustomLayer.js 203 KiB [emitted] [minimized] (name: main)
orphan modules 3.53 MiB [orphan] 335 modules
runtime modules 670 bytes 3 modules
cacheable modules 736 KiB
  ./index.js + 97 modules 687 KiB [built] [code generated]
  ./locations.json 39.5 KiB [built] [code generated]
  ./node_modules/rbush/rbush.min.js 9.7 KiB [built] [code generated]
webpack 5.74.0 compiled successfully in 31026 ms

C:\Users\wwuser\Desktop\ngfr61>
```

If the compile is successful, this creates a **CustomLayer.js** file in the **dist** folder.

5. Copy all of your other resource files--JSON files, image files, CSS files, and so on--into the folder with **CustomLayer.js**.
6. Make sure the relative path is correct, then combine all of the files in the **dist** folder into a ZIP file.

You can now use the ZIP file when creating a custom layer as one of the source maps for the MapApp.

Important: The ZIP file must be less than 2MB in size, or you will not be able to use it in the MapApp.

Zoom layers

Zoom layers allow you to control which graphics appear on a map when it is zoomed to a particular percentage. Configuring zoom layers is a two-step process:

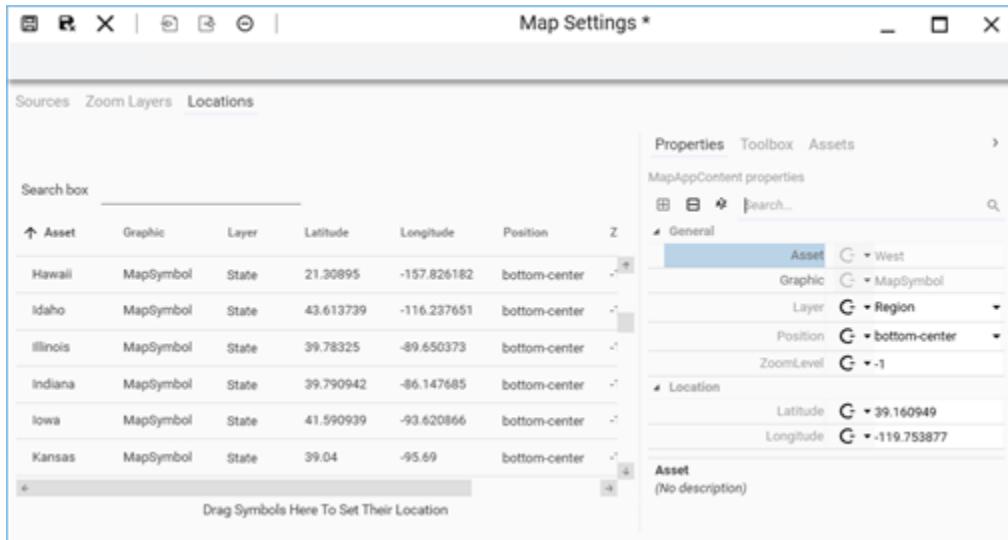
- Create one or more zoom layers: Each layer defines a minimum and maximum zoom percentage. At run time, if the zoom percentage is within these limits, the graphics that are assigned to this zoom layer appear on the map. You can define as many zoom layers as you like, and the zoom ranges that you assign to layers can overlap. For example, if you create one zoom layer for the range 10% to 30% and another for the range 25% to 50%, when a map is zoomed to between 25% and 30%, it includes graphics assigned to either zoom layer.
- Assign graphics to each zoom layer: You can assign one or more graphics to each zoom layer. When you assign a graphic, you select the zoom layer and assign the graphic to a position by specifying its latitude and longitude. You can assign a graphic to more than one zoom layer, and you can assign a graphic to more than one location on the same zoom layer. For example, if you have a graphic that represents a warehouse, you can assign that to a zoom layer at several locations, showing each warehouse in a city or state.

It may take some experimenting to get the transitions between zoom layers set to give the optimal appearance at run time. How close should a map have to be zoomed in before you show individual locations, or assets within those locations, instead of larger state or region symbols? You can modify existing zoom layers, or create new ones, at any time. Overlapping the boundaries of zoom layers can also help, so that at some percentages the user can see, for example, both smaller regions and larger individual locations,

Note: If you change the zoom layer definitions or graphics, you need to redeploy any existing ViewApps that use the MapApp to make them reflect those changes.

Location assets

You can drag and drop graphics on the **Locations** grid to place them on a map layer at specified coordinates. Each row of the **Locations** grid shows the assigned configuration values of a graphic that will appear on a map during run time.



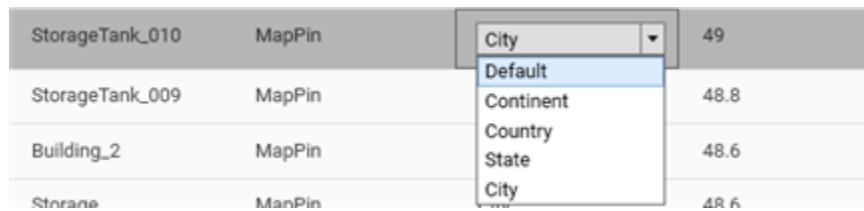
To place graphics on a map

1. Select the **Locations** tab from **Map Settings**.
2. Locate the graphic that you want to place on a map from the **Toolbox** or **Assets** tabs.
3. Drag the graphic and drop it in the **Locations** grid area.

If you drop a graphic from an asset instance, the **Asset** field shows the name of the asset. If you drop a graphic from the **Toolbox** tab, the **Asset** field is empty. In either case, the **Graphic** field shows the name of the graphic.

Asset	Graphic	Layer	Latitude	Longitude
Building_1	MapPin	Default	0	0
	CapperDriver	Default	0	0

4. Select the graphic in the list.
A dark background identifies the selected row.
5. Select the **Layer** column for the graphic to show a list of the defined map zoom layers.



Each graphic is initially assigned to the **Default** zoom layer.

6. Select the zoom layer where you want to place the graphic. The graphic will appear during runtime only when a map is zoomed to a percentage within the range for that zoom layer.

7. Select the **Properties** grid to show fields to specify the latitude and longitude where the graphic will appear on a map.

The center bottom position of a graphic is placed on a map at the specified location.

Coordinate values can be specified as static values or as dynamic values with binding to a UDA by the position of the **Latitude** and **Longitude** switches.

Static Coordinate Values	Dynamic Coordinate Values
Latitude 47.617	Latitude MyObject.TagName
Longitude -95.645	Longitude MyObject.OtherTag

Static coordinates are specified as:

- Latitude: +/- 0-90.0
- Longitude: +/- 0-180.0

Dynamic coordinates are specified as:

Asset_Name.Reference_Name

8. Enter **Latitude** and **Longitude** values.

You should see the coordinate values you entered appear in the **Latitude** and **Longitude** columns of the **Locations** grid. Click once in the grid to refresh the grid values.

9. Save your changes.

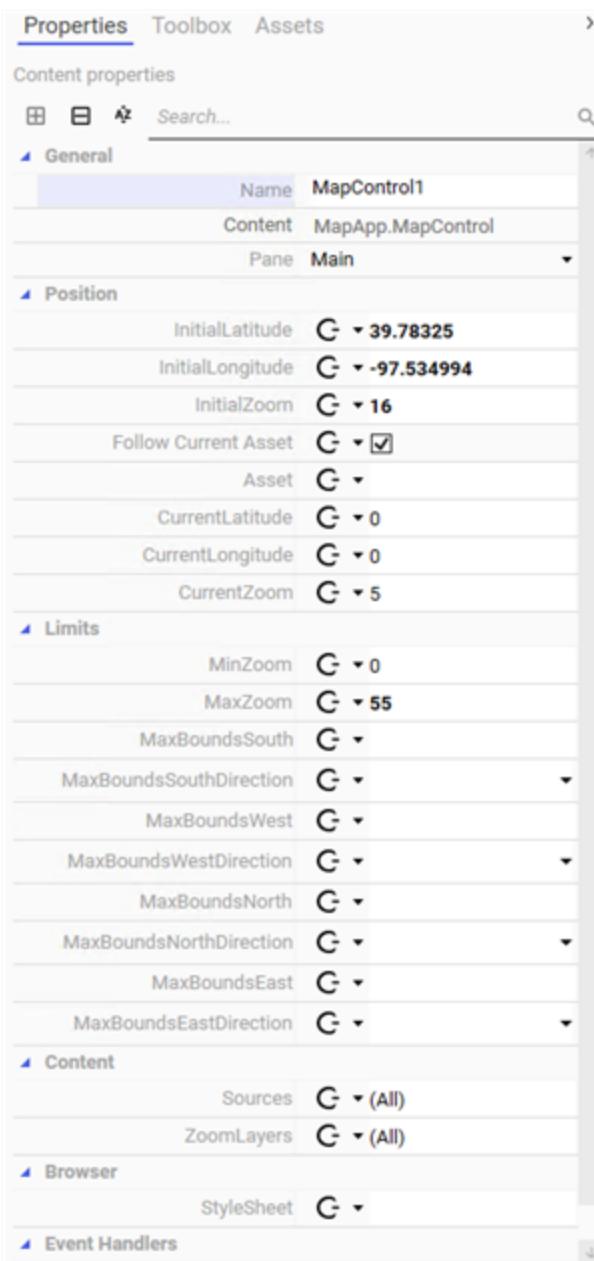
Configure a MapControl

You place a MapControl into a ViewApp by selecting it from the **Toolbox** tab and dragging it to a layout pane.

Note: You must configure MapApp global settings before you can use a MapControl in a layout pane. See [Map source properties](#).

To configure MapControl properties

1. Open the Layout Editor or ViewApp Editor.
2. Select the **Toolbox** tab and locate the MapApp in the list of controls and graphics.
...\\Default Content\\4. Apps\\AVEVA OMI Apps\\MapApp
3. Select **MapApp** from the list of AVEVA OMI apps to show the **MapControl** thumbnail at the bottom of the **Toolbox** pane.
4. Drag and drop the **MapControl** thumbnail to a layout pane.
5. Select the **MapControl** thumbnail within the pane and select the **Properties** grid to show the MapControl's properties.



6. Assign a value to each MapControl property. The following table describes each property.
7. After specifying values for all properties, save your changes.

The following table contains descriptions of MapControl properties. You can reset any property to its default value by selecting the arrow before the entry field and selecting **Reset**.

Property	Description
InitialLatitude	Latitude of the initial center point map position in decimal degrees. Valid latitude values are +/- 0-90.
InitialLongitude	Longitude of the initial center point map position in decimal degrees. Valid longitude values are +/- 0-180.

InitialZoom	Zoom level percentage of a map when initially displayed during run time.
LinkCurrentAsset	Like all AVEVA OMI Apps, the MapApp can follow the currently selected asset (context) and will automatically pan and zoom the map to display the asset's location and associated graphic. If you select LinkCurrentAsset , map navigation shows the following zoom and location behaviors when an asset is selected: <ul style="list-style-type: none"> • Centers the map to a selected asset when the map is opened. You can use the assets to navigate within a ViewApp. For instance, you can display all the states and show a marker for each one of them. Then, by selecting a state from a map, you can set the focus of the ViewApp to a separate pane that shows details about the state. • The map will zoom to the selected asset's zoom layer plus 1 percent. • If no asset is selected or a selected asset is not located on a map, the map shows the initial zoom layer and map center point.
MinZoom	The minimum zoom percentage to which the map can be zoomed out (0-100%) during run time.
MaxZoom	The maximum zoom percentage to which the map can be zoomed in (0-100%) during run time.
MaxBoundsSouth	Latitude of the southern map boundary in decimal degrees (+/- 0-90) to constrain the vertical panning movement of the screen viewport midpoint to the bottom boundary of a map.
MaxBoundsSouthDirection	Cardinal direction of the south boundary value (S/N).
MaxBoundsWest	Longitude of the western map boundary in decimal degrees (+/- 0-180) to constrain the horizontal panning movement of the screen viewport midpoint to the left boundary of a map.
MaxBoundsWestDirection	Cardinal direction of the west boundary value (E/W).
MaxBoundsNorth	Latitude of the northern map boundary in decimal degrees (+/- 0-90) to constrain the vertical panning movement of the screen viewport midpoint to the

	top boundary of a map.
MaxBoundsNorthDirection	Cardinal direction of the north boundary value (S/N).
MaxBoundsEast	Longitude of the eastern map boundary in decimal degrees (+/- 0-180) to constrain the horizontal panning movement of the screen viewport midpoint to the right boundary of a map.
MaxBoundsEastDirection	Cardinal direction of the east boundary value (E/W).
Sources	<p>Source maps configured in the MapApp settings. The default is (All). To show only some source maps, specify the source maps by name, separated by commas. For example:</p> <p>OSM,Bing,TemperatureOverlay</p> <p>Note: To set the value to All, you must include the parentheses: (All)</p>
ZoomLayers	<p>Map zoom layers configured for the Map app. (All) is the default value, MapApp zoom layers to show symbols from. The default is (All). To show symbols from only some zoom layers, specify the zoom layers by name, separated by commas. For example:</p> <p>country,state,city</p> <p>Note: To set the value to All, you must include the parentheses: (All)</p>
Style Sheet	Sets the Cascading Style Sheet (CSS) for the map webpage.

More details on bounding box properties

Together, a MapControl's MaxBounds directional properties construct a bounding box to restrict the viewable area within a map that can be shown by user's screen viewport, regardless of the map zoom level.



Rather than restrict the viewable area of a map to the area strictly within the bounding box, the MapApp enables users to extend the screen viewport to its center point beyond the borders of the bounding box.



When a zoom level is set to a very low percentage to show the maximum map area, a bounding box does not significantly reduce the viewable map area because moving the center of the screen viewport to the border of a bounding box will likely show the majority of the map.

Specify zoom level percentage ranges of the layers of your map to minimize the amount of zooming or panning required to see the entire area within the bounding box. Set the zoom percentage layer ranges more narrowly to be able to show assets at different map layers with reduced zooming and panning.

Graphic limitations of graphics in the MapApp

The MapApp can show industrial graphics on a map. The MapApp relies on an HTML5 rendering engine to show the mapped industrial graphics. The HTML5 engine has some limitations and does not fully support all industrial graphics properties or animations.

This topic describes all known limitations of the HTML5 engine when it renders industrial graphics properties or animations. The HTML5 rendering engine will be enhanced in subsequent product releases to provide additional support for graphic properties and animations.

Properties supported by all graphic elements

The following graphic element properties are supported for all graphics that appear in a MapApp.:

- Angle
- X
- Y
- Width
- Height
- Start
- End
- Enable
- Visible
- AbsoluteAnchor
- RelativeAnchor
- Transparency
- ElementStyle
- OwningObject
- Custom Property Overrides

Note: The ElementStyle property enables users to select a defined element style and apply it to a graphic element or animation. An element style includes color properties like FillColor, LineColor, or UnFilledColor, which are unsupported by the HTML5 rendering engine. These limitations still apply if you select an element style that contains unsupported color properties.

Properties supported by all graphic elements with some limitations

The following table lists the properties of graphic elements that can be incorporated in the MapApp with some HTML5 rendering limitations.

Graphic element property	Property limitations
FillColor	The Pattern and Texture properties are not supported.
UnFilledColor	The same gradient color limitations of the FillColor property apply to the UnFilledColor property. There is no support for the Pattern and Texture properties.

Graphic element property	Property limitations
LineColor	The Pattern and Texture properties are not supported.
Font	TheMapApp only supports the Font, Size, and Bold or Regular Font Style options of the Font property.
TextColor	Only the Solid Color option of the TextColor property is supported.
FillOrientation	RelativeToScreen is not supported.
FillBehavior	Will always be set to Both.

Supported graphic elements and additional properties

The following table lists the graphic elements that can be incorporated with some limitations. Any graphic element not listed in this table is not supported.

The table includes a column that lists any limitations of the supported graphic element properties. When using graphics that incorporate color properties, the colors have the same limitations applicable for FillColor, LineColor, and other related graphic properties.

Graphic elements	Supported properties	Limitation notes
Rectangle Rounded Rectangle Ellipse Closed Polygon 2 Point Pie 3 Point Pie Closed Curve Path 2 Point Chord 3 Point Chord	Fill Style: HorizontalDirection HorizontalPercenFill Line Style: LineWeight LinePattern	No Limitations
Line Polyline Curve 2 Point Arc 3 Point Arc Connector	Line Style: LineWeight LinePattern StartCap EndCap	StartCap: Only Flat is supported. EndCap: Only Flat is supported.
Button	ButtonStyle Text	ButtonStyle: Only Standard button style is supported. Word Wrap: Any caption that

Graphic elements	Supported properties	Limitation notes
	Word Wrap FillOrientation FillColor Alignment	exceeds the button width will be truncated. FillOrientation: Only RelativeToGraphic is supported FillColor: Only Solid color is supported. If you have gradients with multiple colors selected, they will be converted to a single color using the first color. Alignment: Only Centers alignment is supported.
Text	Caption Alignment	Alignment: Only left-top supported.
Image	HasTransparentColor TransparentColor ImageStyle	ImageStyle: Only Stretch is supported.
Text Box	Text TextFormat WordWrap LineWeight LinePattern Alignment Font	WordWrap: Only the True option is supported.
Status	Graphics Expression	Status Style: Only Default configuration is supported.

Graphic elements	Supported properties	Limitation notes
Web Alarm Client (EAC)	Alarm Mode SAL Graphics Column Details Queries and Filters Time Settings Run-Time Behavior	Alarm Mode <ul style="list-style-type: none"> Client Mode: Only supports Current Alarms, Recent Alarms and Events. Alarm Query: Supports InTouch and ArchestrA alarms Use Default Ack Comment: Only supports comments showing Alarms and Events. SAL Graphics: SA_Alarm_RuntimePage and SA_Alarm_RuntimeBanner are not supported. Column Details: Does not support sorting. Queries and Filters: Only supports selected Query and Filter at design time. Does not support runtime selection of different query or filter. Does not support relative references in alarm query. Time Settings: Not supported. Run-Time Behavior: Only supports show heading, show grid and allow column resizing. Does not support Ack, Shelve, Unshelve, Hide, Unhide, and Freeze operations. Note: WindowViewer must be running to view InTouch or Application Server alarms. Note: Does not support embedded alarm client's methods or properties.

Graphic elements	Supported properties	Limitation notes
Web Trend Client	Pens Appearance Options	Pens Does not support historical data backfill. Only supports Show, Pen Name (As description), Expression or Reference Pen Details: Min, Max Pen Options: Color Plot type(all will be treated as Line) Appearance PlotArea: Single tag mode, Grid (Show vertical grid, show horizontal grid, Color) X time axis: Show cursor(Cursor1:Color), Number of values Y value axis: Number of values, Value axis label (Multiple Scales, Single Scale) Options Only supports Retrieval:Trend Duration Historical Sources: Not Supported Data Binding: Not Supported Event: Not Supported Note: Does not support the Trend Client's methods and properties.
Trend Pen	Data Source	Does not support historical data backfill. Trend time Period: Does not support Fixed Type.
Connection Point		Size Property: Is not supported Will not be displayed if no connector is using it.

Supported animations

The following table lists animations supported by the MapApp. When using animations that incorporate color properties, the selected animation colors have the same limitations applicable for FillColor, LineColor, and other related graphic element properties. Furthermore, the HTML5 rendering engine does not support writing values.

Any type of animation that updates attribute or custom property values is not supported.

Note: The following table lists only supported animations. Any animations not listed are not supported by the MapApp.

Animation	Limitations
Alarm Border	Alarm Border animation shows an alarm border around a graphic element whose visible appearance represents the graphic element's alarm severity and acknowledgement status. The appearance of each alarm border is defined by an associated element style. Not all element style properties are supported. For instance, if Gradient is specified as the value of an element style's Line Color Override option in an alarm border element style, it will not be supported.
Element Styles	<p>Refer to the graphic elements limitations regarding color, line, and font. The same limitations applicable to graphic elements also apply to Element Style animation.</p> <p>Expression Or Reference: Time and Elapsed Time are not supported.</p> <p>Style only gets refreshed when changes are made in the Galaxy Style Library and the ViewApp is redeployed.</p>
Visibility	No limitations.
Fill Style Line Style Text Styles	<p>Refer to graphic element limitations regarding color, line, and font. The same limitations applicable to graphic elements also apply to Element Style animation.</p> <p>Expression Or Reference: Time and Elapsed Time states are not supported.</p> <p>TextStyle: Text Style animation cannot be applied to CheckBox or ComboBox graphic elements.</p>
% Fill Horizontal % Fill Vertical	Orientation: only Relative to Graphic is supported.
Slider Horizontal Slider Vertical	<p>Does not support CursorAnchor.</p> <p>WriteData:only On Mouse Release is supported when writing to a custom property.</p> <p>Does not support ShowTooltip</p>

Animation	Limitations
Width Height	Origin is not a supported value of the Anchor property.
Orientation	The Relative Anchor or RelativeOrigin are not supported when setting Orientation animation. (Only supported anchor position is Center).
Value Display	Text Format: If you specify integer attributes in an expression, no decimal points appear in the formatted number. States: The Time and Name states are not supported.
Disable	The RadioButton graphic element does not support the Disable Animation option.
Point	Point animation is not supported by the Curve and ClosedCurve graphic elements.
Tooltip	A MapApp does not support Tooltip animation. An InTouch Web Client supports Tooltip animation except for Image, Radio Button, Button, and Tex graphic elements.
Action Scripts	Actions scripts are supported except for functions that invoke dialogs. The exception is the ShowGraphic animation, which will replace the current graphic by the one specified in the call. Tooltip animation is not supported on a grouped or embedded graphic.
ShowSymbol	Title: Only supports the default option. Type: Supports Modal and Modeless when the Close button option is selected. Position: Only supports Center when the Client Area x and y values are 0,0. Size: Only supports the Relative to Symbol option. Shortcut: Not supported.
Blink	Group and embedded graphics are not supported.
Push Button	Only supports binding with a custom property. In various states the push button animation supports different actions:

Animation	Limitations
	<ul style="list-style-type: none"> • States: Boolean; Action: Only supports Toggle • States: Analog; Action Only supports Toggle/ Increment/Decrement/Multiply/Divide; Send Value: "Continuously while button is pressed" not supported • States: String; Action: Direct/Set not supported; Send Value: "Continuously while button is pressed" not supported <p>Shortcut: Is not supported.</p>

Major unsupported industrial graphic features

The HTML5 rendering engine will be enhanced with each product release. The following list summarizes the current major limitations of the current HTML5 rendering engine.

- Setting graphic properties using a script: Only the following properties are supported for reading or writing to a graphic element property using a script:
 - Position (X,Y)
 - Angle
 - Size(Height,Width)
 - Angle
 - StartAngle
 - SweepAngle
 - Text
 - Visible
 - Transparency
 - ElementStyle
- Graphic Library

Several graphics in the Industrial Graphics library incorporate graphic elements or color settings that are not yet supported by the HTML5 rendering engine.

- Writing values

The current HTML5 rendering engine cannot write values back to graphic or animation properties. All graphics that appear on a map are read only.

- Multiple ViewApps on the same node

The MapApp retrieves map content and graphics from a web server. The web server starts on demand when the first ViewApp loads the MapControl.

If changes are made to a graphic when running multiple ViewApps simultaneously, the ViewApp that started the web server must be deployed. Otherwise, the ViewApps do not show the graphic updates during run

time.

MapApp enhancements

The MapApp includes several enhancements that make it easier to add map pins to a map using an action script and the ability to display assets on a map by zoom levels.

Enhanced MapApp scripting support

You can dynamically add pins to a map displayed by the MapApp using an action script. The AddContent() function has been added to the MapApp.

```
MyContent.MapControl1.AddContent(Map_information)
```

Action script to add one map pin to a map.

```
' Use "AddContent" to add one map pin for US
dim usMapInfo = new aaGraphic.GraphicInfo;
usMapInfo.GraphicName = "MECRE_KSA.zMAP_ICON_Bottom_Left_3";
usMapInfo.Identity = "US_MapPin1";
usMapInfo.Latitude = 35.99;
usMapInfo.Longitude = -97.99;
' Support MaxZoom/MinZoom
usMapInfo.MinZoom = 15;
MyContent.MapControl1.AddContent(usMapInfo);
```

As MinZoom=15 in runtime the map pin will only appear if the minimum zoom level is 15. If the MaxZoom is also set then the map pin will only appear between the MinZoom and MaxZoom values.

Action script using an array to add multiple map pins to a map.

```
' Use "AddContent" with array parameter to add two map pins for Asia
dim asiaMapInfo1 = new aaGraphic.GraphicInfo;
asiaMapInfo1.GraphicName = "MECRE_KSA.zMAP_ICON_Bottom_Left_3";
asiaMapInfo1.Identity = "ASIA_MapPin1";
asiaMapInfo1.Latitude = 30.42;
asiaMapInfo1.Longitude = 149.46;
dim asiaMapInfo2 = new aaGraphic.GraphicInfo;
asiaMapInfo2.GraphicName = "MECRE_KSA.zMAP_ICON_Bottom_Left_3";
asiaMapInfo2.Identity = "ASIA_MapPin2";
asiaMapInfo2.Latitude = 35.42;
asiaMapInfo2.Longitude = 149.46;
dim asiaMapInfo[2] = as aaGraphic.GraphicInfo;
asiaMapInfo[1] = asiaMapInfo1;
asiaMapInfo[2] = asiaMapInfo2;
MyContent.MapControl1.AddContent(asiaMapInfo);
```

Action scripts can include the Set zoom level range to show or hide map pins that are added.

You can dynamically remove a pin from a map displayed by the MapApp using an action script and the RemoveContent() function. You can also use the ClearAllContents() function to clear all the pins added using the AddContent() function. The RemoveContent() allows you to remove one map pin at a time while the ClearAllContents() function will clear all the pins in one function call.

Map App zoom level enhancements

You can set the map zoom levels for an asset from the **Locations** tab of the MapApp.

Asset	Graphic	Layer	Latitude	Longitude	Position	ZoomLevel
Enterprise	MapPin	Default	33.650066	-117.6931	bottom-center	-1
Site	MapPin	Continent	33.650066	-117.6931	bottom-center	-1
Plant	MapPin	Country	33.650066	-117.6931	bottom-center	-1
Plant_Area	MapPin	State	33.650066	-117.6931	bottom-center	-1
Sys	MapPin	City	33.650066	-117.6931	bottom-center	-1

If an asset is assigned a zoom level of -1, the map will follow the current zoom level. If the assigned zoom level exceeds the maximum zoom level of the map, the map will be assigned the maximum zoom level of the map.

Current limitations of the MapApp widget

Element	Limitations
MapApp widget script statements in the form: <code>MyContent.<MapControlName>.AddContent(info)</code>	Limited by the lack of layout scripting support. The AddContent and RemoveContent functions only work correctly for the MapApp control running in AVEVA OMI ViewApps.
MapApp script statements in the form: <code>Support <WidgetName>.AddContent()</code>	Limited by the lack of support in an OMI runtime environment, which does not support the MapApp widget. The AddContent and RemoveContent functions are only supported on the WebClient.
MapApp widget script statements in the form: <code><WidgetName>.RemoveContent()</code> , and MapApp control script statements in the form: <code>MyContent.<MapControlName>.RemoveContent(info)</code>	Do not support a reference value for the identity. Only support removing the map pin added using the AddContent() function. Locations added using the Map Settings will not be removed.

Export and import map source data

The MapApp includes separate import and export utilities to be able to transfer your map source data. **Import** and **Export** buttons appear on the menu bar of the **Map Settings** dialog box.



Both the import and export utilities use a comma-delimited CSV file to save the map source data. You can edit the file with Excel or text editors. When you export map source data, you must select a folder location where the

export CSV file should be saved. Likewise, you must navigate to the folder containing the CSV file when you want to import the contents of the file to **Map Settings**. You see a message after your map settings file was successfully imported or exported.

The import/export file contains separate sections for the data specified in the tabbed **Sources**, **Zoom Layers**, and **Locations** areas of the **Map Settings** dialog box. Each section shows how the map source data should be ordered within a row, which corresponds to the order of the grid column headings. For example, the **Sources** section of the file shows the sources map data should be ordered as:

Name, Provider, Configuration

which correspond to the grid column headings of the **Sources** area.

Created on: 3/7/2017 7:17:49 PM

:SOURCES

:Name,Provider,Configuration

OSM,Open Street Maps,"{""layerType"":""Base Map""，《"url": "openstreetmaps.org"}"

Bing,Bing,"{""aerial"":false,""aerialWithLabels"":false,""apiKey"":""An6Grh5_YKz-CYnREoNn3nOaBU_r1nVEoc7o8HpSj3GWcvjseEguPvN3rJkZ95T""，《"colinsBart"":false,""layerType"":""Base Map""，《"ordnanceSurvey"":false,""road"":false}"}

WMS,WMS,"{""customData"": "", "layer": "", "layerTitle": "WMS", "layerType": "Overlay", "op":

:LAYERS

:Name,MinZoom,MaxZoom,AssetsQuery,NavZoom,Default

Continent,0,8,,,False

Country,5,16,,,True

State,14,31,,,False

City,29,99,,,False

:LOCATIONS

:Asset,Graphic,GraphicType,Layer,Mode,Latitude,Longitude,NavigationZoom,Position

Building_1,MapPin,Symbol,State,Static,37,-122,,Center

Building_2,MapPin,Symbol,State,Static,34,-118,,Center

Building_3,MapPin,Symbol,State,Static,32,-117,,Center

Storage,MapPin,Symbol,Continent,Static,39,-98,,Center

StorageTank_001,MapPin,Symbol,City,Static,37.1,-122.1,,Center

StorageTank_002,MapPin,Symbol,City,Static,37.11,-122.11,,Center

StorageTank_003,MapPin,Symbol,City,Static,37.12,-122.12,,Center

StorageTank_004,MapPin,Symbol,City,Static,37.13,-122.13,,Center

StorageTank_005,MapPin,Symbol,City,Static,37.14,-122.14,,Center

StorageTank_006,MapPin,Symbol,City,Static,37.15,-122.14,,Center

NavigationView

The NavigationView provides a pair of controls to enable users to set the focus within a ViewApp to show content from different navigation items of a navigation model.

- Configure the Breadcrumb control

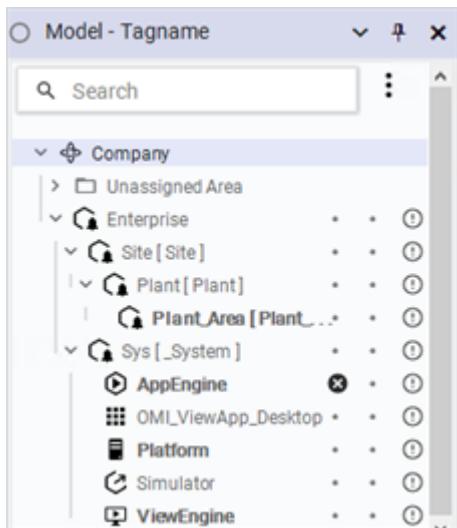
The NavBreadcrumbControl shows a user-selected sequential path through ViewApp navigation model. Each segment shown in a breadcrumb display shows a selected navigation item.

- About ViewApp navigation hierarchical display

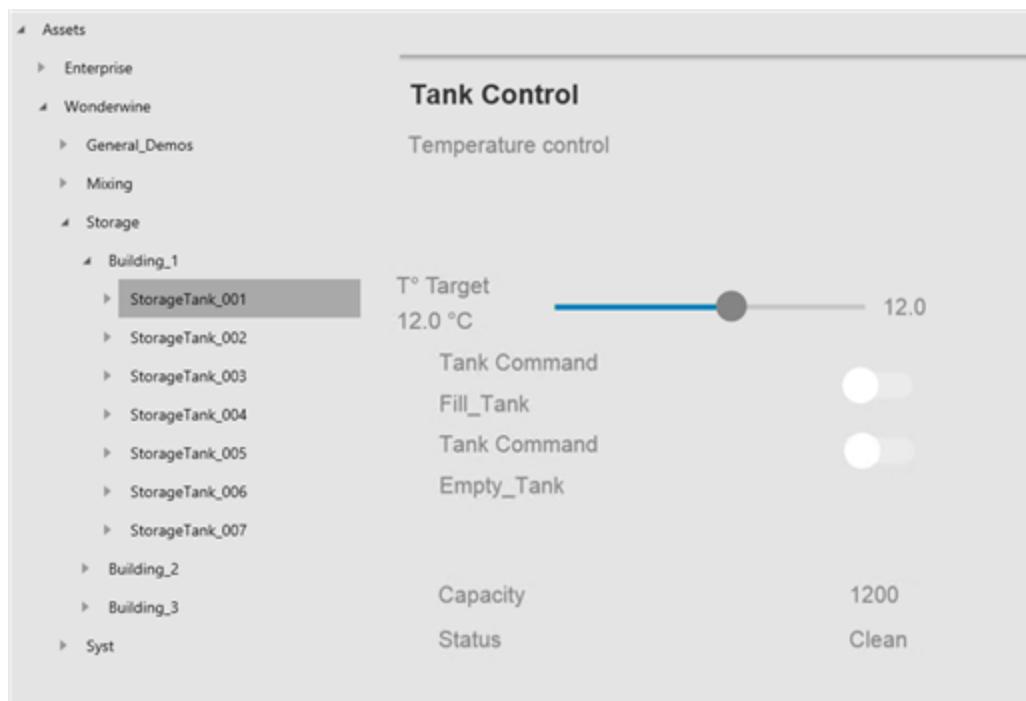
The NavigationTree shows a familiar hierarchical arrangement of parent and child navigation items that together represent a ViewApp navigation model. Users select an item from the tree to show its content.

About ViewApp navigation hierarchical display

AVEVA OMI includes a NavTree app that shows a ViewApp's navigation items arranged in a hierarchical tree. During configuration time in the ViewApp Editor, the NavTree app shows this navigation item hierarchy based on the Model view of object instances. This view represents the navigation model of the ViewApp being edited.

Model View	NavTree App in ViewApp Editor
	<ul style="list-style-type: none">AssetsEnterpriseSitePlantPlant_AreaSysAppEnginePlatformSimulatorViewEngine

During runtime, users can select the chevron icon next to a navigation item to expand or collapse the tree of navigation items. When an item is selected from the tree, the item's navigation commands are executed and associated content appears in layout panes that meet the command requirements.



About display of controls in the ViewApp navigation hierarchy

Aside from the standard **Content** and **Pane** properties of all objects and graphics, the NavTree app includes sets of **Behavior**, **Alarms**, **Appearance**, and **Search** properties:

- Enable the **Show Root** property to show the root navigation item of the navigation model during runtime.
- Enable the **Show Alarms** property to show any active local and aggregated alarms on the individual items within the navigation model.
- Enable the **Show Search Bar** property to show a search field above the runtime Navigation tree to enter a search string. For more information about searching a ViewApp's navigation model during runtime, see *Search View App Content During Runtime* in the AVEVA OMI help.
- The **Appearance** properties determine the color, size, and font family of text shown on the navigation tree. Each property contains a drop-down list of binding options. Some properties only provide a **Reset** option to reset a property value to its default.

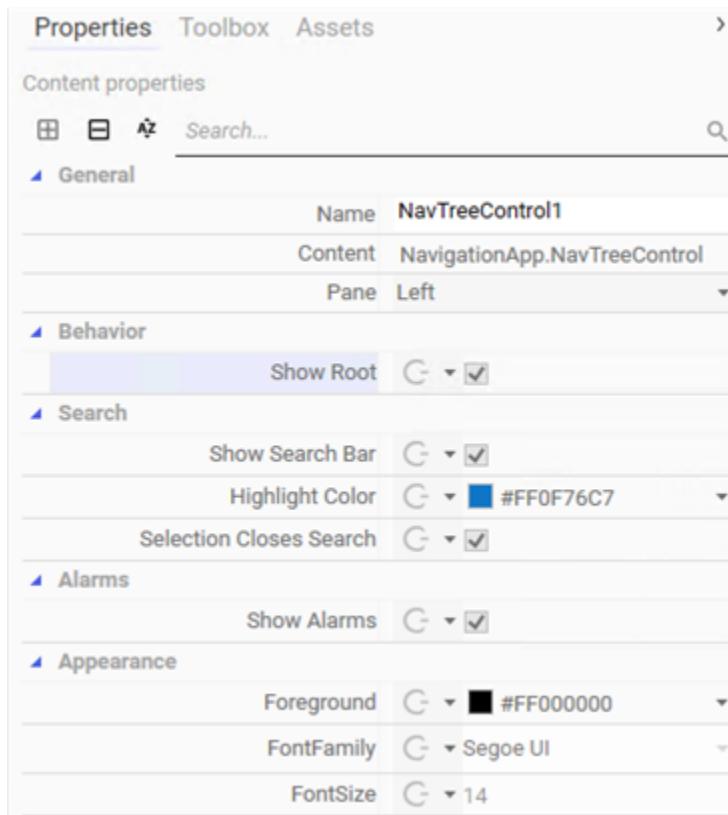
In addition to a static binding, dynamic binding values can also be configured.



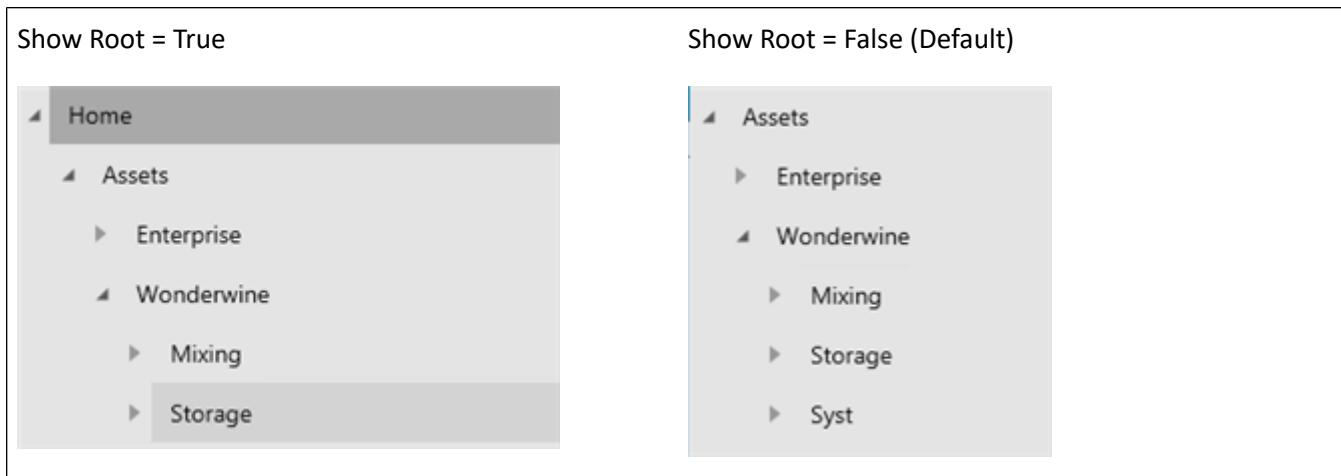
The type of control property /attribute binding can be specified by selecting a value from the drop-down list.

Constant	Static binding to the specified path. Enter a value in
-----------------	--

	the data entry field.
In	Dynamic root path with read only binding by the control to a reference.
Out	Dynamic root path with write only binding by the control to a reference.
InOut	Dynamic root path with read/write binding by the control to a reference.
Reset	Reset the current property to its default value.



When **Show Root** is selected, the root navigation item appears at the top of the NavigationTree's hierarchical list of navigation items. The root navigation item is the default, selected startup item when a ViewApp starts.



The root navigation item is a custom navigation item, which supports the following common behaviors of all navigation items:

- The default name of the root navigation item is Home.
- The root navigation item can be renamed like any navigation item.
- The root navigation item can be a reference to an asset or a graphic item.
- The root navigation item can hold custom actions defining layouts and content in layouts.

The following behaviors are specific to the root navigation item:

- The root navigation item cannot be deleted.
- Sibling items to the root cannot be created.

When **Show Alarms** is selected (it is selected by default), alarm counts from the **AlarmCntsBySeverity** attribute are displayed on any NavigationTree item with an active alarm. If the alarm has been aggregated from a contained object, alarms are shown at 50% opacity (transparency) on the containing object. If alarm is on the object, it is shown as a solid color. The alarm color and graphic reflect the severity of the alarm. Graphics and animations for the alarm counts are taken from the Galaxy style library.

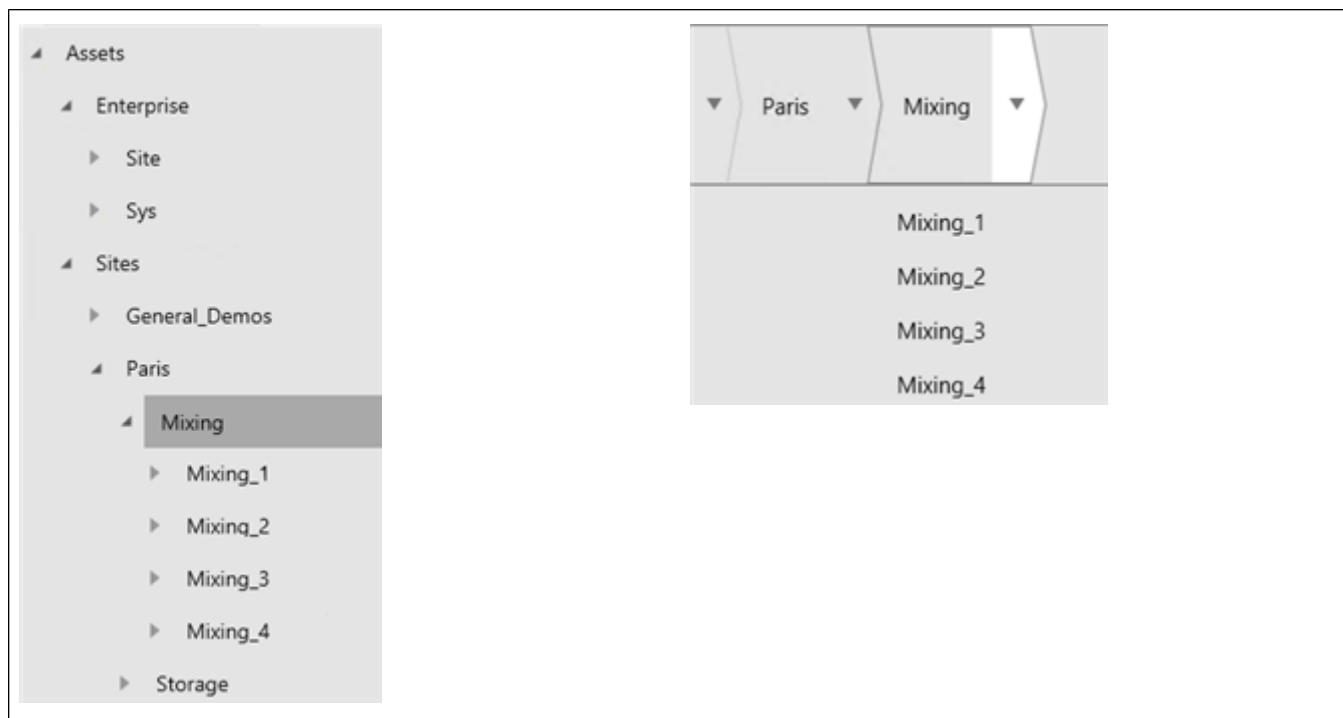
Set values for the color and font properties. Select the color property and then select the small triangle at the right of both property fields to show the color picker.

If you want to create event handler scripts, select the **Available Events** field to show a list of NavTreeControl events.

Configure the Breadcrumb control

The NavBreadcrumbControl can be placed in a layout pane to enable path-style navigation. Instead of a hierarchical tree representation of a ViewApp navigation model, a breadcrumb shows a sequential path. Each segment shown in a breadcrumb represents a user selected navigation item in the path.

The following examples of tree and breadcrumb navigation apps show the different visual representations of the current location within a ViewApp navigation model. Both apps show the focus of the ViewApp is currently at the Mixing navigation item. In the case of the NavBreadcrumbControl, the user selects the triangle area within a breadcrumb segment to show the child navigation items beneath the current item. Then, the user continues the navigation path by selecting a child item beneath the current parent item shown in the breadcrumb control.



The NavBreadcrumbControl includes a property to show an Alarm Adorner. An Alarm Adorner shows aggregate alarm counts within the navigation pathway. The alarm count includes the number of alarms by severity (1-4). Semi transparent alarm counts indicate the source of the alarms is a child navigation item of the current item shown in the NavBreadcrumbControl.. Solid Alarm Adorner colors indicate the navigation item that is the source of the alarms.



How to configure the NavBreadcrumbControl

Aside from the standard **Content** and **Pane** properties of all objects and graphics, the NavBreadcrumbControl includes several sets of properties:

- **Navigation:** This set of properties determine if the navigation root and path are shown and the starting point of the breadcrumb path within the navigation model.
- **Alarms:** The value of the **Show Alarms** property determines if the NavBreadcrumbControl includes an Alarm Adorner or not.
- **Appearance:** This set of properties specify the font size of text and the foreground and background colors shown on the BreadcrumbControl.

Each property contains a drop-down list of binding options. Some properties only provide a **Reset** option to reset a property value to its default. Other properties allow you to configure dynamic binding values in addition to a static binding.



The type of control property /attribute binding can be specified by selecting a value from the drop-down list.

Constant	Static binding to the specified path. Enter a value in the data entry field.
In	Dynamic root path with read only binding by the control to a reference.
Out	Dynamic root path with write only binding by the control to a reference.
InOut	Dynamic root path with read/write binding by the control to a reference.
Reset	Reset the current property to its default value.

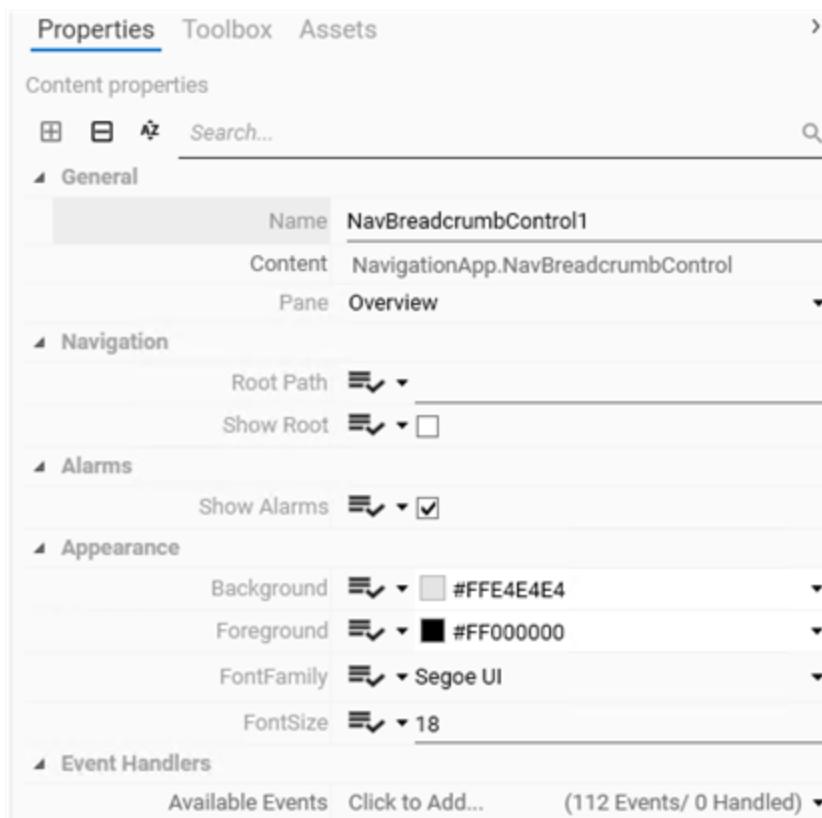
To configure the NavBreadcrumbControl

1. Open the Layout or ViewApp Editor and select the **Toolbox** tab.
2. Open the **AVEVA OMI Apps** folder to show the default apps.
3. Select **NavigationApp** from the **Toolbox** list.

The NavBreadcrumbControl and NavigationTree thumbnails appear beneath the **Toolbox** list.

4. Drag and drop the NavBreadcrumbControl thumbnail on a layout pane.
5. Select the pane containing the NavBreadcrumbControl and select the **Properties** grid.

The **Properties** grid shows a list of NavBreadcrumbControl properties.



6. Select **Root Path** to load the NavBreadcrumbControl to a specified path. Enabling **Root Path** sets the initial context view of the ViewApp to the specified path.
7. The default is static binding that begins the navigation path at the root of the navigation model.
8. Select **Show Root** if you want the root of the navigation model to appear in the NavBreadcrumbControl. By default, the root is not shown.
9. Select **Show Alarms** if you want to show aggregate alarm counts for each navigation node shown by the the NavBreadcrumbControl.
10. Assign values to the **Appearance** properties for color and fonts.
11. If you want to create event handler scripts, select the **Available Events** data entry field to show a list of NavBreadcrumbControl events. For more information about creating event handler scripts, see [Add an Event Handler Script](#).
12. Save your changes.

PDFViewerApp

The PDFViewerApp shows a Portable Document Format (PDF) document within a pane of a ViewApp in preview mode or while a ViewApp is running. The PDFViewerApp initially shows the first page of the document at 100 percent zoom scale.



The PDFViewerApp includes a set of visual controls that appear above the displayed PDF document on the menu bar. Users can select these controls to change the visual appearance of a PDF document while it is being displayed, but cannot make any permanent editing changes.



	Zoom controls to zoom in or out of the current page shown from a PDF document.
	Find control to search the contents of the displayed PDF document.
	Page controls to scroll forward or backward through a PDF document.
	Orientation controls to rotate the PDF document 90 degrees clockwise or counterclockwise.

	Copy control to copy selected content from a displayed PDF document.
	Page view controls to set the number of pages shown by PDFViewerApp and the type of scrolling.

Change the view of a PDF document

The PDFViewerApp provides several methods to set the zoom level of the page shown from a ViewApp. A page can be zoomed out to a minimum of ten percent of its original size or zoomed into a maximum of 500 percent.

- Zoom in and Zoom out controls



The control shown at the right includes a drop-down list to select a specific zoom level. The list also includes other options to set the zoom level to:

- Actual Size
- Zoom to the Page Level
- Fit Width
- Fit Visible
- Keyboard shortcuts

Users can select keyboard shortcuts to change the zoom level of a page.

- Zoom in by pressing Ctrl + key showing the plus sign (+)
- Zoom out by pressing Ctrl + key showing the minus sign (-)
- Zoom to 100 percent by pressing Ctrl + 1
- Zoom to page level by pressing Ctrl + 0
- Zoom to fit width by pressing Ctrl + 2
- Zoom to fit visible by pressing Ctrl + 3

Search in a PDF document

The PDFViewerApp includes the capability to search for a word or phrase within an open PDF document during preview mode or runtime. Users invoke the search utility by selecting the **Find** control above the open document or using the **Ctrl + F** keyboard shortcut.



A search field appears to type the search item after selecting **Find**.



Search: Previous Next

If the PDF document includes the word or search phrase, the view shifts to the first page containing the search

item and highlights the matching word or phrase on the page. Users can select the **Previous** or **Next** search options to continue the search within the document backward or forward from the current displayed page.

Search: vacuum pump **Settings** **Previous** **Next**

R, 25 CU IN VAC PUMP	4434313	I
W/LITE, 1,000 ID	1760006	I
W/LITE, 1,250 ID	1760033	I
ER DRIVE	4226900	2
UPPORT, PRIMER	4036000	I
C PUMP, 12VDC	4227001	I
VACUUM PUMP, 25 CU IN	1021201	I
.M X .75 F	3500543	I
NPTM, TSJICM	3501553	2
NP - 1810/1710/1610	4824200	I
1.00 ID X 1.00 NPT MALE	3502031	I
VAC PUMP, 25 CU IN	2813801	I
.20 X .63, GR 5 SST	5400603	2
.18 X 0.75, SST	5400641	I
.20 X 2.75, GR5	5400118	I
GED, .313-18 X .750	5402833	6
GED, .313-18 X 1.00	5402843	3
GED, .375-16 X .88	5402835	I
GED, .375-16 X 1.25	5402856	I
P.E. .75 X CLOSE, BR	1081048	I
.313-18, NYLOC, SST	5403400	I
.75 x 4.94 x 0.09	3601114	I
S, .5"	4401501	2
R, .125 SST HAIRPIN	3605213	2
IMER, PUSH BUTTON	1978500	I
IMER LUBRICANT TANK	1963400	I
IMER DRIVE	4826200	I
BEARING, PRIMER CLUTCH	2314400	I
M, MECHANICAL, 1.250	1843403	I
.20 x 0.37 SST	5401400	I
.06 X .88 X .38	3309000	I
TCH ANTI-ROTATION	4838110	I
H, PUSH-BTN	2600008	I
ER, 4 QUART, W/CAP	4403703	I
ER CONNECT	4434800.25M2	I
D, 0.19 X 6", NEO	1101937	I
G - STR, .25 X .19	3500021	I
UATED, 12/24VDC, .75NPT FF	5209404	I
R, 25 CU IN, .1905 THICK	4434200	4
.00 x 1.50 x .273 SS	3603342	I
AT, .375 ID	3603820	I
AT, .0.31 ID, SST	3603812	I
CK, 0.250 ID, SST	3603532	2
ESS, NC/WO,15 PSI,I	2600167	I

NOTES:

- 1) PRIMER ASSEMBLY IS EITHER 12 VDC OR 24 VDC AND MUST BE SPECIFIED UPON ORDER.
- 2) BELTS MUST BE INSTALLED IN THE ORIENTATION SHOWN.
- 3) TIGHTEN A NEW BELT UNTIL THERE IS 30 TO 60 LBF STATIC TENSION PER BELT. THIS CORRESPONDS TO A 2.2 TO 4.1 LBF DEFLECTION FORCE RESULTING IN A .16 INCH DEFLECTION AT THE BELT MIDSPAN.
- 4) IF A SONIC TENSION METER IS USED FOR INITIAL BELT TIGHTENING, 68 HZ WOULD BE THE RECOMMENDED MINIMUM FREQUENCY.
- 5) PRIME GREEN IS OUR RECOMMENDED PRIMER LUBRICANT.
- 6) PRIMING MUST BE DONE WITH A DRIVELINE SPEED OF 1000 RPM OR LESS.

REFERENCE NOTES:

- 1) THERE IS A 2.1998:1 SPEED INCREASE BETWEEN THE DRIVELINE AND THE PRIMER.
- 2) A 1550 SERIES JOINT REQUIRES A 4-2-689 FLANGE YOKE.
- 3) A 1610 SERIES JOINT REQUIRES A 5-2-379 FLANGE YOKE.
- 4) A 1710 SERIES JOINT REQUIRES A 4819900 (4-2-739-1) FLANGE YOKE.
- 5) A 1760 SERIES JOINT REQUIRES A 4820000 (4-3-2-19-1) FLANGE YOKE.
- 6) A 1810 SERIES JOINT REQUIRES A 4820100 (4-5-2-329-1) FLANGE YOKE.
- 7) PRIMER EVACUATES A THEORETICAL 51.28 CUBIC INCHES PER REVOLUTION.

**TIGHTEN UNTIL LOCK NUT ENGAGES
DO NOT CINCH DOWN
BOLT ONLY KEEPS CLUTCH FROM ROTATING**

31 LB-FT 18 19 13 43 32 21 42 14 4 100 LB-IN

Navigate in a PDF document

Based on the current page shown by the PDFViewerApp, users can select a visual scroll control to show the next page or backward to the previous page. The PDFViewerApp also supports page scrolling by pressing the Page Up and Page Down keys or the Up and Down arrows keys. Users can also scroll rapidly through the pages of a document by rotating their mouse wheel forward or backward, or by moving the vertical window scrollbar by touch or a mouse.

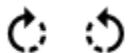


The PDFViewerApp also includes a control to show a specific page of a document by entering the page number and pressing the Enter key.

29 / 119

Change the orientation of a PDF document

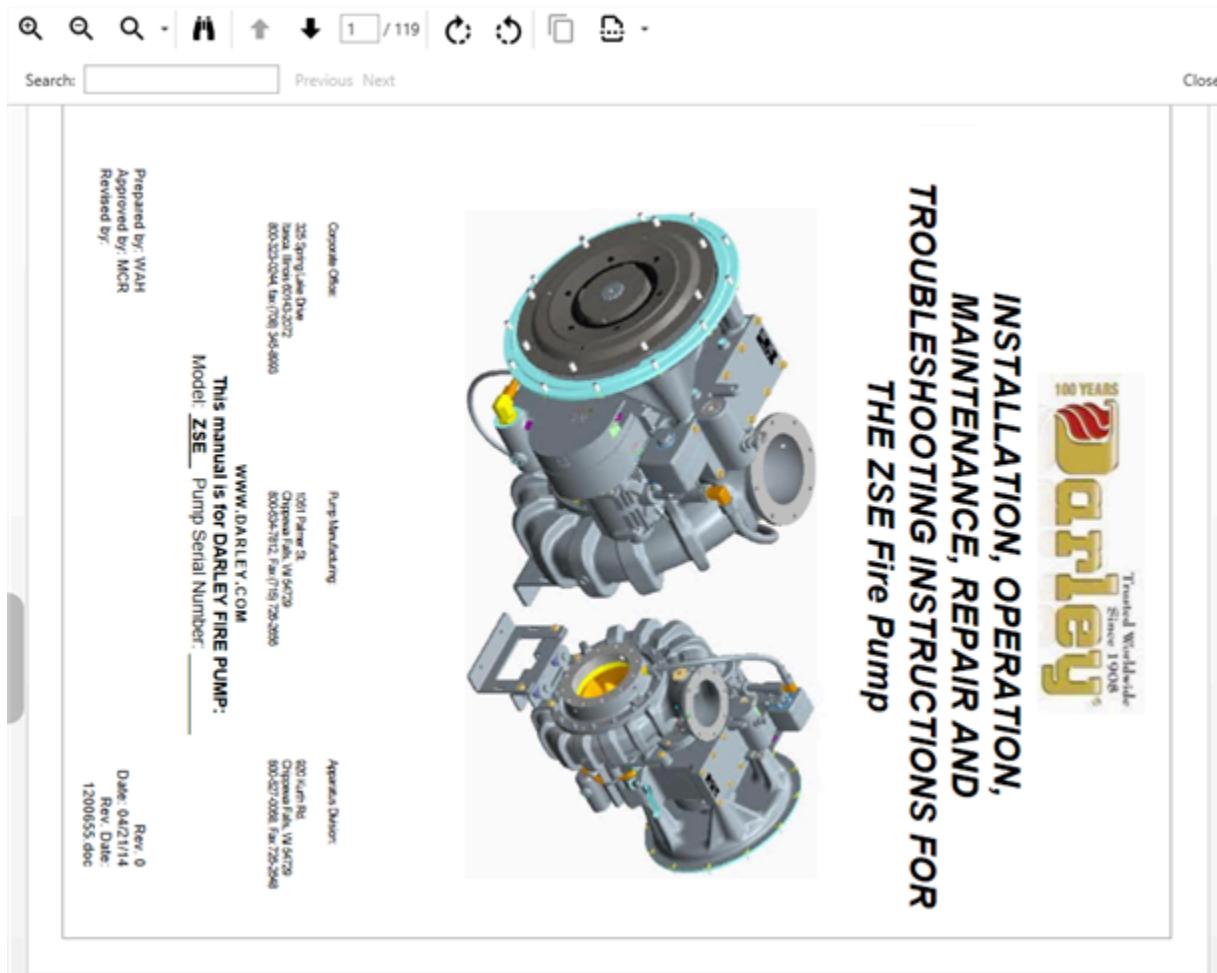
The PDFViewerApp includes a set of controls to change the orientation of a PDF document while it is being displayed in preview mode or while the ViewApp is running.



A document can be rotated in 90 degree clockwise or counterclockwise increments each time an orientation control is selected.

Users can also use keyboard shortcuts to rotate clockwise or counterclockwise

- 90 degree clockwise rotation: Ctrl + Shift + OemPlus
- 90 degree counterclockwise rotation: Ctrl + Shift + OemMinus



Copy and paste content from a PDF document

Users can copy selected text or graphics from an open PDF document and paste it into another open application like WordPad or Word.

Important: Graphics and text from a PDF document must be copied separately.

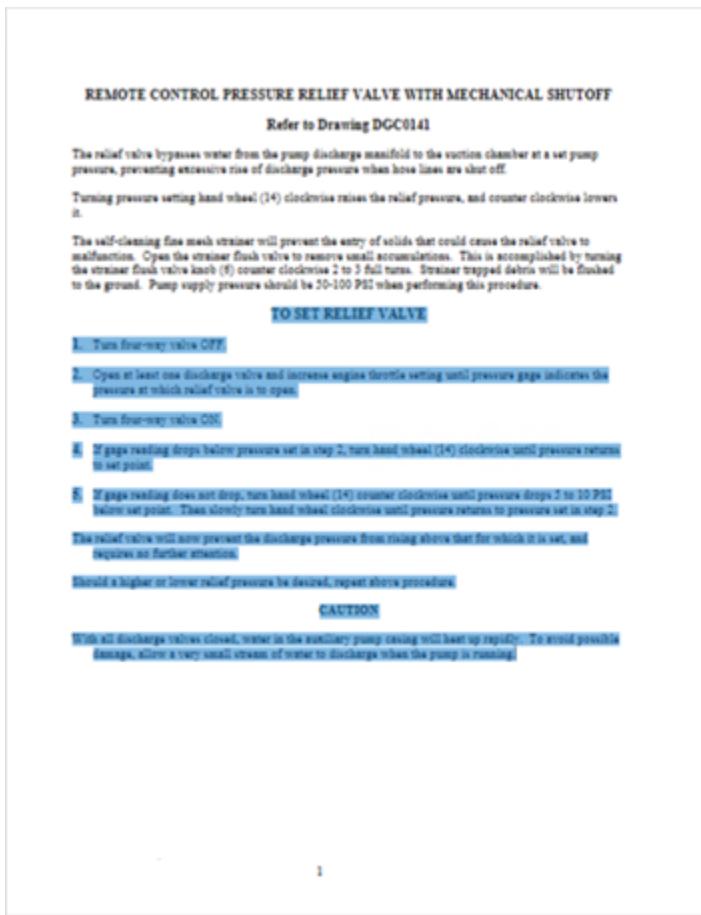
To copy a graphic from a PDF document

1. Open an application like Word or WordPad to paste content from the PDF document open in the PDFViewerApp.
2. Select the graphic that you want to copy from the PDF document.
Blue highlighting appears around the graphic to indicate that it is selected.
3. Select the **Copy** control from the PDFViewerApp.
4. Select the location in the open application where you want to place the graphic.
5. Press **Ctrl + V** to paste the graphic into the open application.

To copy text from a PDF document

1. Select the text to be copy.

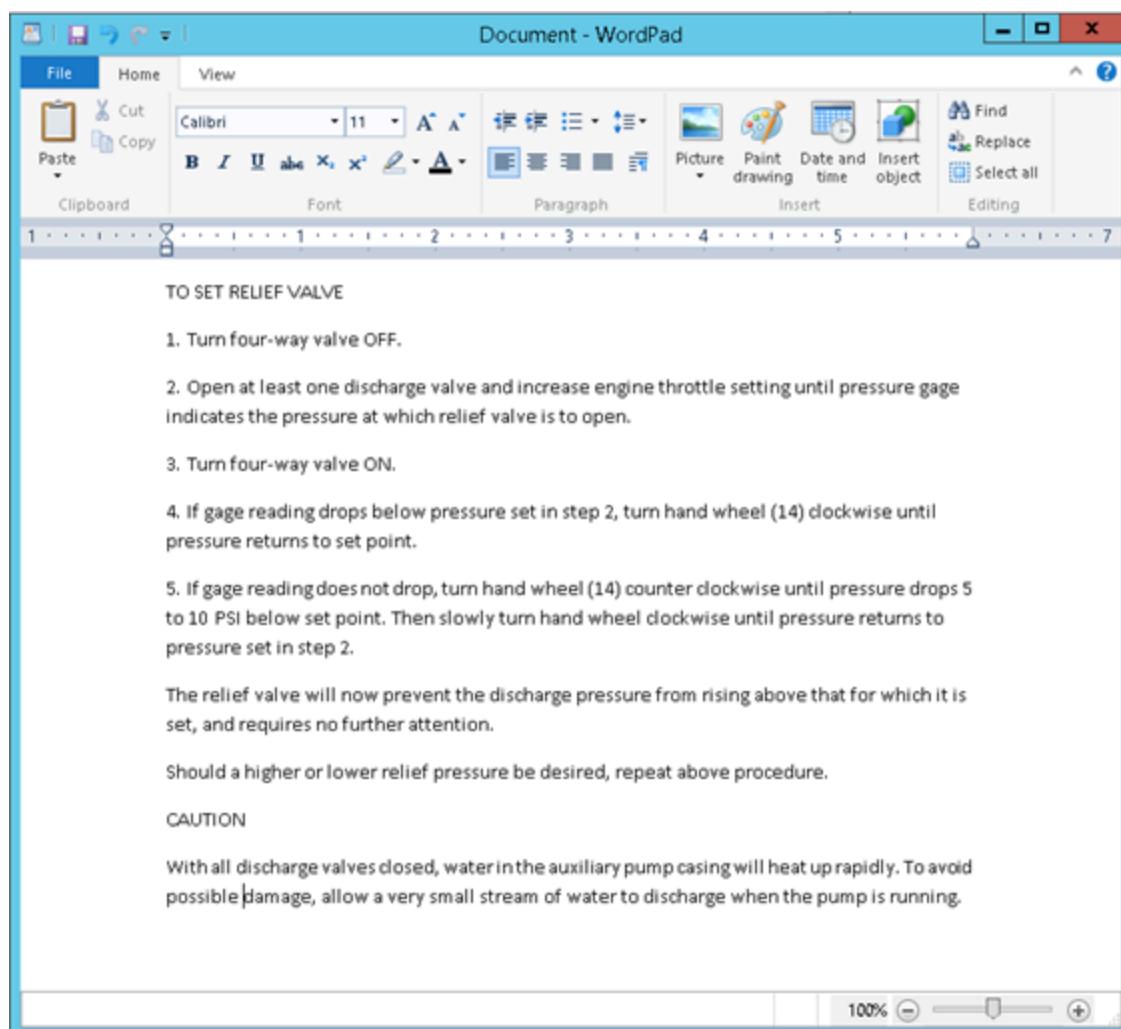
The selected text appears with blue highlighting to indicate that it has been selected.



2. Select the **Copy** icon above the open page or press the **Ctrl + C**.

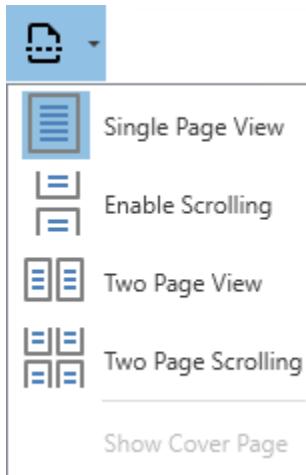


3. In the application to copy the text to, place the insertion point where you want to insert the text. Press **Ctrl + V** to paste the text.



Change the page view of a PDF document

The PDFViewerApp includes a **Page Display** control to set the view of pages within a PDF document and the type of scrolling. The default is a single page view with scrolling.



Other options include a two page view with two page scrolling.

The screenshot shows a two-page scrollable interface. The left page contains a section titled "Section 1" and a "Definition of Symbols and Immediate Safety Information". The right page contains an "IMPORTANT" section with safety symbols and descriptions, followed by sections for "DANGER", "WARNING", "CAUTION", and "WARNING" again, each with a detailed description. At the bottom of the right page, there is a "SAFETY" note and a "WARNING" note with specific instructions. The interface has a standard Windows-style scroll bar on the right side.

Touch support

The PDFViewerApp supports single finger gestures on touch screens or portable devices.

- A PDF document can be swiped or flicked vertically to show the previous or next pages of a document.
- A single finger tap can be used to select the visual icons shown on the menu bar of the PDFViewerApp.

See [Optimize the PDFViewerApp for a touch device](#) for instructions to enlarge the size of the visual icons to make it easier to select them while using touch gestures.

Optimize the PDFViewerApp for a touch device

The visual icons shown within the viewer window of the PDFViewerApp can be enlarged to make them easier to select on touch screens or portable devices.

The following screen captures show the default size of the PDFViewerApp's visual icons and their size after they have been enlarged for touch. AVEVA OMI includes the `OptimizeForTouch` Boolean attribute that belongs to the `MyViewApp.Settings` namespace. The default value of `OptimizeForTouch` is false. When `OptimizeForTouch` is set

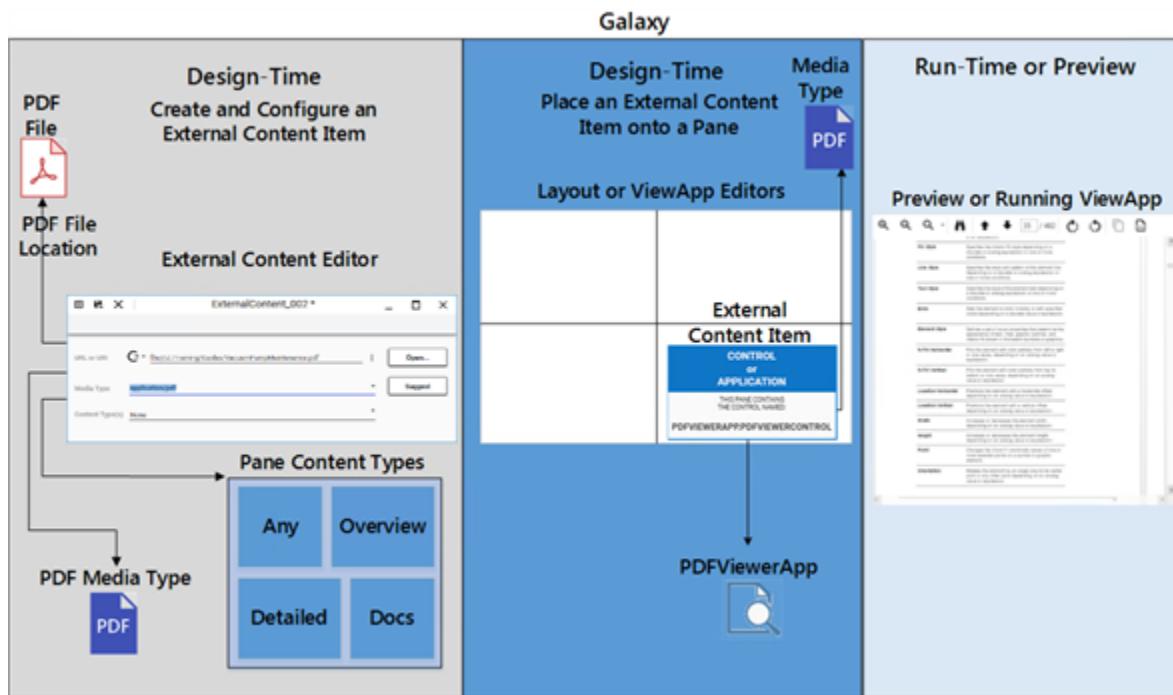
to true, the size of the visual icons is enlarged.

OptimizeForTouch Attribute Value	Example Icons
MyViewApp.Settings.OptimizeForTouch=False	Regular Size (Non-Touch)
MyViewApp.Settings.OptimizeForTouch=True	Optimized for Touch

A common way to dynamically adjust the size of the visual icons in a running ViewApp is to place a graphic element like a button on a pane and then associate user input animation to toggle the state of the OptimizeForTouch attribute.

Configure the PDFViewerApp

The following figure summarizes using an External Content item to point to a folder location containing a PDF document that explains how to perform maintenance on a pump. An External Content item is configured during design time to identify the location of the PDF document and its media type. The External Content item is associated with an asset or a user defined object. The asset or object is selected from the **Toolbox** tab and placed onto a layout pane. During runtime, the user selects an asset associated with a navigation item to view training procedures described in the PDF document.



The major steps to implement the PDFViewerApp to show PDF documents in a ViewApp are summarized in the following list:

1. Place a PDF document on a computer accessible from your Galaxy.
2. Create an External Content item.
3. Configure the External Content item that identifies the location and media type of the PDF document.
4. Associate an External Content item to a graphic or object.
5. Place an asset or graphic on a ViewApp pane associated with the External Content item.

Before you begin

Before starting the steps to configure an External Content item, ensure the following prerequisites have been met:

- If you are placing the PDF document on a remote computer in your network, ensure connectivity between the computer where the ViewApp will run and the computer where the remote PDF document has been saved.

Mapping a network drive is a typical way to connect to a remote computer hosting documents that you want show in a running ViewApp. For this configuration to work successfully, it is important that the computer to which the ViewApp is launched from has the same drive letter mapped to a location having the referenced document.

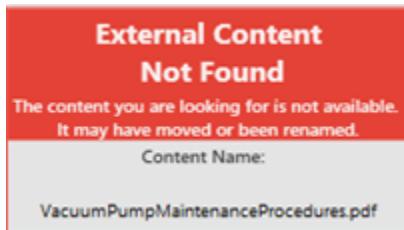
The PDFViewerApp shows an error message to the user if the specified URI connection to the PDF file is bad. For more information about the error message, see [PDFViewerApp error message](#).

- Save the PDF document in the presentation format that you want to display in the DocViewApp.

Typically, you want to format the PDF document in its intended final form without showing any revision markup.

PDFViewerApp error message

An External Content item includes an **URL or URI** option to specify the location of the PDF file to be shown by the PDFViewerApp during runtime or in preview mode. The specified location of the file must be complete and accurate. If the file is moved, deleted, or renamed, the PDFViewerApp shows an error message to the user during runtime, but the ViewApp continues to run.



Create an External Content item

You create an External Content item from the Visualization folder of the System Platform IDE. Initially, the External Content item defaults to a set of values that can be changed using the External Content editor.

To create an External Content item

1. Open the IDE and select the **Visualization** tab.

2. Select a folder within the **Visualization** folder if you want to create a new External Content item at a specific location.
3. On the **Home** ribbon, in the **Create** area, select **External content**.

You can also create an External Content item by other methods:

- **Keyboard Shortcut**

Press Ctrl + Shift + C

- **Shortcut Menu**

Right-click a folder of the **Visualization** folder to show a shortcut menu. Select the **New** option, and then select **External Content**.

A new **External Content** item is created in the **Visualization** folder.



The name of the new item follows a default naming convention of appending a three-digit number to the word ExternalContent.

4. Rename the External Content item.

After you create an External Content item, it must be configured to specify the remote location of the content and its media type.

Configure the External Content item for a PDF document

Each External Content item has three properties that must be assigned values:

- **URL or URI**
- **Media Type**
- **Content Type(s)**

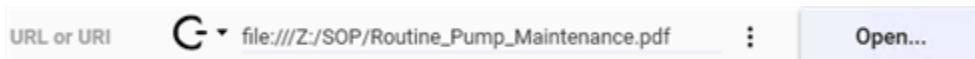
An External Content item is modified from the External Content editor. The editor can be opened by double-clicking an item from the Visualization folder. Also, an External Content item that has been placed onto a layout pane can be edited from the Layout and ViewApp editors by selecting the item from the Actions list.

To configure an External Content Item

1. Open the IDE and select the **Visualization** tab to show the External Content items available in the Galaxy.
2. Double click an External Content item to open within the External Content editor.
For more information about each field in an External Content item, see [Details of the External Content editor](#).
3. Enter a URI string or web site URL in the **URL or URI** field where the document is saved.

The value you enter is the location of the external content specified by a URI-formatted string. A media location must be specified.

You can browse for the document file by selecting the vertical dots icon to the right of the data entry field. A drop-down list includes a **Browse for file** option that enables you to browse your network and select a media file. The **URL or URI** field updates and shows a formatted URI path to the file.



You can also enter a URL to a web site by selecting the **http://** or **https://** options from the drop-down list. The **http://** and **https://** options assist the user when entering URLs by prefixing the URLs with the selection. See [More information about specifying a URI](#) for information about the format of a URI string.

4. Select the **Open** button to validate the URI or URL you entered.

The **Open** button launches the appropriate application associated with a pdf file type, which is usually Adobe Reader to display the document specified in the **URL or URI** field.

5. Select the type of media associated with documents from the drop-down list of the **Media Type** field.

The value you enter is the MIME type of a PDF document. The PDF MIME type provided by the PDFViewerApp is:

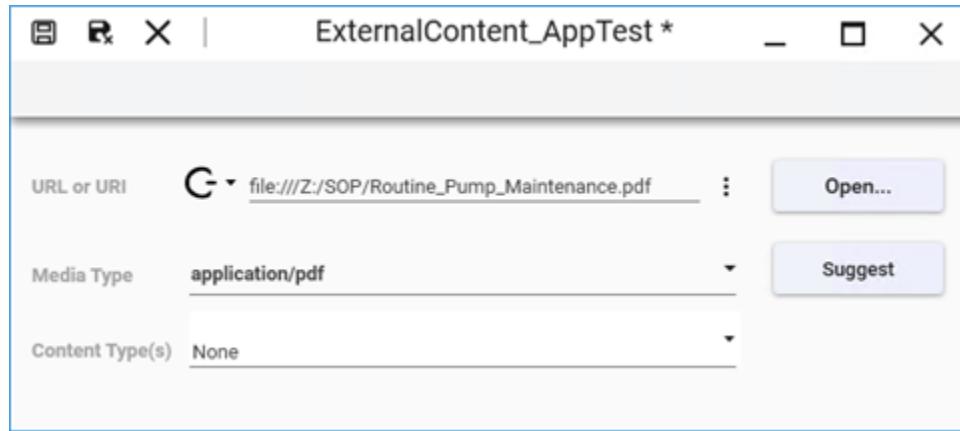
application/pdf

See [More information about media types](#) for more information about supported media types.

6. Select the type of content from the drop-down list of the **Content Type(s)** field.

The value you enter is used by content-placement algorithms within AVEVA OMI during run time to determine the placement of content when AutoFill or ShowContent calls are invoked.

7. The configured External Content item for a PDF document should look like the following screen capture.



8. Save your changes and exit from the External Content item.

Link an External Content item to an object or template

There are three ways to integrate an External Content in a ViewApp.

- Associate an External Content item to an object template, which is described in the following procedure
- Associate an External Content item to a custom navigation item that appears in the navigation model of the ViewApp.
- Place the External Content item on a layout pane, which is the simplest way and is described in [Adding an External Content item to a ViewApp](#).

To link an External Content item to an object template

After you have created and configured an External Content item, you can link it to an object template. External Content items can be linked only to objects as opposed to graphics, which can be linked to or owned by an object. All External Content items reside in the Visualization folder. A single External Content item can be linked to multiple objects.

1. Open an object template in the **Object Editor**.
2. Select the **Attributes** tab.
3. In the **Content** pane, select the **Link Content**  button.
The Galaxy Browser opens.
4. Navigate to the folder that contains the **External Content** item to be linked.
5. Select the item, then click **OK**.

The item is added to the object and appears in the **Content** tab.

Note: After linking an External Content item, you can modify it by selecting it and then pressing the **Edit** button. However, any changes you make will apply to all objects that link to the item.

6. Save your editing changes to the object template and exit from the Object Editor.
7. Create an instance of the linked object template.
8. Create a layout which contains pane with a Content Type that matches the content type of the External Content item.
9. Add a navigation tree to the layout
10. Create a ViewApp that incorporates the layout.
11. Deploy the ViewApp.

In runtime, navigate to the object or any instance created from the template. External Content item will auto-fill the matching pane to show the external content.

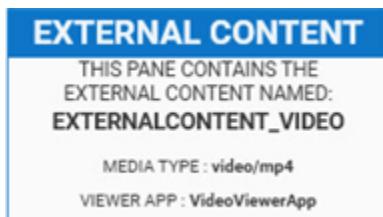
Add an External Content item to a ViewApp

You can place an External Content item directly onto a pane from either the Layout or ViewApp editors . External Content items are listed in the **Toolbox** or **Assets** tabs of either editor.

To associate external content to a ViewApp

1. Open the System Platform IDE and select the **Visualization** tab to see the list of layouts.
2. Select a layout that you want to insert external content and open it in the Layout editor.
3. Select the **Toolbox** tab of the Layout editor to show the list of content accessible from the Galaxy.
4. Locate the External Content item you want to place in the layout you selected.
External Content thumbnails appear beneath of the list of content shown in the **Toolbox** list.
5. Select an External Content thumbnail from the **Toolbox** area, and then drag and drop it onto a pane of a layout incorporated into a ViewApp.

The External Content thumbnail appears in the pane containing the name of the External Content item, the media type of the external content, and the viewer app that will display the media during runtime. Also, the layout **Actions** area shows the name of the External Content item and the pane that it was placed in.



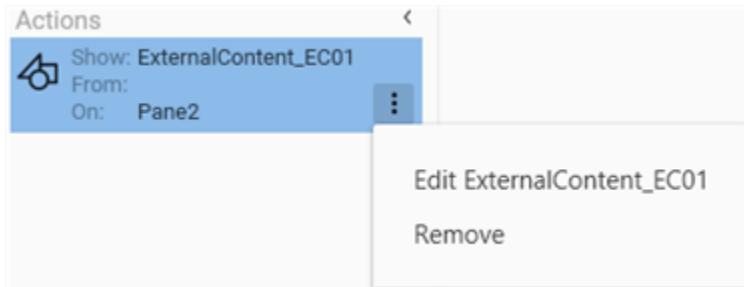
Editing an External Content item after it is placed onto a pane

An External Content item can be modified after being placed in a layout or a ViewApp, which automatically propagates the changes

To edit an External Content item in a layout

1. If necessary, open the layout containing an External Content item.
2. Locate the item in the layout **Actions** list.
3. Select the **Options** icon shown at the right of the item in the **Actions** list.

A drop-down list shows an Edit External Content command.



4. Select the **Edit** command.

The External Content editor opens the item for editing.

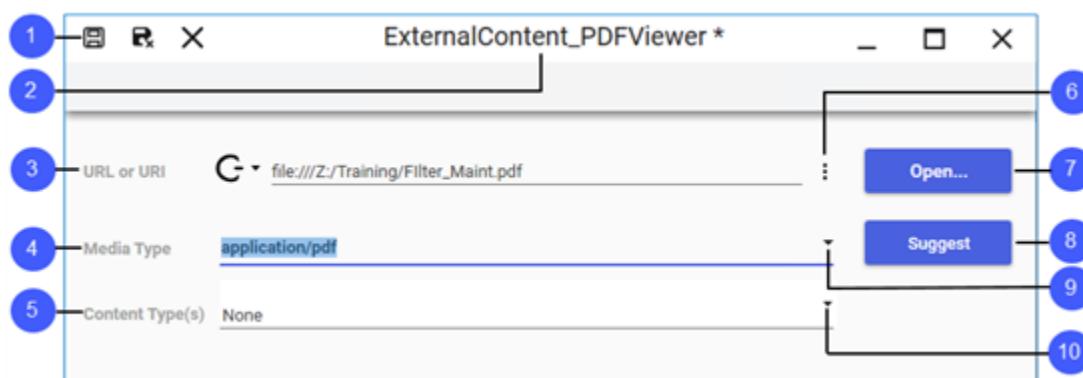
5. Update the values assigned to the item and save your changes.

The External Content thumbnail placed in a pane updates to reflect the changes made to the item.

Details of the External Content editor

You use the External Content editor to configure an External Content item for a document. Before you configure an External Content item, you must know:

- Network location of a document
- Media type of the document
- Content type(s) of the layout pane that will host the External Content item



1	Commands to save, save and close, and close the
---	---

	External Content editor.
2	Name of the External Content item.
3	URL or URI that specifies the location of external media using a standard format. See Specify the URL of External Content for the format of a URI or URL. A warning message appears if the document cannot be located during runtime.
4	A media type is a two-part identifier that specifies the type of application required to process or view remote content. A media type can be entered in the field or selected from a drop-down list. See Specify media types for External Content for the format of a Media type value. A warning message appears if a media type is specified that does not have an associated viewer application.
5	Content type assigned to the external media that enables ViewApp algorithms to place content in specific panes during runtime.
6	Drop-down list with options to browse for an external content file or specify a URL using HTTP or HTTPS.
7	Validates the location of external media specified in the URL or URI field. An attempt is made to display the external media in an application assigned as the default by the operating system, not the app specified for the external content media type. A warning message appears if the content cannot be found at the location specified in the URL or URI field.
8	Suggest is optional. When selected, it automatically selects a media type based on the entry in the URL or URI field. The entered value can be changed if the suggestion does not match the expected media type. Note: Ensure the suggested media type is correct. The external media does not appear if the media type is incorrect.
9	Drop-down list of commonly used media types. Media types that have apps in the Galaxy capable of

	<p>servicing them are shown in bold text.</p>
10	<p>Drop-down list of content types that can be used at preview\runtime to assign the external content to panes of a matching content type.</p> <p>Note: None is not a content type. Instead, it simply means a content type has not been assigned to the pane.</p>

More information about media types

A media type is a standard two-part string analogous to a MIME type that identifies external content file types and their format. System Platform uses a media type to identify external content and the type of app required to display media during runtime.

A media type consists of a type and a subtype, which is further structured into a tree.

```
type "/" [tree "."] subtype ["+" suffix] *[ ";" parameter]
```

For more information about the format of media types, see <https://www.iana.org/assignments/media-types/media-types.xhtml>

System Platform provides several default media types that include associated apps to display media. The drop-down list of the **Media Type** field shows the default media types in bold text to indicate a viewer app is available in the Galaxy that can service external content with these media types. Other listed media types that appear in plain text require an app to be created and imported into a Galaxy to play the specified media.

More information about specifying a URI

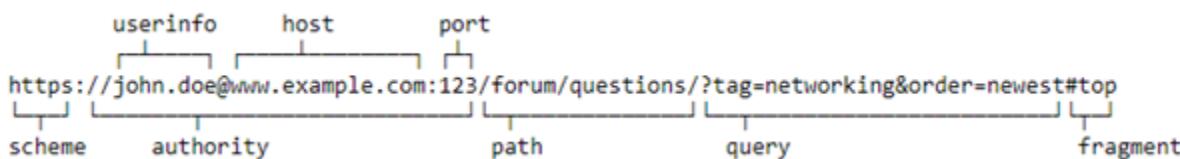
In the context of External Content, a Uniform Resource Identifier (URI) is a string that identifies the pathway to specific content that can appear in a running ViewApp.

All URIs adhere to predefined syntax rules.

- Generic URI syntax consists of a hierarchical sequence of components

```
URI = scheme:[//authority]path[?query][#fragment]
```

Example



- Each URI begins with a scheme name followed by a colon
- Examples of common scheme names include http:, https:, and file:
- An optional authority component preceded by two slashes (//)
 - Use information component consisting of a user name and optional password preceded by a colon followed by an at symbol (@)
 - `//username:password@`

- Host subcomponent consisting of a registered hostname or IP address
- A path component consisting of a sequence of path segments separated by a slash (/). A path is always defined for a URI, though the defined path may be empty
- An optional query component preceded by a question mark (?), containing a query string of non-hierarchical data.

For more information about the format of a URI, see <https://tools.ietf.org/html/rfc3986>

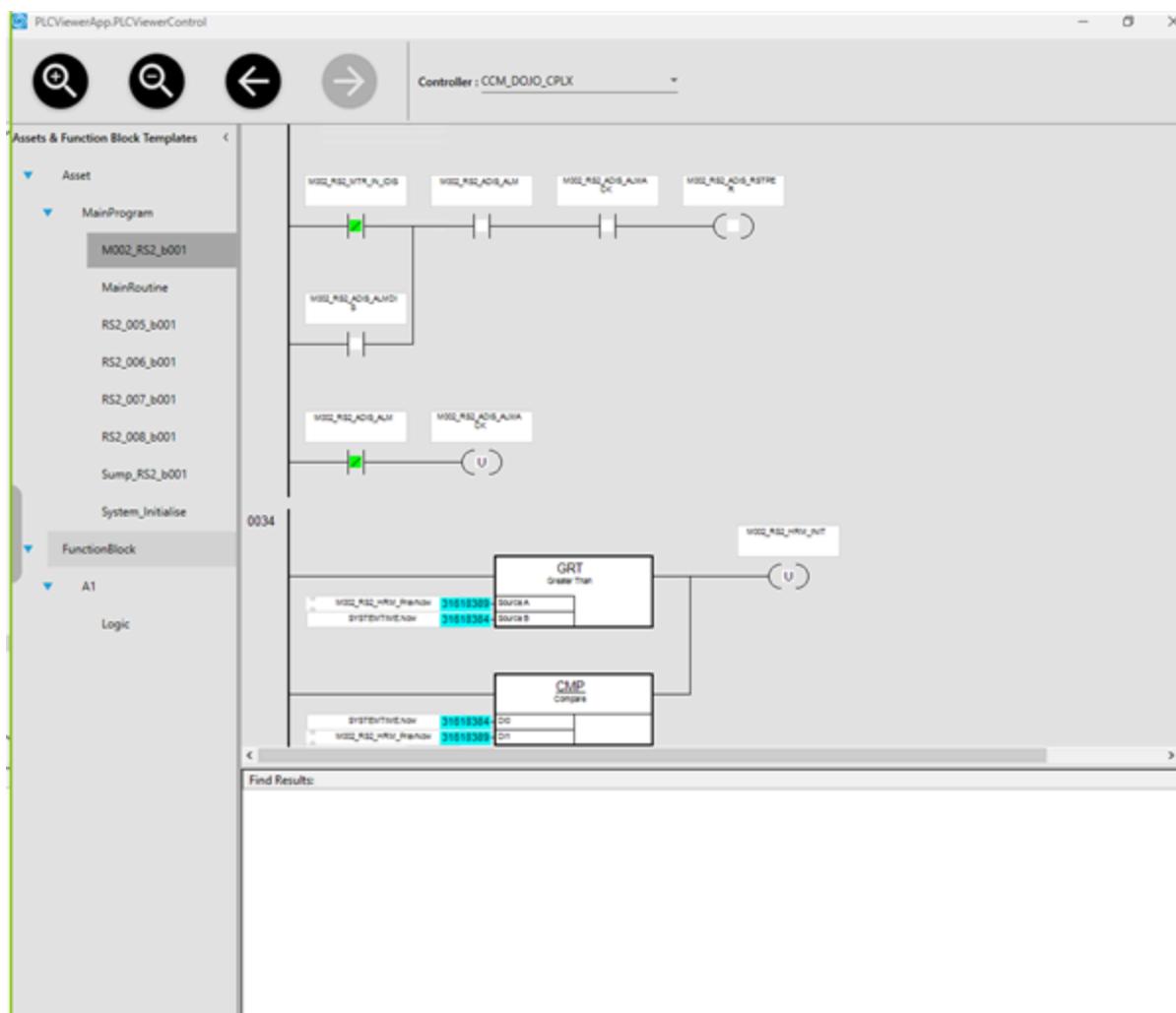
The following examples show common URIs that identify External Content within a Galaxy

- `ftp://ftp.is.co.za/rfc/rfc1808.txt` -- ftp scheme for File Transfer Protocol services
- `file:///Z:/Media/Runtime%20Language%20Switching.mp4`
- `http://www.exampleserver.com/documents/index.html`

PLCViewerApp introduction

While an InTouch OMI ViewApp is running, the PLCViewerApp shows a current asset's PLC control logic with real-time logic animation.

The PLCViewerApp provides a view within a layout pane to easily identify the current control states of a PLC. In the following example, current values associated with a function block have a blue background. Function blocks are annotated with descriptive comments that appear when hovering the mouse cursor directly over a function block shown in the app view.



The PLCViewerApp shows a three-pane window. The left navigation pane shows a vertical list of controller components by assets and function blocks. Users select an item from the component list to show its details in the right pane. The detail pane includes scroll bars to move the view horizontally and vertically. Controls are included to zoom the view in or out. The **Find Results** pane shows the results of a search.

Users can switch the view by selecting another controller from the **Controller** field shown above the two PLCViewerApp panes. The **Scope** field appears when selecting add-on instructions to enable selecting a specific instance to animate.

PLCViewerApp requirements

You should have already imported the PLCViewerApp to your computer hosting the Galaxy Repository. Refer to the *AVEVA PLC Viewer OMI App Readme* for instructions to import the app.

This version of the PLCViewerApp must adhere to the following requirements:

- The PLCViewerApp is targeted for System Platform 2017 Update 3 Service Pack 1 and later.
This version of the PLCViewerApp can be upgraded to subsequent versions of System Platform when they become available.
- This version of the PLCViewerApp supports Rockwell, Schneider Electric, and Siemens Step7 PLCs.

- The PLCViewerApp requires controller project data from a source logic file, which can be created by each PLC vendor's software utilities. A logic file contains the following data referenced by the PLCViewerApp:
 - Ladder programs
 - Controller and program tags
 - Add-on instructions
 - User function blocks
 - I/O tags
 - User defined data types

For Siemens Step7 PLCs, an export utility application has been provided to export the project configuration into a portable file for each CPU using the Siemens API.

Rockwell RSLogix 5000-compatible products

PLCs compatible with the Rockwell RSLogix 5000 family of products require a logic file that must be exported as a .L5X file.

A RSLogix source file is an ASCII file whose data is formatted using Extensible Markup Language (XML). A logic file contains exported controller configuration information and ladder logic of a controller project. You must use your RSLogix-compatible software to export a L5X source logic file.

Schneider Electric Unity Pro-compatible products

Controllers compatible with the Schneider-Electric Unity Pro family of products require a logic file that must be exported as a .zef file. This version of the PLCViewerApp supports Unity Pro 11 to the latest version known as EcoStruxure™ Control Expert.

*.ZEF files are used for exporting projects with global DTMs configuration and are compatible with all versions of Unity Pro.

You must place the source logic file in a folder accessible to the computer hosting the System Platform IDE. Unity Pro needs to be installed on the PC where the PLCViewerApp is configured to access Unity Pro library files. The PLCViewerApp only accesses the XML files included by the installer and does not require the application to be licensed.

Note: If Unity Pro is not installed, a browser dialog appears to enable sourcing the Unity FBLIBSET folder from a user specified location.

- When using context following with the PLCViewerApp, you can optionally create an asset mapping file to alias the context string to a PLC tag and set the context to a specific controller in a comma separated value (CSV) file.

You must place the asset mapping file in a folder accessible to the computer hosting the System Platform IDE.

For more information about an asset mapping file, see [Create an asset mapping file](#).

Siemens Step7-compatible products

Projects configured using Siemens Step7 must be exported using the PLCViewerAppExporter provided with the package. The PLCViewerAppExporter uses the Siemens Step7 API to export each Stations to an encrypted file.

You must place this file in a folder accessible to the computer hosting the System Platform IDE

To export a Siemens project:

1. Copy the PLCViewerAppExporter folder containing the PLCViewerAppExporter.exe and ArchestrA.Prometheus2.Shared.dll to the computer where Siemens SIMATIC Manager is installed and licensed.
2. Confirm the project is accessible in SIMATIC Manager
3. Export SFC source if the source is not current
4. Close SIMATIC Manager
5. Open PLCViewerAppExporter.exe
6. Select the Controllers to export
7. Override the Block and Symbolic name by entering a Display Name (optional)
8. Select File – Import – Display names (*.csv) to load previously saved Display Names (optional)
9. Select File – Export – Display names (*.csv) to save overridden Display Names (optional)
10. Select File – Export – Project (*.S7X) to export controllers to disk
11. Check Exclude unreferenced programs to include programs not being executed (optional)
12. Select OK
13. Select the folder location to save the controller export files

NOTE: Siemens GRAPH (SFC) source files may require updating manually using SIMATIC Manager to ensure the logic presented in the PLCViewerApp is up to date. A message will appear in the output window for each GRAPH file that requires manual export.

When using context following with the PLCViewerApp, you can optionally create an asset mapping file to alias the context string to a PLC tag and set the context to a specific controller in a comma separated value (CSV) file.

You must place the asset mapping file in a folder accessible to the computer hosting the System Platform IDE.

For more information about an asset mapping file, see [Create an asset mapping file](#).

PLCViewerApp configuration

This chapter describes how to configure the PLCViewerApp after it has been imported to a computer hosting a Galaxy Repository on a supported version of System Platform. Refer to the *AVEVA PLC Viewer OMI App Readme* for instructions to import the PLCViewerApp.

Configuring the PLCViewerApp consists of two separate workflows.

- [Configure PLC data sources](#)

Configuring PLC data sources is required and must be completed before configuring PLCViewerApp properties. You must have a source logic file and an asset mapping file ready.

- [Configure PLCViewerApp properties](#)

PLCViewerApp properties specify the default runtime behavior and visual color scheme of the app. These properties have been assigned default values that can be accepted without making any changes.

PLCViewerApp required and optional information

Before you configure the PLCViewerApp, you must gather information about the PLC environment whose code execution you want to monitor while running a ViewApp.

- You are expected to be knowledgeable about your production environment. You must specify the names of the following components to configure the PLCViewerApp:
 - Communication Driver Object, also known as a DI Object (OPCClient or DDESuiteLink Client)
 - OI Server path
 - Scan group
- Source logic file (Required)

You must export a Rockwell, Schneider Electric, or Siemens Step7 source logic file (L5X, ZEF, or S7X) that contains data from your PLC environment and place it in a folder location accessible to the computer hosting the System Platform IDE.

- Asset mapping file (Optional)

The asset mapping file provides alias information for mapping context to a tag name in a specific controller. You can create an optional asset mapping file and place it in a folder location accessible to the computer hosting the System Platform IDE. For instructions to create an asset mapping file, see [Create an asset mapping file](#).

- Translation file (Optional)

The PLCViewerApp includes a default translation file that defines function block layouts for Rockwell 30 ladder functions. If you want to show additional data associated with data structures or add layouts for functions that are not included in the default translation file, you can create an optional translation file and add it to the PLCViewerApp. For instructions to create a custom translation file, see [Create a translation file](#). For instructions to add a custom translation file to the PLCViewerApp, see [Integrate information from a custom translation file](#).

Configure PLC data sources

You can specify three different files to configure the data sources for the PLC Viewer App:

- An RSLogix 5000, Schneider Electric Unity Pro, or Siemens source logic file (Required)
- Translation file (Optional)
- Asset mapping file (Optional)

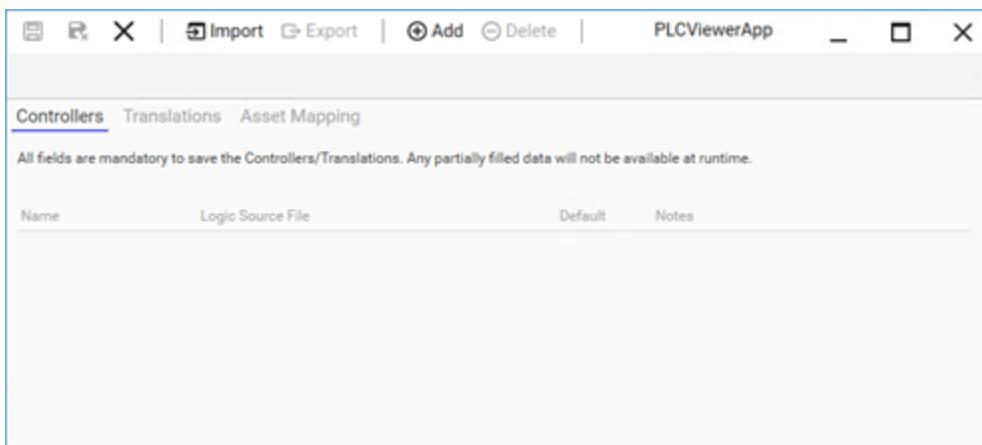
You must place these files at a known folder location that is accessible to the computer hosting the System Platform IDE.

To incorporate source data information into the PLCViewerApp

1. Open the System Platform IDE and select the **Graphic Toolbox** tab to show a list of folders containing symbols and apps.
2. Locate the PLCViewerApp control within the **Toolbox**.

By default, the PLCViewerApp control is placed at the root folder of the Galaxy when it is imported into the System Platform IDE.

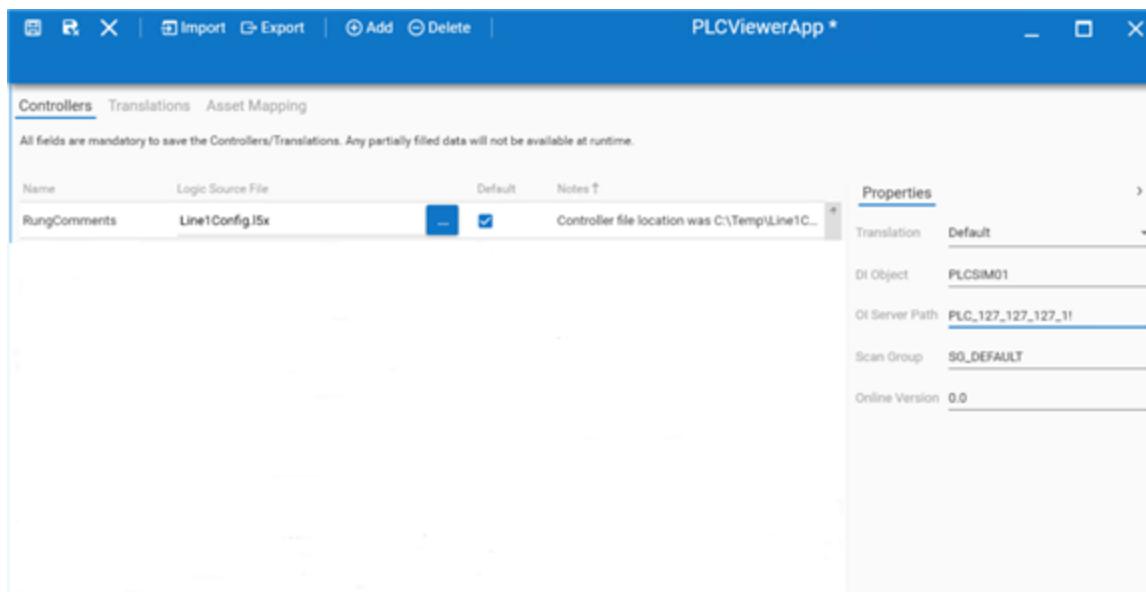
3. Double-click the PLCViewerApp control listed in the **Toolbox** view to show the **PLCViewerApp** dialog box.



4. Select the **Controllers** tab.
5. Click **Add** to add a row to the page.
6. To change the controller name, double-click the assigned name, type the new name, and press **Enter**.
7. Select the browse button at the right of the **Logic Source File** field and locate the folder where the source logic file has been saved.
8. Select the source logic file from the folder and click **Open**.
The **Properties** area of the dialog shows a list of properties.
9. Enter values for the **DI Object**, **OI Server Path**, and **Scan Group** properties.

Important: All properties must be assigned values. No property can be left blank.

- **DI Object:** Name of the DIOBJECT (Communication Driver Object) in the Galaxy configured to communicate to the controller. The DIOBJECT must have a configured Scan Group.
- **OI Server:** Format specific:
 - DDESuiteLinkClient (Rockwell) – *attribute()*
 - OPCClient OI.ABCIP (Rockwell) – <Port>.<Module>.<Backplane>.<Processor>
 - OPCClient Schneider-Aut.OFS (SE) - <OFS Device Alias>!
 - OPC Client (Siemens Step7) - <path_to_server> (For example, POrt.legacy)
- **Scan Group**
 - DDESuiteLinkClient (Rockwell) – <Device Groups Name> configured in the OIServer Processor
 - OPCClient Scan Group (Schneider or Siemens Step7)



- Enter the **Online Version** of your RSLogix or Schneider Electric controller.

An **Online Version** is entered as a single value as:

RSLogix: x.y (major.minor) For example: 1.01

Unity Pro: word1.word2 For example: 24754.6346 or %SW94.%SW95

For more information about checking the online controller version, see [PLCViewerApp runtime version check](#).

Note: The online version is not supported for Siemens controllers. You do not need to enter anything in this field.

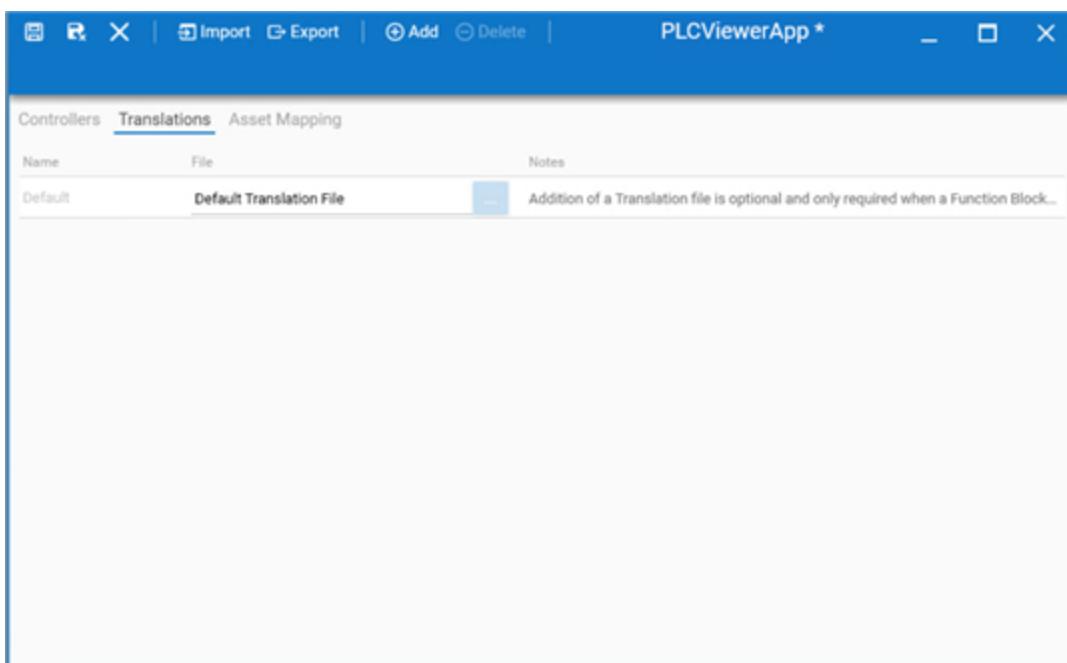
- Repeat steps 2-10 if you want to add information from another logic source file.
- Select the checkbox beneath the **Default** field for the controller that you want to assign as the default.

Controllers Translations Asset Mapping				
All fields are mandatory to save the Controllers/Translations. Any partially filled data will not be available at runtime.				
Name	Logic Source File	Default	Notes	
Line4	Line4_FinalWrap.lsx	<input checked="" type="checkbox"/>	Controller file location was C:\Temp\PL...	
Line2	Line2_PreWrap.lsx	<input type="checkbox"/>	Controller file location was C:\Temp\PL...	

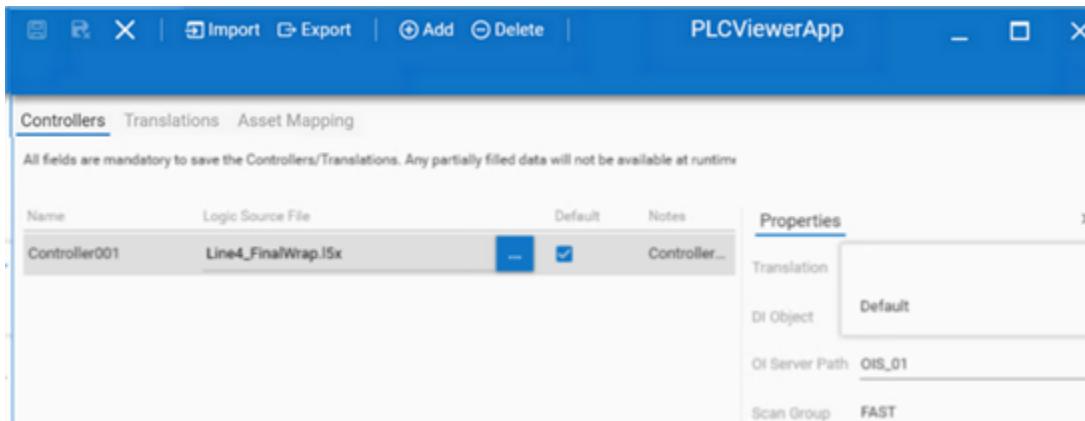
The PLCViewerApp will initially open in runtime showing data from the default controller.

- Select the **Translations** tab.

A translation file is assigned by default. You can accept the default translation file or add a translation file that has been imported. The rest of this procedure explains how to configure the PLCViewerApp using the default translation file. For the steps to add a translation file, see [Create a translation file](#).



14. Select the **Controllers** tab and select a controller from the list.
15. Select **Default** from the drop-down list of the **Translation** field.

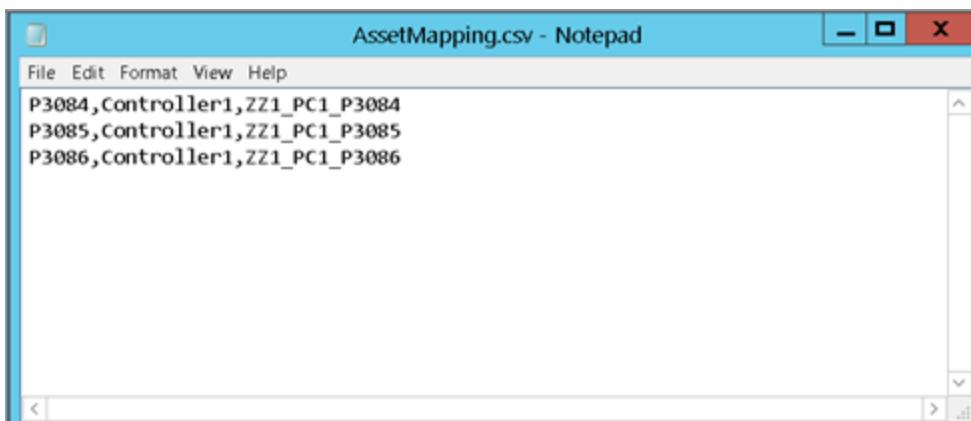


16. Select the **Asset Mapping** tab
17. Select **Add** to add a row to the page
18. Select the browse button at the right of the **File** field and locate the folder containing the asset mapping file.
19. Select the asset mapping file and click **Open**.
20. Save your changes and close the dialog box.

Proceed to [Configure PLCViewerApp properties](#) to complete the remaining configuration tasks.

Create an asset mapping file

An asset mapping file is a comma separated value file with a .CSV file extension that can be created with Excel, Notepad, or other text editors.



An asset mapping file contains a list of AVEVA assets that represent physical objects like a pump or a valve in your production environment that are monitored and managed by a running ViewApp. Each listed asset includes the controller that manages data from the asset and an asset identifier of data collected from the asset.

An asset mapping file enables you to map an AVEVA asset name to a PLC tag name in a specific controller. For example, you have an asset Pump001 and your PLC environment contains multiple controllers with the tag PMP001. You can add a map for Pump001 that searches controller PLC01 for the tag PMP001 using the following line in the mapping csv:

Pump001,PLC01,PMP001

After you create an asset mapping file, you need to place the file in a folder location that is accessible to the computer hosting the System Platform IDE. You specify the asset mapping file while configuring the data sources for the PLCViewerApp described in [Configure PLC data sources](#).

Create a translation file

A translation file is an XML formatted file associated with the Rockwell Automation RSLogix 5000 ladder logic programming package. The PLCViewerApp includes a default XML translation file that specifies the parameters that appear on a function block for RSLogix v30. If you want to show additional parameters on a function block, support additional versions of RSLogix, or support blocks not included in the default library, you can create a custom translation file.

A quick and simple way to create a custom translation file is to copy the default translation file and modify it to specify additional function block parameters.

To create a copy of a translation file

1. After importing the PLCViewerApp, locate the default translation file.
 - a. Using Windows Explorer on the Galaxy Repository computer, proceed to the following folder:
C:\Program Files (x86)\ArchestrA\Framework\FileRepository\Galaxy_Name\ObjectFileStorage2
where *Galaxy_Name* is the assigned name of your Galaxy.
 - b. Type plc in the search field near the top right of the Windows Explorer window.
A list of files containing plc in their names appears.
 - c. Right-click a listed file to show a shortcut menu and select **Open file location**.
You should see the default translation file listed in the folder.

Name	Date modified	Type	Size
AppManifest.xml	3/30/2020 2:52 PM	XML File	1 KB
ArchestrA.Apps.PLCViewerApp.dll	4/21/2020 11:34 A...	Application extens...	99 KB
ArchestrA.Apps.PLCViewerAppCommon.dll	4/21/2020 11:34 A...	Application extens...	31 KB
ArchestrA.Apps.PLCViewerAppCommon....	4/21/2020 11:34 A...	PDB File	74 KB
ArchestrA.Apps.PLCViewerAppEditor.dll	4/21/2020 11:34 A...	Application extens...	95 KB
ArchestrA.Client.RuntimeData.dll	4/2/2020 2:56 PM	Application extens...	46 KB
ArchestrA.Prometheus2.LoaderCommon....	4/23/2020 10:25 A...	Application extens...	635 KB
ArchestrA.Prometheus2.LoaderCommon....	4/23/2020 10:25 A...	PDB File	1,034 KB
CCMLoadLib.dll	4/23/2020 10:25 A...	Application extens...	121 KB
Default.xml	4/23/2020 10:24 A...	XML File	50 KB
OpcComRcw.dll	4/2/2020 2:55 PM	Application extens...	89 KB

2. Copy the default translation file and place the copy in an accessible folder.
3. Rename the copy of the default translation file.
4. Modify the copied translation file to access more data from the RSLogix applications.
5. Save your changes.

Continue to [Integrate information from a custom translation file](#) for instructions on configuring a custom translation file.

Integrate information from a custom translation file

After creating a custom translation file, you need to integrate the data from it into the PLCViewerApp. You follow a procedure similar to accepting the default translation file.

To integrate information from a translation file to the PLCViewerApp

1. Place your custom translation file in a folder that is accessible to the computer containing the imported PLCViewerApp.
2. Open the System Platform IDE and select the **Graphic Toolbox** tab to show a list of folders containing symbols and apps.
3. Locate the PLCViewerApp control within the **Toolbox**.
By default, the PLCViewerApp control is placed at the root folder of the **Toolbox** when it is imported into the System Platform IDE.
4. Select the PLCViewerApp control to show the **PLCViewerApp** dialog box.
5. Select the **Translations** tab
6. Select **Add** to add a row to the **Translations** field.
7. Double-click in the gray area beneath the **Translation Name** column to show a data entry field.
8. Type the name of the translation file.

Controllers [Translations](#)

Translation Name

Line4_p01

9. Click the browser button at the right of the **Translation File** field and locate the folder where the XML translation file has been saved.
10. Select the XML translation file from the folder and click **Open**.
11. Repeat steps 6 through 10 to add other translation files as required.
12. Save your changes.

Configure PLCViewerApp properties

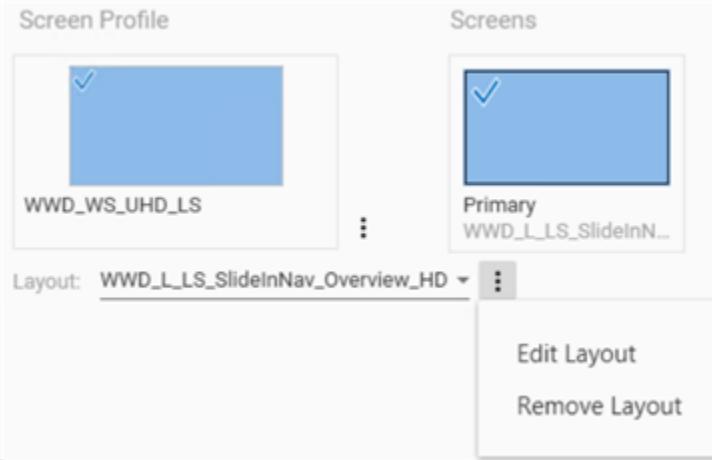
You place the PLCViewerApp into a ViewApp by selecting it from the Toolbox and dragging and dropping it onto a layout pane. After you place the PLCViewerApp on a pane, you can view its properties. Properties are assigned default values that can be modified or accepted without changes.

Note: You must configure PLC data sources before placing the PLCViewerApp on a layout pane. See [Configure PLC data sources](#)

Placing the PLCViewerApp onto a layout pane can be done from the Layout editor or ViewApp editor. The following procedure explains how to integrate the PLCViewerApp into a ViewApp from the Layout editor.

To configure PLCViewerApp properties

1. From the System Platform IDE, select the **Template Toolbox** tab to show the list of asset templates.
2. Open the ViewApp template instance that you want to place the PLCViewerApp in.
3. From the **Layout** field, open the drop-down list and select **Edit Layout**.



The Layout editor appears with a set of tabs at the right of the editor window.

4. Select the **Toolbox** tab and locate the PLCViewerApp in the list of controls and symbols.

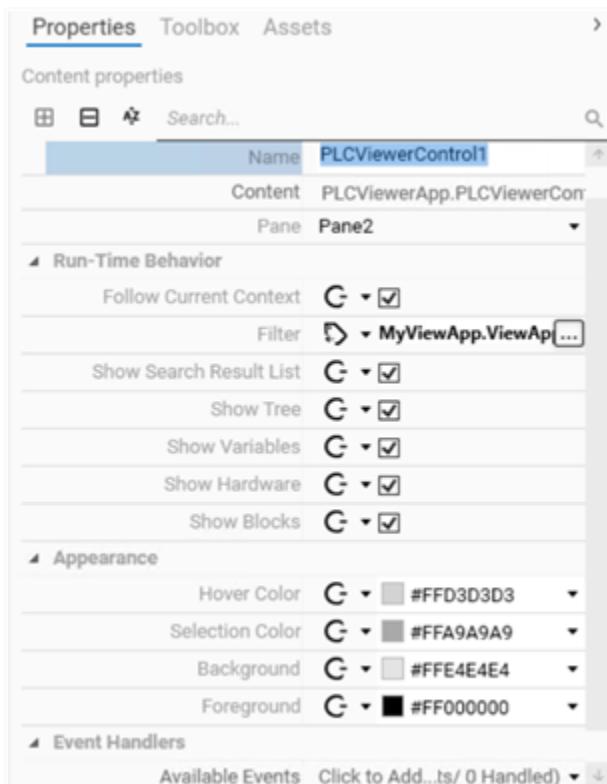
The default app location in the System Platform IDE is:

Toolbox > _Default Content > 4. Apps > InTouch OMI Apps

5. Select the PLCViewerApp. This displays its thumbnail graphic beneath the **Toolbox** list of controls and symbols.



6. Drag and drop the PLCViewerApp thumbnail onto the desired layout pane.
7. Select the PLCViewerApp thumbnail in the layout pane and select the **Properties** tab to show its properties.



These properties specify the runtime and appearance of the PLCViewerApp while running in a ViewApp. Each property has been assigned a default. You can accept the default or select another value.

The table following this procedure describes the **Run Time Behavior** and **Appearance** properties.

8. Save your changes and close the Layout editor to return to the ViewApp editor.
9. Save your changes and close the ViewApp editor.

Follow Current Context	Boolean value that determines whether the app view follows the current context of the selected components or not.
Filter	String, expression, or a regular expression that filter out PLCViewerApp components from the view except those whose asset names match the filter value in the current context.
Search Result List	Boolean value that determines whether the Find Results area appears within the PLCViewerApp window or not.

Show Tree	Boolean value that determines whether the Assets and Function Block Templates tree area appears within the PLCViewerApp window or not.
Show Variables	Boolean value when set to true the Variables node appears in the Controller navigation tree to navigate within the Variables grid.
Show Hardware	Boolean value when set to true the Hardware node appears in the Controller navigation tree to navigate within the Hardware grid.
Show Blocks	Boolean value when true the Function Block node appears in the Controller navigation tree to navigate within Function Blocks .
Hover Color	Background color of a listed item in the Assets and Function Block Templates tree in which the mouse is hovering over.
Selection Color	Background color of a selected item in the Assets and Function Block Templates tree.
Background	Background color of the PLCViewerApp window except the Find Results area, which is white.
Foreground	Font and icon color that appears in the Assets and Function Block Templates tree.

Continue the standard System Platform IDE workflows to place an instance of the ViewApp beneath a ViewEngine or WebViewEngine in the **Deployment** view of the IDE. Then, deploy the ViewApp and its supporting assets.

PLCViewerApp runtime version check

This version of the PLCViewerApp includes a runtime version check to alert the user when the online program has been modified. During configuration, the user enters the online version values of a controller. These values can be read from the object viewer or by opening the running PLCViewerApp and reading the Online Version error tooltip.

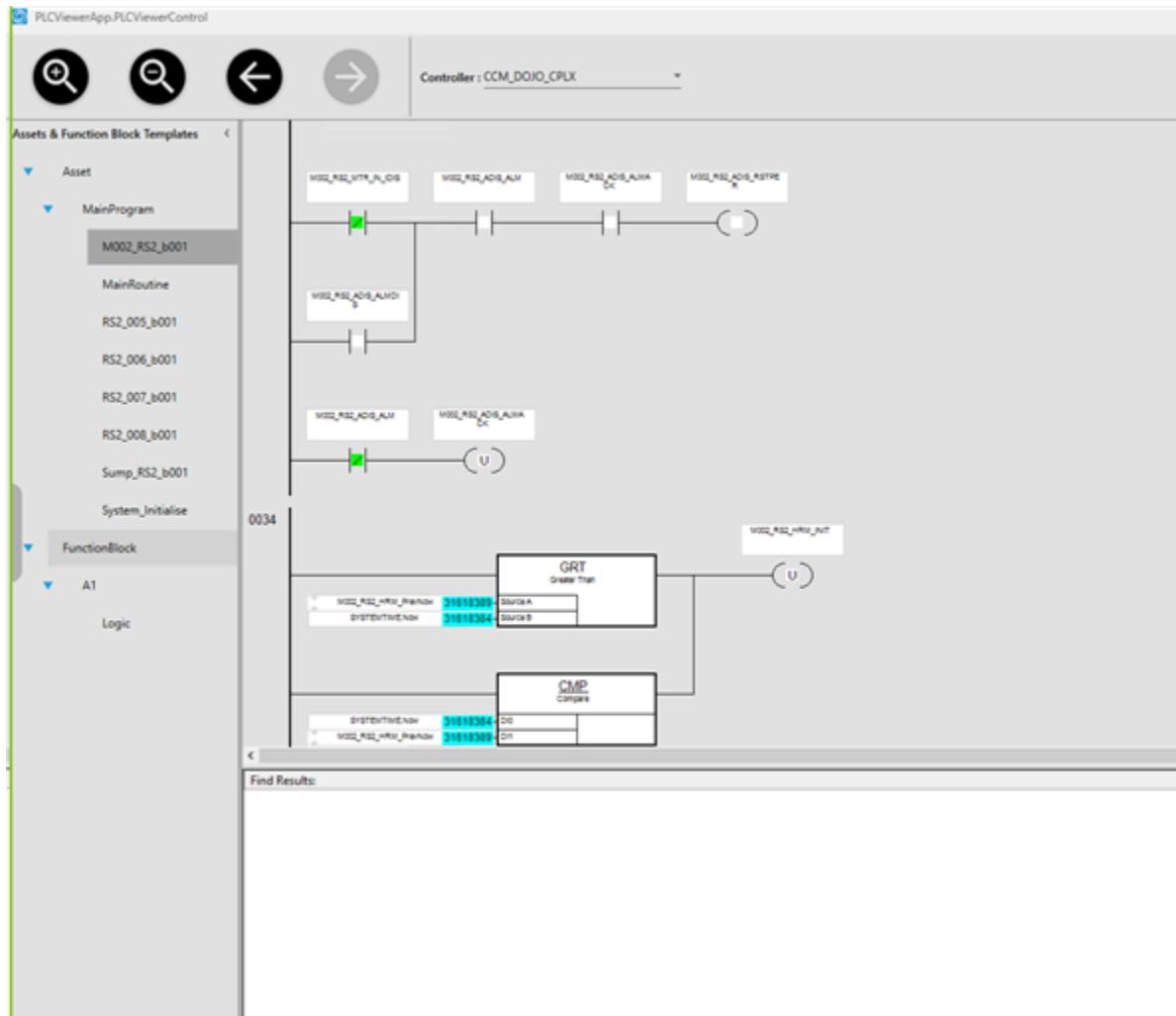
Also, version values can be copied to the clipboard by selecting the Online Version error and right-clicking to show a shortcut menu and selecting the **Copy** command.

The Rockwell runtime version change detection has some limitations:

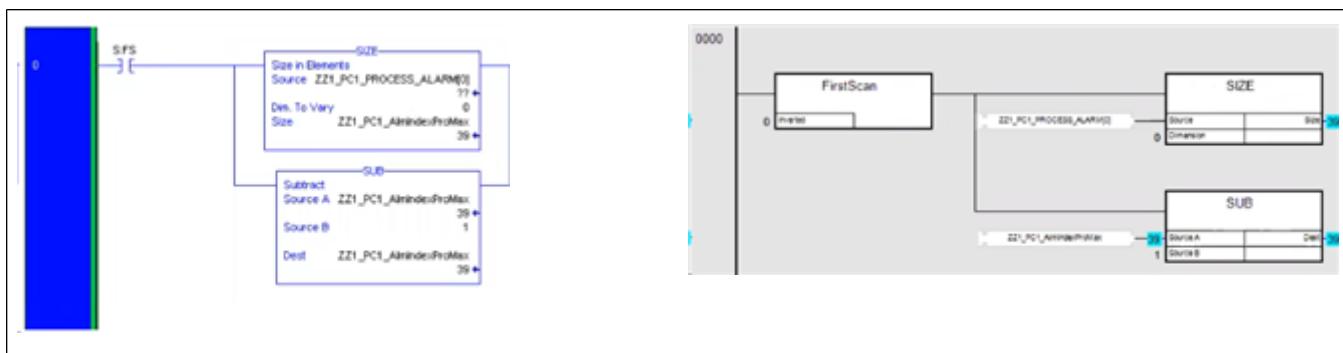
- Only online changes to the variable database are detected.
- Setting the online version to “1.0” will not detect changes due to subsequent downloads.

PLC ViewerApp during runtime

The following figure shows an example of the PLCViewerApp during runtime. The app starts with the default controller assigned during configuration.



The **Assets & Function Block Templates** column at the left shows a hierarchical navigation tree of the controller's Rockwell programs. For example, the figure at the left below shows a Rockwell alarm handler routine. The figure at the right shows the same routine displayed by the PLCViewerApp. The SIZE and SUB function blocks show the same values in both figures. In the PLCViewerApp, current asset values appear next to the function blocks with a blue background.



- Users select an item listed in the **Assets & Function Block Templates** column to set the context of controller information shown by the PLCViewerApp.

The PLCViewerApp incorporates color animation to identify the current state of data:

- Grey: Binding
- Purple: Bad data
- Cyan: Good data (background color is used to highlight animated values)
- Blue: Private (indicates the data is not accessible to the HMI)

Runtime enhancements of the PLCViewerApp

This version of the PLCViewerApp includes a new set of runtime enhancements in addition to the runtime features introduced in Version 1.

Additional support for languages

This version of the PLCViewerApp includes support for the following languages:

- Sequential Function Chart (SFC)
 - Function Block Diagrams (FBD)
 - Structured Text (ST)
- Variables tree grid

This version of the PLCViewerApp shows all controller variables with a check box option. When the check box is selected, the PLCViewerApp displays the current online value of each variable.

Hardware tree grid

This version of the PLCViewerApp incorporates a runtime tree grid that displays the current values of hardware controller racks, modules, and other devices in the network topology.

DDESuiteLinkClient support

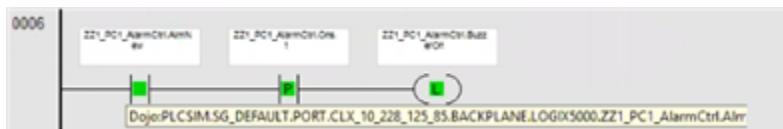
DDESuiteLinkClient support has been added for RSLogix controllers. The IO Server Path has been updated to support the attribute() method to call a tag reference as ".attribute(tagname)".

See more information from the PLCViewerApp

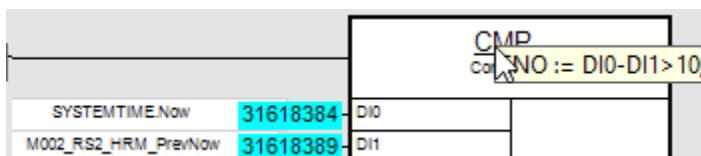
In addition to showing current PLC controller values during runtime, the PLCViewerApp provides other

information about the executing Rockwell programs shown in the current view context.

The PLCViewerApp includes a system of flyover help, which is shown by hovering your mouse directly over a function block to show the asset identifier associated with the block.

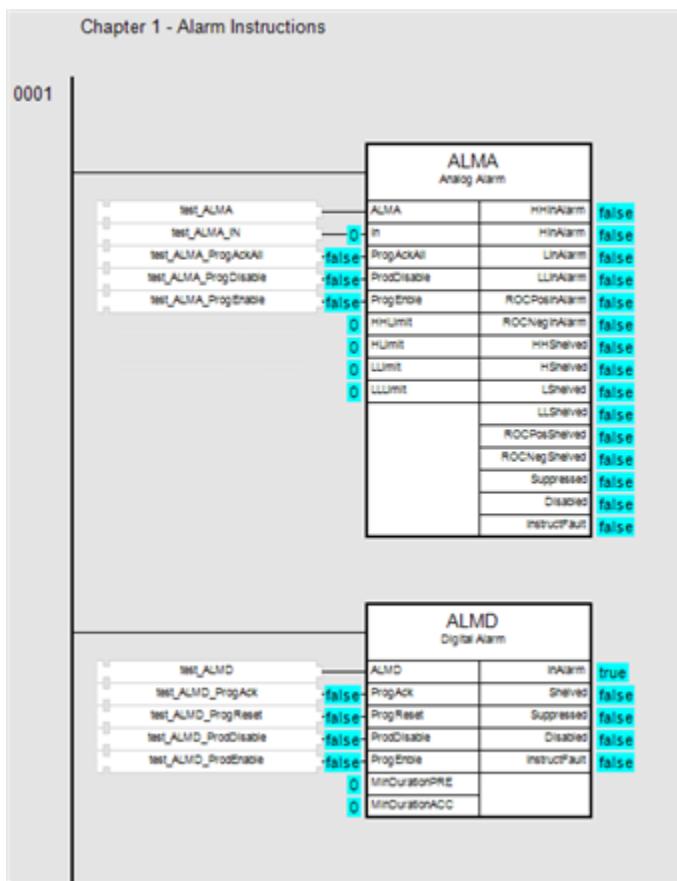


The flyover help can also show embedded Rockwell Compute expressions by hovering the mouse over a compute block title.



The PLCViewerApp shows the expression in its parameterized form.

Comments that appear inline in Rockwell code also appear in the same context in the PLCViewerApp.



Set view context

Users can select items from the **Assets & Function Block Templates** area of the PLCViewerApp window to set the context of a view. Also, users can select navigation items from a running ViewApp to set the context of a view. For more information about showing current PLC values of a ViewApp navigation item, see [Navigation from a](#)

[ViewApp.](#)

The PLCViewerApp includes a set of navigational and context controls in a menu bar near the top of the window.



The **Zoom In** and **Zoom Out** controls at the left enable you to zoom in or out to show more or less detail in the view window.

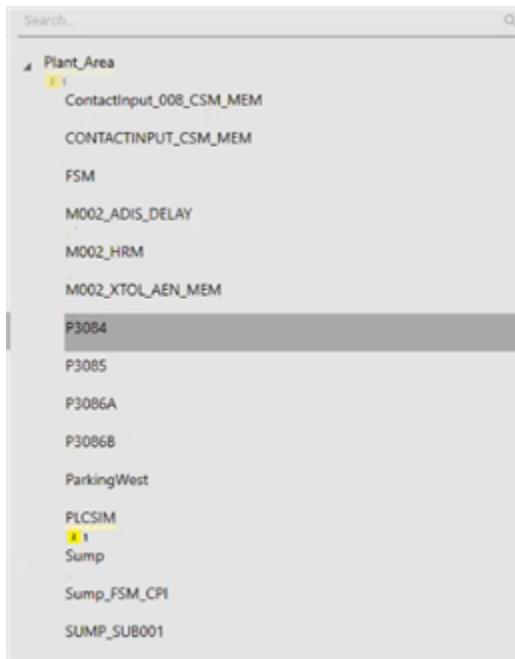
The **Back** and **Forward** controls enable you to move between previously shown context views in your current session. For example, if you have looked at three views in the PLCViewerApp, selecting **Back** will show the second context view. Using **Back** and **Forward** controls eliminates the need to scroll through the **Assets & Function Block Templates** area to reselect an item to show its view again.

The **Controller** field provides a drop-down list of controllers whose source logic files were added to the PLCViewerApp while configuring data sources. The source logic file includes the controller's programs. The PLCViewerApp can only show program data from the selected controller during runtime.

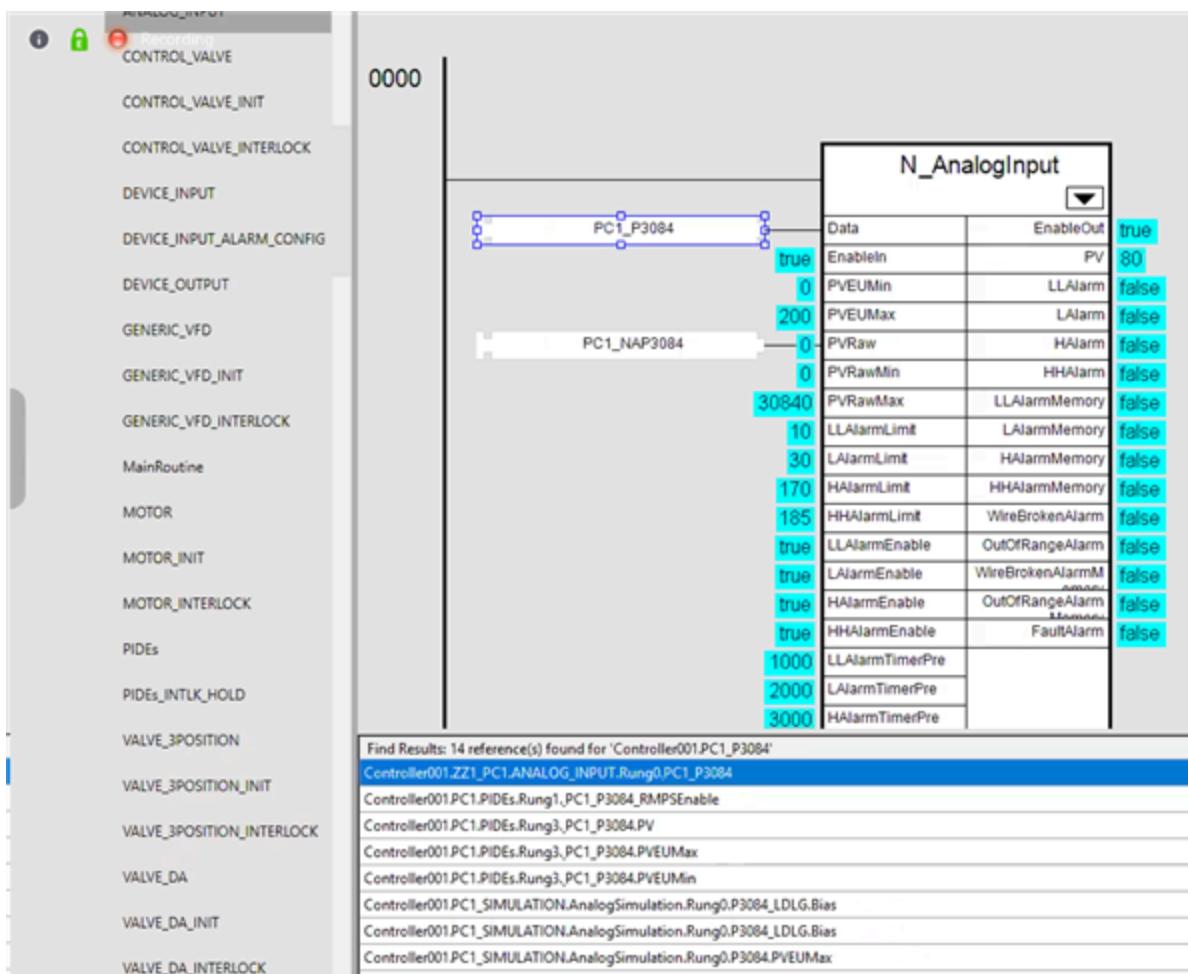
The **Scope** field appears when selecting an add-on instruction to enable selecting a specific instance of an add-on instruction in the control for animation.

Navigation from a ViewApp

The PLCViewerApp works in conjunction with other apps that are included in a running ViewApp. The following figure shows asset P3084 has been selected from the NavTree app placed in a layout pane of a ViewApp. A NavTree app shows the navigation model of a ViewApp and P3084 has been selected.

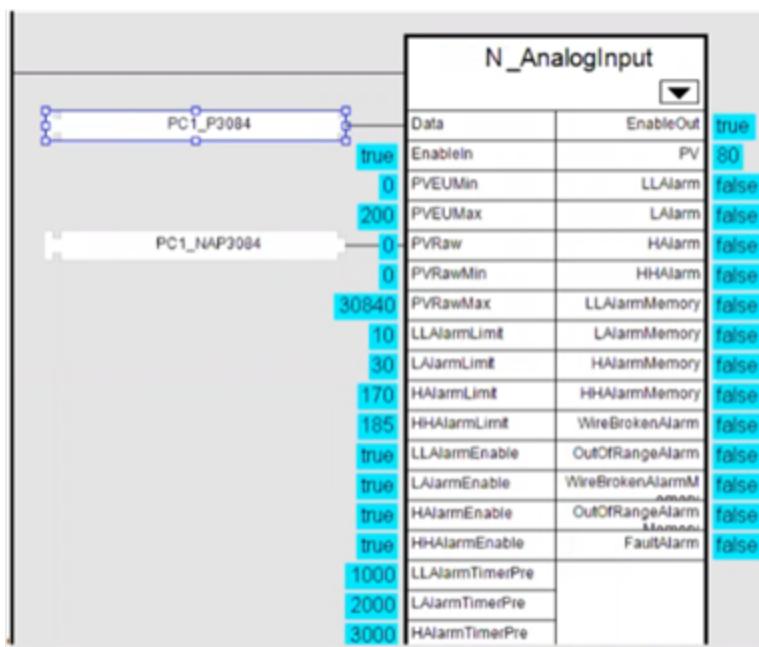


The ViewApp focus is set to the pane hosting the PLCViewerApp, which sets the view context to a function block in which PC1_P3084 is a data input



The **Find Results** field shows all instances of P3084 in Controller001. The Asset Mapping table with its list of associated asset names, controllers, and asset identifiers narrows the number of P3084 instances to the matching N_AnalogInput function block.

This function block shows the attributes and their values from the data of PC1_P3084 in a single view.



SpreadsheetViewerApp

The SpreadsheetViewerApp shows an Excel spreadsheet within a pane of a ViewApp while the ViewApp is running or is in preview mode. The SpreadsheetViewerApp initially shows the first worksheet of the spreadsheet at 100 percent zoom scale. No permanent editing changes can be made to a spreadsheet while it is displayed in the SpreadsheetViewApp.

A	B	C
1 Maintenance Period	Type of Maintenance	Maintenance Performed
2 Weekly: 12-07-2018/12-14-2018	Weekly	Changed pump oil and inspected bearings
3 Weekly: 12-15-2018/12-21-2018	Weekly	Changed pump oil and inspected bearings
4 Weekly: 12-22-2018/12-28-2018	Weekly	None. Christmas break.
5 Weekly: 12-29-2018/01-04-2019	Weekly	Replaced number 3 bearing
6		
7		
8		
9		
10		
11		
12		
13		

Below the spreadsheet, there is a status bar with tabs: Weekly Maintenance, Monthly Maintenance, Annual Maintenance, Defect Tracking, Part Replacement, Down Time, and Hour Logs. There are also zoom controls (magnifying glass and plus/minus) above the status bar.

The SpreadsheetViewerApp includes a set of visual controls that appear above the displayed spreadsheet. These controls can be used to change the zoom level of a spreadsheet.



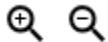
The SpreadsheetViewerApp also includes a slider zoom control in the status bar beneath the displayed document. Moving the slider left or right zooms the spreadsheet view in or out.



Change the zoom level of a spreadsheet

The SpreadsheetViewerApp provides several methods to set the zoom level of the worksheet shown in preview mode or a running ViewApp. A worksheet can be zoomed out to a minimum of ten percent of its original size or zoomed in to a maximum of 400% using the slider magnification tool, or to an infinite percentage using the zoom control magnifier icon.

- Zoom in and zoom out controls



These controls appear above the displayed document on the menu bar. Each time the user selects a zoom in or zoom out control, the view magnification increases or decreases by 15 percent.

- Zoom slider control



The slider control appears beneath the displayed document on the status bar. Moving the slider right or left increases or decreases the magnification of the document view.

Users can also select the + or - controls at the ends of the slider. Each press of the + or - control increases or decreases magnification in increments of 10 percent.

Touch support

The SpreadsheetViewerApp supports single finger gestures on touch screens or portable devices.

- A spreadsheet can be flicked vertically or horizontally to move to different areas of the spreadsheet.
- A single finger tap can be used to select the visual icons shown within the viewer window of the SpreadsheetViewerApp.

See [Optimize the SpreadsheetViewerApp for a touch device](#) for instructions to enlarge the size of the visual icons to make it easier to select them while using touch gestures.

Optimize the SpreadsheetViewerApp for a touch device

The visual icons shown within the viewer window of the SpreadsheetViewerApp can be enlarged to make them easier to select on touch screens or portable devices.

The following screen captures show the default size of the SpreadsheetViewerApp's visual icons and their size after they have been enlarged for touch. AVEVA OMI includes the `OptimizeForTouch` Boolean attribute that belongs to the `MyViewApp.Settings` namespace. The default value of `OptimizeForTouch` is false. When `OptimizeForTouch` is set to true, the size of the visual icons is enlarged.

OptimizeForTouch Attribute Value	Icon Size	Example Icons
MyViewApp.Settings.OptimizeForTouch=False	Regular Size (Non-Touch)	 
MyViewApp.Settings.OptimizeForTouch=True	Optimized for Touch	 

A common way to dynamically adjust the size of the visual icons in a running ViewApp is to place a graphic element like a button on a pane and then associate user input animation to toggle the state of the OptimizeForTouch attribute.

Compatibility of the SpreadsheetViewerApp

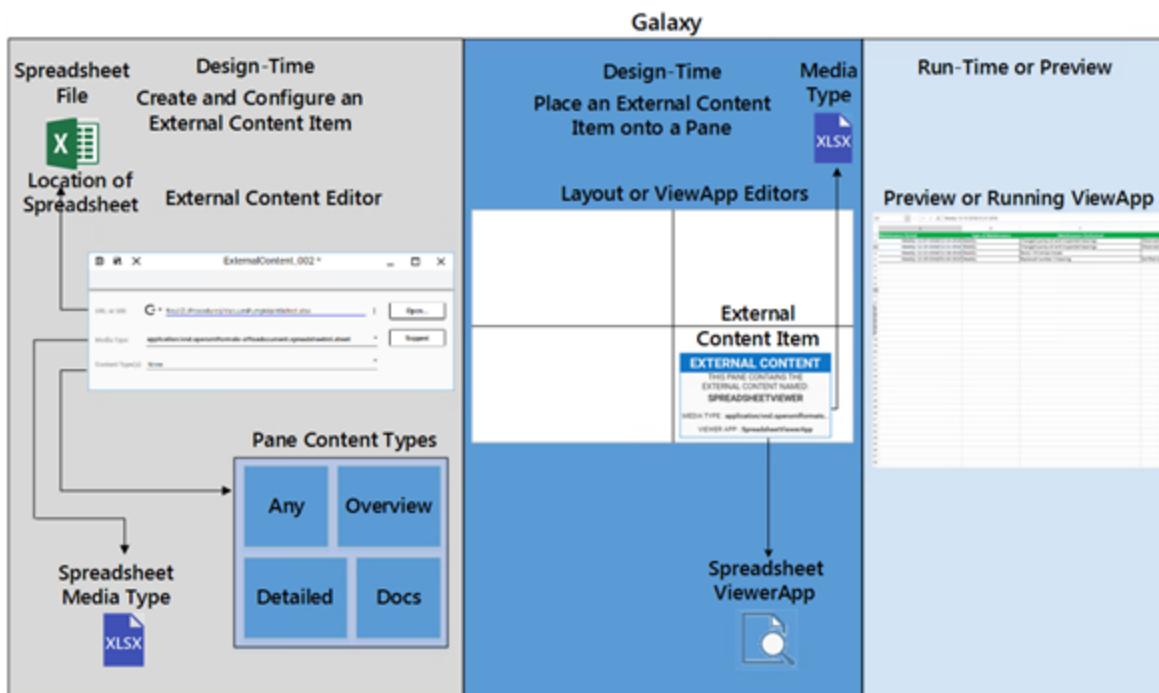
The SpreadsheetViewerApp includes compatibility with current and future versions of System Platform.

- The SpreadsheetViewerApp package must be installed on a computer running System Platform 2017 Update 3. The SpreadsheetViewerApp cannot be installed on any other version of System Platform except Update 3.
- After the SpreadsheetViewerApp is installed on a System Platform 2017 Update 3 system, the SpreadsheetViewerApp will be able to migrate to future versions of System Platform InTouch OMI when they become available.

Configure the SpreadsheetViewerApp

The following figure summarizes using an External Content item to point to a remote computer containing a spreadsheet file. An External Content item is configured during design time to identify the location of the spreadsheet and its media type.

The External Content item is associated with an asset or a user defined object. The asset or object is selected from the **Toolbox** tab and placed onto a layout pane. During runtime, the user selects an asset associated with a navigation item to view the documentation included in the spreadsheet.



The major steps to implement the SpreadsheetViewerApp to show spreadsheets in a ViewApp are summarized in the following list:

1. Place a spreadsheet on a computer accessible from your Galaxy.
2. Create an External Content item.
3. Configure the External Content item that identifies the location and media type of the spreadsheet.
4. Associate an External Content item to a graphic or object.
5. Place an asset or graphic on a ViewApp pane associated with the External Content item.

Before you begin

Before starting the steps to configure an External Content item, ensure the following prerequisites have been met:

- If you are placing the spreadsheet on a remote computer in your network, ensure connectivity between the computer where the ViewApp will run and the computer where the remote spreadsheet has been saved.

Mapping a network drive is a typical way to connect to a remote computer hosting documents that you want show in a running ViewApp. For this configuration to work successfully, it is important that the computer to which the ViewApp is launched from has the same drive letter mapped to a location having the referenced document.

The SpreadsheetViewerApp shows an error message to the user if the specified URI connection to the Excel spreadsheet file is bad. For more information about the error message, see [SpreadsheetViewerApp error message](#).

- Save the spreadsheet in the presentation format that you want to display in the SpreadsheetViewerApp.

Important: The SpreadsheetViewerApp only supports the xlsx Excel file type. The earlier xls Excel file type is not supported.

SpreadsheetViewerApp error message

An External Content item includes an **URL or URI** option to specify the location of the Excel file to be shown by the SpreadsheetViewerApp during runtime or in preview mode. The specified location of the file must be complete and accurate. If the file is moved, deleted, or renamed, the SpreadsheetViewerApp shows an error message to the user during runtime, but the ViewApp continues to run.



Create an External Content item

You create an External Content item from the Visualization folder of the System Platform IDE. Initially, the External Content item defaults to a set of values that can be changed using the External Content editor.

To create an External Content item

1. Open the IDE and select the **Visualization** tab.
2. Select a folder within the **Visualization** folder if you want to create a new External Content item at a specific location.
3. On the **Home** ribbon, in the **Create** area, select **External content**.

You can also create an External Content item by other methods:

- **Keyboard Shortcut**

Press Ctrl + Shift + C

- **Shortcut Menu**

Right-click a folder of the **Visualization** folder to show a shortcut menu. Select the **New** option, and then select **External Content**.

A new **External Content** item is created in the **Visualization** folder.



The name of the new item follows a default naming convention of appending a three-digit number to the word ExternalContent.

4. Rename the External Content item.

After you create an External Content item, it must be configured to specify the remote location of the content and its media type.

Configure an External Content item for the SpreadsheetViewerApp

Each External Content item has three properties that must be assigned values:

- **URL or URI**

- **Media Type**
- **Content Type(s)**

An External Content item is modified from the External Content editor. The editor can be opened by double-clicking on an item from the Visualization folder. Also, an External Content item that has been placed onto a layout pane can be edited from the Layout and ViewApp editors by the selecting the item from the Actions list.

To configure an External Content item

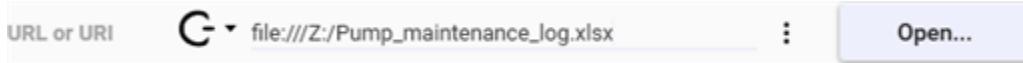
1. Open the IDE and select the **Visualization** tab to show the External Content items available in the Galaxy.
2. Double click an External Content item to open it in the External Content editor.

For more information about each field in an External Content item, see [Details of the External Content editor](#).

3. Enter a URI string or web site URL in the **URL or URI** field where the spreadsheet is saved.

The value you enter is the location of the external content specified by a URI-formatted string. A media location must be specified.

You can browse for the spreadsheet file by selecting the vertical dots icon to the right of the data entry field. A drop-down list includes a **Browse for file** option that enables you to browse your network and select a media file. The **URL or URI** field updates and shows a formatted URI path to the file.



You can also enter a URL to a web site by selecting the **http://** or **https://** options from the drop-down list. The **http://** and **https://** options assist the user when entering URLs by prefixing the URLs with the selection.

See [More information about specifying a URI](#) for information about the format of a URI string.

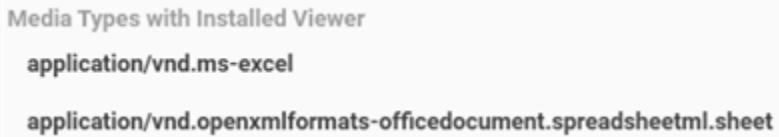
4. Select the **Open** button to validate the URI or URL you entered.

The **Open** button launches the appropriate application associated with a xlsx file type, which is usually Excel to display the document specified in the **URL or URI** field.

5. Select the type of media associated with spreadsheets from the drop-down list of the **Media Type** field.

The value you enter is the MIME type of a spreadsheet. The SpreadsheetViewApp provides two Excel spreadsheet MIME types, as shown in the following screen capture.

- The **application/vnd.ms-excel** Media Type is for legacy .xls Excel files.
- The **application/vnd.openxmlformats-officedocument.spreadsheetml.sheet** Media Type is for .xlsx Excel files.

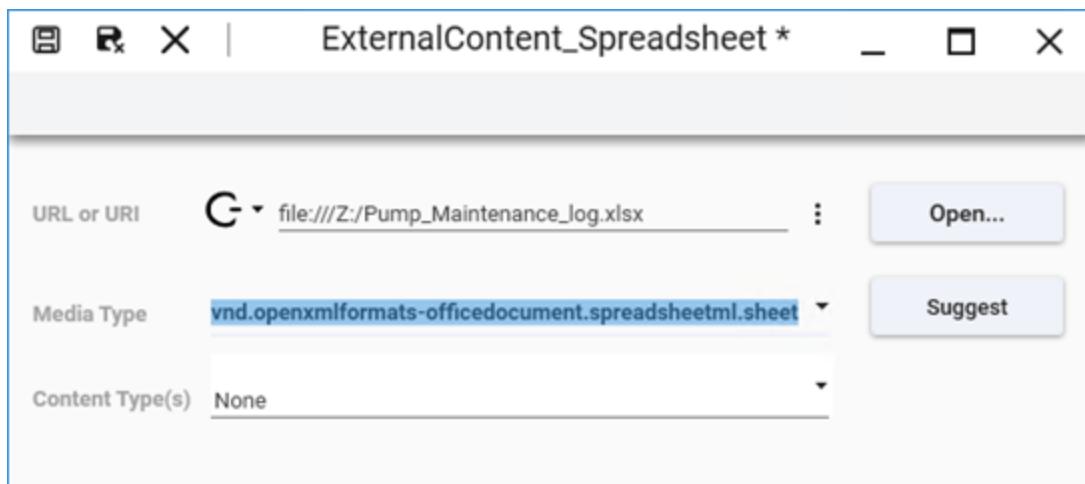


See [More information about media types](#) for information about supported media types.

6. Select the type of content from the drop-down list of the **Content Type(s)** field.

The value you enter is used by content-placement algorithms within AVEVA OMI during run time to determine the placement of content when AutoFill or ShowContent calls are invoked.

The following screen capture shown an External Content item configured for a spreadsheet.



7. Save your changes and exit from the External Content item.

Link an External Content item to an object or template

There are three ways to integrate an External Content in a ViewApp.

- Associate an External Content item to an object template, which is described in the following procedure
- Associate an External Content item to a custom navigation item that appears in the navigation model of the ViewApp.
- Place the External Content item on a layout pane, which is the simplest way and is described in [Adding an External Content item to a ViewApp](#).

To link an External Content item to an object template

After you have created and configured an External Content item, you can link it to an object template. External Content items can be linked only to objects as opposed to graphics, which can be linked to or owned by an object. All External Content items reside in the Visualization folder. A single External Content item can be linked to multiple objects.

1. Open an object template in the **Object Editor**.
2. Select the **Attributes** tab.
3. In the **Content** pane, select the **Link Content**  button.
The Galaxy Browser opens.
4. Navigate to the folder that contains the **External Content** item to be linked.
5. Select the item, then click **OK**.

The item is added to the object and appears in the **Content** tab.

Note: After linking an External Content item, you can modify it by selecting it and then pressing the **Edit** button. However, any changes you make will apply to all objects that link to the item.

6. Save your editing changes to the object template and exit from the Object Editor.
7. Create an instance of the linked object template.
8. Create a layout which contains pane with a Content Type that matches the content type of the External Content item.

9. Add a navigation tree to the layout
10. Create a ViewApp that incorporates the layout.
11. Deploy the ViewApp.

In runtime, navigate to the object or any instance created from the template. External Content item will auto-fill the matching pane to show the external content.

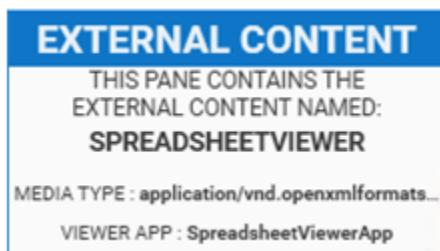
Add an External Content item to a ViewApp

You can place an External Content item directly onto a pane from either the Layout or ViewApp editors . External Content items are listed in the **Toolbox** or **Assets** tabs of either editor.

To add external content to a ViewApp

1. Open the System Platform IDE and select the **Visualization** tab to see the list of layouts.
2. Double-click the layout that you want to add external content to. It opens in the Layout editor.
3. Select the **Toolbox** tab of the Layout editor to show the content accessible from the Galaxy.
4. Locate the External Content item you want to place in the layout you selected.
External Content thumbnails appear beneath of the list of content shown in the **Toolbox** list.
5. Drag and drop an External Content thumbnail from the **Toolbox** area onto a pane of the layout.

The External Content thumbnail appears in the pane containing the name of the External Content item, the media type of the external content, and the viewer app that will display the media during runtime. Also, the layout **Actions** area shows the name of the External Content item and the pane that it was placed in.



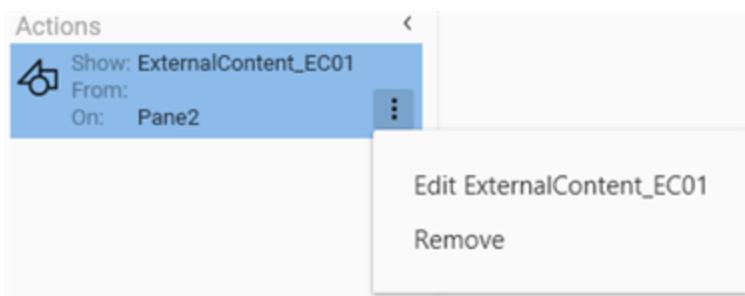
Editing an External Content Item After Being Placed onto a Pane

An External Content item can be modified after being placed in a layout or a ViewApp, which automatically propagates the changes

To edit an External Content item in a layout

1. If necessary, open the layout containing an External Content item.
2. Locate the item in the layout **Actions** list.
3. Select the **Options** icon shown at the right of the item in the **Actions** list.

A drop-down list shows an Edit External Content command.



4. Select the **Edit** command.

The External Content editor opens the item for editing.

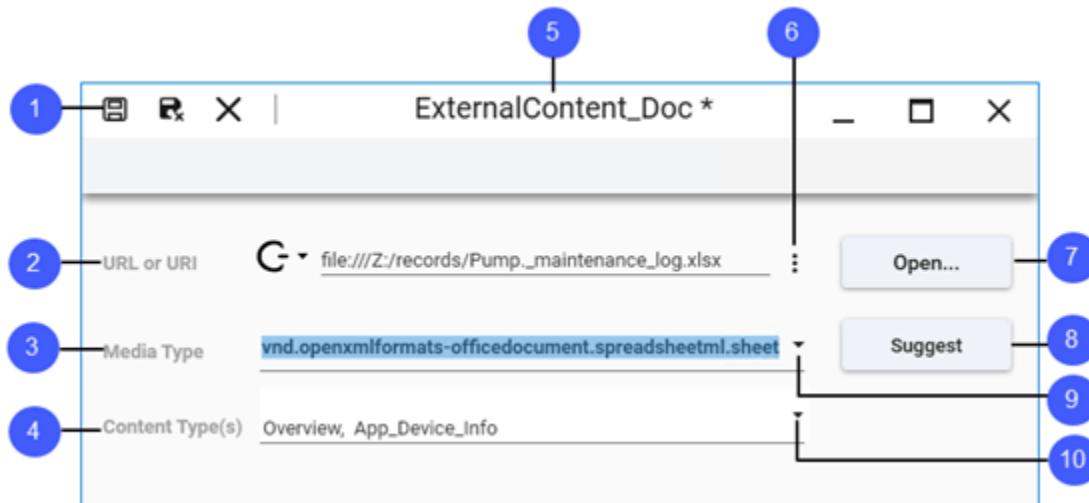
5. Update the values assigned to the item and save your changes.

The External Content thumbnail placed in a pane updates to reflect the changes made to the item.

Details of the External Content editor

You use the External Content editor to configure an External Content item for a document. Before you configure an External Content item, you must know:

- Network location of a document
- Media type of the document
- Content type(s) of the layout pane that will host the External Content item



1	Commands to save, save and close, and close the External Content editor.
2	URL or URI that specifies the location of external media using a standard format. See Specify the URL of External Content for the format of a URI or URL. A warning message appears if the document cannot be located during runtime.

3	<p>A media type is a two-part identifier that specifies the type of application required to process or view remote content. A media type can be entered in the field or selected from a drop-down list. See Specify media types for External Content for the format of a Media type value.</p> <p>A warning message appears if a media type is specified that does not have an associated viewer application.</p>
4	<p>Content type assigned to the external media that enables ViewApp algorithms to place content in specific panes during runtime.</p>
5	<p>Name of the External Content item.</p>
6	<p>Drop-down list with options to browse for an external content file or specify a URL using HTTP or HTTPS.</p>
7	<p>Validates the location of external media specified in the URL or URI field. An attempt is made to display the external media in an application assigned as the default by the operating system, not the app specified for the external content media type.</p> <p>A warning message appears if the content cannot be found at the location specified in the URL or URI field.</p>
8	<p>Suggest is optional. When selected, it automatically selects a media type based on the entry in the URL or URI field. The entered value can be changed if the suggestion does not match the expected media type.</p> <p>Note: Ensure the suggested media type is correct. The external media does not appear if the media type is incorrect.</p>
9	<p>Drop-down list of commonly used media types. Media types that have apps in the Galaxy capable of servicing them are shown in bold text.</p>
10	<p>Drop-down list of content types that can be used at preview\runtime to assign the external content to panes of a matching content type.</p> <p>Note: None is not a content type. Instead, it simply means a content type has not been assigned to the</p>

pane.

More information about media types

A media type is a standard two-part string analogous to a MIME type that identifies external content file types and their format. System Platform uses a media type to identify external content and the type of app required to display media during runtime.

A media type consists of a type and a subtype, which is further structured into a tree.

type "/" [tree "."] subtype ["+" suffix] *[";" parameter]

For more information about the format of media types, see <https://www.iana.org/assignments/media-types/media-types.xhtml>

System Platform provides several default media types that include associated apps to display media. The drop-down list of the **Media Type** field shows the default media types in bold text to indicate a viewer app is available in the Galaxy that can service external content with these media types. Other listed media types that appear in plain text require an app to be created and imported into a Galaxy to play the specified media.

More information about specifying a URI

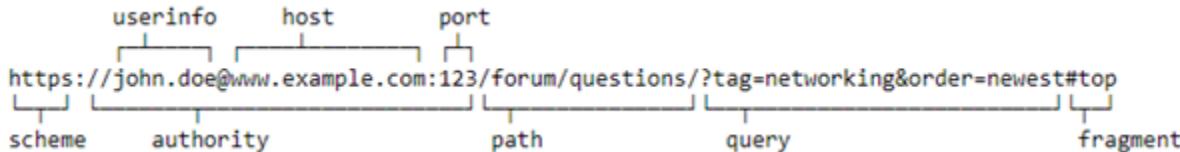
In the context of External Content, a Uniform Resource Identifier (URI) is a string that identifies the pathway to specific content that can appear in a running ViewApp.

All URIs adhere to predefined syntax rules.

- Generic URI syntax consists of a hierarchical sequence of components

URI = scheme:[//authority]path[?query][#fragment]

Example



- Each URI begins with a scheme name followed by a colon
Examples of common scheme names include `http:`, `https:`, and `file:`
- An optional authority component preceded by two slashes (`//`)
 - Use information component consisting of a user name and optional password preceded by a colon followed by an at symbol (`@`)
`//username:password@`
 - Host subcomponent consisting of a registered hostname or IP address
- A path component consisting of a sequence of path segments separated by a slash (`/`). A path is always defined for a URI, though the defined path may be empty
- An optional query component preceded by a question mark (?), containing a query string of non-hierarchical data.

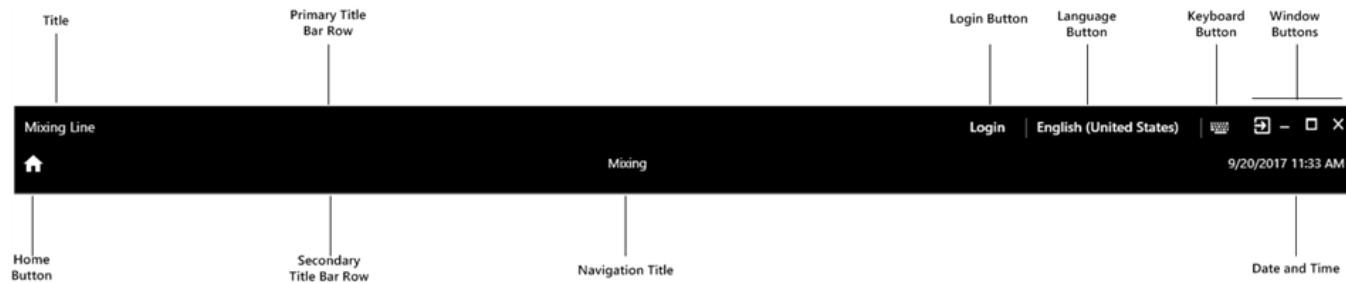
For more information about the format of a URI, see <https://tools.ietf.org/html/rfc3986>

The following examples show common URIs that identify External Content within a Galaxy

- `ftp://ftp.is.co.za/rfc/rfc1808.txt` -- ftp scheme for File Transfer Protocol services
- `file:///Z:/Media/Runtime%20Language%20Switching.mp4`
- `http://www.exampleserver.com/documents/index.html`

TitleBarApp

The TitleBarApp appears in a pane typically placed at the top of a ViewApp window. The app includes a set of properties to add functional components to the title bar. Many of the properties include positional options to place these components at specific locations on the title bar.



Configure a title bar

The TitleBarApp includes several sets of properties whose values determine the components that appear on a title bar, the component functionality, and how these components are positioned on a title bar.

To configure the TitleBarControl

1. Open the Layout Editor and create a pane at the top of the intended ViewApp window.
The pane width can extend to the full width or only partial width of the window.
2. Adjust the height of the pane.
3. Show the items listed in the **Toolbox** tab.
4. Open the **AVEVA OMI Apps** folder to show the list of default apps.
5. Select **TitleBarApp** from the **Toolbox** list.
A TitleBarApp thumbnail graphic appears at the bottom of the **Properties** grid area.
6. Drag and drop the TitleBarApp thumbnail to the layout pane you created.
7. Select the thumbnail within the pane and select the **Properties** grid.
The **Properties** grid shows the properties of the TitleBarApp. See [TitleBarApp properties](#) for the details of each property.
8. Assign values to the properties.
9. Save your changes.

TitleBarApp properties

TitleBarApp properties appear on the **Properties** page by the following functional groups. Many of the properties include fields containing options to position functional components at a specific location on the title bar. You can reset any property by selecting the arrow before the text entry field and selecting **Reset**.

The screenshot shows the AVEVA Properties page with the 'Properties' tab selected. The 'Content properties' section is expanded, displaying various configuration options for a TitleBarControl2 component. The 'Appearance' group includes settings for primary and secondary font families, sizes, and colors. The 'Content and Display Area' group includes settings for title, home, navigation title, keyboard, and various date/time fields. The 'Window Buttons' group includes checkboxes for minimize, maximize, close layout, and close viewapp. The 'Event Handlers' group is partially visible at the bottom.

Category	Setting	Value
General	Name	TitleBarControl2
	Content	TitleBarApp.TitleBarControl
	Pane	KPIs
Appearance	Primary Font Family	Roboto
	Primary Font Size	14
	Has Secondary Row	✓
	Secondary Font Family	Roboto
	Secondary Font Size	10
	Background	#FF000000
	Foreground	#FFFFFF
Content and Display Area	Title	AVEVA OMI
	Home	SecondaryLeft
	Navigation Title	SecondaryLeft
	Keyboard	PrimaryRight
	Login	PrimaryRight
	Style Libraries	Reset
	Language	None
	Date	None
	Time	None
Window Buttons	Show Minimize	✓
	Show Maximize	✓
	Show Close Layout	✓
	Show Close ViewApp	□
Event Handlers		...

Appearance Properties

Primary Font Family	Option to specify the font family for text shown on the primary row of a title bar. Roboto is the default font.
Primary Font Size	Point size of text shown on the first row of a title bar. 14 points is the default font size.
Has Secondary Row	Option to include or exclude a secondary title bar row immediately beneath the primary row. The default is to include a secondary title bar row. Note: Has Secondary Row must be selected to show the two Secondary Font properties listed below.
Secondary Font Family	Option to specify the font family for text shown on the secondary row of a title bar. Roboto is the default font.
Secondary Font Size	Point size of text shown on the secondary row of a title bar. 10 points is the default font size.
Background	Color of the TitleBarApp background shown in the pane of a running ViewApp. Select the Background field to show a small triangle at the right. Select the triangle to show a color picker to set the background color. Black or #FF000000 is default background color.
Foreground	Color of the TitleBarApp foreground shown in the pane of a running ViewApp. Select the Foreground field to show a small triangle at the right. Select the triangle to show a color picker to set the foreground color. White or #FFFFFF is the default foreground color.

Content and Display Area Properties

Title	Static text string to include on a title bar. AVEVA OMI is the default title.
Home	Shows a Home button, which the user selects to set the focus of the ViewApp to the root of the

	<p>ViewApp navigation model. The default is to show the Home icon.</p> <p>The property includes a drop-down list to select the location within a title bar to place the Home button. SecondaryLeft is the default location of the Home icon.</p>
Navigation Title	<p>Shows the title of the current selected navigation item. The default is to show the Navigation Title.</p> <p>The property includes a drop-down list to select the location within a title bar to place the Navigation Title. SecondaryLeft is the default location.</p>
Keyboard	<p>Shows a keyboard button, which the user selects to show an on-screen keyboard within a pop-up window. The default is to show the Keyboard icon.</p> <p>The property includes a drop-down list to select the location within a title bar to place the Keyboard button. PrimaryRight is the default location.</p> <p>Based on a retentive attribute setting in the MyViewApp.ViewApp namespace, you can show a standard Windows keyboard (default) or the built-in AVEVA OMI keyboard. See ViewApp Namespace Attributes in the AVEVA OMI SDK Help for more information.</p>
Login	<p>Shows an optional Login button, which the user selects to show a log in dialog box to enter a username and password. The default is to show the Login button.</p> <p>When a user logs in to a ViewApp, the Login button and drop arrow are shown. Pressing the button automatically logs out the current user from the ViewApp. Pressing the drop arrow shows options “Login” and “Logout”. The “Login” option enables a different user to log in to the ViewApp.</p> <p>The default position of the Login button on a title bar is PrimaryRight.</p>
Language	Shows an optional Language button, which the

	<p>user can select to show a list of languages configured for the ViewApp. The current language is the caption of the Language button. Selecting another language from the list changes the language used by the running ViewApp. The default is None to hide the Language button.</p> <p>The property includes a drop-down list to select the location within a title bar to place the Language button.</p>
Date	Shows the current date formatted as MM/DD/YYYY. The default is None to hide the Date value.
Time	Shows the current time formatted as HH:MM AM/PM. The default is None to hide the Time value.
Date and Time	<p>Shows the current date and time formatted as MM/DD/YYYY HH:MM AM/PM.</p> <p>The property includes a drop-down list to select the location within a title bar to place the date and time. SecondaryRight is the default location.</p>

Window Buttons Properties

Show Minimize	Inserts a Minimize icon  on the title bar that minimizes a running ViewApp window to an item on the window's task bar when selected. The default is to show the Minimize icon.
Show Maximize	<p>Inserts a Maximize icon  on the title bar that maximizes a running ViewApp window to full screen size when selected. The default is to show a Maximize icon.</p> <p>The Maximize icon changes to a Restore Down icon  on the title bar after the ViewApp window has been maximized. Selecting the Restore Down icon restores the ViewApp window to its previous size before it was maximized.</p>
Show Close Layout	Inserts a Close Layout icon  on the title bar, which closes the current layout running in a ViewApp window when selected. The default is to show the Close Layout icon.

Show Close ViewApp	Inserts a Close ViewApp icon  on the title bar that closes an entire running ViewApp when selected. The default is to hide the Show Close ViewApp icon.
--------------------	---

Event Handlers

Available Events	Lists all available events of the TitleBarControl from a list that appears by double-clicking on the data entry field. Select one or more events from the list to create event handlers that can be used in scripting. For more information about event handler scripts, see Write an event handler script .
------------------	--

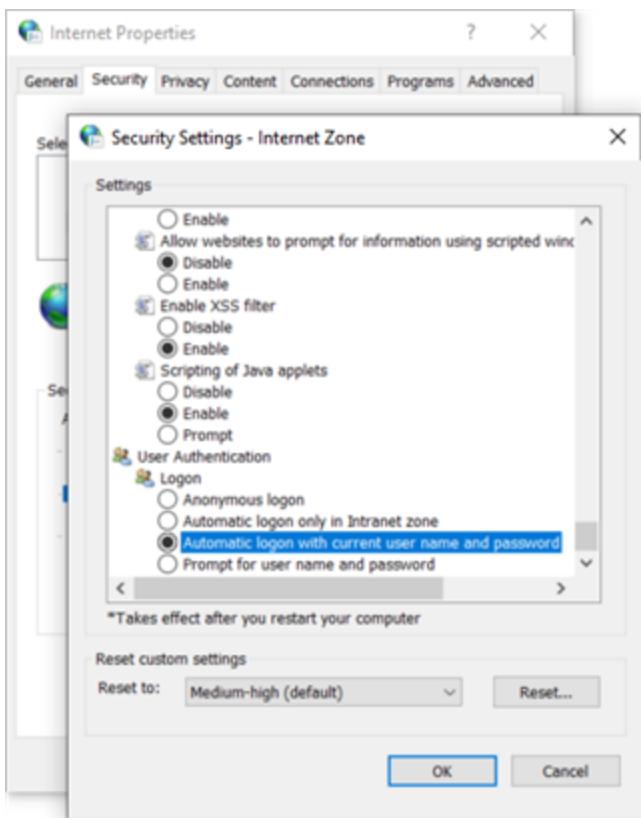
WWWebAppControls

AVEVA OMI provides a set of controls to show web-based content within a running ViewApp. WWWebAppControls includes two web controls, **Sharepoint** and **WebBrowser**, that you can add to a layout. With web-based content, the destination server determines how requests for data are processed. The credentials for the current logged-in Windows user are used to access the server. Therefore, you may have to change the internet properties of the computer running the ViewApp to allow automatic logon with the current user. If you get a message that you cannot access content in the Sharepoint app, WebBrowser app, or other app that uses web controls, you may need to change the web security settings on your computer.



To change internet security settings:

1. Open Windows **Internet Options**.
2. On the **Security** tab, select **Internet** zone, then select **Custom Settings**.
3. Scroll to **User Authentication**.
4. Change the logon setting to **Automatic logon with current user name and password**. The default setting for this zone is "Automatic logon only in Intranet zone."



- **SharePoint**

SharePoint is a web-based application that integrates with Microsoft Office and primarily functions as a document management and collaboration tool. SharePoint is highly configurable and its usage varies substantially between organizations.

Note: Configure SharePoint permissions to enable users to open and immediately show data in a running ViewApp. Otherwise, a message appears during run time asking users to log on to Sharepoint before viewing any content. In this document, SharePoint (with capital P) refers to the Microsoft Office application.

Sharepoint (lowercase p and bolded) refers to the OMI app.

- **Web browser**

AVEVA OMI includes a full-featured standards-compliant web browser, which supports WinForms and WPF apps. The web browser is based on Chromium Embedded Framework (CEF), the open source version of Google Chrome.

Note: Place the WebBrowser app in a sufficiently large ViewApp pane to reduce user scrolling to view web content.

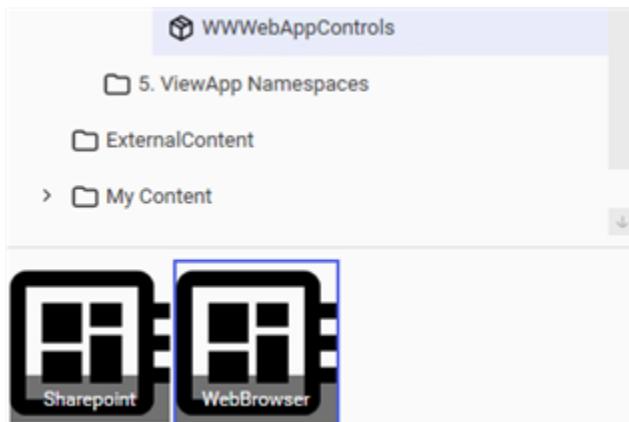
Configure a web control

All web controls include a similar **Address** property to assign the URL or IP address of a web server that provides the data shown by the Web control.

Important: You should complete all user security or authentication tasks before configuring the web control.

To configure the web control properties

1. Open the Layout or ViewApp Editor and show the items listed in the **Toolbox** tab.
2. Select WWWebAppControls from the **Toolbox** list to show its preview thumbnails.



3. Select the preview thumbnail of the web control that you want to place in your ViewApp.
4. Drag and drop the thumbnail on to a layout pane.
5. Select the pane containing the web control and select the **Properties** grid.

The **Properties** grid shows the **Address** property.

Property	Value
Name	WebBrowser1
Content	WWWebAppControls.WebBrowser
Pane	Faceplate
Address	(dropdown menu open)
StyleSheet	(dropdown menu open)

6. Enter the URL of the computer providing data for the Web control.

You can configure a static or dynamic URL.

Constant	Static binding to the specified URL. Enter the URL in the data field of the Address property.
In	Dynamic URL with read only binding by the control to a reference.
Out	Dynamic URL with write only binding by the control to a reference.

InOut	Dynamic URL with read/write binding by the control to a reference.
Reset	Reset the Address property to remove any specified URL.

7. If you want to create event handler scripts, select the **Available Events** data entry field to show a list of events. For more information about creating event handler scripts, see [Add an Event Handler Script](#).
8. Save your changes.

Using AuthenticationProviders mode with WebAppControls

The security mode for OMI apps is set by the app developer. Most of the default apps bundled with System Platform use OS User or OS Group security (Kerberos-based) for user authentication.

A new authentication mode, AuthenticationProviders, has been added in System Platform 2023. App developers can override this setting to allow multiple authentication modes for their custom apps. However, the authentication mode must be explicitly set for the app. See [WebControls SDK](#) in the *OMI SDK Help* for more information.

Sharepoint App

When AuthenticationProviders security mode is implemented, the **Sharepoint** control does not necessarily follow the security mode implemented by the ViewApp, which uses the security mode set in the System Platform IDE. As a result, users may see their Azure AD credential shown in the **Sharepoint** pane, but their Kerberos credentials will be shown for the ViewApp logged-in user.

Import OMI Apps (WebHelp)

An OMI App is collection of one or more controls primarily developed with Windows Presentation Foundation (WPF) for use in AVEVA OMI ViewApps. Other technologies such as Windows Forms (WinForms) and HTML5, can be incorporated into the app . You can create your own OMI Apps, which can be imported into an Application Server Galaxy.

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

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

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

To import a custom control

1. Assemble the control and all required dependencies and related files into a folder.
2. Place the folder containing the control at a folder location accessible to the computer hosting the System Platform IDE.
3. Open the System Platform IDE.

4. From the ribbon, select **Galaxy**, then select **Import**.
5. Select **Visualization**, then **OMI App**.
The **Import OMI App** dialog box appears.
6. Locate and select the folder containing the control to import.
7. Select **Select Folder**.

The control with other supporting files in the folder are imported. After the import process completes successfully, a new object appears in the Visualization folder, named *FolderName00n*, where *n* is 1 or the next available integer.

The imported control can be renamed, deleted, or exported as an .aaPKG file for use in another Galaxy. See [Export objects](#) for additional information.

Troubleshoot AVEVA or WPF apps that fail to import

There are two major factors that may stop an AVEVA or WPF app from successfully importing.

Expose controls and properties: Make sure that you have included an AppManifest.xml file with the app, and that all controls and properties that are needed in your app are explicitly exposed in it. For additional information about the AppManifest.xml file, see [AVEVA app development guidelines](#) and the [AVEVA OMI SDK Online Help](#).

Block business logic from executing during design time: If you have business logic executing in the control constructor, or in static properties or static constructors, the import operation will not be successful. The control constructor is invoked when the control is instantiated. Therefore, do not include business logic within the control or static constructors, or in static properties.

- If you cannot remove business logic from a control or static constructor, or a static property, make sure the property is not invoked or that the property does not execute while in design mode.
- To block execution of logic within a constructor while in design time, add the following line to the constructor:

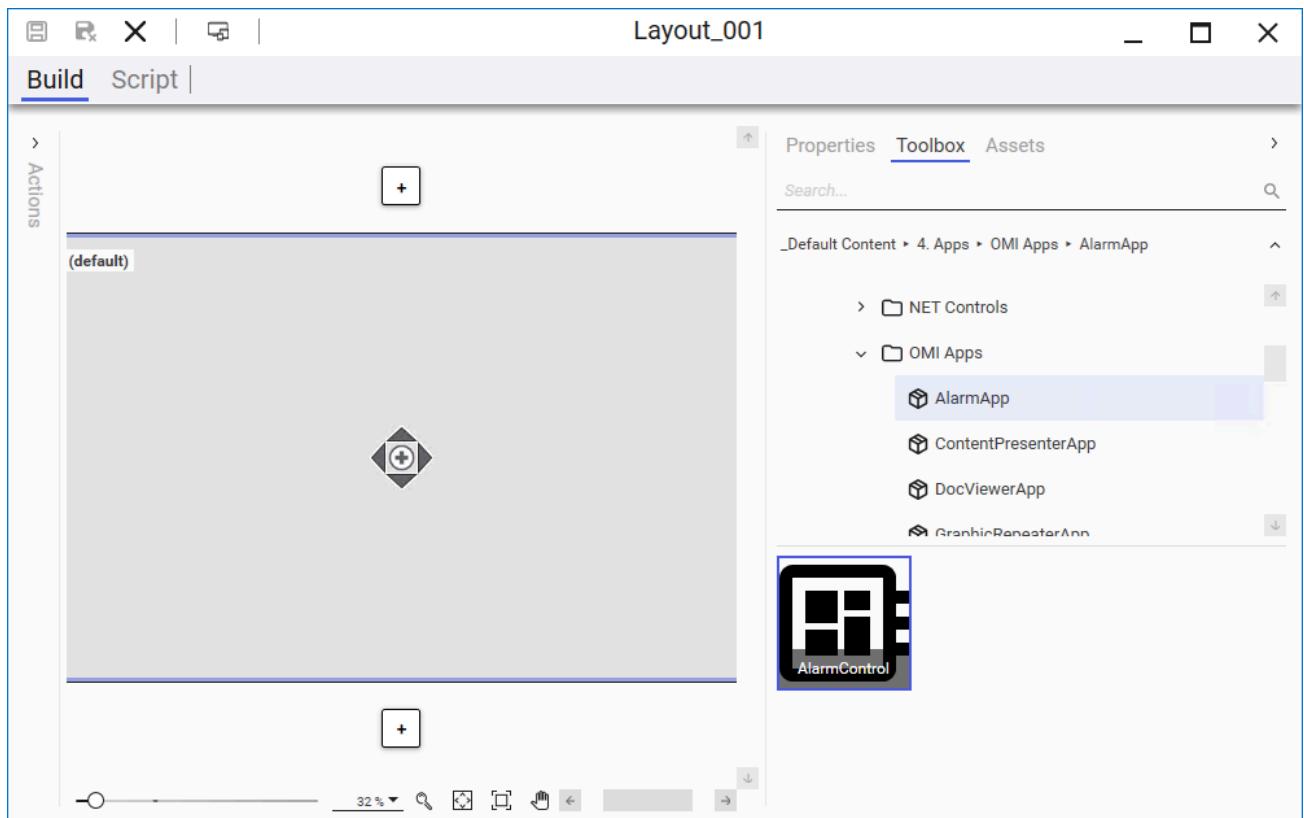
```
!DesignerProperties.GetsInDesignMode
```

Add a control from an AVEVA app to a ViewApp

Controls from the imported App appear in the **Toolbox** tab of the **Layout** or **ViewApp** editors.

To assign a control from an App to a ViewApp

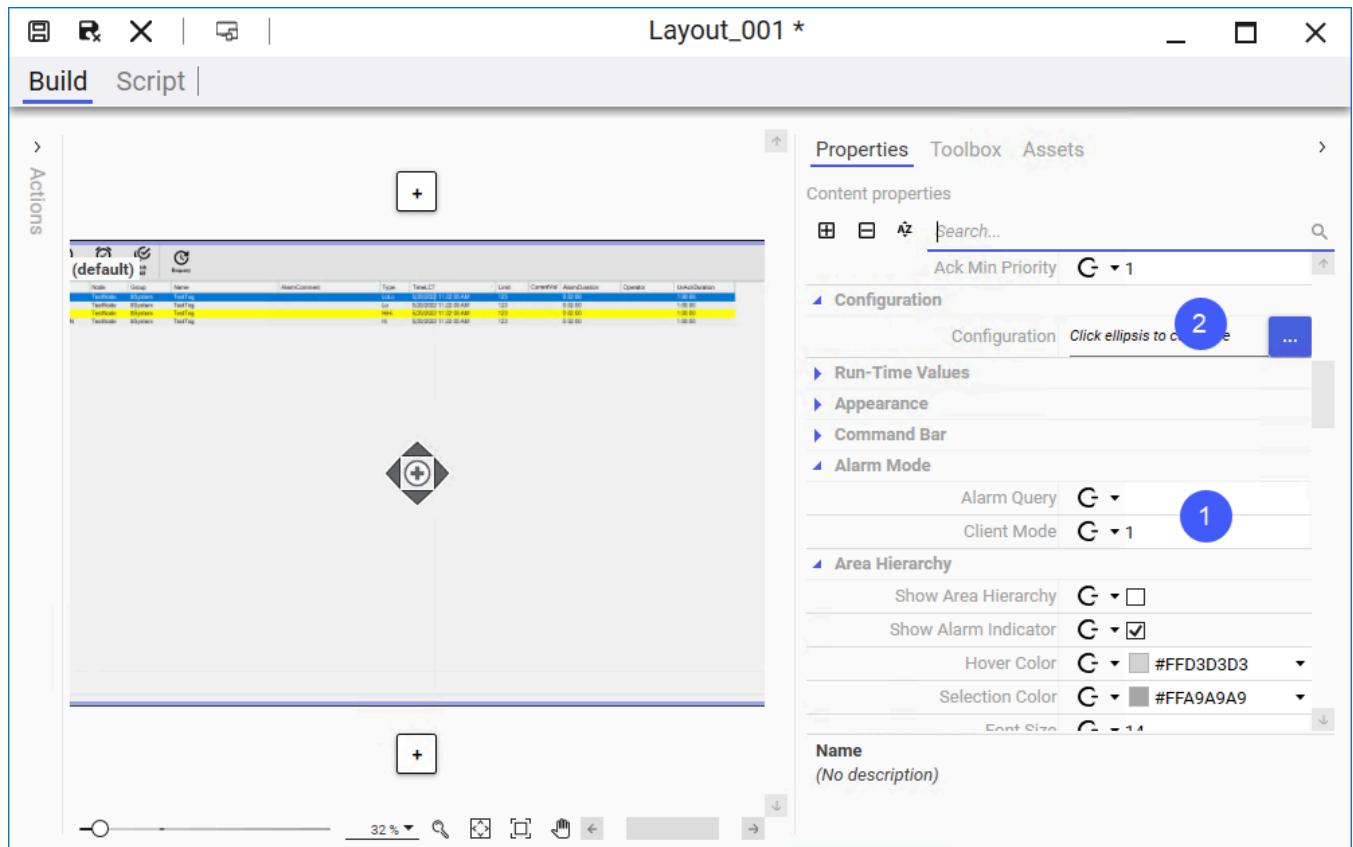
1. Open a layout in the **Layout** Editor and select the **Build** tab at the left of the editor.
2. Select the **Toolbox** tab on the right side of the editor and select a folder that contains the AVEVA Apps you want to add.
3. Select the AVEVA App to be added. The controls available in the App are shown.
4. Select a control and drag it onto a pane.
5. To view or override the control's public properties, select the **Properties** grid. Enter overrides as needed.
6. Save the layout. You can then use the layout that contains the embedded control in a ViewApp.



Configurable properties in AVEVA apps

AVEVA apps can contain two types of configurable properties:

- ① — Dependency properties
- ② — CLR properties (.NET framework properties)



Configurable Property Data Types

- System.Boolean
- System.Byte
- System.Char
- System.Decimal
- System.Double
- System.Int16
- System.Int32
- System.Int64
- System.SByte
- System.Single
- System.String
- System.UInt16
- System.UInt32
- System.UInt64
- System.DateTime
- System.Drawing.Color
- System.Windows.Media.Color

- System.Windows.Media.Brush
- Any property that can be converted to or from string

CLR properties can only be set at configuration (design) time, while dependency properties can be bound to any attribute and can be set at runtime. If a dependency property is changed at runtime, the new value is propagated to the attribute. Dependency properties are data-bindable at design time, and you can set the binding direction to in (read), out (write), or both. To change the binding direction, click on the arrow to toggle the binding direction.

For dependency properties, you must specify the attribute to which you are binding the property, as well as the direction of the binding (in (read only), out (write only) or both. The attribute naming convention is the same you use, for example, when configuring an animation in the Graphic Editor. Note that the ViewApp Editor does not provide syntax validation for attribute names, but the Logger will list any configuration errors.

Binding is only supported for the following basic data types.

DataBindable Properties

- System.Boolean
- System.Byte
- System.Char
- System.Decimal
- System.Double
- System.Int16
- System.Int32
- System.Int64
- System.SByte
- System.Single
- System.String
- System.UInt16
- System.UInt32
- System.UInt64

Any data type can be converted to and from the System.String type property.

AVEVA app development guidelines

The following types of properties are available for configuring WPF-based displays:

- Public writable CLR properties on the display that are not hidden using the browsable attribute
- Dependency Properties
- Nested Properties which satisfy the criteria of configurable properties
- Enum and Flag properties
- Any property that can be converted to or from string

The following types of properties are NOT available for configuring WPF-displays:

- Any property defined by the System.Windows.UIElement class
- Any property defined by the System.Windows.FrameworkElement class
 - The Tooltip property is an exception to this and can be configured
- The following properties defined by the System.Windows.Controls.Control class
 - IsTabStop
 - TabIndex
 - Template
- Any property defined by the System.Windows.Controls.ContentControl class

Optional Filter File: AppManifest.xml

You can include an optional file called **AppManifest.xml** in your imported AVEVA App. See the *AVEVA OMI SDK Help*, located in the **AVEVA Documentation** folder, for additional information about AppManifest.xml. This file will allow you to filter properties when you are configuring the app. The basic file structure is:

- DLL name
- Control name
 - Property

The following is a sample AppManifest.xml file, from the Hamburger App. The files specifies controls and properties that are exposed to System Platform IDE users as they configure a layout or ViewApp.

```
<?xml version="1.0"?>
<AppManifest AppVersion="1.0">
    <Filters>
        <!--List all the Controls that need to be exposed to the System Platform IDE
        user to utilize them in a View Application.
        Only the controls listed here will be available for the IDE users to place
        them on panes within a Layout or a ViewApp.-->
        <Control AssemblyName="AVEVA.Apps.HamburgerApp"
ControlFullName="AVEVA.Apps.HamburgerApp.HamburgerButton">
            <!--List the public properties on this control that need to be exposed
            when user is configuring this control in a Layout or
            a ViewApp. The Property Editor within the Layout Editor and ViewApp
            Editor includes these properties when this control is
            selected and allows the user configure new default values.-->
            <Property Name="HamburgerColor" />
            <Property Name="SlideInPanePosition" />
            <Property Name="Background" />
        </Control>
    </Filters>
</AppManifest>
```

AVEVA app limitations and restrictions

- Events and scripting are not supported
- Only basic datatype properties configuration

- Apps cannot be imported from a network share location
- Collections and arrays are not supported
- Upgrading Apps is not supported
- Nested properties can be viewed and configured, but they cannot be configured with data binding
- To host the .NET controls, use the WPF element **WindowsFormHost** to wrap the control into WPF.

```
<UserControl x:Class="AVEVA.Visualization.TrendApp.TrendControl"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:d="http://schemas.microsoft.com/expressionblend/2008"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="300">
    <Grid>
        <WindowsFormsHost x:Name="FormsHost" />
    </Grid>
</UserControl>
```

Expose underlying controls of a custom app in layout scripting

If you have created a custom app that contains an underlying .NET Winforms or Windows Presentation Foundation (WPF) control, you can show your control's properties, methods, and events in the Layout or ViewApp editors.

You must do the following:

- Include an `ExpandableObjectConverter` attribute in your control code
`[TypeConverter(typeof(ExpandableObjectConverter))]`
- Expose the control's methods, properties, and events using a `public` declaration similar to the following example code snippet.

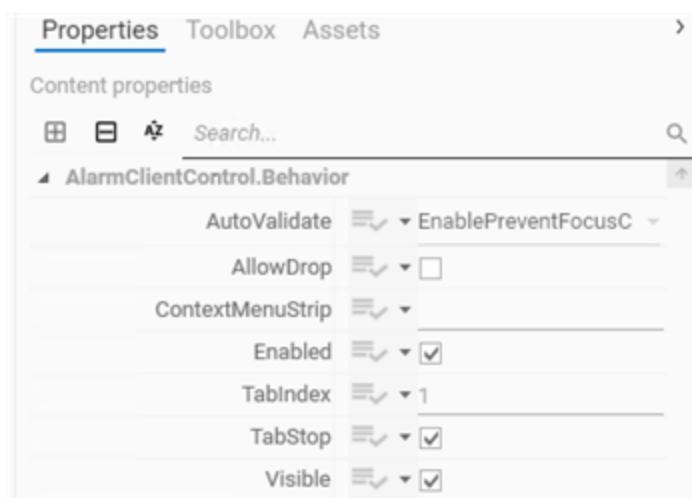
Example

```
[TypeConverter(typeof(ExpandableObjectConverter))]
public AlarmAck AlarmAckControl
{
    get => this.AlarmAckControl;
    set => this.AlarmAckControl = value;
}
```

- Add this public property in the app's manifest file if you have one to enable control properties and events from the **Properties** grid.

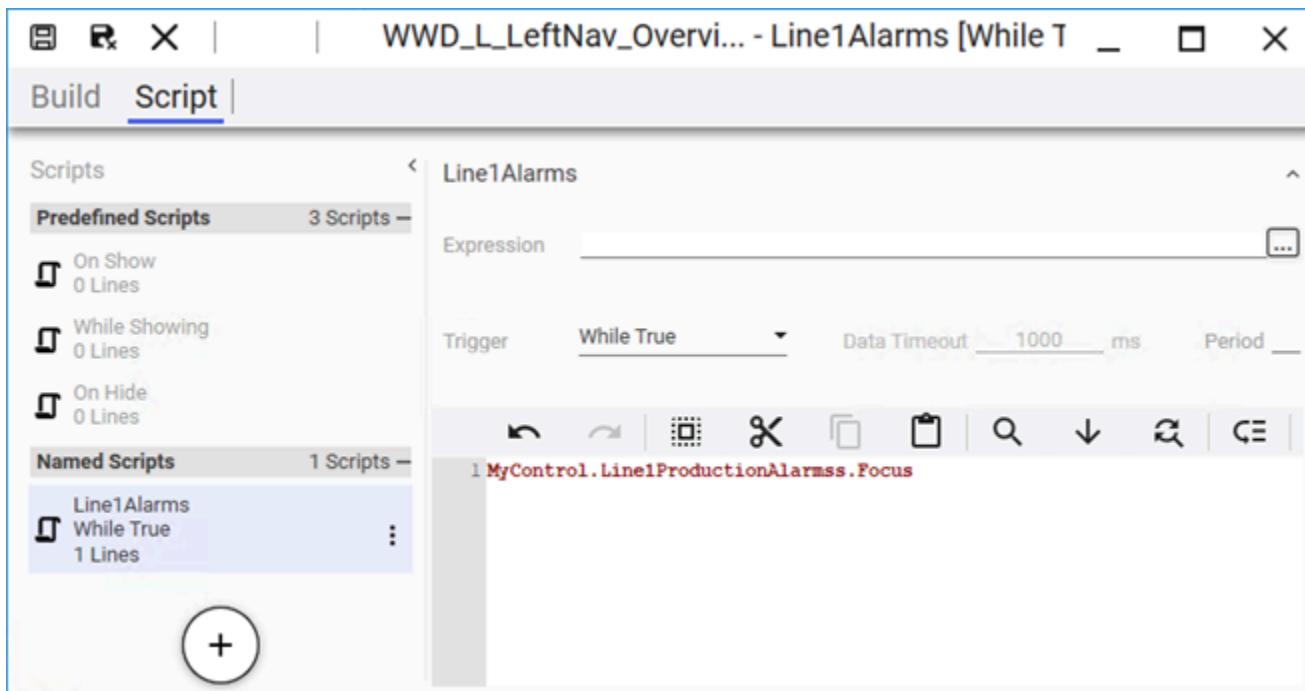
After making the required code changes to your custom app, import the app into the Galaxy.

Place your app onto a layout pane and select **Properties** to show your app's properties. The following example shows the properties of the AlarmApp. Notice the control name (`AlarmClientControl`) appended to the property group as a prefix.

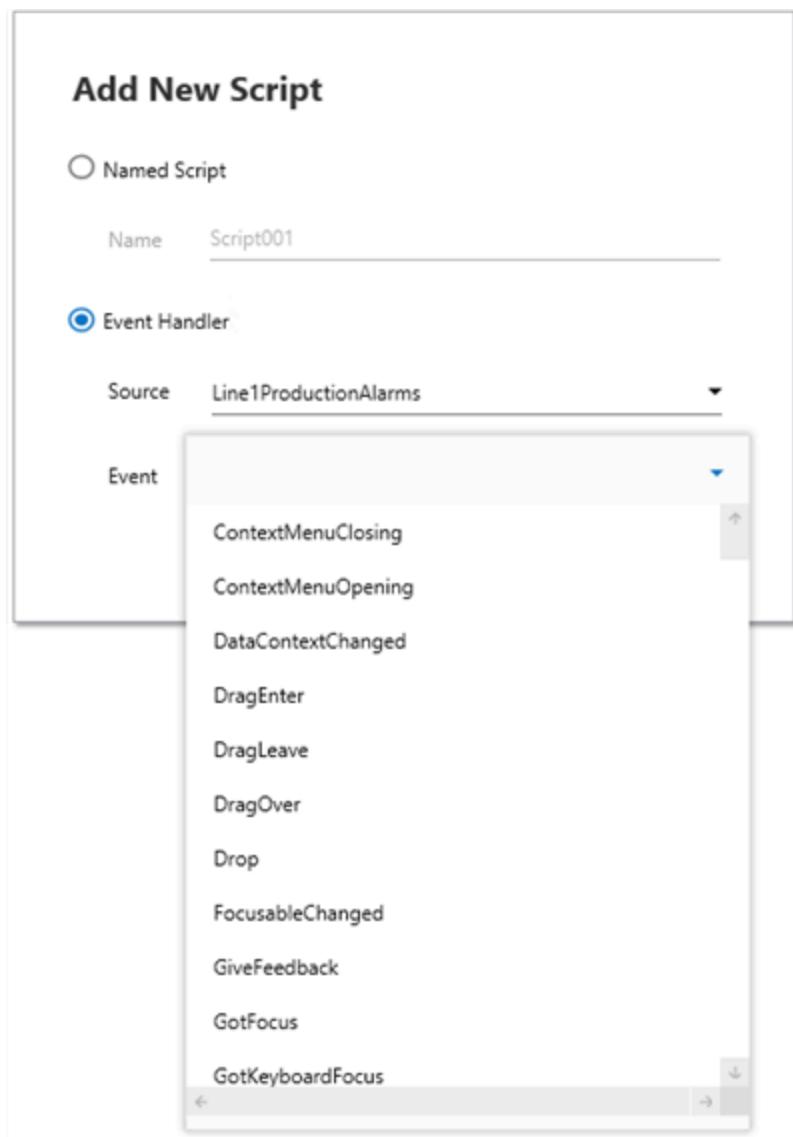


At the bottom of the Properties grid, the **Available Events** property shows a list of all events of the control when you select the data entry field.

When you open the Layout script editor, and create a named script, the editor's Autocomplete feature presents a selectable list of methods and properties as you complete a statement. Notice the MyContent namespace, the instance name of the control, and the custom control's exposed .NET Focus method in the following example.



You can also create event handler scripts that incorporate exposed events from a custom control that is selected as the source.



About industrial widgets

AVEVA widgets are roughly the HTML 5-based equivalents of AVEVA OMI Apps. However, there are some widgets for which there is no equivalent OMI App, and widgets which do have matching OMI Apps may not have all the functionality of the OMI App.

The Visualization folder includes a set of default widgets, which you can place in a ViewApp by dragging and dropping them onto a pane in the Layout or ViewApp editors. The default widgets are listed below. For complete information about the default widgets, click any of the links below to go directly to information about a specific widget.

[Carousel widget](#)

[Map_App widget](#)

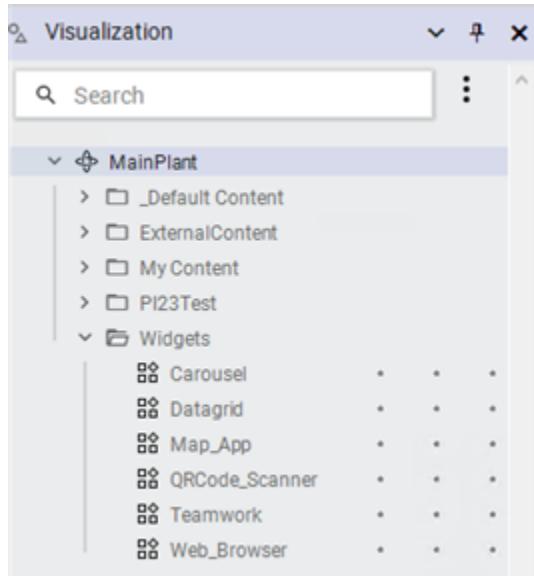
[DataGrid widget](#)

[QRCode_Scanner widget](#)

[Teamwork widget](#)

[Web_Browser widget](#)

The following figure shows the Visualization folder path to the default widgets in in the System Platform IDE.



Web widgets

AVEVA OMI apps were built originally using Microsoft WPF technologies. WPF and WinForms cannot be viewed in Web Browsers. To address this limitation, corresponding HTML5-based web widgets have been developed for most WPF based Apps. When a user tries to view an OMI ViewApp that incorporates WPF based apps in the web browser, the corresponding web widget is used in the background. These widgets eliminate the need to re-engineer applications.

Note: Not all WPF properties are supported as widget HTML5 properties.

An HTML5-based widget requires web browser control to be hosted in an on-premise AVEVA OMI ViewApp. From performance point of view, this will be slower to load when compared to a WPF app, but users can continue to use existing OMI Apps.

You can configure a web widget for a web-enabled application by two methods based on the AVEVA product you are using.

- [Configure a widget for a web-enabled AVEVA OMI ViewApp](#)

Placing a web-enabled widget directly on a layout pane can be done only for AVEVA OMI ViewApps

- [Configure a widget for other web-enabled AVEVA products](#)

The following list shows other AVEVA products that can support web-enabled widgets by embedding them into Industrial Graphics:

- AVEVA OMI

Note: AVEVA OMI ViewApps can support widgets embedded within Industrial Graphics. However, if you are using an on-premise ViewApp, you are advised to continue using AVEVA OMI apps instead.

- AVEVA InTouch HMI

- AVEVA Citect SCADA
- AVEVA Edge
- AVEVA OASyS SCADA
- AVEVA Insight

Follow the hyperlinks shown above for descriptions of the workflows to embed a web widget in an AVEVA application.

Configure a widget for a web-enabled AVEVA OMI ViewApp

Each widget incorporated in an AVEVA OMI web-enabled ViewApp must be individually configured to assign values to the widget's properties after being placed on a layout pane. For more information about widget properties, see [Configurable properties in widgets](#).

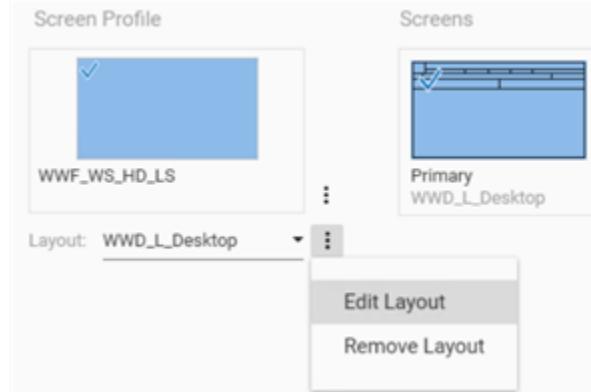
Note: AVEVA OMI widgets are fully supported for web-enabled ViewApps. However, if you are using an on-premise ViewApp, you are advised to continue using AVEVA OMI apps instead.

To configure a widget for a web-enabled AVEVA OMI ViewApp

1. Open the ViewApp you intend to run on the web with one or more widgets.

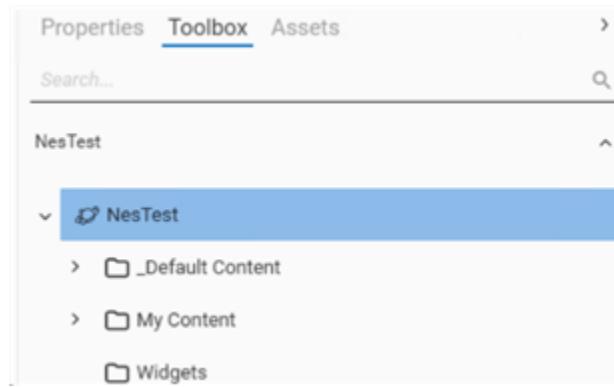
The ViewApp editor opens to show the layout of the open ViewApp.

2. Select **Edit Layout** from the drop-down list of the **Layout** field.



The Layout editor opens to show the panes of the layout assigned to the ViewApp.

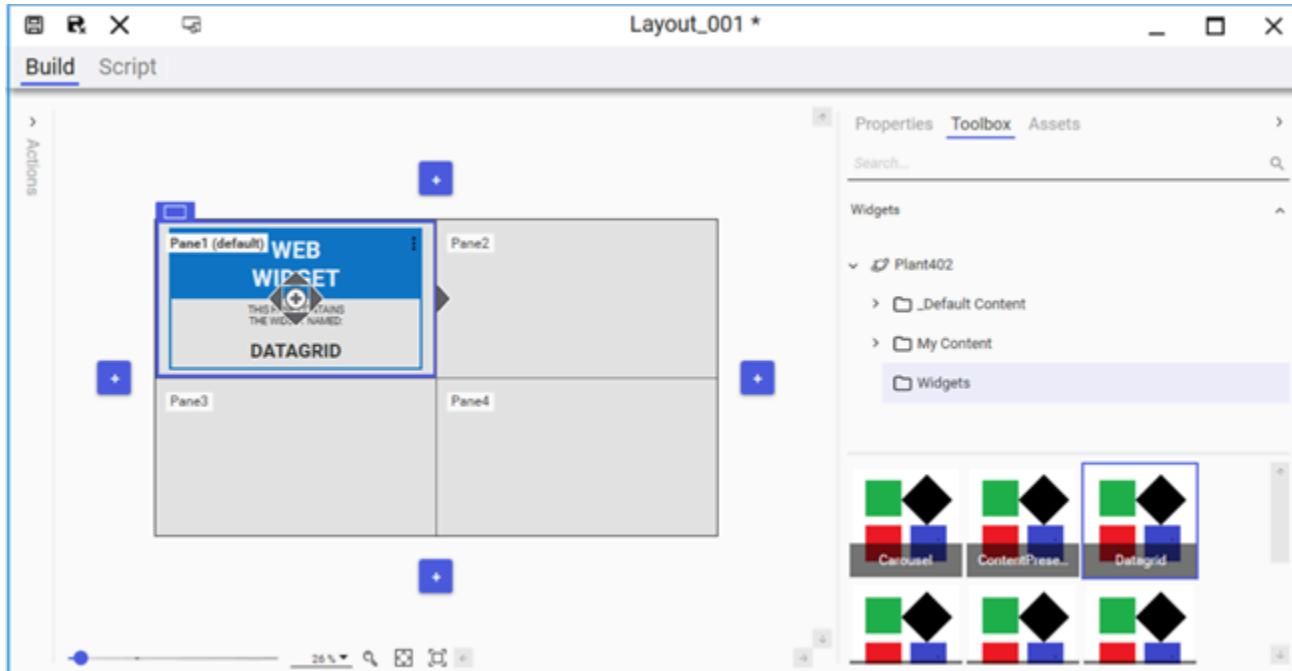
3. Select the **Toolbox** tab to show the **Widgets** folder.



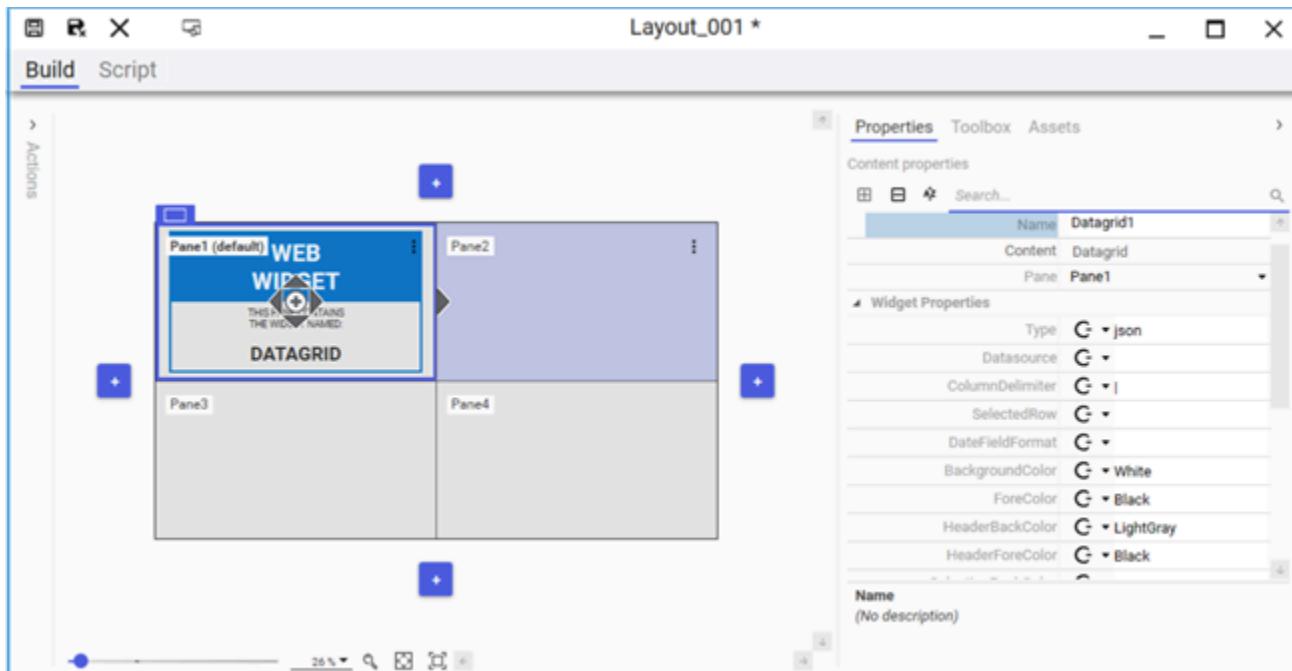
4. Open the Widgets folder to show the widgets within it.

Widget thumbnails appear near the bottom of the **Toolbox** area of the Layout editor.

5. Select the pane within the layout that you want to place the widget.
6. Select the widget that you want to place in the pane and use a drag and drop mouse movement to place the widget on the pane.



7. Select the widget on the pane and then select the **Properties** tab to show the properties of the selected widget.



8. Assign values to the widget's properties.

Note: Widget color properties must be specified by the RGBA format. This is different from OMI App that specify color using the ARGB format.

See Widget Properties for a description of each widget property.

9. Repeat this procedure for each widget to be placed in a web-enabled ViewApp.

Configure a widget for other web-enabled AVEVA products

The AVEVA product list shown in Apps and HTML5 Widgets support web-enabled widgets by embedding them into Industrial Graphics. The following procedure describes the major steps of a workflow to embed a web-enabled widget within an Industrial Graphic.

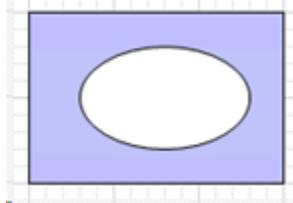
To embed a widget in an Industrial Graphic

1. Open the Industrial Graphic Editor with your AVEVA product.
2. Create a new graphic and open it on the Industrial Graphic Editor canvas.
3. Select the **Toolbox** tab and select the **Widgets** folder to show the set of widgets.
Widget thumbnails appear near the bottom of the **Toolbox** area.
4. Drag and drop the widget you want to embed within the graphic you created.
5. With the embedded widget selected, view the **Properties** area to show the list of widget properties.
6. Configure the widget's properties and save your work.

A Limitation of Industrial Graphics in a Web-Enabled ViewApp or Application

A limitation of Industrial Graphics in a web-enabled ViewApp or application is the inability to invoke an animation from a graphic embedded within another graphic.

In the following example, the ellipse is embedded within the rectangle and the rectangle is lower in the z-order. The rectangle has interactive animation in which the user performs some type of action like push a button to start the animation.

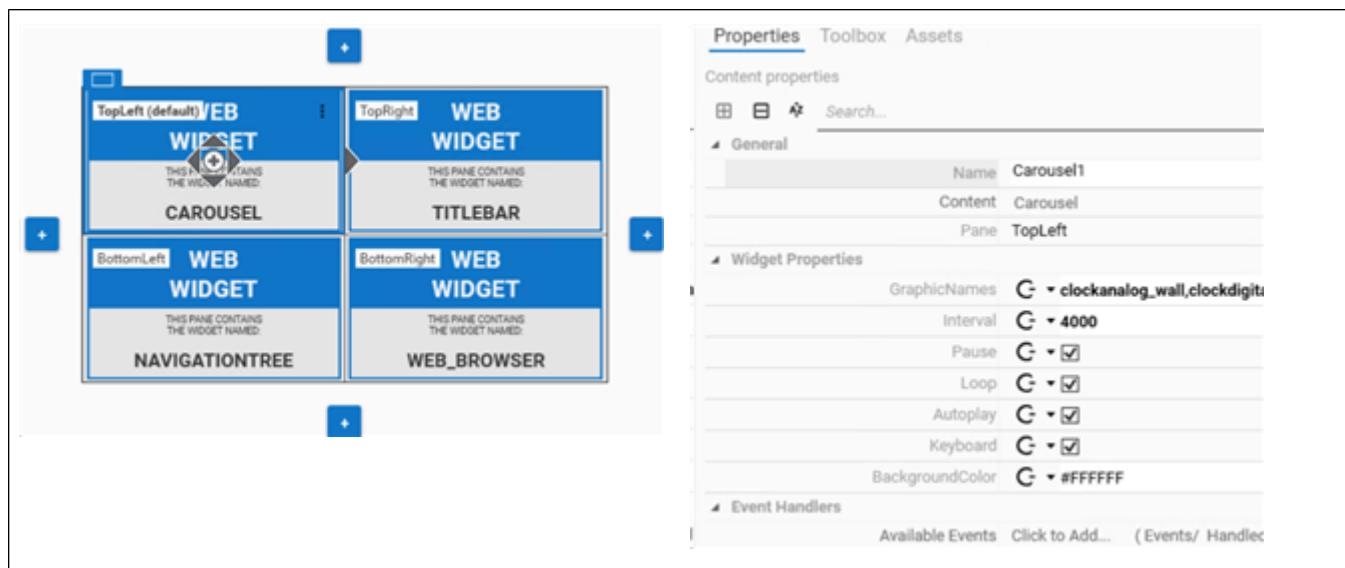


If the ellipse has been assigned a Tooltip animation, clicking on the ellipse in a web-enabled AVEVA OMI ViewApp does not invoke interactive animation configured on the underlying rectangle.

An interim solution for this issue is to configure Tooltip animation directly on the rectangle element in this example.

Configurable properties in widgets

Widget properties can be shown in the Layout or ViewApps editors by placing a widget on a layout pane. After selecting the widget, its properties appear in the **Properties** area of the Layout or ViewApp editors. For more information about configuring a widget workflow, see [General Workflow to Configure a Widget](#).



Widget properties

The topics in this section describe the properties you can set in each of the available widgets. These widgets are included in the standard System Platform installation.

Carousel widget

A Carousel widget shows a set of visual items within a ViewApp pane. The carousel performs like a photo carousel to show the next item in the set for a specified interval. The Carousel widget can be used to display dashboards, alerts, or alarms that are updated continuously.

Properties

You configure specific properties of the Carousel widget from the Layout or ViewApp editors.

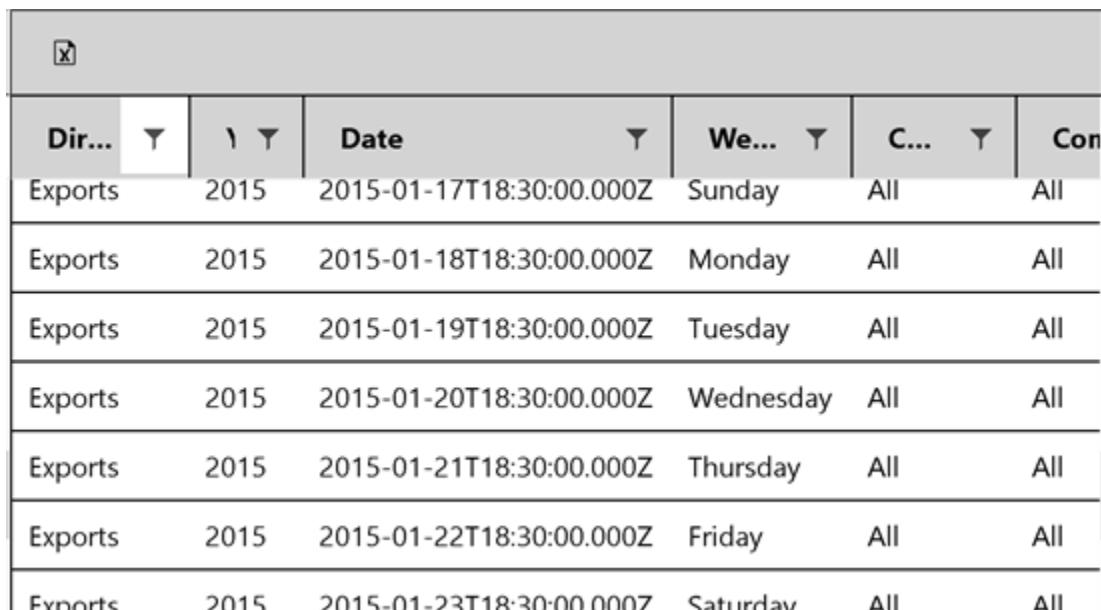
Name	Description	Default
GraphicNames	A comma separated list of graphics the carousel displays in runtime.	Empty
Interval	The display period in milliseconds the Carousel widget shows each specified graphic.	5000
Pause	If the Pause property is set to true, the carousel pauses the cycling of the graphics when it detects the mouse hovering or a touch down event. The carousel resumes cycling when the mouse is moved away.	True

Name	Description	Default
Loop	If the Loop property is set to true, the carousel cycles through the set of graphics continuously. When Loop is set to false, the carousel stops after a single cycle.	True
Autoplay	If the Autoplay property is set to true, the Carousel widget automatically starts when the ViewApp starts running. If Autoplay is set to false, the user must select the next item to start the carousel.	True
Keyboard	If the Keyboard property is set to true, the carousel responds to keyboard inputs.	True
BackgroundColor	Sets a background color for the widget. Specify the color value in RGB, HTML Code (#FF0000) or a valid HTML color name.	White

The carousel widget is based on the Bootstrap 4.0 Carousel component. For more information on bootstrap, go here: <https://getbootstrap.com/docs/4.0/components/carousel/>

DataGrid widget

The DataGrid widget shows tabular data within a pane of a AVEVA OMI ViewApp during runtime. Data shown in the widget can be filtered by the user. The widget also includes a set of visual properties to customize the appearance of the widget and its displayed tabular data.



Dir...	Y	1 Y	Date	Y	We...	Y	C...	Y	Con
Exports	2015	2015-01-17T18:30:00.000Z	Sunday	All	All	All	All	All	All
Exports	2015	2015-01-18T18:30:00.000Z	Monday	All	All	All	All	All	All
Exports	2015	2015-01-19T18:30:00.000Z	Tuesday	All	All	All	All	All	All
Exports	2015	2015-01-20T18:30:00.000Z	Wednesday	All	All	All	All	All	All
Exports	2015	2015-01-21T18:30:00.000Z	Thursday	All	All	All	All	All	All
Exports	2015	2015-01-22T18:30:00.000Z	Friday	All	All	All	All	All	All
Exports	2015	2015-01-23T18:30:00.000Z	Saturday	All	All	All	All	All	All

Filter data during runtime

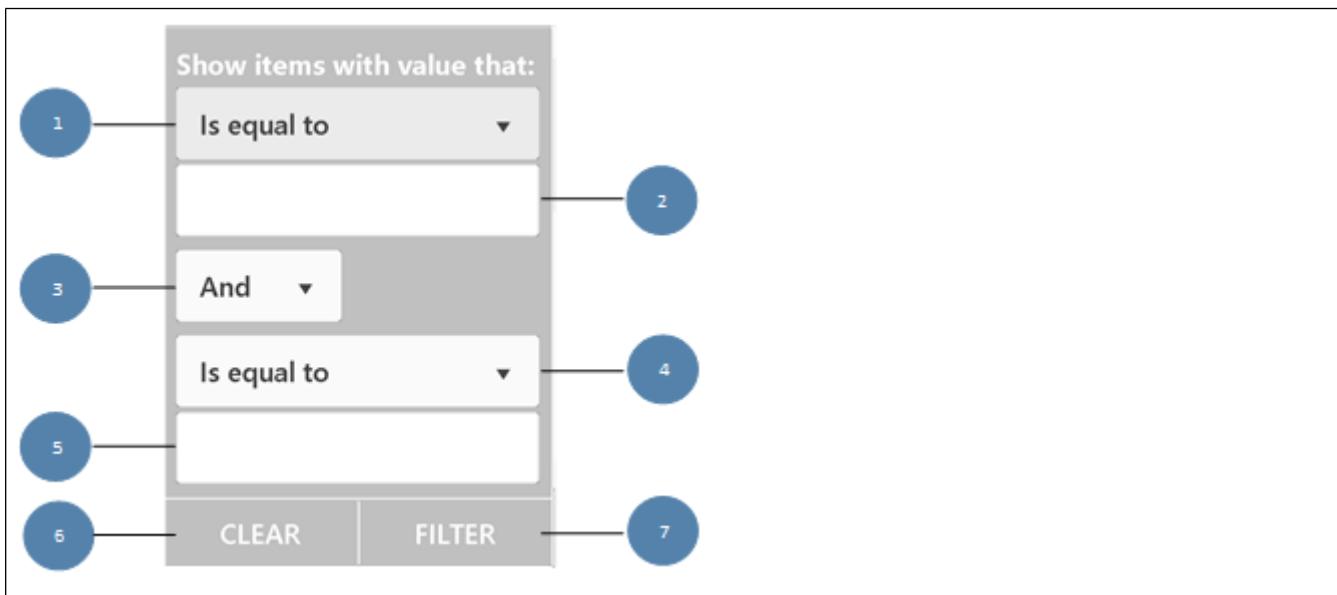
Each column of data shown in the grid can be filtered. The Data Grid widget only shows the data from the row whose column item matches the filter criteria when the filter is applied.

The following example shows tabular data before and after applying a filter to the **Week** column to filter the data to show only Thursday.

Dir...	1	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...	1	Date	Week...	C...
Exports	2015	2014-12-31T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-01T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-02T18:30:00.000Z	Thursday	All
Exports	2015	2015-01-03T18:30:00.000Z	Thursday	All

Select the filter icon within each column header to display a drop-down pane to specify filter criteria.



1	Drop-down list of filter operators. The selected operator performs a comparison to the value entered in the paired data entry field (2). If column data matches the filter statement, the grid shows the data for that row. Unmatched data is removed and not shown in the grid. The Data Grid widget supports the following filter operators: <ul style="list-style-type: none">• Is equal to• Is not equal to• Starts with• Contains• Does not contain• Ends with• Is null• Is not null• Is empty• Is not empty• Has no value• Has value
2	Data entry field to type a filter value that is evaluated by the filter operator selected in (1).
3	Filters logic to evaluate filter statements 1 and 2 with Boolean operators (AND, OR). If filter statement 2 is blank, no filters logic is applied.
4	Same as 1. If the paired data entry field (5) is blank, the filter operator is not evaluated.
5	Same as 2. Can be left blank.
6	Select CLEAR to remove filter criteria and show all data on the grid.
7	Select FILTER to apply the filter criteria.

Properties

The following table lists the properties of the DataGrid widget. All properties are exposed as dynamic runtime

properties in which they can be bound to a reference to a ViewApp namespace as an:

- Attribute
- Object attribute
- MyContent attribute
- Compound expression

Data Grid Widget Properties	Descriptions
Type	Gets or sets the format of source data strings. <ul style="list-style-type: none"> • csv • json
AllowExport	Allows the export of the data to a file. The default value is true.
Datasource	Gets or sets the string of JSON or CSV data shown by the Data Grid widget.
ColumnDelimiter	Gets or sets the delimiter used to separate field values in the selected grid row. <ul style="list-style-type: none"> • Comma for csv strings • Vertical pipe () for json strings
SelectedRow	Gets or sets the string of the selected row field values.
DateFieldFormat	Gets or sets the date format of a date. Supports all date format strings.
BackgroundColor	Background color of the widget.
ExportFileName	If AllowExport = True, specifies the name of the file exported.
FilterMode	Gets or sets the filter mode of the grid; either default or checklist.
ForeColor	Gets or sets the text color of the widget.
HeaderBackColor	Background color of the tabular header in which column names appear.
HeaderForeColor	Text color of the header column names.
SelectionBackColor	Background color of a selected row within the grid.
SelectionForeColor	Text color of the selected row within the grid.

RowHoverBackColor	Background color of the row within the grid in which the mouse is hovering.
RowHoverForeColor	Text color of the row within the grid in which the mouse is hovering.
Font Family	Name of the font family used by the widget to show text.
FontSize	Gets or sets the font size in pixels of text shown in the Data Grid widget.
PageSize	Gets or sets the number of records to display at a time.

Configure the Data Grid widget

Preparing the Data Grid widget to run in an AVEVA OMI ViewApp includes several configuration workflows.

- Assign values to Data Grid widget properties
- Prepare source data to be shown in the Data Grid widget.

The following sections of this chapter describe each configuration workflow.

Configure Data Grid widget properties

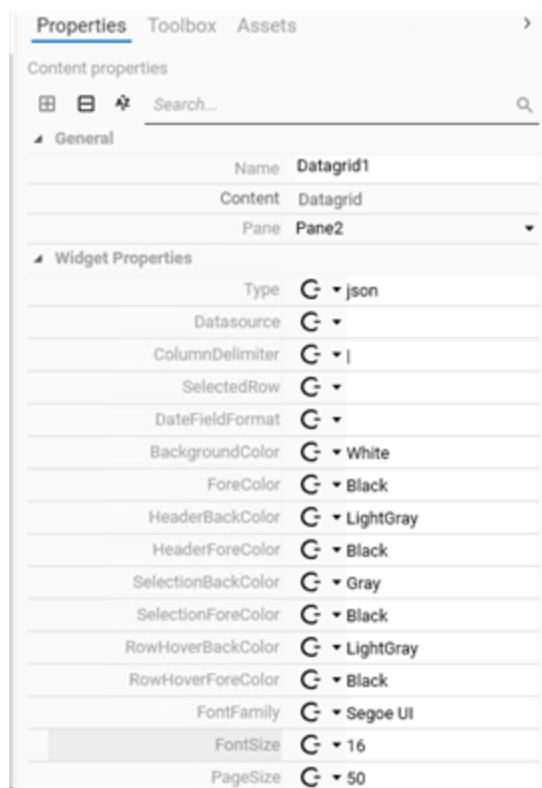
Data Grid widget properties can be shown in the Layout or ViewApp editors by placing the widget on a layout pane. After selecting the widget on a pane, its properties appear on the **Properties** page and can be assigned values.

To configure Data Grid Widget properties

1. Open the Layout or ViewApp Editor and show the items listed in the System Platform IDE **Toolbox** tab.
2. Open the folder in the **Toolbox** where the Data Grid Widget was placed when it was imported.
By default, the Data Grid widget is placed in the **Widgets** folder.
3. Select the Data Grid Widget thumbnail.
4. Drag and drop the Data Grid Widget thumbnail onto a layout pane.
5. Select the Data Grid Widget thumbnail on the pane and select the **Properties** tab.

The **Properties** tab shows the properties of the Data Grid widget. The following screen capture shows the default values assigned to the widget.

For more information about Data Grid widget properties, see [Data Grid properties](#).



6. Assign values to properties and save your changes.

Data Grid properties

The following table lists the properties of the Data Grid widget. All properties are exposed as dynamic runtime properties in which they can be bound to a reference to a ViewApp namespace as an:

- Attribute
- Object attribute
- MyContent attribute
- Compound expression

Data Grid Widget Properties	Descriptions
Type	Gets or sets the format of source data strings. <ul style="list-style-type: none"> • CSV • json
Datasource	Gets or sets the string of JSON or CSV data shown by the Data Grid widget.

Data Grid Widget Properties	Descriptions
ColumnDelimiter	Gets or sets the delimiter used to separate field values in the selected grid row. <ul style="list-style-type: none"> • Comma for csv strings • Vertical pipe () for json strings
SelectedRow	Gets or sets the string of the selected row field values.
DateFieldFormat	Gets or sets the date format of a date. Supports all date format strings.
BackgroundColor	Background color of the widget.
ForeColor	Text color of the widget.
HeaderBackColor	Background color of the tabular header in which column names appear.
HeaderForeColor	Text color of the header column names.
SelectionBackColor	Background color of a selected row within the grid.
SelectionForeColor	Text color of the selected row within the grid.
RowHoverBackColor	Background color of the row within the grid over which the mouse is hovering.
RowHoverForeColor	Text color of the row within the grid over which the mouse is hovering.
Font Family	Name of the font family used by the Data Grid widget to show text.t

Data Grid Widget Properties	Descriptions
FontSize	Font size in pixels of text shown in the Data Grid widget.
Pagesize	Number of grid rows shown the widget display.

Prepare source data for the Data Grid widget

Source data displayed by the Data Grid widget can be formatted as CSV or JSON strings.

For source data formatted as a CSV string, the first line is considered as the header (column names). Subsequent lines are data. Generally for CSV formatted files, a comma is used as the separator between data elements in a row.

```
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
```

For source data formatted as a JSON string, the data is parsed as an array of dictionary objects. The first item in the array is considered as the header (column names) of the Data Grid widget.

As part of parsing the data, the data type of the fields is also resolved. The resolved data types can be any of the following:

- String
- Number
- Date

The parser identifies a string as a Date only if the value is either in the ISO Date (e.g. 2021-08-26T07:44:36.079Z) or in the format specified by the **DateFieldFormat** property of the widget.

Map_App widget

AVEVA OMI and InTouch HMI include a Map_App widget to show a map containing graphics within a running application. During run time, the map provides controls and touch support to enable users to pan to different areas of the map and zoom in or out to show more or less map detail. Graphics placed in a map typically represent business assets located within an area shown by the map. These graphics can include alarming to show the current state of processes at each business location.

Properties

For AVEVA OMI, you configure specific properties of the Map_App widget from the Layout or ViewApp editors. For InTouch HMI, you configure specific properties of the Map_App widget from the Industrial Graphic editor.

Property	Description
ConfigName	Name of the Map_App widget global configuration file.
InitialLatitude	Latitude of the initial center point map position in decimal degrees. Valid latitude values are +/- 0-90.
InitialLongitude	Longitude of the initial center point map position in decimal degrees. Valid longitude values are +/- 0-180.
InitialZoom	Zoom level percentage of a map when initially displayed during run time.
MinZoom	The minimum zoom percentage in which the map can be zoomed out (0-100%) during run time.
MaxZoom	The maximum zoom percentage in which the map can be zoomed in (0-100%) during run time.
MaxBoundsSouth	Latitude of the southern map boundary in decimal degrees (+/- 0-90) to constrain the vertical panning movement of the screen viewport midpoint to the bottom boundary of a map.
MaxBoundsWest	Longitude of the western map boundary in decimal degrees (+/- 0-180) to constrain the horizontal panning movement of the screen viewport midpoint to the left boundary of a map.
MaxBoundsNorth	Latitude of the northern map boundary in decimal degrees (+/- 0-90) to constrain the vertical panning movement of the screen viewport midpoint to the top boundary of a map.
MaxBoundsEast	Longitude of the eastern map boundary in decimal degrees (+/- 0-180) to constrain the horizontal panning movement of the screen viewport midpoint to the right boundary of a map.
Asset	Name of an asset selected from a displayed map.
CurrentLatitude	Current latitude of a selected item shown on a map.
CurrentLongitude	Current longitude of a selected item shown on a map.
CurrentZoom	Current zoom level of a displayed map.
FollowCurrentAsset	Set this property to true to enable the MapApp widget to follow the currently selected asset (context) and automatically pan and zoom the map to display the

Property	Description
	<p>asset's location and associated graphic (the asset should be added into the location tab in map app editor page and asset level should be as "-1").</p> <p>Centers the map to a selected asset when the map is opened. You can use the assets to navigate within a ViewApp. For instance, you can display all the states and show a marker for each one of them. Then, by selecting a state, from a map, you can set the focus of the ViewApp to a separate pane that shows details about the state.</p> <p>The map will zoom to the selected asset's zoom layer plus 1 percent.</p> <p>If no asset is selected or a selected asset is not located on a map, the map shows the initial zoom layer and map center point.</p> <p>If the Asset property is configured, set the FollowCurrentAsset property to false to enable the map to follow the asset configured to load map using the asset's location and associated graphic.</p>
Sources	<p>Map data sources configured in the app's map settings. (All) is the default value, which includes all map data sources specified for the Map app.</p> <p>Note: All must be placed within parentheses (All) as a Sources property value.</p> <p>If you want to restrict a Map app to show data from only some map sources, use a comma delimited string to specify multiple sources.</p> <p>OSM,Bing,TemperatureOverlay</p>
ZoomLayers	<p>Map zoom layers configured for the Map app. (All) is the default value, which includes all zoom layers specified for the Map app.</p> <p>Note: All must be placed within parentheses (All) as a ZoomLayers property value.</p> <p>If you want to restrict a Map app to show data from only some zoom layers, use a comma delimited string to specify the zoom layers by name.</p> <p>country,state,city</p>

QRCode_Scanner widget

The QRCode_Scanner widget connects to a camera to scan for a QR code and returns the resulting string.

Properties

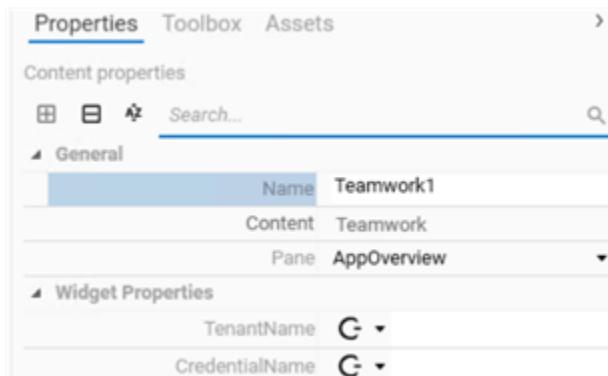
The following table shows a list of properties for the QRCode_Scanner widget.

Property Name	Description	Default Value
QRCode	The resulting string of the scanned QR code. The default value is empty.	Empty
AutoStart	If set to true, the camera starts automatically after detecting a QR code.	True
AutoStop	If set to true, the camera stops after scanning a QR Code.	True
StartStop	If set to true, the camera starts scanning. If set to false, the camera stops scanning.	False
BackgroundColor	Sets the background color of the widget within a layout pane. Specify the color value in RGB, HTML Code (#FF0000) or a valid HTML color name.	Black
DefaultCamera	Sets the default camera on the connected device. True selects the front camera; false selects the back camera.	False

Teamwork widget

The Teamwork widget includes two properties:

- **TenantName (Required)** needs to be set to your assigned tenant name.
- **CredentialName (Optional)** needs to be set to a user credential name or left blank.



Properties	Descriptions	Data Type	Default Value
TenantName	Name of a tenant to show a specific Teamwork page during runtime. A tenant name is unique for each tenant. Consult with your administrator to obtain your tenant name.	String constant or reference	<Empty string>
CredentialName (Optional)	The CredentialName property is not currently supported. To automatically log on to Teamwork, users should use the SSO method.	String constant or reference	<Empty string>

Web_Browser widget

AVEVA OMI includes a full-featured standards-compliant web browser widget. The web browser is based on Chromium Embedded Framework (CEF), the open source version of Google Chrome.

Properties

The following table shows a list of properties for the Web_Browser widget.

Properties	Descriptions
URL	URL of the web site to be shown by the Web_Browser widget.

OMI Web Client

The OMI web client uses cross-platform technology to display ViewApps. Therefore, it supports some features differently than the OMI desktop client, and some features available in the desktop client are not available in this version of the OMI web client. When building a ViewApp that will be used with the OMI web client, and especially when beginning to use a pre-existing ViewApp with the Web Client, you need to keep these limitations and differences in mind. We strongly suggest you test existing ViewApps carefully before using them with the web client in a production environment.

Considerations when creating a ViewApp for the OMI web client

There are inherent limitations when using a browser-based application, compared to using a native desktop

application. For example, the technology that underlies the web client does not allow the web client to open a process on the device on which it is running. If this were allowed, it could potentially be used to circumvent system security.

The considerations and limitations are divided into these types:

- Prerequisites for using the OMI web client
- Security Considerations
- General limitations of the OMI web client
- Limitations when using AVEVA Industrial Graphics
- OMI app limitations
- Widget limitations
- Layout Script limitations
- ViewApp Namespace limitations
- Responsive Layout limitations

Note: Some of the limitations described here may not apply in future releases.

Prerequisites for using the OMI web client

To be able to use the OMI web client to display OMI ViewApps, AVEVA System Platform must be registered with AVEVA Identity Manager. Perform this action in the Configurator either during or after installation.

Before you can make OMI ViewApps available to users, you must define and deploy at least one instance of a WebViewEngine. Assigning ViewApp instances to a WebViewEngine makes them available from the OMI web client. See [Prepare to deploy a ViewApp: assign a ViewApp to an engine](#) for more information.

Required web browsers

The OMI web client is supported with the latest versions of the Google Chrome and Microsoft Edge web browsers.

Preparation needed on remote devices

Any remote device where you will be using the OMI web client to run ViewApps must have a security certificate installed. This security certificate ensures private communication between the remote device and the WebViewEngine, enhancing the security of your data.

A remote device is any device that is not part of the same System Platform installation as the platform that hosts the WebViewEngine. This includes:

- Computers that have System Platform installed but point to a different System Management Server than the one that the platform hosting the WebViewEngine uses. This should occur only if you have two completely separate System Platform installations and want to access the OMI web client on one of those installations from a device hosting the other installation.
- Any device--such as a computer, workstation, tablet, or phone--that does not have System Platform installed.

Depending on procedures and policies at your organization, this security certificate may be installed by the IT

department on all eligible remote devices, or users may need to install the certificate themselves before using the OMI web client for the first time. For instructions on installing the certificate on a Windows device, see [Install a security certificate on Windows](#).

Note: For more information on security certificates, be sure to check the user documentation applicable to your operating system.

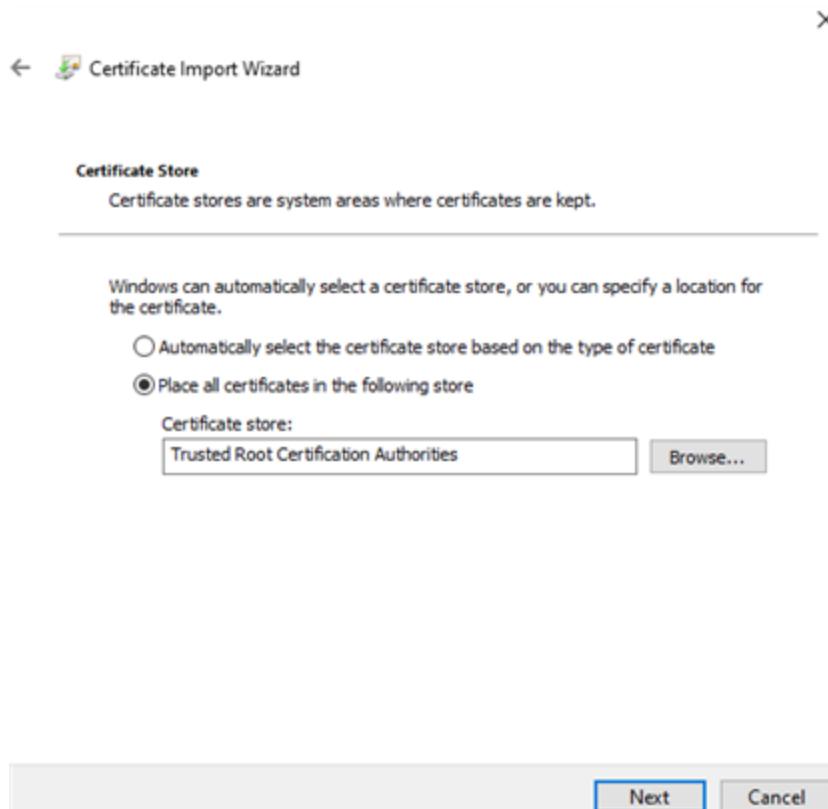
Install a security certificate on Windows

As described in [Prerequisites for using the OMI web client](#), before you can start the web client on any remote device, you must install a security certificate. A remote device is defined as any computer or other device (e.g., smart phone, tablet, etc.) which does not have System Platform installed, or which has System Platform installed but uses a different System Management Server (SMS). The most reliable way to do this is to export the security certificate from the SMS and copy the exported certificate file to the remote device. For instructions on how to export the certificate, see [Export the security certificate](#).

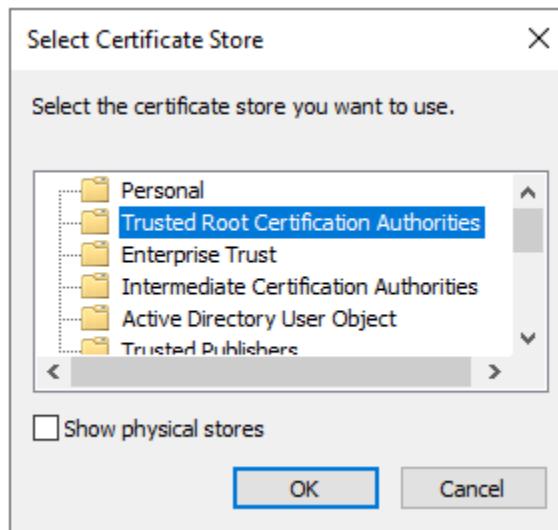
After you have copied the certificate file from the SMS to the remote device, install it to the Trusted Root Certificate Authorities store on the device. The steps below are an example; the precise steps on your version of Windows may be different.

To install security certificate on Windows

1. Open File Explorer and navigate to the certificate file you copied to the device. Double-click the certificate to open it. The **Certificate** dialog box opens.
2. From the **Certificate** dialog box, select **Install certificate**. The **Certificate Import Wizard** opens.
3. For Store Location:
 - Choose **Current User** if you want the certificate to be accessible only to the currently logged-in user.
 - Choose **Local Machine** if you want the certificate to be available to all users on the computer,Select **Next** after setting the Store Location.
4. On the **Certificate Store** page of the wizard, select **Place all certificates in the following store**.



5. Select **Browse**. The **Select Certificate Store** dialog box opens.



6. Select **Trusted Root Certification Authorities**, then select **OK**.
7. Select **Next**. The **Completing the Certificate Import Wizard** dialog box opens.
8. Select **Finish**.
9. After the certificate is imported, close and re-open your browser if it was open. You should now be able to open the OMI web client by going to `https://<web_client_host>/omi`.

HistoricalTrendApp and TrendPen configuration

In some cases, in the OMI web client, the TrendPen graphic element and the HistoricalTrendApp OMI ViewApp cannot retrieve historical data. There are two reasons for this:

- The Historian Server node name and port are not set in the application settings file for the proxy server on the WebViewEngine platform.
- The Historian Server does not use the security certificate from the System Management Server. This means the OMI web client does not have a trust relationship with the Historian Server and will not accept data from it.

Perform Workaround 1 to correct the first issue. For the second issue, perform one of the two procedures in Workaround 2.

Workaround 1: Update the application settings file for the WebViewEngine:

1. On the WebViewEngine node, in a text editor, open the file **C:\Program Files (x86)\Common Files\Archestra\Services\proxyserver\appsettings.json**.
2. Find this section of the file:

```
"history-data": {  
    "HttpRequest": {  
        "Version": "1.1",  
        "VersionPolicy": "RequestVersionExact"  
    },  
    "Destinations": {  
        "destination1": {  
            "Address": "https://localhost:32573/"  
        }  
    }  
}
```

3. Change localhost to the name of the node where your Historian Server is installed. If Historian is not using the default port of 32573, also set the correct port number.
4. Save and close the file.

Workaround 2: To fix the security certificate issue, you have two choices:

- Install each Historian's certificate on every remote node that will use the OMI web client. You do not need to shut down or disable Historian to do this, but each node or remote device must have the requisite Historian certificate installed before it can use the OMI web client.
- Change the Historian Server node to use the certificate of the System Management Server. You need to shut down and disable Historian during the change, but this method requires changes only on the Historian Server node.

Decide on the method you wish to use, and follow the appropriate procedure below.

To install Historian certificates on Windows nodes or remote devices

Note: Depending on your organization's network configuration, your IT department may be able to install the certificate on all devices remotely. Users would then not need to perform this procedure.

1. For each web client node, in a supported web browser, go to https://<historian_node>:32573/. (If Historian is not using the default port of 32573, use the Historian port number for your installation.)
2. Select the **Not secure** message, then, in the dialog box that opens, select **Certificate is not valid**.

3. On the **Certificate Viewer** dialog box, on the **Details** tab, under **Certificate Hierarchy**, select the line that ends in ... **CA**. Select **Export**.
4. Save the Historian certificate to a file. You can place it in any folder and give it any name, but do not change the file type.
5. Install the Historian certificate on your node or device. You can do this either by executing the certificate file you just created, or by using Windows Certificate Manager from the **Start** menu. Once you have started the process, follow the wizard prompts. Be sure to place the certificate in the **Trusted Root Certification Authorities** store.
6. Close, and re-open your browser.

You should now be able to open the OMI web client, start the ViewApp, and see historical data.

Note: For remote devices which use an operating system other than Windows, see the manufacturer's instructions for installing a certificate.

To change the certificate on the Historian Server node

1. On the Historian Server node, use the OCMC to stop and disable Historian:
 - a. Start the OCMC, and expand the **Historian** node in the left pane until you see the Historian Server node name.
 - b. Expand the node name. Right-click **Management Console**, select **All Tasks**, then **Shutdown (and disable) Historian**.
 - c. On the **Shutdown and Disable Historian** dialog box, make sure the node name is correct.
 - d. Select **OK**.
2. From the Windows start menu, select **AVEVA**, then **Configurator**.
3. Under **AVEVA Historian**, select **Server** then **Rest Configuration**.
The Rest Configuration dialog box opens.
4. For Certificate Source, select **Provided by IT**.
5. In **Certificate**, select <hostname> ASB, where <hostname> is the name of the node where your Historian Server is installed.
6. Select **OK**, then select **Configure**.
7. Select **Close** to close the Configurator.
8. In the OCMC, restart Historian:
 - a. Expand the Historian node again.
 - b. Right-click **Management Console**, select **All Tasks**, then select **Enable (allow to run) Historian**.

Security considerations for the OMI web client

The OMI web client allows users to open OMI ViewApps from workstations or other devices (such as a phone or tablet) where no System Platform components are installed. To support secure access from non-System Platform nodes, anonymous access is not allowed and an administrator must configure a System Management Server.

You cannot use OMI web client to open another process on the device on which it is running. For example, you cannot open a native desktop application or a command window on the device.

When the Galaxy uses operating system group-based security, the OMI web client does not support user role and

access level-based navigation. All navigation nodes will always be displayed regardless of access levels or user roles that may be assigned to them. Any user who can open the ViewApp can access all nodes and see all objects tracked by the ViewApp.

You cannot use the OMI web client if you are using Galaxy security as the authentication mode for your Galaxy. You must use one of the other authentication modes: None, OS User-based, OS Group-based, or Authentication Providers.

Note: The OMI web client only supports authentication using AIM, and the authentication happens in the AIM node.

If you have configured your Galaxy to use OS user security, and a user tries to log in as a local OS user (such as localhost/user1) to an OMI web client app running on a remote node, authentication may fail. This is because the localhost will be interpreted as the machine which AIM is running, and the localhost/user1 of the AIM machine may not have permissions set up properly in the Galaxy.

ViewApp namespace attributes related to security do not show the correct values based on the logged-in user.

General OMI web client limitations

The following list summarizes the major general limitations in the current release of the OMI web client.

- Multi-Galaxy configurations are not supported with this version of the web client.
- Secured and verified writes are not supported in this version of the web client.
- You cannot customize the web client URL used to call a specific ViewApp directly.
- You cannot customize the graphics used to indicate ViewApps on the initial OMI web client page.
- Embedded .NET controls will not render correctly on non-Windows devices. Only Windows devices support rendering of .NET controls.
- Security certificates with Public Domain Names are currently not supported by the OMI web client.
- History playback is not supported in this version of the web client.
- The web client cannot show multiple screens at once. If a ViewApp uses multiple screens, you can switch between screens in the web client window. See [ViewApp behavior in the web client](#) for instructions.
- You cannot preview an OMI ViewApp based on how it will look in the web client. The **Preview** button on the ViewApp editor shows a desktop client preview only.
- You cannot generate a URL for a page displayed in the web client and then embed that URL in another product to access the content. This is possible in the InTouch web client, but **not** in the OMI web client.
- By default, you cannot place the web client window in an iFrame, as doing so could make the web client vulnerable to a "clickjacking" attack. If you need to put the web client in an iFrame, see [Placing the Web Client in an iFrame](#).
- In very complex graphics with a large number (thousands) of references, some references may take a very long time to resolve when compared to the OMI desktop client.
- If a user increases the size of a layout during runtime to exceed the defined size of the panes in the layout, the panes will expand to fill the layout, even if they are set to fixed size.
- If you have upgraded the GR node from SP 2020 R2 SP1 to SP 2023 R2 SP1, but have not yet upgraded remote run-time nodes, communication between the GR node and run-time nodes will be lost until the remote nodes are upgraded. Therefore, ViewApp run-time data will not be updated on non-upgraded remote nodes until the remote nodes are upgraded to SP 2023 R2 SP1.

- This version of the OMI web client does not support InTouch-specific references. As a result, the web client will not display run-time data from InTouch applications.

Important! Several graphics in the Visualization folder incorporate elements or settings that are not supported by the OMI web client. Before using a system graphic in a ViewApp that will be used with the web client, make sure that it will display properly on all devices where it may be accessed.

When loading a graphic in the web client, unsupported features are logged as warnings in the Logger. A generated report will include the warning path.

Place the web client in an iFrame

By default, you cannot place the OMI web client in an iFrame because doing so could make it vulnerable to a "clickjacking" attack. If you need to place the web client in an iFrame, you must allowlist the web page that will be hosting the iFrame.

To allow the OMI web client to be placed in an iFrame

1. On the node hosting the WebViewEngine, make a copy of the proxy server **AppSettings.json** file for safekeeping. The default location of this file is **C:\Program Files (x86)\Common Files\Archestra\Services\proxyserver**.

Note: You need elevated privileges to save changes to the **AppSettings.json** file.

2. Open **AppSettings.json** in a text editor.
3. Find this section of the file:

```
// List down all URLs separated by space. The web sites with the urls are allowed
// to embed this OMI web client inside an iframe.
// 'self' is by default allowed. Make sure you postfix the whole value by ";" 
// semicolon.
// i,e. "'self' https://www.google.com https://www.aveva.com;" 
// i,e. "'none';"
// Proxy Service must be restarted after you update frame-ancestors value.
"ContentSecurityPolicy": {
    "frame-ancestors": "'self'"}
```

4. The comments in this section describe how to make the needed changes. Add each URL that should be allowed to host the web client in an iFrame to the **frame-ancestors** list, separating each URL with a space and ending the list with a semi-colon (;). The entire list must be enclosed in quotation marks.
5. Save your change and close the file.

Known issues in the OMI web client

This version of the OMI web client contains the known issues listed below. Some of these issues are also described in other places in this help file.

- When opening the web client home page in Chrome, the user may receive an authentication error. This happens if System Platform was not registered with AVEVA Identity Manager in the Configurator. To correct this:
 - Open the Configurator.
 - Under **AVEVA System Platform**, select **Identity Manager Registration**.

- Select **Configure**.

After this procedure, there should be a green check mark next to **Identity Manager Registration** in the tree view.

Industrial Graphics limitations in the OMI web client

When including AVEVA Industrial Graphics in an OMI ViewApp for use with the OMI web client, there are some limitations and behavioral differences which you should keep in mind.

In general, be sure to test your ViewApp using each browser and type of device (for example, workstation, tablet, and/or phone) that it may be used on. This is especially true when the ViewApp was first developed for the OMI desktop client or when it includes any of the industrial graphics supplied as defaults with System Platform. Some of the supplied graphics use features or settings which are not supported in this version of the OMI web client. Also, some graphic elements may be displayed in the wrong location in the web client.

General limitations when using industrial graphics in the OMI web client include:

- Custom properties with History Summary data type are not supported. All other custom properties are supported.
- Text in a graphic may not wrap properly even when word wrap is set to true.
- Graphics which extend to the very edges of a layout pane may have their edges cut off in the web client.

For information about specific limitations and differences when using industrial graphics, see these topics:

- [Graphic element limitations](#)
- [Animation limitations](#)
- [Embedded widget limitations](#)
- [Script limitations](#)
- [Font limitations](#)

Graphic element limitations

These graphic elements are not supported and will not work in the OMI web client:

- Multi Pens Trend
- Trend Control
- Alarm Client
- Trend Client

Note: While the trend-related graphic elements shown above are not supported, the Trend Pen element and the HistoricalTrendApp OMI ViewApp are supported.

While the Alarm Client graphic element is not supported, the AlarmApp OMI ViewApp is supported.

All other graphic elements are supported, with the limitations described below.

These properties are not supported for any graphic element:

- TabOrder

- TabStop

All other properties are supported.

Gradients and patterns

Because the OMI web client uses different technology to render elements on different platforms, we strongly suggest you check the appearance of gradients and patterns on each type of platform where a ViewApp may be accessed.

A gradient fill may sometimes extend beyond the element it is supposed to fill.

Gradients do not always display correctly when used with arc, pie, or chord-based elements, such as pipe connectors.

Animation limitations

The OMI web client has some limitations which apply to all animations when used with any Industrial Graphic Editor element and other limitations which apply only to a specific animation.

The following list shows the limitations for all supported animations. *Any animation not listed is not supported by the OMI web client.*

All Animations

- Keyboard shortcuts are not supported.

Action Script Animation

- Animations with a While Center Down trigger type do not work the same as in the desktop client. The script does not stop triggering at the proper time based on the mouse position.

Blink Animation

- Blinking may stop after a period of time. This has been observed with the CalendarApp and with the ComboBox element.

Push Button Animation

No limitations.

Slider Animation

No limitations.

ToolTip Animation

- Because the OMI web client uses different technology to render elements on different platforms, we suggest

you check the appearance of tooltip animations on each type of platform where a ViewApp will be accessed.

User Input Animation

- You cannot encrypt string input.
- You cannot define a character to use to hide password input. The default # character is always used.
- You cannot input a multi-line string.

Embedded widget limitations

When embedding a widget into an Industrial Graphic for use with the OMI web client, keep the limitations below in mind. Other than these limitations, widgets embedded in graphics work in the OMI web client as they do in the OMI desktop client.

Map_App

Embedding the Map_App widget in an Industrial Graphic is not supported.

Displaying Industrial Graphics on an embedded widgets

Some widgets, such as the Carousel widget, support displaying Industrial Graphics in the widget. This does not work with the OMI web client.

Script limitations

- The OMI web client uses cross-platform technology to render visuals on non-Windows platform. This causes some limitation in script usage in graphics. If a client script in a graphic uses a .NET namespace that is not compatible with the .Net standard namespace, the script will not function correctly. Where possible, you may need to refactor the script to use the .Net standard namespace for proper script execution. If there is no equivalent member in the standard namespace, the script is not supported in the web client.
- OMI web client does not support the EmbedContent() function.
- The OnHide script for a graphic or layout is not triggered if you close the browser tab (or the entire browser window) where the ViewApp is running. The OnHide script is triggered correctly if you close just the graphic or layout.
- Action scripts configured to trigger while the center mouse button is down do not work. All other trigger types do work.
- SQL scripts and scripts that access the device's file system are not supported. You may be able to work around this by moving the scripts' logic to object scripts and bind visual elements to object attributes.

In addition, the following script methods are not fully functional when used in an OMI web client application. Click the method name to see its limitations.

[HideGraphic\(\)](#)

[ShowContent\(\)](#)

[ShowGraphic\(\)](#)

HideGraphic() limitations

The name of a graphic is used as the hide identity.

ShowContent() limitations

The ShowContent() method does not support the **Name** property.

ShowGraphic() limitations

Scripts that include the ShowGraphic() method have the following limitations when used in OMI ViewApps accessed through the OMI web client:

- Do not support Identity (only support is ShowSymbol using the graphic name as the identity).
- Do not support the **Name** property.
- Do not support the **Owning Object** property.
- Do not support the **ResizeWindow** property.
- Do not support the **HasTitleBar** property. Applications always show a title bar.
- Do not support the **HasCloseButton** property. Screens always show a close button.
- Do not support the **WindowTitle** property.
- Do not support **ScalePercentage** property.
- **Center** is the only supported value of the **Position** property.
- Do not support using the same interface element to open several graphics.

Popup limitations

When ShowGraphic() produces a popup window, the following limitations are present:

Relative to Symbol:

- The size of a popup window is always the same as the graphic's original default width and height with scaling at 100 percent.
- If a graphic's original default size is larger than the screen, the popup window appears in full screen mode as well as relative to the desktop.
- The popup window is always placed at the center of the screen.

Relative to Desktop:

The popup window is always shown in full screen mode.

Custom Width and Height:

The width and height of the popup window cannot be customized.

Font limitations

For OMI ViewApps opened in Google Chrome or Microsoft Edge on Windows, local operating system fonts are supported in most cases. However, this font support has these limitations:

- The user must accept the font use permission message from the browser. If the user does not, only one font is provided for each CJK language and Roboto is used for all other languages.
- The actual font must exist for any configured style used, such as bold or italic. For example, if Arial Italic is configured, Arial must exist on the workstation, with a defined Italic style. The OMI web client does not perform software calculations to change the regular font to italic. The characters will be shown in the regular font.
- The configured font must support the language being used. There is no fallback. For example, if you configure Chinese characters to use the Arial font, the OMI web client will not show the correct Chinese characters.

With any other browser or operating system, only one font is provided for each CJK language and Roboto is used for all other languages.

Custom fonts are supported. It is the customer's responsibility to acquire the necessary licenses for any custom font used.

Note: These font limitations apply to all fonts used in a ViewApp, including those in font style animations and those applied to any element of a graphic either directly or through an element style.

Fonts and Language Settings

If you are concerned about the possibility of missing characters or other issues because the font being used in a ViewApp may not support a user's language, you can set the ViewApp to always use the font defined for the current language in the Galaxy configuration. If you set this option, all fonts defined for any part of any ViewApp are ignored and the OMI web client uses only the font defined for the current language on the **Languages** pane.

For more information, see [Set additional properties for a ViewApp](#).

OMI App limitations in the OMI web client

The following OMI Apps that are included in the Galaxy templates provided by AVEVA do not work in an OMI ViewApp used with the OMI web client:

- DocViewerApp
- ImageViewerApp
- InsightApp
- PDFViewerApp
- SpreadsheetViewerApp

Limitations

All other OMI Apps that are included in the Galaxy templates provided by AVEVA are supported, but have these limitations:

- Event scripts are not supported
- Keyboard navigation is not supported.

The OMI Apps listed below have some additional limitations when used with the web client:

- [AlarmApp](#)
- [ContentPresenterApp](#)
- [GraphicRepeaterApp](#)
- [HistoricalTrendApp](#)
- [MapApp](#)
- [TitlebarApp](#)
- [WebBrowserApp](#)

AlarmApp limitations

Only the properties listed below are supported when using the AlarmApp with the OMI web client:

- Follow Current Context
- Only show Alarms on current Asset
- Show Grid
- Configuration: Alarm mode, Colors
- Client Mode supports only mode 1 (Current Alarms)

No other AlarmApp properties are supported.

The AlarmApp cannot query InTouch alarms.

ContentPresenterApp limitations

For the **Size Mode** property,only **Scaled** is supported. **Fixed** mode is not supported with the OMI web client.

When configuring the content of a ContentPresenter instance, we recommend you limit the number of graphics as much as possible. The web client loads all of the graphics into memory, so large numbers of graphics, or very complex graphics, can use a great deal of memory. Be sure to check the performance of ViewApps that use the ContentPresenter before you place them into production for use with the web client.

GraphicRepeaterApp limitations

These GraphicRepeaterApp properties are not supported with the OMI web client.

- Sort By
- Sort Order
- Live Sorting

HistoricalTrendApp and TrendPen limitations

Only these HistoricalTrendApp design-time properties are supported with the OMI web client:

- Follow Current Context
- Pens

- Grid Horizontal
- Grid Vertical
- Grid Visible
- ValueAxisLevel

In addition, during runtime, the user can set:

- Start time
- End time
- Duration
- Mode

No other HistoricalTrendApp properties are supported.

Historian system tags are not supported.

During runtime, be aware of these additional limitations:

- The label for the x-axis will not wrap.
- The pen color is randomly generated. If you refresh the browser page, it may change.

In some cases, in the OMI web client, the TrendPen graphic element and the HistoricalTrendApp OMI ViewApp cannot retrieve historical data. For more information, see [HistoricalTrendApp and TrendPen configuration](#).

MapApp limitations

These MapApp properties are not supported with the OMI web client:

- Browser Stylesheet
- Event Handler Available Events

In addition, the Sources property supports only string constants, not reference values.

Defining a reverse proxy for the MapApp

Any map source that needs a key or credential to be used with the MapApp must use a reverse proxy to connect to the OMI web client. Configuring this for the MapApp requires updates to a settings file on the WinPlatform that hosts the WebViewEngine. You must also be sure to enter the correct information when configuring the MapApp in its layout pane.

Important! Only these map sources require a credential or key: Google, Bing, and sometimes ArcGis, Baidu, XYZ, and WMS. You do not need to perform these steps for other map sources.

The general procedure for configuring any map source to use a reverse proxy is:

1. Enter the necessary information in the source **Properties** fields when configuring the MapApp in its layout pane.
2. On the platform that hosts the WebViewEngine, edit the **C:\Program Files (x86)\Common Files\Archestra\Services\proxyserver\appsettings.json** file:

- a. In the `Routes` section for the map source, find and uncomment the section for your map source.
 - b. In the uncommented section, set the correct `Path`, `PathRemovePrefix`, and `Append` values for the map source.
 - c. In the `Clusters` section for the map source, enter the URL for the map source.
3. Clear the browser cache on the WebViewEngine after making the changes to `appsettings.json`.

The sections below describe the specific changes for each type of map source.

Google Maps

Properties fields: Enter the API key.

Routes section of `appsettings.json`:

- `Path` and `PathRemovePrefix`: Enter 1 for the first Google map. If there is another map configured in the editor, enter 2. For example:

```
"path": "/Map/google/1/{**catch-all}"  
"PathRemovePrefix": "/Map/google/1"
```

- `Append`: Enter your Google API key as the value.

Clusters section of `appsettings.json`: The `Address` value should already be the URL for the Google map. Correct it if it is not.

Bing Maps

Properties fields: Enter the API key.

Routes section of `appsettings.json`:

- `Path` and `PathRemovePrefix`: Enter 1 for the first Bing map. If there is another map configured in the editor, enter 2. For example:

```
"path": "/Map/Bing/1/{**catch-all}"  
"PathRemovePrefix": "/Map/Bing/1"
```

- `Append`: Enter your Bing API key as the value.

Clusters section of `appsettings.json`: The `Address` value should already be the URL for the Bing map. Correct it if it is not.

XYZ Maps

Properties fields: Enter the source URL. If there is no key parameter in the URL, this map does not need a reverse proxy, and you do not have to make any changes to `appsettings.json`. If there is a key parameter, make the changes below.

Routes section of `appsettings.json`:

- `Path` and `PathRemovePrefix`: Enter 1 for the first XYZ map. If there is another map configured in the editor, enter 2. For example:

```
"path": "/Map/XYZ/1/{**catch-all}"  
"PathRemovePrefix": "/Map/XYZ/1"
```

- `Append`: Enter your XYZ API key as the value.

Clusters section of `appsettings.json`: The `Address` value should already be the URL for the XYZ map. Correct it if it

is not.

WMS Maps

Properties fields: Enter the source URL. If the URL does not require a credential, this map does not need a reverse proxy, and you do not have to make any changes to appsettings.json. If it does need a credential, make the changes below.

Routes section of appsettings.json:

- Path and PathRemovePrefix: Enter 1 for the first WMS map. If there is another map configured in the editor, enter 2. For example:

```
"path": "/Map/WMS/1{**catch-all}"  
"PathRemovePrefix": "/Map/WMS/1"
```

- Append: Set the Basic value to the Base64String encoded username:password for the map URL.

Clusters section of appsettings.json: Set the Address value to the URL for your map server.

Baidu Maps

Properties fields: Enter the source URL. If there is no ak parameter in the URL, this map does not need a reverse proxy, and you do not have to make any changes to appsettings.json. If there is an ak parameter, Make the changes below.

Routes section of appsettings.json:

- Path and PathRemovePrefix: Enter 1 for the first Baidu map. If there is another map configured in the editor, enter 2. For example:

```
"path": "/Map/Baidu/1{**catch-all}"  
"PathRemovePrefix": "/Map/Baidu/1"
```

- Append: Enter your Baidu key as the value.

Clusters section of appsettings.json: The Address value should already be the URL for the Baidu map. Correct it if it is not.

ArcGis Maps

Properties fields: Enter the source URL. If the URL does not require a credential, this map does not need a reverse proxy, and you do not have to make any changes to appsettings.json. If it does need a credential, make the changes below.

Routes section of appsettings.json:

- Path and PathRemovePrefix: Enter 1 for the first ArcGis map. If there is another map configured in the editor, enter 2. For example:

```
"path": "/Map/arcgis/1{**catch-all}"  
"PathRemovePrefix": "/Map/arcgis/1"
```

- Append: For ArcGis versions before 10.9, set the Basic value to your ArcGis token. For older versions of ArcGis, set the Append value to Bearer followed by your ArcGis token.

Clusters section of appsettings.json: Set the Address value to the URL for your map server.

TitleBarApp limitations

The Keyboard property is not supported with the OMI web client. You cannot show an on-screen keyboard when using the TitleBarApp in the web client.

WebBrowserApp limitations

When using the OMI web client, you cannot show a web page that has a Content Security Policy configured with frame ancestors unless the web client site is in the list. You also cannot open a page that is configured with X-Frame-Options.

Widget limitations in the OMI web client

When placing a widget into a layout pane for use with the OMI web client, keep the limitation below in mind. Other than this limitation, widgets work in the OMI web client as they do in the OMI desktop client.

Breadcrumb widget

Alarm adorners do not work in the Breadcrumb widget.

Map_App widget

Embedding the Map_App widget in an Industrial Graphic is not supported in the OMI web client.

Displaying Industrial Graphics on an embedded widgets

Some widgets, such as the Carousel widget, support displaying Industrial Graphics in the widget. This does not work with the OMI web client.

Layout script limitations in the OMI web client

Embedded .NET controls will not render correctly in the OMI web client on non-Windows devices. Only Windows devices support rendering of .NET controls.

In addition, scripts used in layouts have all of the same web client limitations as scripts used in graphics. For information on the limitations when using layout scripts in either layouts or graphics for use with the web client, see [Script limitations](#).

ViewApp namespace attribute limitations

Attributes in the default ViewApp namespace have the following limitations when the ViewApp is opened in the OMI web client:

- Attributes which are set to retain their values do not do so. If you close and reopen the ViewApp, the attribute will be reset to its default.

- MyViewApp.Language.CurrentLanguage does not show the language's full name. For example, for American English it shows en (US).
- The following namespace attributes are not supported:
 - MyViewApp.Graphic.ShowHelp
 - Security-related attributes

OMI web client responsive layout limitations

When using a responsive layout in a ViewApp through the OMI web client, the Column Push and Column Pull features are not supported. Do not rely on these features when designing a layout that will be used in a ViewApp through the OMI web client.

For more information about responsive layouts, see [About the responsive layout mode](#).

About use of touch screens in ViewApp

Touch screens enable users to interact with a ViewApp similarly to applications running on touch-enabled laptops, tablets, or smart phones. On a touch screen, users can manage a running ViewApp by:

- Single tap to push a button, start action scripts, set focus within a window, select an item from a list
- Touch and swipe on vertical or horizontal scroll bars to change the viewing area within a pane
- Pan the content of a pane by swiping a screen with a single finger
- Trigger animations configured for graphic elements, ACF-apps, or NET Controls.
- Zoom a screen view in or out using two finger gestures.
- Double tap to restore the display to its original size
- Display slide out panes using three finger swipe gestures.
- Close a windows using a 5 finger scrunch gesture.
- Interact with user controls that do not natively support touch.

ViewApps can be configured with screen locks to prevent accidental screen interactions. When enabled, a screen lock must be deactivated when interacting with a touch-enabled ViewApp.

Build a ViewApp that supports touch gestures during runtime

Building a ViewApp that supports touch gestures during runtime requires configuration of the screen profile and the layouts that are associated with the ViewApp.

- Screen Profile

When editing a screen profile, you can enable touch on a screen and set values to a Touch Lock that restricts touch interactions to ensure the ViewApp runs safely.

For more information about screen profile properties, see [About screen properties](#).

- Layout

Each pane of a layout includes a set of properties. Based on values set to these properties, touch interactions

can be defined at the pane level.

For more information about layout Appearance properties, see [Set pane properties 1](#).

Software Requirements for Touch ViewApps

Touch supports in running ViewApps requires the following system software:

- Client computer operating system: Windows 10 or higher
- Server computer operating system: Windows 2012 R2 or higher.
- Remote Desktop Protocol (RDP) connectivity between supported client and server computers

Recommendations for Touch ViewApps

Follow these recommendations to create content intended for touch interactions.

- The minimum size of on screen controls should be at least 23X23 pixels (13X13 DLUs).
- Commonly used screen controls should be larger at least 40X40 pixels (23X22 DLUs).
- Include a five pixel border around UI elements that are touch enabled to reduce the likelihood of accidentally touching other nearby screen content.
- Use the Hamburger app navigation control to save space for ViewApps designed to run on small screens.

About ViewApp touch interactions

ViewApp touch interactions are summarized in the following tables based upon the type of finger gesture and the target content shown in a running ViewApp. For more detail about touch gestures during runtime, see [About touch interactions supported during runtimes](#).

Content Level Gestures

Content level gestures are single finger gestures applied to a graphic or control in a screen that trigger configured animations or predefined actions.

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Touch down	Touch down		<ul style="list-style-type: none">• Action Scripts• User Input• Push Button• Show / Hide Graphic	<ul style="list-style-type: none">• Starts action scripts• Sets ViewApp focus
Touch move	Touch down & drag		<ul style="list-style-type: none">• Slider control• Pane scroll bar.	<ul style="list-style-type: none">• Moves slider control• Moves panel scroll bar

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Touch up	Touch up		<ul style="list-style-type: none"> Action Scripts Push button 	<ul style="list-style-type: none"> Stops action scripts Releases button
Long Press	Touch, wait and lift		Action Scripts	Functionally equivalent to a mouse right-click
Double tap	Two touches in quick succession		Pane	<ul style="list-style-type: none"> Stops action scripts, Fits display to pane

Pane Level Gestures

Pane level gestures are two finger gestures applied to a pane that change the view of content.

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Pinch to zoom	Spread fingers apart to zoom in. Pinch fingers together to zoom out.		Pane	Zoom in or out Maximum zoom in is 500 percent and the minimum zoom out is 100 percent based on pane size
Swipe	Touch + drag (+ lift)		Pane	<ul style="list-style-type: none"> Pan the window Hierarchy navigation
Flick	Touch, make a quick flick.		Pane	The inertia and acceleration of a flick gesture distinguishes it from a swipe gesture.

Layout or ViewApp Level Gestures

Layout or ViewApp level gestures require two or more fingers and affect the behavior of content shown in a screen window.

Gesture	Description	Gesture Illustration	Touch Gesture Target	Effect of Gesture
Multi-finger swipe	Touch three fingers, drag and lift		Pane.	Three finger gesture shows a slide out pane from the edge opposite to the direction of the swipe.
Scrunch	Five finger touch and squeeze.		<ul style="list-style-type: none"> Single pane Multiple panes 	On scrunch, the window's content scales until it reaches 70 percent of its original size at which point the window closes. In a multi-pane application, a scrunch gesture can be performed within a single pane or across multiple panes.

About touch interactions supported during runtime

This topic provides some detail about unique touch behavior during runtime.

Multi-Touch Gestures

- Within a layout, content and pane level gestures are supported. However, if a content level gesture is in progress, no other types of gestures function in other panes during the duration of the content level gesture.
- Excluding a scrunch gesture, all other types of gestures act on the contents of the pane within which the gesture occurs. Touch interactions within one pane are independent of interactions in other panes.
- Touch gestures interact with only a single WCC\WPF at a time. However, other non-WCC process graphics can interact with touch, at the same time.
- For WPF controls implementing iSupportTouch, the WPF control is responsible for ensuring multi-touch support.

Touch Support for Apps

- The NavTreeControl and the NavBreadcrumbControl support single finger swipe and flick gestures. The hierarchical tree of the NavTreeControl can be scrolled up or down and left or right. Drop-down lists of the NavBreadcrumbControl can be shown or hidden by the supported gestures.
- Single finger swipe and flick gestures can be used to navigate the alarm hierarchy of the AlarmApp.
- Single finger swipe and flick gestures can be used to show or hide drop-down lists of the ContentPresenter

and TitleBar apps.

- Native touch support on WPF controls is not supported. That is, if a user has implemented recognition of touch gestures within WPF, unless the user implements our `ISupportTouch` interface, it will not work.
- For WCC and WPF controls, without `ISupportTouch` implementation, touch will be converted to equivalent mouse gestures, where possible. These controls will respect the state of the Touch Lock if it does not implement `ISupportTouch`.

Legacy ArchestrA .NET controls accessible from the **Tools** area do not implement `ISupportTouch`.

- Touch interactions along with mouse\keyboard actions are not supported.

Scope of Touch Interactions

- Touch interactions initiated from within a pane continue to function when the gesture continues outside the boundaries of the pane.
- In the case of overlapping graphics, the topmost element configured with input animation is the target of the touch interaction.
- The behavior of double tap and long press are customizable through the Windows Control panel. AVEVA OMI ViewApps support operating system customizations.

Runtime touch interaction limitations

Some touch limitations are apparent during runtime.

- A double tap gesture to select text in an Edit Box is difficult to achieve.
- Using mouse and touch simultaneously is not supported. Either interaction type may cancel the other, but no serious errors occur that prevents the ViewApp from continuing to run.
- Touch gestures are not supported when an app has been placed in a multi-content layout pane and hierarchy or content navigation is enabled.
- Pan and zoom gestures are not supported by apps.
- Content Tooltip animations do not appear when using touch because there is no concept of hovering directly over a screen item with touch.
- A layout supports only one long press or double tap gesture at a time. You cannot perform a double tap or long press simultaneously on a layout.

Note: A layout can support multiple simultaneous touch down or touch up gestures.

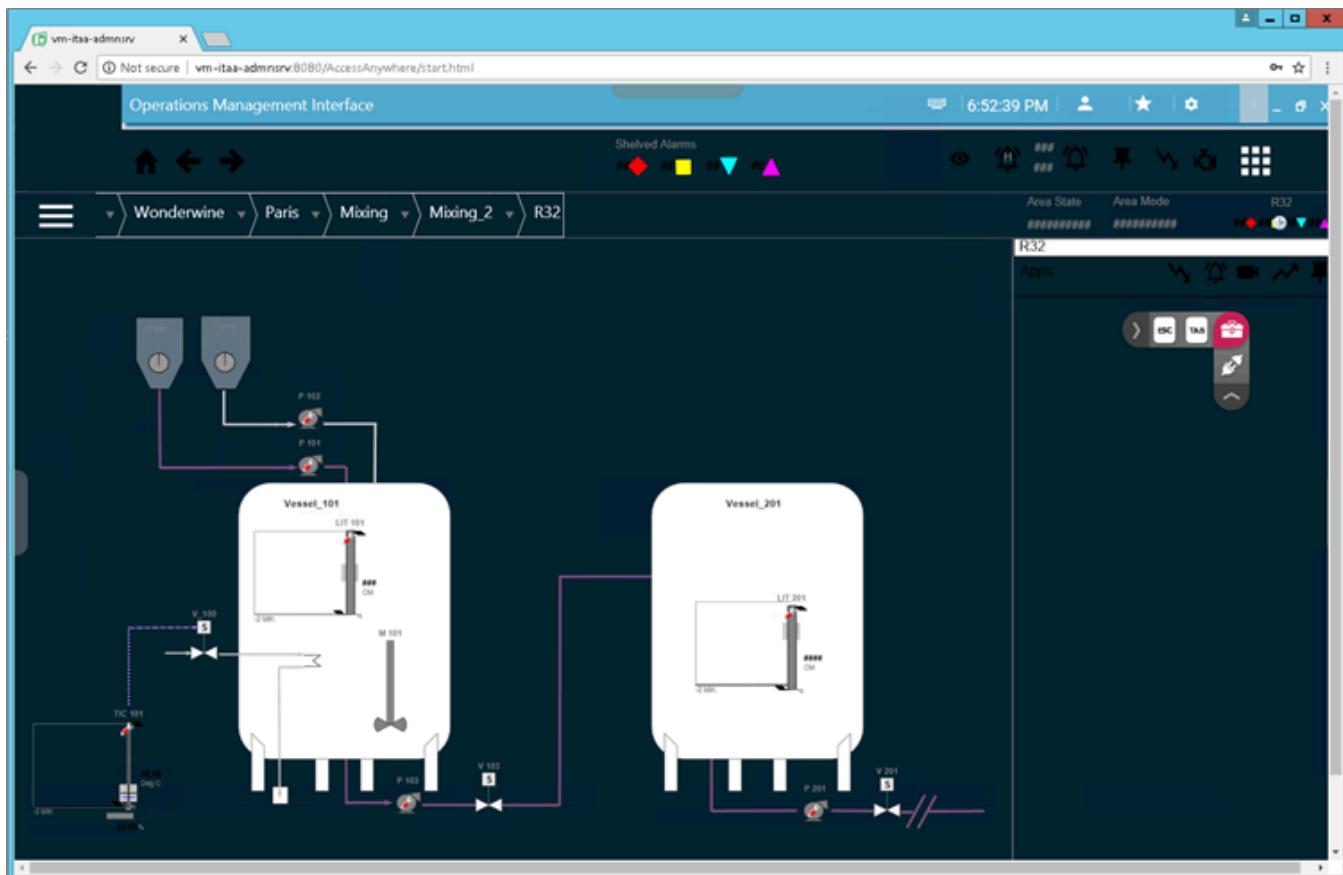
- Multi-touch gestures do not work on WCC controls.
- WinForm controls have severe touch limitations:
 - A WinForm does not respond to a Touch Lock interaction. When Touch Lock is used with WinForm controls, touch gestures become inoperable.
 - Pan and Zoom gestures do not work on layout panes if a pane contains a WinForm control.
 - Hierarchy Navigation does not work on layout panes if a pane contains a WinForm control.
 - A scrunch gesture does not work if a finger touches any WinForm control.
 - Three finger gestures do not work when one of the fingers touches a WinForm control
- WPF control specific limitations
 - Touch Lock cannot be used to control message box dialogs thrown from a WPF application such as those

created from System. The user's finger must be removed from the Touch Lock when working in Command Lock or Two Handed Operation mode to interact with these dialogs.

- WPF touch events are not supported. Any touch interaction with a user control will be promoted to equivalent mouse events. To support touch, the control must implement the ISupportTouch interface. Refer to AVEVA OMI SDK documentation for details.

About InTouch Access Anywhere

InTouch Access Anywhere enables users to remotely view InTouch applications or AVEVA OMI ViewApps from a desktop computer or mobile devices including tablets, smartphones, or laptops. You view and control applications or ViewApps through a secure HTML5 web browser. Using InTouch Access Anywhere, a separate client application does not need to be installed on your portable device or desktop computer to show a running ViewApp.



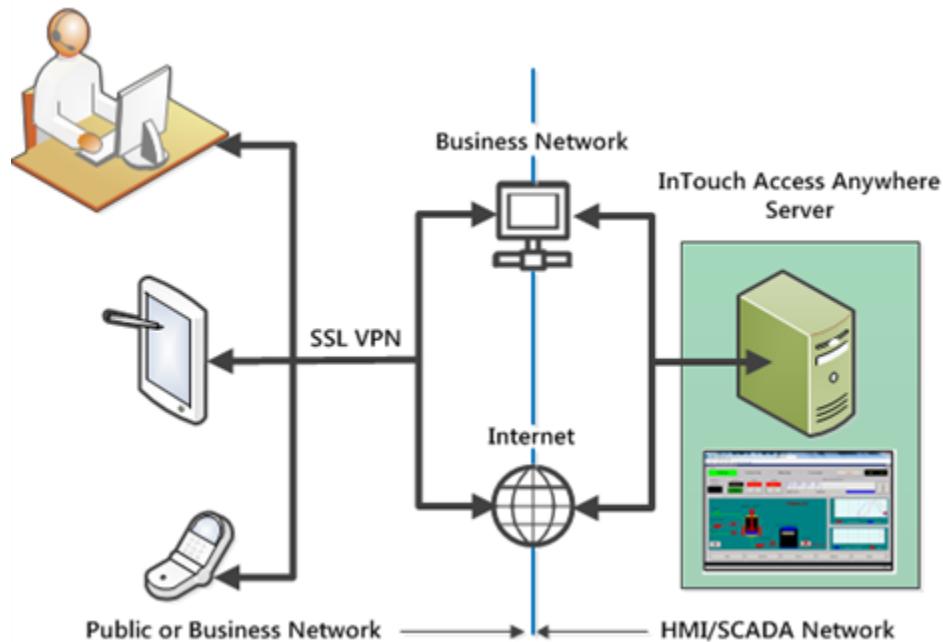
InTouch Access Anywhere provides the following features:

- Provides secure and remote access to InTouch applications or AVEVA OMI ViewApps
- Incorporates image compression, packet shaping, and whole frame rendering to improve network performance
- Automatically adjusts the size of an InTouch Access Anywhere session to the web browser window showing the application
- Supports touch gestures on touch screen devices

- Works on devices that only support web applications like Chromebook
- Provides an expandable session toolbar with icons to disconnect InTouch Access Anywhere sessions, access system keys, and copy application data to the Windows clipboard.

About InTouch Access Anywhere architecture

The InTouch Access Anywhere server is installed on a RDP host in your company's data center. One or more AVEVA OMI ViewApps are deployed to this host computer known as the InTouch Access Anywhere server. When you log on to the InTouch Access Anywhere server, you select the ViewApp you want to view. Typically, you will use a VPN from a public or business network that supports websockets to securely connect to the InTouch Access Anywhere server.



For more information about configuring the InTouch Access Anywhere server , see [Configure InTouch Access Anywhere](#).

Verify an InTouch Access Anywhere installation

InTouch Access Anywhere is installed as part of the overall installation of System Platform. For specific instructions to install InTouch Access Anywhere, see the *AVEVA System Platform Installation Guide*.

After installing the component parts of InTouch Access Anywhere, you need to verify the installation before starting the configuration tasks.

To verify your InTouch Access Anywhere Server installation

1. Before using InTouch Access Anywhere to connect to your Remote Desktop server, log on using a standard Remote Desktop Client, select a ViewApp from Application Manager, and launch it. As a best practice, complete this step for each user expected to display ViewApps through InTouch Access Anywhere.

This configures the initial setup and enables InTouch Access Anywhere clients to determine the list of available ViewApps.

The InTouch Access Anywhere Server can be used immediately after installation.

2. Open an HTML5-compliant browser and enter the URL of the InTouch Access Anywhere Server:

`http://machinename:8080/ or http://IPaddress:8080/`

This URL automatically redirects to the full URL:

`http://machinename:8080/AccessAnywhere/start.html`

The InTouch Access Anywhere Server port must be specified in the URL to tell the browser to use the web server that is built into the InTouch Access Anywhere Server service. HTTPS may also be used, but will prompt you to continue without a secured certificate.

Configure InTouch Access Anywhere

This set of topics provides information about the initial configuration of the InTouch Access Anywhere server after installation. For information about pre-installation requirements and the installation procedure, see the *AVEVA System Platform Installation Guide*.

The InTouch Access Anywhere Server installation attempts to create exceptions within the Windows Firewall to allow the necessary network connections.

If you experience problems connecting to InTouch Access Anywhere after installation, ensure that the InTouch Access Anywhere Server is configured to allow connections through port 8080, and that its executable is allowed to communicate by configuring the Windows Firewall as follows.

Configure InTouch Access Anywhere server

InTouch Access Anywhere Server is a server-side service that translates RDP into WebSocket communication. The InTouch Access Anywhere Server is installed on a RDS host.

The remote client running on a browser connects to the InTouch Access Anywhere Server service using WebSockets directly or through the Secure Gateway.

About configuring a firewall port exception

By default, a client (browser) connects to an InTouch Access Anywhere Server using port 8080 for both encrypted and unencrypted WebSocket communication. This port number can be changed using the InTouch Access Anywhere Server Configuration utility.

To enable direct connection from the client to the InTouch Access Anywhere Server (without using the Secure Gateway), the server must be directly accessible from the client using port 8080.

You can open a port through the firewall either through the command line, or through the Windows firewall configuration. The command line option is presented for advanced users as a quicker way to configure the firewall versus using the GUI.

Use a command prompt to configure a firewall port exception

To configure the firewall through the command line, first open the Windows **Command Prompt**. Then, type the

following command:

```
netsh.exe advfirewall firewall add rule name=<Description> dir=in action=allow  
protocol=TCP localport=<PortNumber>
```

Where:

<Description> = the description used to describe this firewall rule

<PortNumber> = the TCP port to open

For example:

```
netsh.exe advfirewall firewall add rule name ="Open Port 8080 for InTouch Access Anywhere"  
dir=in action=allow protocol=TCP localport=8080
```

Use the Windows GUI to configure a firewall port exception

If the Windows firewall is enabled on the same computer where the InTouch Access Anywhere Server is installed, make sure to configure it to enable the InTouch Access Anywhere client connection.

1. Open the Windows **Control Panel** and then **Windows Firewall**.
2. Select **Advanced Settings** and select **Inbound Rules**.
3. Click **New Rule**.



4. Select **Port** and click **Next**.
5. Enter the specific port: 8080.



6. Click **Next** and select **Allow the connection**.
7. Click **Next** and select to apply the rule on the Domain, Private, and Public networks.
8. Click **Next**, assign a name for the rule, and click **Finish**.

Configure a firewall program exception

In addition to adding an exception for connections on Port 8080, the InTouch Access Anywhere Server program must be added to the list of programs able to communicate with the network.

Use the Server Configuration console for settings

The Server Configuration console shows a series of tabs that enable an administrator to configure various settings of the server service.

You can launch the InTouch Access Anywhere Server Configuration tool from the **Start** Menu Program Group or on the **Apps** view.

In general, changing the InTouch Access Anywhere Server configuration is not required. It is recommended to use the default settings.

Note: It is recommended to hide the Server Configuration application from end users to prevent unexpected changes to the server's settings.

The following topics describe the different configuration tabs of the InTouch Access Anywhere Server.

Use the Server Configuration console General tab

The **General** tab provides functions to start and stop the InTouch Access Anywhere Server service. For certain configuration changes, a service restart is required. This page also displays the number of active InTouch Access Anywhere Server client sessions connected to this computer.

Note: Whenever the InTouch Access Anywhere Server service is restarted, all sessions on the server are disconnected.

Use the Server Configuration console Performance tab

The **Performance** tab displays current performance statistics related to InTouch Access Anywhere connections.

Use the Server Configuration console Communication tab

The **Communication** page provides options to change the InTouch Access Anywhere Server port and the address of the host computer running RDS.



When using an InTouch Access Anywhere Server listening port other than the default (8080), the port number must be explicitly specified in the client address field (for example, <http://<machine name>:5678/>).

When running InTouch Access Anywhere Server on a computer with multiple network cards, change the RDP host address. Change this address from localhost to the IP or DNS address of the network card that has RDP access to the system.

Changes to either setting require a service restart. The **General** tab provides buttons to start or stop the service. You can also start or stop the service using the Windows Service Manager.

Note: If you change the port number, ensure that you make the corresponding changes to the config.js file's "wsport" setting. For more information, see [About static configuration of the config.js file](#)

Use the Server Configuration console Acceleration tab

The **Acceleration** tab provides options to change the Acceleration or Quality level and disable dynamic compression.



When the **Override client acceleration/quality settings** check box is selected, all sessions use the configured setting, and all client settings are ignored. When selecting or clearing this setting, the service must be restarted for the change to become effective. When the setting is enabled, changing the acceleration level does not require a service restart, but active users must reconnect to use the new setting.

Dynamic Compression identifies small graphic elements within an application screen and compresses them during runtime. The most compression occurs when image quality is set to Low. The best quality images occur when image quality is set to higher than Low. All other graphical objects are compressed at the selected quality. This provides the visual impression of a high quality remote desktop session.

By default, this feature is enabled. To disable dynamic compression, clear the **Use dynamic compression** box.

Use the Server Configuration console Security tab

This **Security** page provides options to configure the InTouch Access Anywhere Server security settings.



Note: InTouch Access Anywhere provides integrated 128-bit SSL encryption. For best performance, set the host's RDP Security Encryption level to Low and change the Encrypt InTouch Access Anywhere communication to Always. Using this configuration, InTouch Access Anywhere SSL encryption will be used instead of the RDP encryption. Do not set this if users will be connecting directly to RDP regularly, as those sessions will end up using Low encryption.

To use a custom or trusted certificate, enter the thumbprint ID in the **Certificate Thumbprint** text box and click **Apply**. The certificate's properties will then appear.

Note: When installing a trusted certificate, the DNS address of the InTouch Access Anywhere Server must match the certificate name. If wildcard certificate is used, the domain must match. For example, if the certificate is for `*.acme.com`, the server name must end with `acme.com`.

Use the Server Configuration console Logging tab

The **Logging** tab provides options to enable or disable logging features. Technical Support may request a debugging log for diagnostic purposes. The debugging log is enabled here.

About advanced server configuration (For Administrator Use Only)

This page provides access to advanced Server settings that are stored in the system's registry.

Export Settings exports the InTouch Access Anywhere Server Registry key to the user's home folder (for example, My Documents).

Import Settings imports previously saved InTouch Access Anywhere Server Registry settings.

Advanced Configuration opens the Registry Editor.

About the InTouch Access Anywhere web component

The web component contains the resources used by a web browser to display an interface for the user to connect to an InTouch application. These resources include HTML pages, JavaScripts, CSS files, and graphic images. Review [Configure InTouch Access Anywhere](#) to modify the appearance and behavior of the web component interface.

About installation of the InTouch Access Anywhere Server

The InTouch Access Anywhere web components are automatically installed with InTouch Access Anywhere Server. The web components are located in the InTouch Access Anywhere Server folder, which by default is:

<drive letter>:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere

Note: Your installation may be located elsewhere depending on selections made during the installation process.

Modify the InTouch Access Anywhere interface

The InTouch Access Anywhere Server start page includes a group of images. All standard images can be edited and replaced with custom images. Keep the replacement images as close to the same dimensions as the original images. A default image is available as the logo.

The default path to the resources folder where the images are stored is:

C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere\resources

Note: Backup the resources folder before making any modifications. To roll-back to the original files, simply copy the original resources folder back to the original location.

InTouch Access Anywhere image files that are commonly customized include the following:

File Name	Description
Ericom.jpg	Logo image at the upper left-hand corner of the InTouch Access Anywhere Server landing page.
\images\Background-neuronal.jpg	Background image for the InTouch Access Anywhere Server landing page.

Note: Unless instructed by our Support group, customizations performed on the InTouch Access Anywhere page not herein described are not supported.

Modify the InTouch Access Anywhere connection name

The InTouch Access Anywhere connection name uses the RDP Host node name by default. The connection name can be modified to a custom string.

To change the connection's name:

1. Open the config.js file and add the name setting if it does not exist.
2. Set the name setting to the desired string enclosed in quotation marks.

```
executerTimeout: 1000,
inTouchAccessPort: 100
name: "testname",
clipboard: true
clipboardTimeoutSeconds: 15.
```

Note: The name setting may also be set using the following cookie: EAN_name.

3. After setting the name parameter, the new label will appear in the connection's browser tab and in the **Establishing connection** dialog box.

About InTouch Access Anywhere secure connections

This section describes secure connection communication between WebSockets to both remote desktops and to the InTouch Access Anywhere Secure Gateway.

About secured WebSocket communication to remote desktops

The InTouch Access Anywhere Server installation includes a self-signed certificate for secure SSL connections. Some browsers, such as Google Chrome, allow self-signed certificates for SSL-encrypted WebSocket connections.

Opera browsers will notify the user that the server certificate is not signed and prompt the user to continue. Chrome OS, Safari 5.x, and Firefox do not allow secure SSL connections using a self-signed certificate.

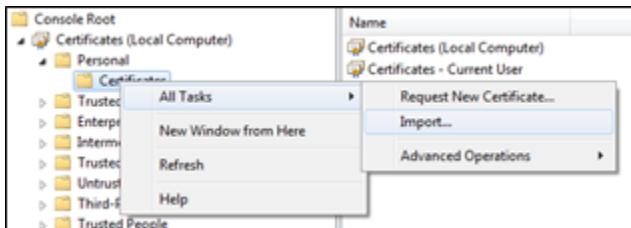
In order to provide connectivity from these browsers, a trusted certificate must be imported into the InTouch Access Anywhere Server or into the InTouch Access Anywhere Secure Gateway if it is being used as a proxy for InTouch Access Anywhere Server. A trusted certificate must be purchased from a trusted certificate authority (for example, VeriSign).

Note: The DNS address of the InTouch Access Anywhere Server or Secure Gateway server must match the certificate name. If a wildcard certificate is being used, the domain must match. For example, if the certificate is for *.acme.com, the server name must end with acme.com.

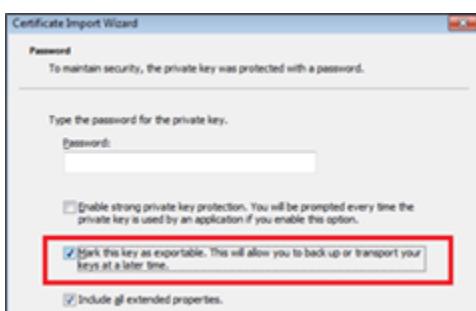
To import a trusted certificate into the InTouch Access Anywhere Server, perform the following steps using the

Microsoft Certificate Manager.

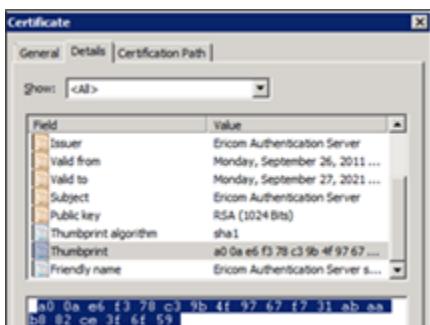
1. Show the Windows Command Prompt running as an Administrator.
2. Type certmgr.msc to show the Certificate Manager.
3. Import the trusted certificate to the Computer (Personal | Certificates) store.



4. Mark the certificate as exportable during the import.



5. Go to the Certificate's Details tab and highlight the **Thumbprint**.



6. Copy the thumbprint (Ctrl+c).
7. Stop the InTouch Access Anywhere Server service.
8. Using the Command Prompt (cmd.exe), go to the folder that contains AccessNowServer32.exe.
9. Run: AccessNowServer32.exe/genbincert <thumbprint of certificate to export enclosed in quotation marks>.

The following is an example import command with thumbprint in quotation marks:

```
c:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server>AccessNowServer
32.exe /genbincert "18 9d f3 52 bb 35 77 12 da 87 e3 85 c6 e2 bc 45 50 50 fd 10"
```

10. After importing the thumbprint, a notification appears confirming the BIN certificate has been successfully created.
11. Start the InTouch Access Anywhere Server service and it will be ready for use.

Secured WebSocket connections using InTouch Access Anywhere secure gateway

The connection between a browser client and the InTouch Access Anywhere Secure Gateway is always secured. The InTouch Access Anywhere Secure Gateway is installed with a self-signed certificate by default, but supports trusted certificates as well. For information about using certificates with the Secure Gateway, see [About InTouch Access Anywhere Secure Gateway ports and SSL certificates](#).

About Using a trusted certificate

Certain browsers permit HTTPS or SSL connections only when a trusted certificate is present. Install a trusted certificate in the InTouch Access Anywhere Secure Gateway or InTouch Access Anywhere Server to ensure safe and reliable connections from a wide range of web browsers. A trusted certificate must be purchased from a trusted certificate authority (i.e., VeriSign).

About InTouch Access Anywhere on MobileDevices

This set of topics provides information about supported browsers and information regarding specific behavior of mobile devices.

Supported Browsers for InTouch Access Anywhere

The following Web browsers have been formally tested and verified to work with InTouch Access Anywhere.

- Microsoft Internet Explorer 11
- Microsoft Edge
- Firefox version 47
- Safari version 8
- Chrome version 51
- Opera version 38

Functionally Compatible Browsers

This list includes HTML5 browsers that should be compatible with InTouch Access Anywhere, but have not been tested.

- Microsoft Internet Explorer 10 if connected through Secure Gateway
- Firefox versions 38 and later
- Safari versions 5 and later
- Chrome versions 12 and later
- Opera versions 11 and later

Refer to the InTouch Access Anywhere Readme for more information regarding tested and supported browsers.

Older versions of Firefox and Opera require WebSocket support to be manually enabled in the browser configuration.

Multiple InTouch Access Anywhere sessions can be opened in different tabs within the web browser, or in different browser windows. When a session is not in use (its tab or window is not displayed) it will reduce its CPU and memory utilization.

Note: Each InTouch Access Anywhere session consumes an RDP session and an InTouch TSE license.

About HTTPS mode for InTouch Access Anywhere in the browser

For environments where WebSockets support is not available, InTouch Access Anywhere can work in HTTPS mode to transmit data by HTTPS only. HTTPS mode is used only if WebSockets support is not available.

WebSockets will be used when available as it will provide better performance. HTTPS mode is required when using Microsoft Internet Explorer 9 with InTouch Access Anywhere Secure Gateway, or with SSL VPNs that only proxy HTTPS traffic.

To enable HTTPS mode, the InTouch Access Anywhere Secure Gateway is required. The InTouch Access Anywhere Server web pages must be delivered using the web server built into the InTouch Access Anywhere Secure Gateway (files are located under the Webserver/InTouch Access Anywhere folder).

Complete the following procedure to enable InTouch Access Anywhere for HTTPS support.

1. Install the InTouch Access Anywhere Server on the desired RDP Host.
2. Install the Secure Gateway on a separate computer located in a DMZ. The Secure Gateway must be installed on a server that is accessible by the target end-user group(s).
3. To connect to the InTouch Access Anywhere Server using HTTPS, enter the InTouch Access Anywhere URL of the Secure Gateway (the Secure Gateway includes the InTouch Access Anywhere web component):
`https://<securegatewayaddress>/InTouch Access Anywhere/start.html`
4. Enter the parameters of the target InTouch Access Anywhere Server in the start.html page.
5. After connecting by HTTPS mode, a '-' character appears as a prefix of the address in the browser tab.

Note: HTTPS mode requires a browser that supports the HTML 5 Canvas. Older browsers, such as Microsoft Internet Explorer 8 (or earlier), do not support the HTML 5 Canvas.

Configure InTouch Access Anywhere

The topics in this section discuss several additional ways to configure aspects of InTouch Access Anywhere.

Modify the InTouch Access Anywhere Interface

Some images can be modified in order to customize the appearance of the interface. Graphics are stored in the "resources" sub-folder of the InTouch Access Anywhere Web Server installation (by default, "C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere\resources"), are most commonly modified:

Note: Back up the **resources** folder before making any modifications. You may then undo the changes by copying the backup to its original location.

File	Description
ericom.jpg	Logo image displayed at top left of InTouch Access Anywhere interface

Note: An experienced web developer can customize more graphics, though these modifications are not supported by AVEVA technical support.

About static configuration of the config.js file

An administrator can modify configuration settings of InTouch Access Anywhere by editing its config.js file that is installed as part of the InTouch Access Anywhere web component. This is a JavaScript file that can be modified using any text editor.

Important: Always create a backup before making any changes to the config.js file.

Most configuration settings in the config.js file have the following format:

name: value,

A value can be a number, a flag (true or false), or text enclosed in quotation marks. Some settings are prefixed by a double slash (//), which means they are disabled. Remove the double slash to assign a value to a setting. JavaScript rules apply in this file and certain characters need to be escaped (for example, backslash).

After the settings are configured, save the file and restart the server.

The config.js file contains the following configuration settings. Setting names are case sensitive. When settings are specified using cookies, setting names are prefixed by EAN_.

address	Address of InTouch Access Anywhere Server. This is always blank for the standard configuration.
audiomode	0 enables audio redirection (default). 1 plays audio on remote computer. 2 disables audio redirection.
blaze_acceleration	True determines if RDP acceleration is used.
blaze_image_quality	Sets the quality level using a numeric. For example: 40 (fair quality), 75, 95 (best).
dialogTimeoutMinutes	Time out period, in minutes, after which an inactive dialog is automatically closed and the session is logged off. The time out period is relevant only for dialogs that have a log off button.
disableToolbar	True (default); set to False to disable the toolbar, which contains shortcut icons and file functions, that appears within an InTouch Access Anywhere session window.
domain	The name of the domain against which the user name and password are authenticated to grant access to the

	Remote Desktop session.
encryption	False determines if encryption is enabled from the client to the InTouch Access Anywhere server.
endURL	URL to open to after the InTouch Access Anywhere session has ended (# value closes window). If there is a prefix with the symbol ^ then this sets the value of window.location instead of top.location. This is useful when the InTouch Access Anywhere session is embedded in a frame.
fulladdress	Address of RDP host. This is always blank for the standard configuration.
gateway_address	Defines the address and port of the Secure Gateway. For example: secure.acme.com:4343
gwport	The default gateway port that will be used if it is not explicitly specified in the address field.
hidden	A comma or space-separated list of field names as they appear in config.js. For example, "username,password, domain". The listed fields are hidden to prevent the user from modifying them. To hide a button, such as the Advanced button, prefix the button text with the word show. For example, "showAdvanced, showAbout" hides both the Advanced and About buttons. All hidden variables will ignore previously saved settings.
leaveMessage	The message shown to the user after navigating away from an active session.
minDesktopWidth	Sets the minimum desktop width (in pixels) that InTouch Access Anywhere will display. The default is 800, which may not display as expected or desired on devices with a display width below 800 pixels.
minDesktopHeight	Sets the minimum desktop height (in pixels) that InTouch Access Anywhere will display. The default is 600, which may not display as expected or desired on devices with a display height below 600 pixels.
minSendInterval	Specifies the minimum duration between mouse position messages sent from the client when the mouse button is pressed. Units are in milliseconds.

name	Defines a custom string for the connection name. By default, the RDP host address is used.
noHTTPS	By default, InTouch Access Anywhere first attempts to connect using WebSockets. If the Secure Gateway is used with InTouch Access Anywhere, the connection will fall back to HTTPS when WebSockets are not available. If this setting is set to true, only WebSockets will be used and HTTPS fallback will be disabled.
onlyHTTPS	By default, InTouch Access Anywhere first attempts to connect using WebSockets. If the Secure Gateway is used with InTouch Access Anywhere, the connection will fall back to HTTPS when WebSockets are not available. If this setting is set to true, HTTPS is used immediately.
overrideSaved	False (default) settings that the user changes are preserved between sessions and override values set in config.js. Change to true for config.js to override preserved settings.
password	Password to pass into an InTouch Access Anywhere session (entered as clear text in config.js file).
reconnectOnDropped	True (default) automatically reconnects a session after recovering from a network outage. Set to False to disable this behavior.
remember	False (default) determines whether the user's password will be saved in the InTouch Access Anywhere page for future use. Set to true to enable password saving (not recommended for kiosk usage).
resolution	Sets the resolution size of the InTouch Access Anywhere screen. The value set must be a valid option under the InTouch Access Anywhere screen resolution setting. For example: "1024,768". For Full Screen, use: screen.
sessionTimeoutMinutes	Time out period, in minutes, after which an inactive session is disconnected. The time out period resets automatically whenever the user clicks on the keyboard or a mouse button. The default value is 0, which disables this feature.
settings (URL parameter only)	Name of the Configuration Group to be used.
settingsURL	URL of the connection settings file.

use_gateway	False (default), set to true to use a Secure Gateway for remote access.
username	Username to pass into an InTouch Access Anywhere session.
wport	<p>The default WebSocket port that will be used by the client. The value specified in the file (8080 by default) is used for both encrypted and unencrypted WebSocket communication. The user can override this value by explicitly specifying another port address in the client user interface (UI).</p> <p>For backward compatibility with older versions of InTouch Access Anywhere Server, this behavior can be modified. If singlePort is set to false, then the port value specified is only for encrypted communication. The value specified in the file plus one (8081 by default) will be used for unencrypted WebSocket communication.</p>

WARNING! Do not attempt to modify config.js settings not listed here unless directed by our Technical Support department.

Define InTouch Access Anywhere configuration groups

All users share configuration settings specified in the config.js configuration file. Special settings can override global settings for certain groups of users. Multiple configuration groups are defined in the configuration file.

For example, if the Marketing group needs clipboard redirection and printing enabled, change config.js as follows:

```
var defaults = { / this already exists in the file
...
"Marketing": { // bold text are new additions
printing:true,
clipboard:true
},
};
```

Note: The quotation marks surrounding Marketing must be identical. If necessary, delete them and re-type them if the text was copied from another source. Also, the last setting of the configuration group should not have a ',' at the end. This comma is placed after the closing bracket '}'.

In the URL to be used by the Marketing group, add the settings parameter:

<http://<machine name>:8080/InTouch Access Anywhere/start.html?settings=Marketing>

About settings precedence for InTouch Access Anywhere

When an InTouch Access Anywhere client starts, it reads configuration information from a variety of sources. If two or more sources contain different values for the same setting, the value used by InTouch Access Anywhere is determined by the following precedence order:

Highest Precedence to Lowest Precedence

- URL parameters
- Cookies
- Saved settings from previous session
- config.js

For example, if the gateway_address is specified to be "server1" in config.js but "server2" in a cookie (EAN_gateway_address), then the value "server2" will be used.

If the setting override Saved is set to true in config.js, then any settings predefined in the config.js file will override previously used settings, and the precedence order will change slightly:

Highest Precedence to Lowest Precedence

- URL parameters
- Cookies
- config.js
- Saved settings from previous session

Note: These settings become effective only after the user starts a new session. In some cases, the local browser must be closed and reopened before changes become effective. The local browser cache may also need to be cleared.

About form posts in InTouch Access Anywhere

When using a third-party authentication entity (such as an SSL VPN) that supports Form Post, the user can sign on to an InTouch Access Anywhere session using the authenticated credentials. The Secure Gateway is required for this feature.

In the authentication entity, there will be a field requesting the Post URL. Enter the SSO URL for the desired product:

AccessNow: <https://sq-address/AccessAnywhere/sso>

Note: The Secure Gateway will auto-redirect the request to the respective default page (start.html).

Include the following fields in the form:

- name="autostart" value="yes"
- name="esg-cookie-prefix" value="EAN_"
- name="username"
- name="password"
- name="domain"

Here is an example from a Juniper SSL VPN:



The value "esg-cookie-prefix" in the graphic above defines the Access Anywhere cookie prefix in the Single Sign-on form. For InTouch Access Anywhere connections, this is a mandatory entry.

If the target is a relative URL, it will replace the "/sso" portion in the path.

If the target is a full URL, it will completely replace the current path.

Embed InTouch Access Anywhere in an iframe

To embed InTouch Access Anywhere within a third-party web page using the iframe mechanism, place an iframe tag within the containing page, and have the SRC attribute of the iframe reference the InTouch Access Anywhere URL.

For example:

```
<body>
<h1>Embedded InTouch Access Anywhere</h1>
<iframe src="http://127.0.0.1:8080/AccessAnywhere/start.html" style="width:1024px;
height:768px"></iframe>
</body>
```

When an InTouch Access Anywhere session ends, it can be configured to send the browser to a specified URL using the endURL setting.

- Specify a simple URL to redirect the iframe.
- Prefix the URL with ^ to redirect the iframe's parent (container).
- Prefix the URL with \$ to redirect the top-most container.
- Specify # and the URL will close the browser tab.

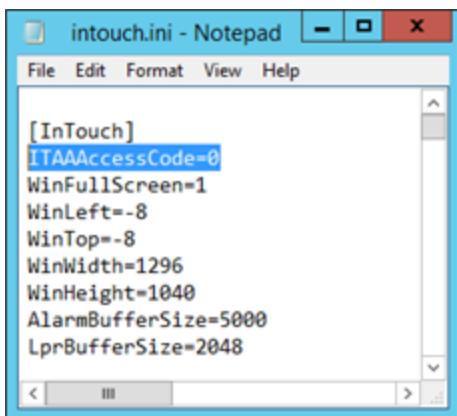
Hide InTouch Applications from InTouch Access Anywhere

By default, all available InTouch applications are accessible by InTouch Access Anywhere.

You can hide an InTouch application from the list of applications provided by the InTouch Access Anywhere Server start page.

To hide an InTouch application

- Browse to the location of the InTouch application in Windows Explorer.
- Open the **intouch.ini** file in a text editor, such as Notepad.
- Find or create the entry **ITAAccessCode** under the [InTouch] section:
 - To hide the application, set **ITAAccessCode=0** as shown below.



- To display the application, set ITAAAccessCode=1. Note that this setting is unnecessary to display the application unless it exists *and* is set to 0.
4. Save the file under its original filename.

Configure gestures for touch devices

InTouch Access Anywhere automatically detects if a portable device is touch capable and automatically uses the built-in virtual keyboard for text input and gesture support for display navigation. InTouch Access Anywhere supports Windows multi-touch gesture redirection. All multi-touch gestures are redirected natively into the Windows session for use by an application running inside an Access Anywhere session.

Multi-touch gesture redirection feature is enabled using the Access Anywhere Toolbar button and by config.js settings.

Activation Criteria

Access Anywhere multi-touch gesture redirection is enabled and activated based on the following criteria:

Multi-touch gesture redirection functionality is enabled if all of the following are true:

- Touch is supported by the remote RDP host
- Touch is supported by the user's computer or mobile device
- Touch redirection is enabled by setting rdpTouchEnabled to True in the config.js file

Multi-touch gesture redirection can be activated if all of the following are true:

- Touch feature is enabled
- Toggle MultiTouch icon is set to active on the Access Anywhere toolbar
- Touch is not suspended by the RDP host

Toolbar button

Users enable and disable the Windows multi-touch gesture redirection feature by toggling the Toggle MultiTouch icon in the Access Anywhere toolbar, which is enabled by default.

Multi-Touch Enabled



Multi-Touch Disabled



Multi-touch Gesture Redirection Settings in the Config.js File

The Config.js file is located in the folder path of the computer hosting the InTouch Access Anywhere server:

C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere

Config.js Settings	Description
rdpTouchEnabled	True (default) - enable remote touch. On the Server: enable the feature, create RDP dynamic virtual channel, and send RDP client touch events. On the Client – enable the feature, process the incoming server touch messages.
rdpTouchActive	False (default) – Sets the default activation state (ignored when not enabled.) On the Client this is the initial state of the toolbar button. If active, send touch events.
rdpTouchAction	0 (default) - Action to be taken if multi-touch redirection is enabled, but is not supported by the server or client device. Action codes and their assigned values: 0 - no action 1 - display an error message 2 - display an error message and disconnect from the session 3 - ask for user confirmation to continue without touch

Conflict with Local Gesture Usage

When multi-touch redirection is enabled, all gestures are redirected to the remote session. However, the user may need to use gestures locally on the device to pan and zoom around the session. When local gesture functionality is required, disable multi-touch redirection temporarily, and re-enable it when it is needed again.

About HTTPS mode for InTouch Access Anywhere in the browser

For environments where WebSockets support is not available, InTouch Access Anywhere can work in HTTPS mode to transmit data by HTTPS only. HTTPS mode is used only if WebSockets support is not available. WebSockets will be used when available as it will provide better performance. HTTPS mode is required when using Microsoft Internet Explorer 9 with InTouch Access Anywhere Secure Gateway, or with SSL VPNs that only proxy HTTPS traffic.

To enable HTTPS mode, the InTouch Access Anywhere Secure Gateway is required. The InTouch Access Anywhere Server web pages must be delivered using the web server built into the InTouch Access Anywhere Secure Gateway (files are located under the Webserver/InTouch Access Anywhere folder).

Complete the following procedure to enable InTouch Access Anywhere for HTTPS support.

1. Install the InTouch Access Anywhere Server on the desired RDP Host.
2. Install the Secure Gateway on a separate computer located in a DMZ. The Secure Gateway must be installed

on a server that is accessible by the target end-user group(s).

3. To connect to the InTouch Access Anywhere Server using HTTPS, enter the InTouch Access Anywhere URL of the Secure Gateway (the Secure Gateway includes the InTouch Access Anywhere web component):
<https://<securegatewayaddress>/InTouch Access Anywhere/start.html>
4. Enter the parameters of the target InTouch Access Anywhere Server in the start.html page.
5. After connecting by HTTPS mode, a '-' character appears as a prefix of the address in the browser tab.

Note: HTTPS mode requires a browser that supports the HTML 5 Canvas. Older browsers, such as Microsoft Internet Explorer 8 (or earlier), do not support the HTML 5 Canvas.

InTouch Access Anywhere server limitations

This topic describes known behaviors and limitations of InTouch Access Anywhere when viewing an application or ViewApp on a portable device.

Networking limitations

- Network quality

Network quality will impact the performance of InTouch Access Anywhere running on mobile devices. Long latencies, limited bandwidth, and poor Wi-Fi coverage of the working area will impact user experience.

We recommend that in the menu of your application you add a heartbeat or a clock that displays time, including seconds, that helps visualize good connectivity.

- InTouch Access Anywhere does not support WindowMaker

InTouch WindowMaker is not supported in a TSE (Remote Desktop) environment. Therefore, InTouch Access Anywhere does not support InTouch WindowMaker. To prevent users from attempting to start WindowMaker from WindowViewer, do not install a license that enables WindowMaker and hide the Fast Switch menu bar in your InTouch applications.

Browser limitations

- Browser Extension Conflicts

Browser extensions and tool bars may inject JavaScript code into web pages, which can adversely impact the behavior of certain web pages. If InTouch Access Anywhere is not working properly, disable or uninstall any active browser extensions or tool bars. Restart the web browser after uninstalling or disabling an extension, and clear the local browser cache, to ensure that it is no longer active.

- HTTPS and SSL Encryption

When the InTouch Access Anywhere page is delivered to the web browser using HTTPS, the SSL encryption setting will be checked by default. Modern browsers usually require WebSocket connections to be encrypted when launched from pages that are delivered using HTTPS.

- Zooming in Browsers

Using the CTRL+ and CTRL- hotkeys to zoom only works with Internet Explorer 10.

Navigational limitations

- Mouse Events

When designing your applications, keep in mind that certain mouse events do not have an equivalent behavior on a touch mobile device, including the following:

- While Left Key Down
- On Right Key Down
- While Right Key Down
- On Right Double Click
- On Right Up
- Mouse Center click
- Pushbutton>Discrete Value>Direct

Other mouse events are triggered with a gesture you must learn. For example, in many mobile devices a mouse over event is triggered by a tap on the screen.

- Right Click on Mac

To perform a right-click on Mac OSX system: Command+ left-click.

- Left Click on iPad

A single tap does not consistently toggle the state of a push button. Tap and hold to toggle a push button.

- Scroll Bars

In some cases, moving a scroll bar in a touch environment can be difficult, particularly when the device has a small screen. As an alternative, try touching the empty area of a scroll bar in the direction you want to move.

- Dialog Boxes

Dragging and dropping a dialog box can also be difficult on a touch device with a small screen. We recommend that you use a stylus to perform these operations for better precision, if possible.

- Using Software Keyboards

InTouch provides the ability to invoke an InTouch keyboard or the Windows On Screen Keyboard from Input Animations. When designing applications to be accessed by InTouch Access Anywhere from mobile devices, keep in mind that these devices have their own software keyboards optimized for their specific form and size. In these cases, invoking the InTouch or the Windows keyboards from your application is not needed. In general, users have a better experience using a device software keyboard.

Also, keep in mind that software keyboards in mobile devices in most cases do not have certain keys available in a physical keyboard, such as F1-F12, CTRL, or ALT. If you already have an application that uses Key Scripts associated with some of these keys, modify your application to use alternate available, supported keys.

Some key combinations may not be available through your mobile device, such as Shift+<letter>, CTRL+Shift, CTRL+ALT.

Network application development limitations

Distributed InTouch applications typically have a central development computer, central data storage, and client workstations that run distributed applications. You use InTouch Network Application Development (NAD) to build and maintain distributed applications. NAD enables client stations to maintain a copy of a single application

without restricting the development of that application. Client stations are automatically notified when the application changes.

NAD applications do not appear on the list of applications shown on the Start.html page when the user logs on to the InTouch Access Anywhere server.



The InTouch Access Anywhere service populates the Start.html file drop-down list of available InTouch applications. This service runs under a local system account by default and cannot communicate with applications or folders on remote computers, which are typical of a NAD environment. This is the reason why NAD applications do not appear in the list of InTouch applications.

If you open Application Manager and see InTouch applications whose folders are not located on the computer's local hard drives, then those applications will also not appear on the drop down list of the Start.html page.

A workaround is to set the InTouch Access Anywhere service to run under a specific domain user account that has sufficient privileges on the Access Anywhere server and can also access the application directory on the NAD primary computer.

Build an InTouch Access Anywhere Secure Gateway

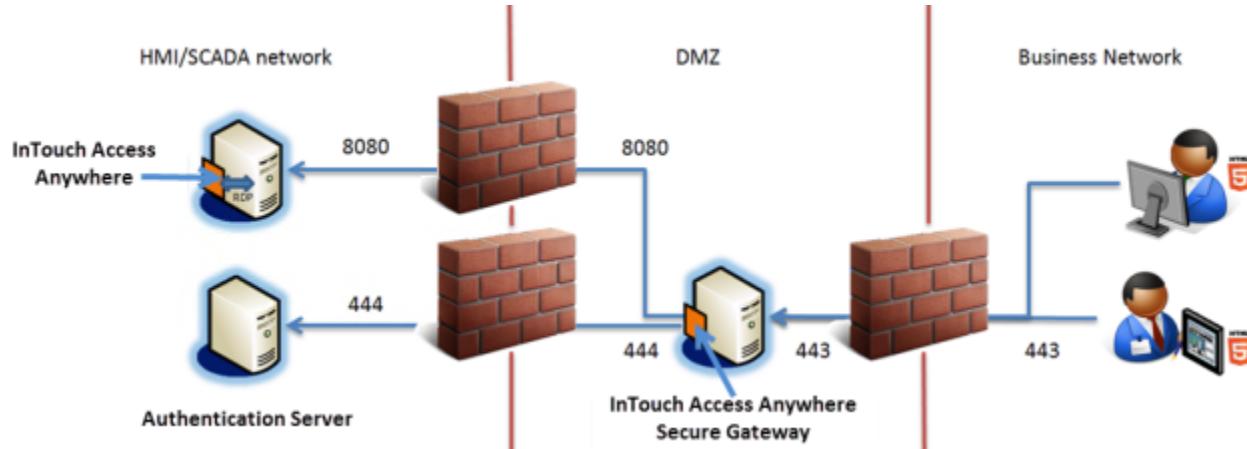
InTouch Access Anywhere Secure Gateway is a complementary component of InTouch Access Anywhere server. The Secure Gateway ensures secure access to InTouch ViewApps by acting as a gateway between users in remote locations and AVEVA OMI ViewApps deployed to hosts running in a control network.

Secure Gateway provides the following benefits:

- Access to InTouch applications running on an internal network using a single secure port
- Eliminates the need to purchase, install, configure, and manage a VPN
- Located in a perimeter network, also known as a DMZ, while all other resources reside securely behind an internal firewall
- Provides the ability to install a single SSL digital certificate on the Secure Gateway node instead of requiring a certificate for every host that needs to be accessed
- Compatible with HTML5 client browsers supported by InTouch Access Anywhere

The following diagram shows the recommended architecture of the Secure Gateway in a production

environment. The Secure Gateway uses a single port for secured remote access to InTouch applications. All web traffic from an external business network is tunneled through a SSL-based connection of the Secure Gateway placed in a DMZ.



The Authentication server is an optional InTouch Access Anywhere component and is disabled by default. The Authentication Server provides an additional layer of security by authenticating end-users before they can contact the Access Anywhere server. When the Authentication Server is enabled, only domain users will be able to authenticate. Local system users (such as Administrator) will not be able to log on through the Authentication Server.

Connect to an InTouch Access Anywhere Server through the Secure Gateway

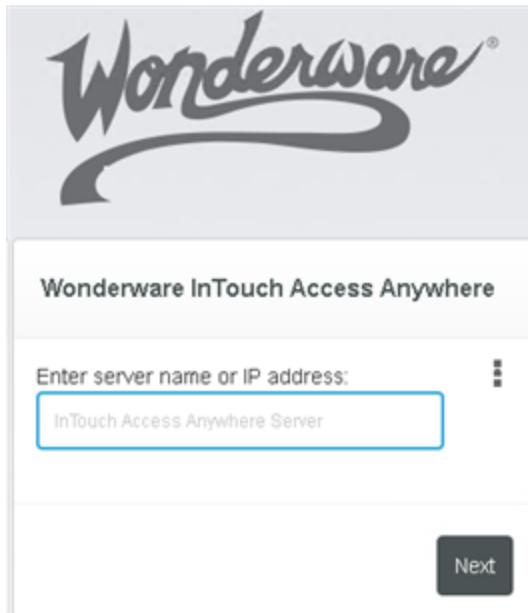
This set of topics describes how to configure the Secure Gateway node to connect to an InTouch Access Anywhere Server.

Connect to an InTouch Access Anywhere server through a secure gateway

The following logon procedures assume the InTouch Access Anywhere Server is installed on Node 1 and InTouch Access Anywhere Secure Gateway is installed on Node 2.

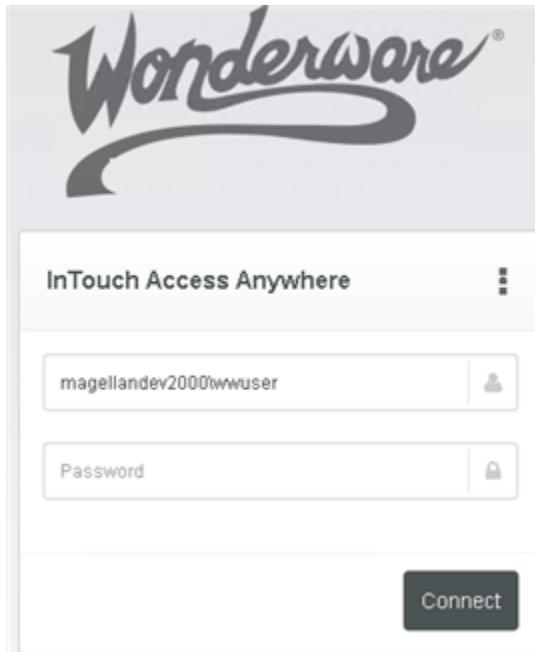
Important: One or more ViewApps must be deployed to the computer running the InTouch Access Anywhere server to be able to

You access the InTouch Access Anywhere Server by first going through the InTouch Access Anywhere Secure Gateway node. When you navigate to <https://<node2 name>/>, the **Connection Details** page appears:



To access InTouch Access Anywhere Server on Node1, enter the computer name or IP address of Node1 in the **Enter server name or IP address** field and click **Next**.

The logon page to the InTouch Access Anywhere Server on Node1 appears with fields to enter a user name and password.



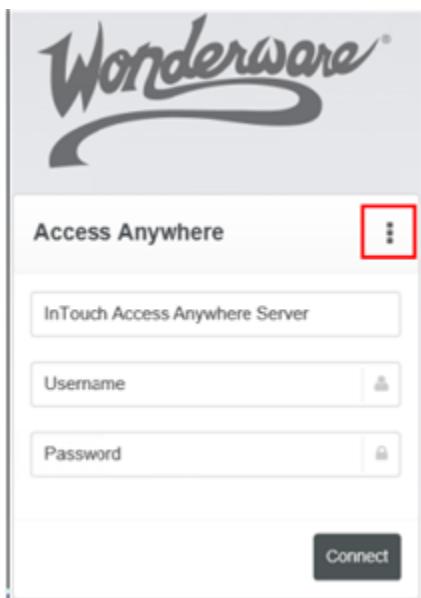
After providing your logon credentials and clicking **Connect**, there are two possible log on scenarios:

Scenario 1: InTouch Access Anywhere Secure Gateway node (Node2) does not show a list of InTouch applications

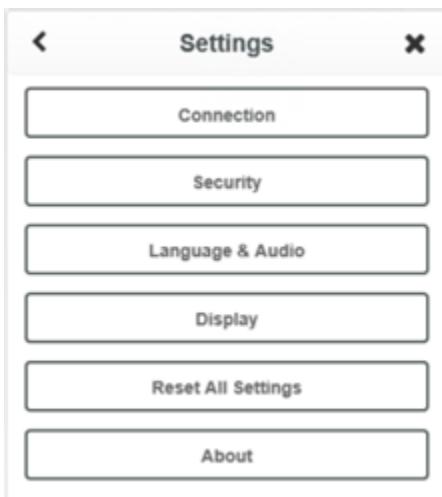
You can perform the following steps when you have a single InTouch Access Anywhere Server and will always run the same ViewApp.

To connect to the ViewApp you want to view:

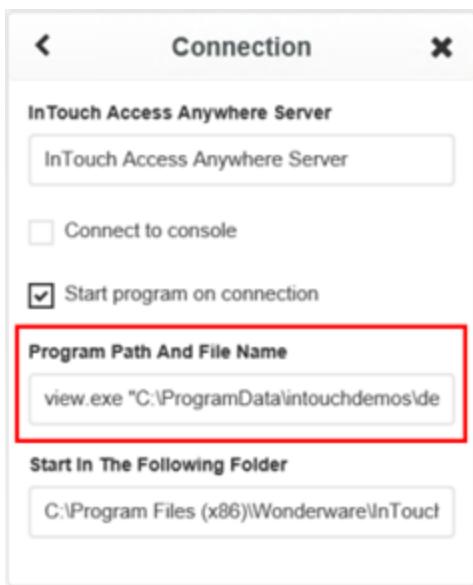
1. Click the **Settings** icon.



The **Advanced Settings** dialog appears.



2. Click the **Connection** option.
3. In the **Program path and filename** field, enter **view.exe** followed by the path to the InTouch application you wish to start on the server, enclosed within quotation marks (as shown below). If the path is not supplied, then WindowViewer will start with the last application it was running (as specified by the per-user win.ini file).



The InTouch installation path is populated by default in the **Start in the following folder** field.

Scenario 2: Secure Gateway node shows a list of InTouch applications

In this scenario, you will be directed to a page that looks similar to the start page for accessing an InTouch Access Anywhere Server. In this case, select the application you want to open in WindowViewer, then click **Connect**.

Configure a secure gateway node to host a single InTouch Access Anywhere server

If your Secure Gateway points to a single InTouch Access Anywhere Server, you do not need to specify the name of the server in the start.html file. Use the following procedure to show a list of InTouch applications available from a single InTouch Access Anywhere Server.

To display a list of InTouch applications from a single InTouch Access Anywhere Server

1. On Node 2, where the Secure Gateway is installed, go to the start.html file in the following directory:
`C:\ ProgramFiles (x86)\Wonderware\InTouch Access Anywhere Secure Gateway\InTouch Access Anywhere Secure Gateway\WebServer\Access Anywhere`
2. Rename the start page to StartOriginal.html.
3. On Node 1, where the InTouch Access Anywhere Server is installed, copy the Start.html file from:
`C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Server\WebServer\AccessAnywhere\start.html`
4. On Node 2, paste the Start.html from Node 1 in the following directory:
`C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Secure Gateway\InTouch Access Anywhere Secure Gateway\WebServer\Access Anywhere`

You can now see the Application Name list with all InTouch applications available on the single Node 1 InTouch Access Anywhere Server.

Configure a secure gateway node to show a list of InTouch applications

You can display a list of your InTouch applications on the InTouch Access Anywhere Server that can be accessed through a Secure Gateway.

1. From Node1, where the InTouch Access Anywhere Server is installed, clone (copy and paste) the Start.html page located in the following directory:
<InTouch Access Anywhere Server installation folder>\WebServer\AccessAnywhere\
2. Rename the cloned file and go to Node2. Paste the file under <InTouch Access Anywhere Secure Gateway installation folder>\Ericom Secure Gateway\WebServer\AccessAnywhere\ folder on the Gateway node (i.e. Node 2).

Note: The start page can be renamed to any valid file name, but for better readability and compatibility, we recommend prefixing the file name with the InTouch Access Anywhere server name. For example, if the server name is Primary01, the start page should be renamed to Primary01_start.html.

3. Open Start.html and locate the following html element:

```
<select id="ITAA ServerList" name="ITAA ServerList" style="visibility:hidden">
  <!-- A sample option element
  <option ServerName="Primary01" IPAddress="xx.x.xx.xx"
  StartPageName="Primary01_Start.html"/> -->
</select>
```

4. Add an option element under the select element (an example is given) and update the property values as follows:
 - The ServerName property value should be set to InTouch Access Anywhere server name (Node1 in our example).
 - The IPAddress property value should be the IP Address of the server. Setting an IP value will allow the page to be accessed when you use IPAddress instead of ServerName.
 - The StartPageName property value should be set to the start page name from step 2 above.
5. Save the changes.

Now you can see the **Application Name** list with all InTouch applications available on the InTouch Access Anywhere Server node.

About the InTouch Access Anywhere Secure Gateway Configuration Portal

The InTouch Access Anywhere Secure Gateway includes a Configuration Portal to enable an administrator to change any related settings. To access the Configuration Portal page, use a web browser and navigate to the Secure Gateway's configuration URL:

- <https://<SG-server-address>:<port-number>/admin>

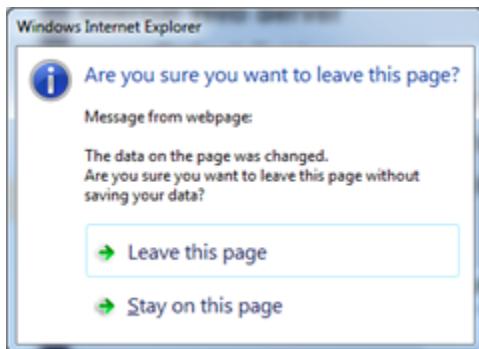
Accessing the Configuration Portal is restricted to only members of the local Administrators group of the InTouch Access Anywhere Secure Gateway server. All log ons are audited in the Secure Gateway log file. Administrators are strongly encouraged to enforce a strong password policy for Secure Gateway administrators.



To log out of the Configuration Portal, click **Logout**.

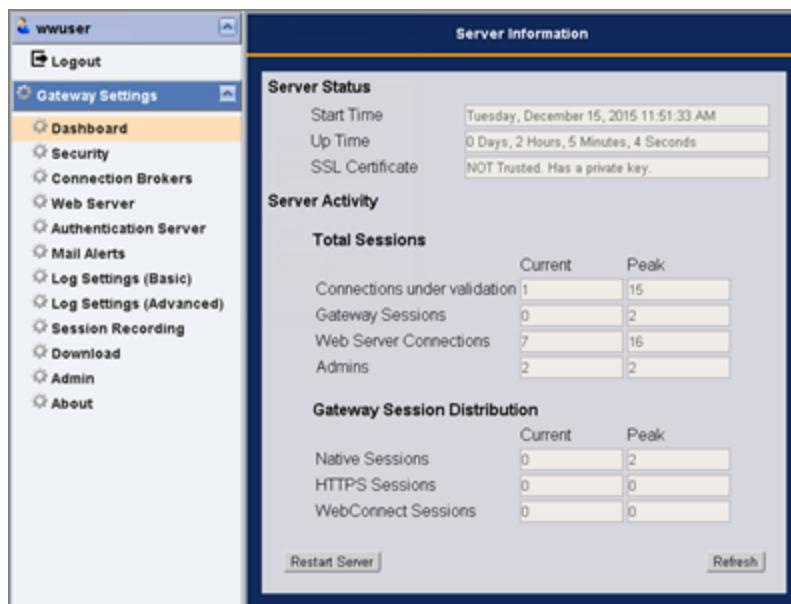


After making changes to any settings, click **Save**. If a different page is selected and the settings are not saved, a warning dialog will appear. Click **Leave this Page** to continue and cancel any changes. Click **Stay on this page** to return to the current page to save changes.



Secure Gateway configuration dashboard

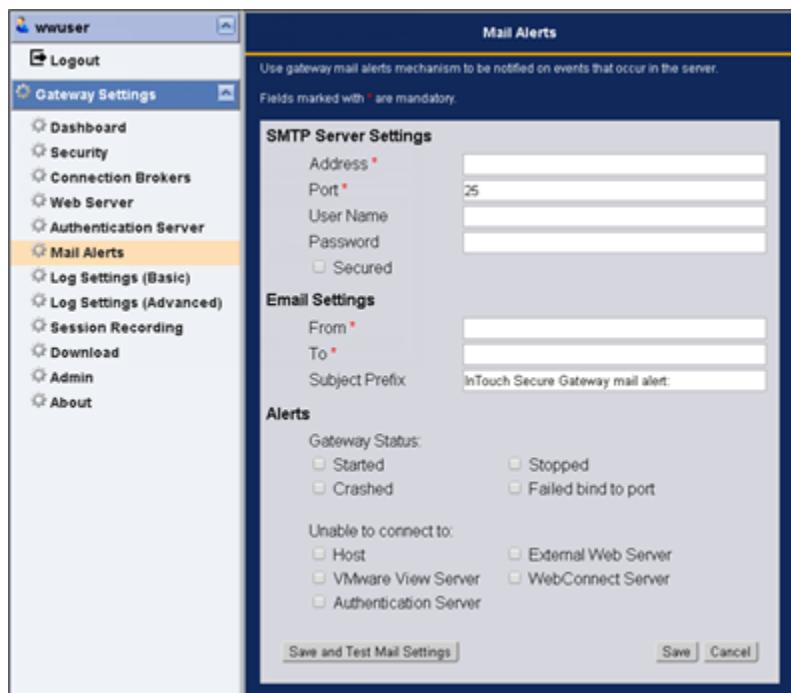
Secure Gateway **Configuration Dashboard** displays useful statistics related to the Secure Gateway operation. Open this page to view server uptime, SSL certificate status, session activity, and to restart the Secure Gateway Server service.



Configure Secure Gateway e-mail alerts

Secure Gateway can be configured to send e-mail alerts when specified system events occur. To configure mail alerts, enter the SMTP information of the e-mail server. Then, check the events that trigger an e-mail alert.

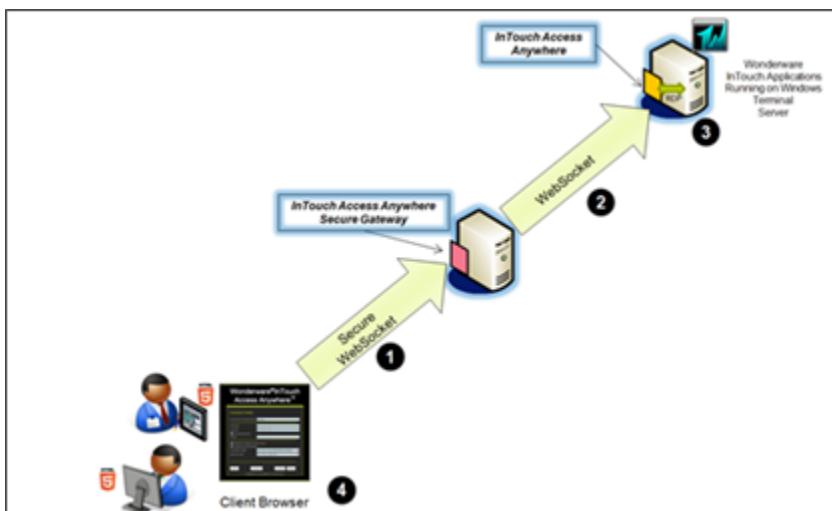
Click **Save** or **Save and Test Mail Settings** to apply the configuration.



InTouch Access Anywhere HTML5 client configuration

InTouch Access Anywhere can use the Secure Gateway to provide secured connections between HTML5 web clients and InTouch Access Anywhere servers. The following diagram shows how these components work

together.



In this configuration, a client browser always establishes a secure WebSocket connection to the Secure Gateway. The Gateway then establishes a WebSocket connection to the InTouch Access Anywhere server.

Whether the WebSocket connection between the Gateway and the Access Anywhere server can be secured or not is based on a configuration setting in the InTouch Access Anywhere client (check the box marked **Enable SSL** for the InTouch Access Anywhere web configuration).

Configure the Access Anywhere server to work with a secure gateway

The InTouch Access Anywhere server includes a set of **Security** options that indicate a Secure Gateway should be used and the address of the computer hosting the Secure Gateway.

The Access Anywhere server always establishes a secure WebSocket connection to the Secure Gateway. The Secure Gateway then establishes a WebSocket connection to the Access Anywhere server.

To configure the Access Anywhere server to work with a Secure Gateway

1. Log on to the computer hosting the Access Anywhere server.
2. Show the Start page by entering the following URL in a web browser.

<https://localhost:8080>

3. Click the Advanced Settings icon on the Start page and select the **Security** option.

The WebSocket connection between the Secure Gateway and the Access Anywhere server can be secured by selecting the **Enable SSL encryption for remote session** option.

4. Select **Use InTouch Secure Gateway** and enter the IP address or name of the server hosting the Secure Gateway in the **Gateway address** field.



About security allowlists

You can configure two types of allowlists:

- End-user Address and Range
- Target Host Address and Range

The target host allowlist is enabled by default, while the client allowlist is disabled by default. When a allowlist is enabled, a list of IP addresses must be specified.

To enable a type of allowlist, change the enabled setting from "false" to "true". For example:

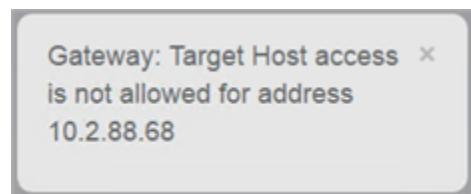
```
<add key="ClientWhitelistByIPAddressesEnabled" value="false"/>
<add key="ClientWhitelistByIPAddressesEnabled" value="true"/>
```

Note: The enabled setting is set to "true" by default for InTouch Access Anywhere.

Addresses are entered in the standard format, for example 10.2.88.1, and are separated by semicolons (;).

Address ranges are defined using a lower IP, the character "-", and the upper IP. For example: 10.2.88.1-10.2.88.5

The IP addresses of each Access Anywhere Server must be configured in the EricomSecureGateway.Config file, or you will be prompted with the below error message:



Note: This is an example IP address. This value will be associated with the InTouch Access Anywhere host you are trying to connect to.

The values below provide an example for how each type of allowlist would be configured in the EricomSecureGateway.exe.config file:

- End-user Address and Range:

```
<add key="ClientWhitelistAllowedIPv4Addresses" value="10.2.88.1-10.2.88.5;10.2.88.10" />
```

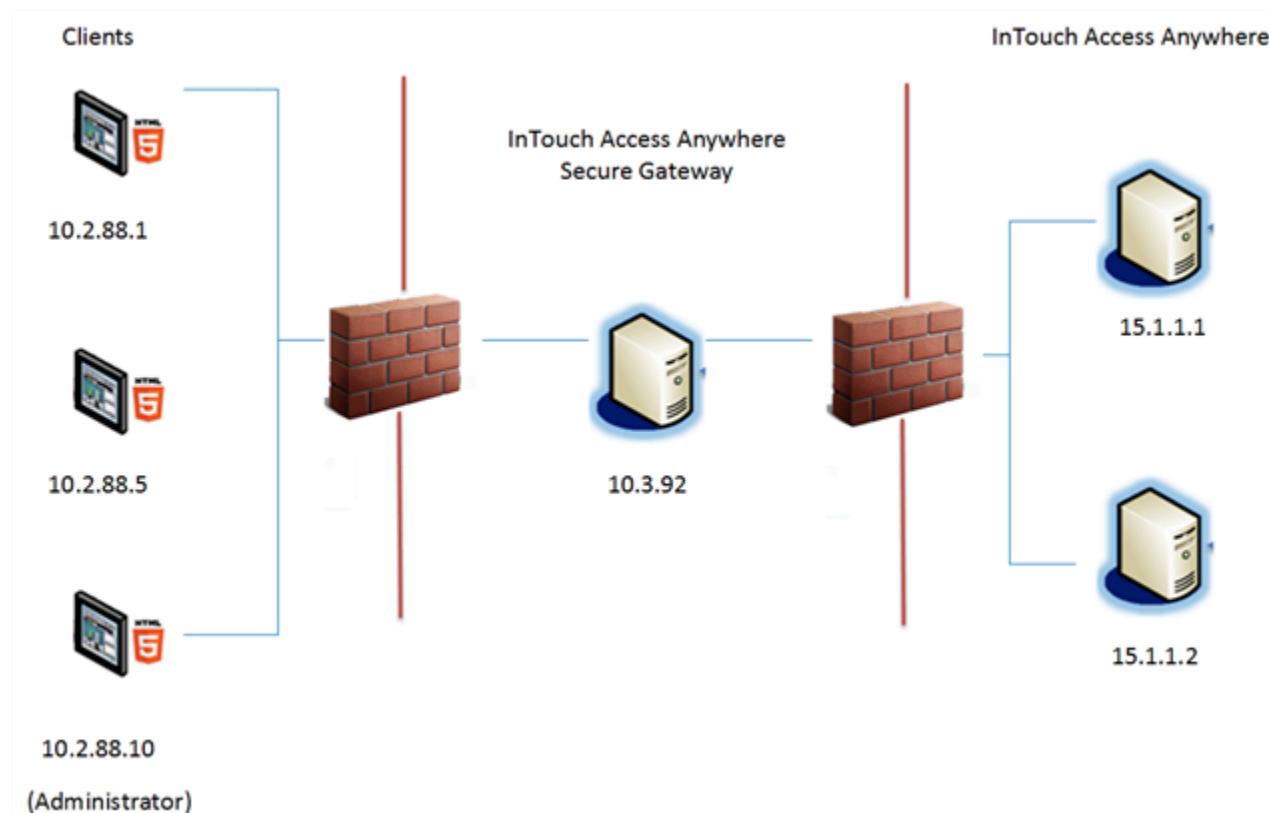
The IP Address of each client node is included.

- Target Host Address and Range

```
<add key="TargetHostWhitelistAllowedIPv4Addresses" value="15.1.1.1;15.1.1.2" />
```

The IP Address of each InTouch Access Anywhere Server(s) connecting through the Gateway.

The following diagram illustrates an example allowlist security configuration for clients, the Gateway, and the Access Anywhere Server. The IP addresses of all allowlisted clients are represented.



The list of all the configuration options is:

```
<Visitor>
<add key="HandshakeTimeoutSeconds" value="60" />
<add key="ClientWhitelistByIPAddressesEnabled" value="false" />
<add key="ClientWhitelistAllowedIPv4Addresses" value="" />
<add key="ClientWhitelistAllowedIPv6Addresses" value="" />
<add key="TargetHostWhitelistByIPAddressesEnabled" value="false" />
<add key="TargetHostWhitelistAllowedIPv4Addresses" value="" />
<add key="TargetHostWhitelistAllowedIPv6Addresses" value="" />
</Visitor>
<Admin>
<add key="InactivityTimeoutMinutes" value="5" />
<add key="WhitelistByIPAddressesEnabled" value="true" />
<add key="WhitelistAllowedIPv4Addresses" value="" />
<add key="WhitelistAllowedIPv6Addresses" value="" />
</Admin>
```

Note: ClientWhitelistByIPAddressesEnabled and the Admin allowlist settings existed in previous versions as "LockdownAllowed****Addresses", if these settings are currently configured, simply copy the parameters to the new values.

Configure the origin header parameter for allowlist security

You can configure a parameter to allowlist the origin header field in incoming HTTP requests to the InTouch Access Anywhere Gateway. The Gateway will check for this parameter upon connection request to the host. If the "origin" HTTP header exists in the connection request, it will verify that it is in the list of trusted addresses. If there is no match, the Gateway will deny the connection request to the host.

Use the following parameter to allowlist the origin header field:

```
<add Key="OriginHttpHeaderWhitelistAddresses" value="" />
```

where value is the Gateway IP address or node name. For example:

```
<add Key="OriginHttpHeaderWhitelistAddresses" value="http://10.3.92" />
```

or

```
<add Key="OriginHttpHeaderWhitelistAddresses" value="http://GatewaynodeName" />
```

If there is a load balancer or proxy server in front of the Gateway, then the value address in this parameter refers to the load balancer or proxy server. In this case, the address can be any URL. For example:

```
<add Key="OriginHttpHeaderWhitelistAddresses" value="http://URL" />
```

For detailed information about HTTP origin header specifications, see section 7 and 8 of the [Internet Engineering Task Force](#)

Configure session cookie timeout

A session cookie is generated when browser clients connect to the InTouch Access Anywhere Secure Gateway.

Use the following parameter to configure the session cookie timeout in the EricomSecureGateway.Config file:

```
<Property name="ClientSessionCookieTimeoutMinutes" type="int" value="60" />
```

Note: The default timeout period is sixty minutes. Do not set this value to 0. A value of 0 disables the cookie timeout interval.

If you try to connect by means of a Websocket or HTTPS (if enabled) and the cookie has expired, the connection will be rejected. You will need to reload the page to re-attempt the login.

The following details apply to the session cookie lease:

- A cookie is cached in the InTouch Access Anywhere Gateway the first time an end-user's browser requests a page.
- The cookie lease duration is defined based on "ClientSessionCookieTimeoutMinutes" value.
- The lease is maintained on the Access Anywhere Server side, not in the browsers, so all browsers are treated as a single browser from your device.
- The cookie value and lease are per client (IP address), so multiple browsers on the same device will use the same cookie value and the same lease.
- The cookie lease duration is not extended each time a page is retrieved. A cookie lease expires only after the configured duration.

This cookie lease duration requires reloading the page after each expiration to contact the Gateway.

Advanced secure gateway configuration

All configurable settings related to the Secure Gateway can be found in the EricomSecureGateway.exe.config file.

This is a text file that can be modified with a text editor. The configuration settings are also defined in the section [About a secure gateway built-in authentication server](#).

Changing parameter values marked as "Reloadable" do not require a service restart. "Not Reloadable" parameters only become effective after restarting the InTouch Access Anywhere Secure Gateway service.

Provide high availability of a secure gateway

To provide high availability of the Secure Gateway, it is recommended that you install two or more Secure Gateways and use a third-party redundant load balancer to manage access.

The load balancer will provide one address for end users. As requests arrive at the load balancer, they are redirected to an available Secure Gateway based on built-in weighting criteria. A basic round-robin load balancer can be used, but it may not detect whether a Secure Gateway is active or not.

Restrict access to and from a secure gateway

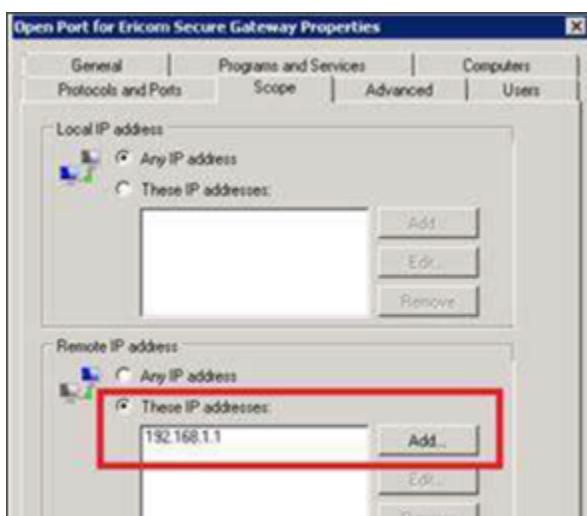
Ericom documentation mentions "Terminal Servers" here too. I assumed those should be removed.

Use the Windows Firewall Scope rules to restrict incoming connections to the Secure Gateway server.

To restrict incoming connections to the Secure Gateway

1. Access the Port rules for Secure Gateway.
2. Click the Scope tab.
3. In the Remote IP address section, click Add.
4. Enter the IP address(es) from which you wish to allow connections.

In the example below, only connections originating from 192.168.1.1 can connect to the Secure Gateway.



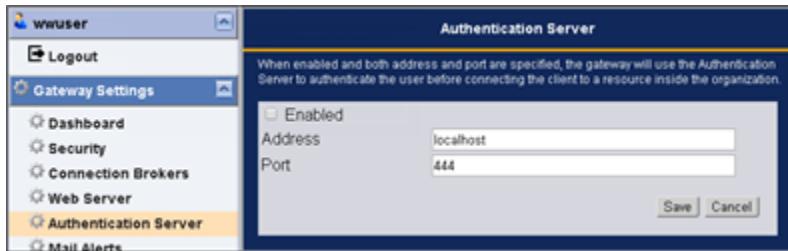
About a secure gateway built-in authentication server

The Secure Gateway includes an Authentication Server that provides a layer of security by authenticating end-users before they can access any internal resource. The Authentication Server is disabled by default and should be installed on a computer on the safe side of the firewall that is a member of the domain and which is

employed to authenticate users.

Note: The Authentication Server can only be configured for one domain at a time.

Use the Secure Gateway **Configuration** page to modify some of the settings of the Authentication Server:



Other configuration settings are specified in the EricomSecureGateway.Config file, which is located at
 C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Secure Gateway\InTouch Access
 Anywhere Secure Gateway

The user configurable settings are located under the **Authentication Server** section of the EricomSecureGateway.Config file and defined in the following table.

Setting	Description
Enabled	Boolean value to enable the Authentication Server or not. True enables the Authentication Server. The default is False.
Address	The IP address of the computer hosting the Authentication Server. Localhost is the default.
Port	This is the port on which the Authentication Server listens. Make sure that no other services on the system are using the same port. A port conflict will interfere with the operation of the Authentication Server. The default port is 444.
CertificateDnsIdentity	The connection between the Secure Gateway and the Authentication Server is secured. In case the Authentication Server is not using its default certificate, this parameter must be updated to include the DNS identity of the alternate certificate.
MaxClockSkewMinutes	The maximum difference in minutes between the clocks of the Secure Gateway and the Authentication Server. The default is 180.
KeepAliveFreqSeconds	The keep alive interval in seconds that maintains the connection between the Authentication Server and the Secure Gateway. The default is 30.

When an Authentication Server is enabled, only domain users will be able to authenticate. Local system users (such as Administrator) will not be able to log on through the Authentication Server.

Disable authentication server when access is through a broker

When all access is through a connection broker and not from any stand-alone clients, the Authentication Server should be disabled, and the "broker only mode" enabled.

To disable the Authentication Server

- At the **Authentication Server** page of the Secure Gateway portal, clear the **Enabled** check box to disable the Authentication Server.



- Make the following changes to EricomSecureGateway.Config file:
 - Under <AuthenticationServer>, change <add key="Enabled" value="true"/> to <add key="Enabled" value="false"/>

```
<Section name="AuthenticationServer">
<Property name="Enabled" type="bool" value="false" />
```
 - Under <Security>, change <add key="ConnectionBrokerOnlyMode" value="false"/> to <add key="ConnectionBrokerOnlyMode" value="true"/>

```
<Section name="Security">
<Property name="CertificateFindBy" type="X509FindType" value="FindByExtension" />
<Property name="CertificateFindValue" type="string" value="1.2.840.113556.1.8000.2554.57748.52896.21682.18417.45066.8514989.679433.2" />
<Property name="ConnectionBrokerOnlyMode" type="bool" value="true" />
```

Making these changes prevents any connections from stand-alone clients through the Secure Gateway. All users will log in only through a connection broker.

About InTouch Access Anywhere Secure Gateway ports and SSL certificates

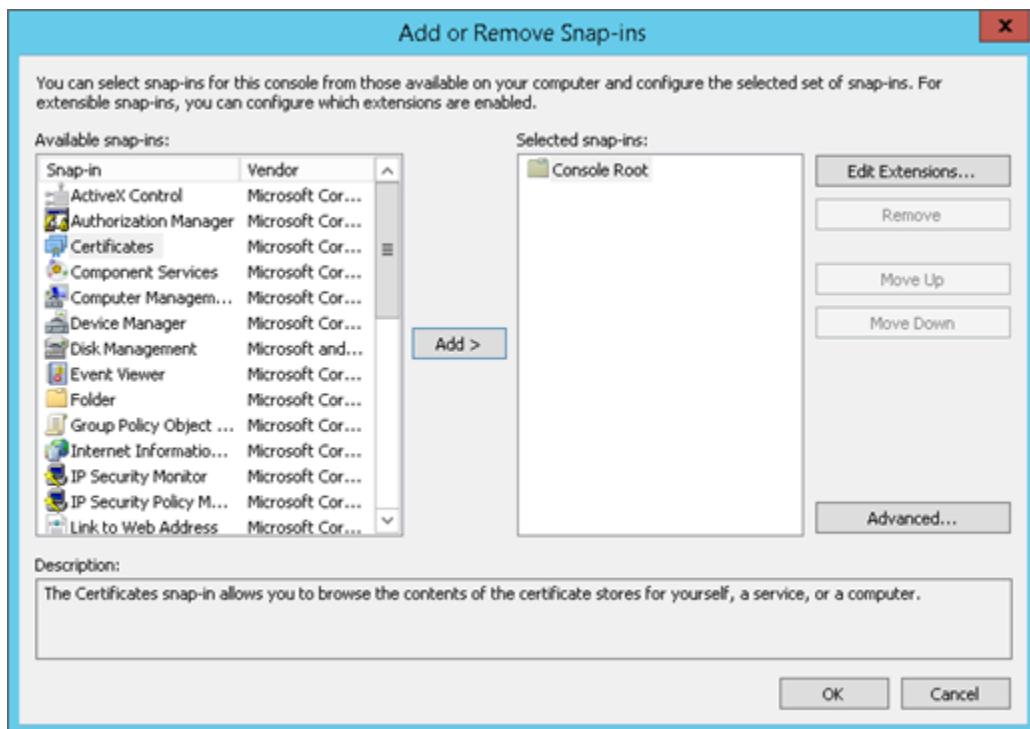
The InTouch Access Anywhere Secure Gateway includes a self-signed certificate. Some web browsers may show a security warning when a self-signed certificate is detected. To remove the warning, install a trusted certificate purchased from a trusted certificate authority (for example, VeriSign).

Important: The signed certificate must have a private key associated with it. A .CER file may not have a private key. Use a signed certificate that includes a private key, which usually has a .PFX extension.

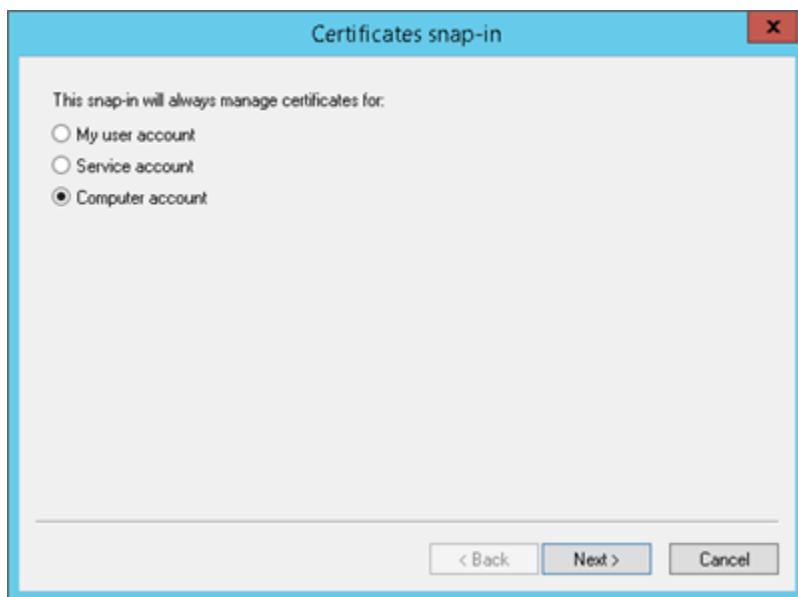
The Secure Gateway uses the certificate in the Windows Certificate Store (Computer Account), which is accessible using the Microsoft Management Console (MMC).

To add, view, or modify certificates

1. Log on as an administrator to the computer hosting the Secure Gateway.
2. From the Windows **Command Prompt**, run the mmc.exe command to show the MMC.
3. Select the **File** option from the menu bar and select **Add Remove Snap-in** to show the **Add or Remove Snap-ins** dialog box.

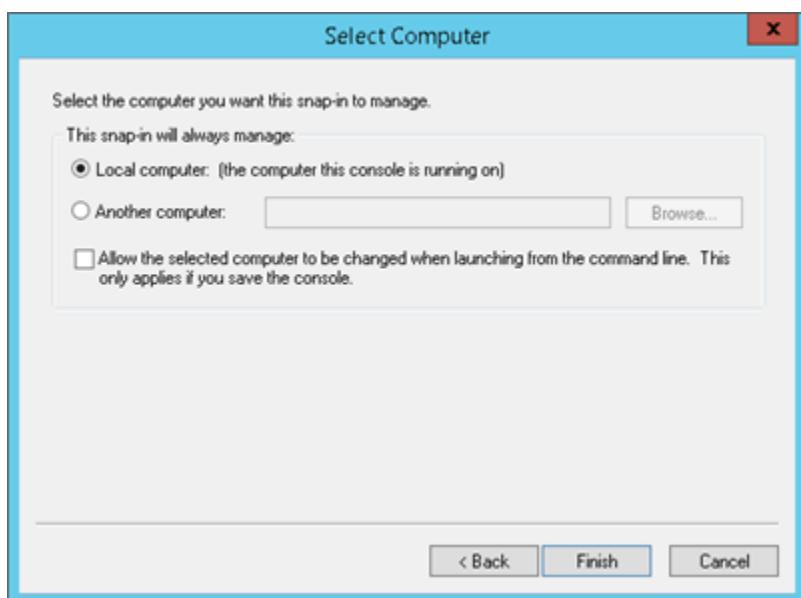


4. Select **Certificates** from the **Available snap-ins** area and select **Add**.
5. Select **Computer Account** from the **Certificates snap-in** dialog box and click **Next**.



The **Select Computer** dialog box appears with options to select a computer account.

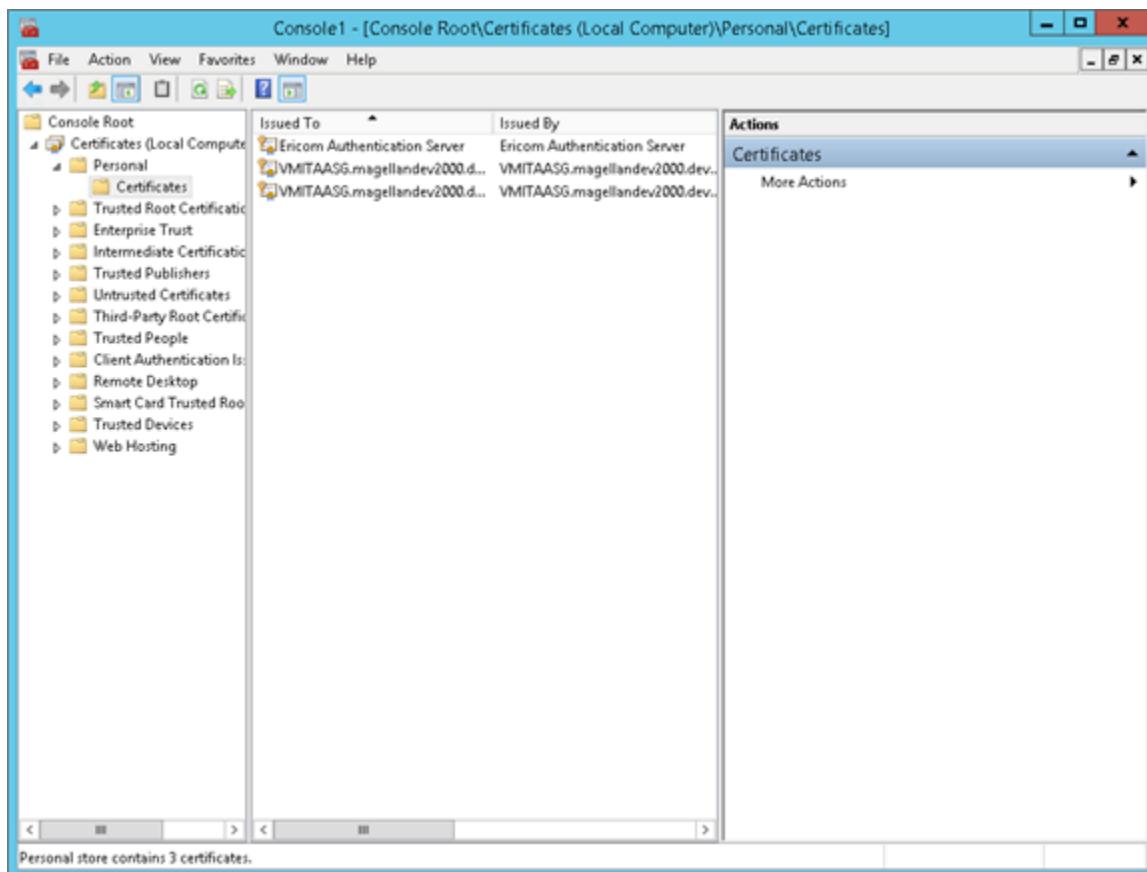
6. Select **Local Computer**.



7. Click **Finish** and then **OK**.

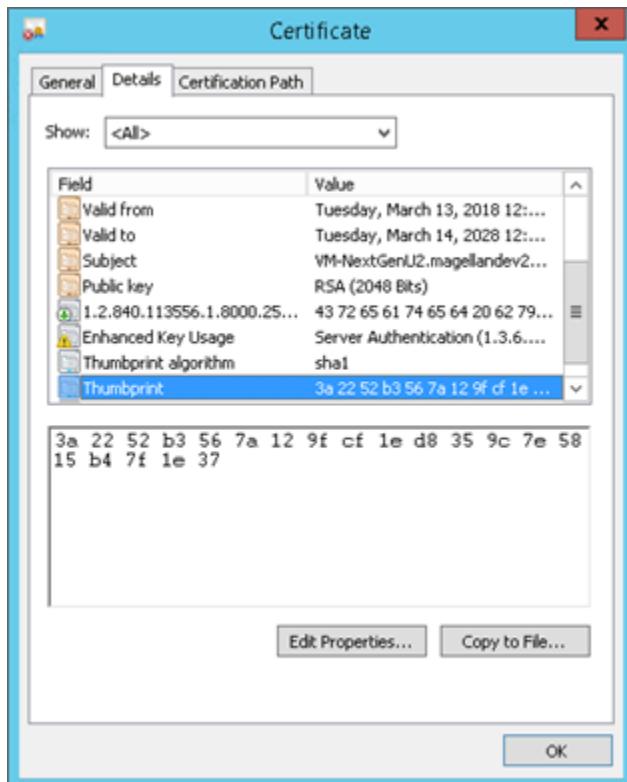
The Console Root shows Certificates (Local Computer) option.

8. Select the icon to the left of the **Certificates (Local Computer)** option to expand the list of sub options.
9. Browse **Certificates | Personal | Certificates** folder to view the available certificates that can be used by the Secure Gateway.



10. If a trusted certificate is used with Secure Gateway, place it in the same location as the Secure Gateway **Certificates | Personal | Certificates**.
11. Browse the **Certificates | Personal | Certificates** folder of the MMC to show a list of certificates.
12. Double-click on the trusted certificate that you want to use with the Secure Gateway.
13. Select the **Details** tab and highlight **Thumbprint**.

The Thumbprint value appears beneath the list of certificate properties.



14. Select the entire thumbprint value.
15. Press CTRL+C to copy it.
The Thumbprint can also be manually typed in.
16. Click **OK** to close the dialog.
17. Open the EricomSecureGateway.Config file, which is located in the following folder of the computer hosting Secure Gateway:
C:\Program Files (x86)\Wonderware\InTouch Access Anywhere Secure Gateway\InTouch Access Anywhere Secure Gateway
18. Locate the Security section of the file.

```
<Section name="Security">
<Property name="CertificateFindBy" type="X509FindType" value="FindByThumbprint" />
<Property name="CertificateFindValue" type="string"
value="3A2252B3567A129FCF1ED8359C7E5815B47F1E37" />
```
19. Ensure the value of the CertificateFindBy property value is set to FindByThumbprint.
20. Delete the existing Thumbprint from the CertificateFindValue property value field.
21. Press CTRL+V to paste the new Thumbprint in the value field of the CertificateFindValue property.

All blank spaces in the thumbprint are removed after pasting it as the value of the CertificateFindValue property.

22. Save the file and the new Thumbprint will be used. Restarting the Secure Gateway service will apply the new certificate immediately.

Note: The DNS address of the Secure Gateway server must match the certificate name. If it does not, a "Connection failed" error message will appear upon attempting a connection.

Add a certificate thumbprint to the EricomSecureGateway.Config file

Another way to add a trusted certificate thumbprint is to dump the certificate values and copy the identify thumbprint to the EricomSecureGateway.Config file. When you are using the extension identity of a certificate, the CertificateFindBy property value of the EricomSecureGateway.Config file should be set to "FindByExtension".

To add a certificate thumbprint to the EricomSecureGateway.Config file

1. Place a X.509 certificate at a known location of the computer running Secure Gateway.

2. Open a Command prompt window and enter the certutil command in the following form:

C:\Temp>certutil -dump CertificateName.cer

where CertificateName is the actual name of the certificate.

3. Find the Certificate Extensions 2 location of the output from the certutil command.

The identify string appears immediately beneath Certificate Extensions 2

Certificate Extensions: 2

1.2.840.113556.1.8000.2554.57748.52896.21682.18417.45066.8514989.679433.2: Flags = 0,
Length = 1a

4. Copy the identity string.

5. Edit the EricomSecureGateway.Config file and locate the Security section of the file.

```
<Section name="Security">
<Property name="CertificateFindBy" type="X509FindType" value="FindByThumbprint" />
<Property name="CertificateFindValue" type="string" value=
```

6. Change FindByThumbprint to FindByExtension and copy the identity string as the value of the CertificateFindBy property.

```
<Section name="Security">
<Property name="CertificateFindBy" type="X509FindType" value="FindByExtension"/>
<Property name="CertificateFindValue" type="string" value=<Certificate Identity
Thumbprint Goes Here> />
```

Example:

```
<Section name="Security">
<Property name="CertificateFindBy" type="X509FindType" value="FindByExtension"/>
<Property name="CertificateFindValue" type="string"
value="1.2.840.113556.1.8000.2554.57748.52896.21682.18417.45066.8514989.679433.2"/>
```

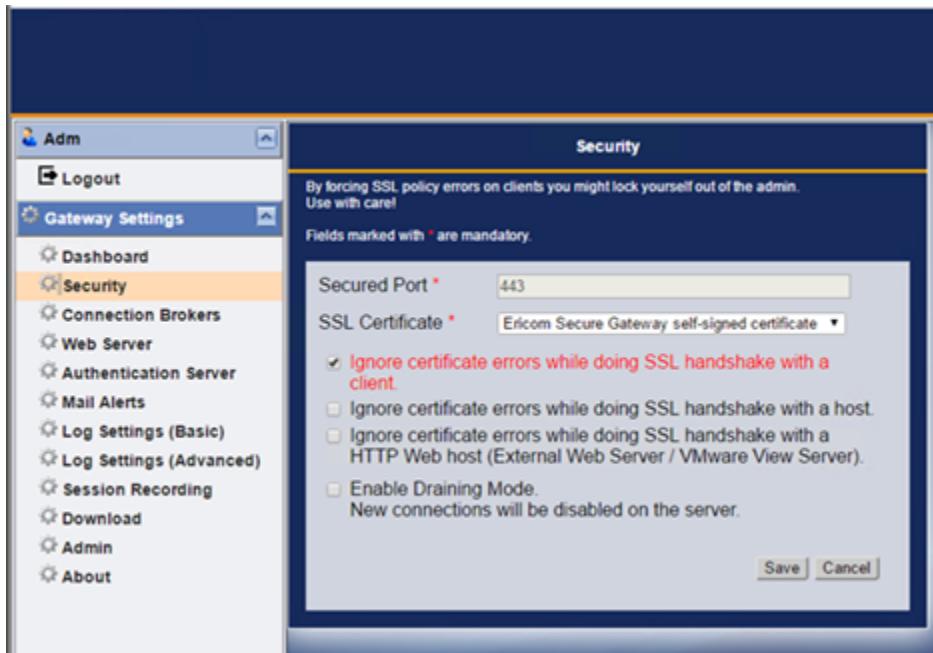
Configure a secured port and SSL certificate

In the Secure Gateway Configuration dashboard, select the **Security** page to modify the port and SSL certificate that will be used by the Secure Gateway.

Note: Before configuring the port, make sure it is not currently in use.

From the **SSL Certificate** field, select the desired SSL certificate to be used by InTouch Access Anywhere Secure Gateway. It is strongly recommended to use a trusted certificate when the InTouch Access Anywhere Secure Gateway is used in production. Verify whether the selected certificate is trusted.

Configure the desired security options. The **Ignore certificate errors while doing SSL handshake with a client** option is selected by default.



The options are as follows:

Note: The recommended deployment method is to have none of the above options checked. The most secure method is to leave all security features enabled.

- **Ignore certificate errors while doing SSL handshake with a client:** this option refers to the certificate verification between any clients connecting to the InTouch Access Anywhere Gateway. If this option is checked, you will not be prompted with a certificate error if a connecting client's trusted certificate is not recognized on the Access Anywhere Gateway node.
- **Ignore certificate errors while doing SSL handshake with host:** this option refers to the certificate verification between the Access Anywhere Gateway and the Access Anywhere Server. If the Access Anywhere Server does not have a trusted certificate recognized on the Access Anywhere Gateway node, the connection will be rejected. If this option is checked, you will not be prompted with a certificate error if the Access Anywhere Server's certificate is not recognized on the Gateway node.
- **Enable Draining Mode:** this option will disable new connects being made to the Access Anywhere Server, and old connects will be closed out.

Configure failover gateways

Multiple Secure Gateways can be configured as a failover chain in the **InTouch Access Anywhere** web client. A failover chain provides improved reliability with redundant Secure Gateways. Alternate Gateways automatically become active when the primary Gateway is unavailable. If the connection to the first Secure Gateway on the list fails, the request is redirected to the next server on the list.

To specify a failover list of Secure Gateways, enter each gateway address separated by a semicolon.

The following list of servers:

Us-bl2008r2;securegateway.domainname.com;192.168.0.3:4343

- The primary gateway is Us-bl2008r2 over port 443.
- The second Secure Gateway is securegateway.domainname.com over port 443.
- The third Secure Gateway is 192.168.0.3 over port 4343 (any port value other than 443 needs to be explicitly specified).

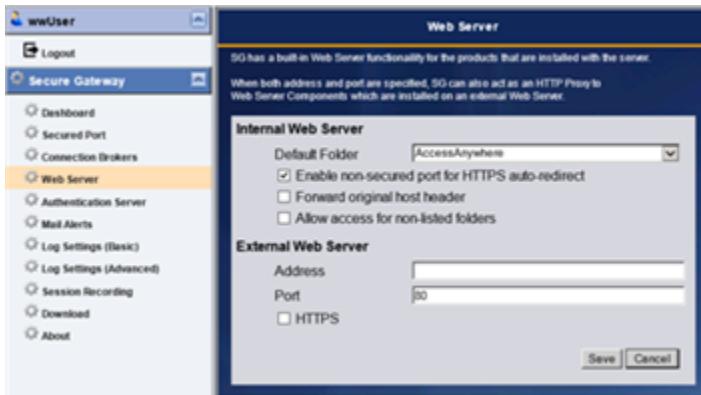
Note: Maintain uptime for the servers at the front of the list to ensure the fastest logon time. If the primary server is unavailable, end-users will experience delays as the log on process must wait for the primary server to time out before attempting to connect to a failover server.

Configure the built-In InTouch Access Anywhere web server

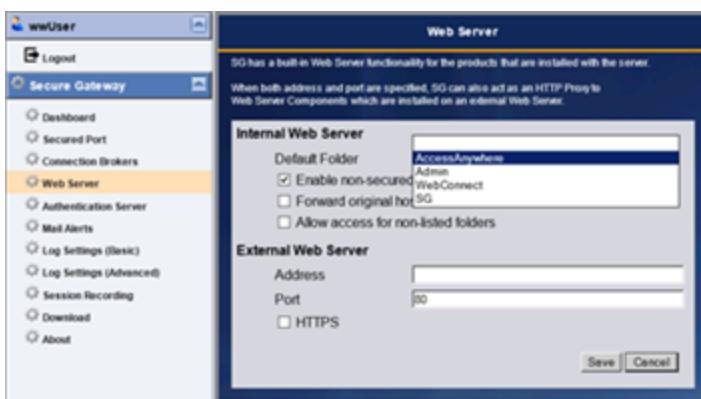
The Secure Gateway has a built-in web server to host web pages for InTouch Access Anywhere. The built-in Web server cannot be disabled and always listens on the Secure Gateway port.

To configure the Web server

1. Open the **Configuration** tool and show the **Web Server** page.



2. Click the **Default Folder** drop down list to select the default URL for the built-in web server.
3. Click **Save**.



When the user goes to the root path of the URL, the selected component will be used. For example, if InTouch Access Anywhere Server is selected, when the user navigates to <https://<sg-server-address>:<port-number>/> the URL will automatically redirect to:

<https://<sg-server-address>:<port-number>/AccessAnywhere/start.html>

Note: The Secure Gateway could technically be used to host non-related pages, but this is not officially supported. Hosted web pages should be of basic static content.

About the InTouch Access Anywhere Secure Gateway web server proxy

The InTouch Access Anywhere Secure Gateway also has a built-in Web server proxy.

Note: Using the Secure Gateway to proxy to pages other than InTouch Access Anywhere is not officially supported.

Connect to an InTouch Access Anywhere server through the Secure Gateway web server

To connect to an InTouch Access Anywhere server available through the Secure Gateway Web server, open a browser and navigate to the desired URL. If a port other than 443 is being used by the Secure Gateway, it must be explicitly stated in the URL. For example: <https://myserver:4343/AccessAnywhere/start.html>

The following URLs are available by default.

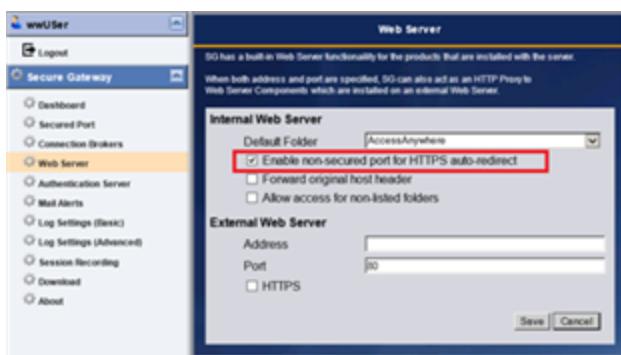
Secure Gateway Welcome Page	https://server:port/ or https://server:port/welcome.html
InTouch Access Anywhere Server	https://server:port/AccessAnywhere/start.html

Enable non-secured port for HTTPS auto-redirect

The InTouch Access Anywhere Secure Gateway Web server listens on port 80 by default. This way, HTTP references to the server will automatically redirect to the HTTPS URL.

Note: This feature only works if the Secure Gateway is listening on port 443. If it is configured to use any other port, the HTTP automatic redirect is not supported.

To enable this feature, select the option: **Enabled non-secured port for HTTPS auto-redirect** (see below).



Configure this feature in the EricomSecureGateway.exe.Config file using: <add key="EnableNonSecuredPortForHttpsAutoRedirect" value="false" /> in the <appSettings> section.

Disable HTTP/HTTPS filtering

Certain types of network traffic are blocked by firewalls. Port 443 on most firewalls is initially reserved for HTTP (and HTTPS) based communication. Most firewalls have a rule to filter out any non-HTTP data. Depending on what the Secure Gateway will be routing, HTTP filtering may need to be disabled on the firewall.

The Secure Gateway can proxy various types of traffic. Some are HTTP based and some are not. The only configuration where HTTP filtering does not need to be disabled is when the Web Application Portal and InTouch Access Anywhere are used together.

This table denotes the protocol used by a connection method:

Communication Type	Protocol Used
Web Application Portal login	HTTP/HTTPS
Application Zone login	TCP
InTouch Access Anywhere RDP session	HTTPS (Secure Gateway required)

Advanced configuration using the EricomSecureGateway.Config file

Back up the current EricomSecureGateway.Config file before making any changes.

To configure the settings of the built-in Web server, open the EricomSecureGateway.Config file using a text editor. Each folder in the WebServer directory may have a default document assigned for it, and may also be restricted so that end users cannot access it.

Name	Date modified	Type	Size
AccessAnywhere	1/3/2018 7:59 PM	File folder	
Admin	1/3/2018 7:59 PM	File folder	
Blaze	1/3/2018 7:59 PM	File folder	
SG	1/3/2018 7:59 PM	File folder	
welcome.html	12/29/2017 10:50 ...	HTML File	1 KB

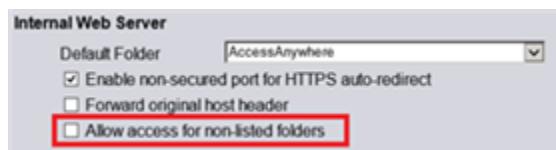
For example, the settings below will configure the following:

- Sets the View folder as the default folder
- Sets the view.html as the default document for the View folder
- Restricts access to any unlisted folders in the directory
- Prohibits access to the Blaze and MyCustom folders.

```
<>Section name="InternalWebServer">
<Property name="Enabled" type="bool" value="true" />
<Property name="ForwardOriginalHostHeader" type="bool" value="false" />
<Property name="ForwardFaviconRequest" type="bool" value="false" />
<Property name="XFrameOptions" type="string" value="" />
<Property name="ContentSecurityPolicy" type="string" value="" />
<Property name="AccessControlAllowOrigin" type="string" value="*" />
<Property name="ClientSessionCookieTimeoutMinutes" type="int" value="60" />
<Property name="AllowAccessForNonListedFolders" type="bool" value="false" />
<Property name="DefaultFolder" type="string" value="AccessAnywhere" />
<Property name="FolderList" type="list(WebServerFolder)">
<Value>AccessAnywhere,start.html,True</Value>
<Value>Blaze,blaze.zip,True</Value>
<Value>Admin,login.html,True</Value>
<Value>WebConnect,start.html,True</Value>
<Value>SG,,True</Value>
</Property>
</Section>
```

Prevent access to non-listed folders

Additional subfolders can be added to the Secure Gateway WebServer folder. These folders can be accessible, even if they are not listed in the internal WebServerSettings list. To prevent access to folders that are not explicitly defined in the internalWebServerSettings list, clear the Allow access for non-listed folders (or set allow_access_for_non_listed_folders="false").

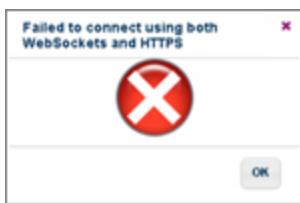


InTouch Access Anywhere Secure Gateway known limitations

This topic describes a number of known behaviors and limitations of Secure Gateway. Refer to InTouch Access Anywhere ReadMe for a more complete list of current known issues in Secure Gateway.

About InTouch Access Anywhere certificate error messages

Most web browsers require a trusted certificate to establish an encrypted network session. If you see an error message similar to the following figure, there could be a problem with the certificate on the InTouch Access Anywhere Secure Gateway server.



If this error message appears, check the address that is being used for the InTouch Access Anywhere Secure Gateway. If it is an IP address, like the image shown below, it may pose a problem.



Rather than using the IP address, use the domain name that matches a trusted certificate that has been configured in the InTouch Access Anywhere Secure Gateway.

For example, instead of using 192.168.1.111, use its domain name: sg.test.com.

In addition, install a trusted certificate on the InTouch Access Anywhere Secure Gateway that matches sg.test.com or *.test.com

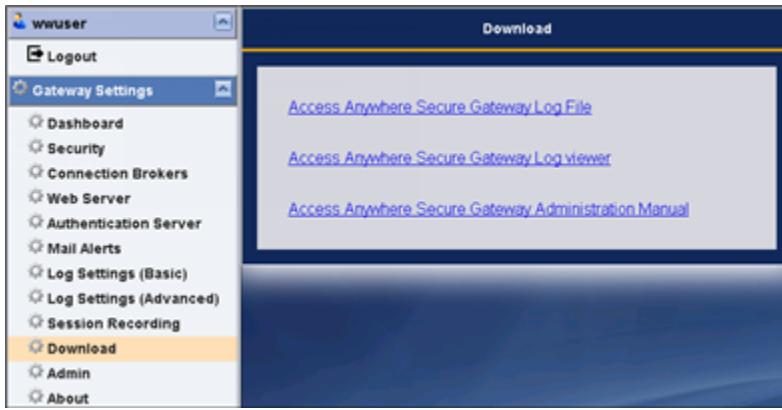
About InTouch Access Anywhere Secure Gateway log files

If you require technical support for the Secure Gateway, log files may be requested.

Note: The logs require a special viewer, which can be downloaded from the [Download](#) page

The current log file is accessible using the **Configuration** page under the **Download** tab. The actual diagnostic information saved in the log file can be set under the two log pages (Log Settings - Basic and Log Settings-Advanced).

Consult with a support engineer on which settings to enable.



About using InTouch Access Anywhere

This set of topics describe how to use InTouch Access Anywhere to remotely connect to your InTouch ViewApps by means of an HTML5-compatible web browser running on your desktop computer or mobile devices like tablets and cell phones. Information is provided for the different methods to log on to a Viewapp with InTouch Access Anywhere and how to use controls to manage a running Viewapp.

About InTouch Access Anywhere Supported Browsers

HTML5 websocket technology makes it possible to open an interactive communication session between your web browser and an InTouch Access Anywhere server. The following versions of HTML5 compatible web browsers have been tested and validated to work with InTouch Access Anywhere:

- Microsoft Edge
- Microsoft Internet Explorer 11
- Firefox version 38 and later
- Safari version 5 and later
- Chrome version 12 and later
- Opera version 11 and later

Although InTouch Access Anywhere supports the listed browsers, you should review the [About running InTouch Access Anywhere on different operating systems](#) for unique behaviors and any alternative solutions for each browser.

Note: The browsers listed above have been tested only on PCs, tablets and SmartPhones. No browser testing has been done on other devices like smart TVs, eBook readers, or game consoles.

Older versions of the listed browsers below are functionally compatible with InTouch Access Anywhere. You may be able to use the following older versions of these browsers, but specific behaviors are unknown because no formal testing has been done with InTouch Access Anywhere.

- Apple Safari, versions 5, 6, and 7
- Google Chrome, versions 11 and later
- Microsoft Internet Explorer, versions 10
- Mozilla Firefox, versions 6 and later
- Opera, versions 11 and later

Important: You are strongly advised to upgrade to a recent version of a supported web browser for use with InTouch Access Anywhere.

About InTouch Access Anywhere and ViewApps

InTouch Access Anywhere can view AVEVA OMI ViewApps created by System Platform 2017 or later.

Contact your IT department to verify the AVEVA OMI ViewApp you want to view is from one of the supported versions System Platform.

View an application with InTouch Access Anywhere

This set of topics describes how to log on to InTouch Access Anywhere from a supported device with an HTML5-compatible web browser. It includes the information you need to log on, a step-by-step log on procedure, advanced connection settings, and how to securely connect to InTouch Access Anywhere.

About information needed to sign on to InTouch Access Anywhere

Before you attempt to log on and begin viewing applications or ViewApps with InTouch Access Anywhere, contact your IT department to gather the following information:

- Has your IT department added you as a member of the Remote Desktop Users group on the InTouch Access Anywhere server? You must be a member to access the InTouch Access Anywhere server with a Remote Desktop Connection client.
- What is the domain name or IP address of your InTouch Access Anywhere server?
- Does your InTouch Access Anywhere server use default port 8080? If not, you need to specify the non-default port number when you log on to the InTouch Access Anywhere server.
- What is the name of the Windows domain specified with your user name to log on to the InTouch Access Anywhere server?
- Verify that your IT department has listed the InTouch applications you want to view on the InTouch Access Anywhere logon page.

Information Needed for an External Connection to InTouch Access Anywhere

- Does your company require a VPN to connect to the InTouch Access Anywhere server from a remote location? InTouch Access Anywhere is compatible with most VPNs that support web sockets.
- Will your data center use an InTouch Secure Gateway with InTouch Access Anywhere? If so, what is the domain name or IP address of the server hosting your Secure Gateway?
- If your data center uses an InTouch Secure Gateway, does it listen on default port 443? If not, you need to specify the non-default port number when you log on to the Secure Gateway.
- Is a trusted certificate installed on the Secure Gateway? By default, a self-signed certificate is installed on the Secure Gateway. Some web browsers show a security warning or prohibit connections to servers using self-signed certificates.

Sign on to InTouch Access Anywhere

You can log on to the InTouch Access Anywhere server by several methods based on whether your company uses a Secure Gateway or not.

Note: If you have any trouble connecting to a remote InTouch Access Anywhere node, see [Troubleshoot sign on problems](#) for help.

Sign on directly to InTouch Access Anywhere

Use the following procedure to log on directly to the InTouch Access Anywhere server. You should be using a VPN for a secure connection from an external network.

To log on directly to the InTouch Access Anywhere server

1. Using your web browser, go to

http://ITAA_Server_Node_Name:8080/

or

http://ITAA_Server_IP_Address:8080/

Note: You may need to enter a different port number if your company changed the default port number assigned to the Access Anywhere server.



The screenshot shows a mobile-style logon screen for InTouch Access Anywhere. At the top, it says "InTouch Access Anywhere". Below that are three input fields: "User name" with a user icon, "Password" with a lock icon, and "Application name" containing the text "InTouchOMI_Control_Room_001" with a dropdown arrow. At the bottom is a large "Connect" button.

A logon page appears with fields to enter your user name, password, and select the InTouch application you want to view.

Connection Details	Description
User Name	Your user name to log on to the Remote Desktop Service (RDS) host. It can optionally contain a domain specification, for example, domain\user. If a domain name is not specified, you will be prompted for it by the RDS host.
Password	Your password to log on to the RDS host. When not specified, you will be prompted for a password by the RDS host.
Application name	A pull down list shows InTouch applications accessible from the InTouch Access Anywhere server that can be selected for viewing.

2. Enter your user name and password.
3. Select the InTouch application you want to view from the drop-down list of the **Application Name** field.
4. Tap or click **Connect** to initiate an InTouch Access Anywhere browser session.

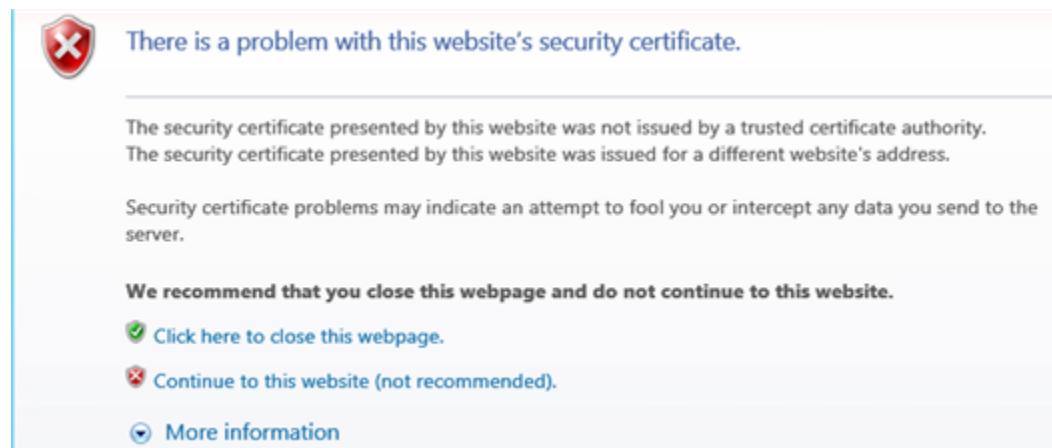
Your initial view will show the InTouch application within the boundaries of your web browser window, which

is the **Fit to browser** Advanced Settings option.

- For more information about setting your InTouch application view, see [Advanced connection settings: display page](#).
- For more information about changing the size of an InTouch application window using the **Fit to browser** setting, see [Resize an application window](#).

Sign on through a Secure Gateway

By default, the Secure Gateway includes a self-signed certificate. Some browsers, such as Google Chrome, allow self-signed certificates for SSL-encrypted websocket connections. Other web browsers like Internet Explorer show a warning message that the server certificate is not signed and prompt you to continue only at a significant risk.



The screenshot shows a browser warning message. At the top, there is a red shield icon with a white 'X' and the text "There is a problem with this website's security certificate." Below this, a detailed message states: "The security certificate presented by this website was not issued by a trusted certificate authority. The security certificate presented by this website was issued for a different website's address." It continues: "Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server." At the bottom, there are three options: a green checkmark next to "Click here to close this webpage.", a red 'X' next to "Continue to this website (not recommended).", and a blue downward arrow next to "More information".

Chrome OS and Safari web browsers prohibit connections using a self-signed certificate. In order to provide connectivity for these browsers, your IT department must import a trusted certificate into the InTouch Access Anywhere server or into the Secure Gateway.

If you encounter certificate problems trying to log on to the Secure Gateway, consider changing web browsers or ask your IT department to install a trusted certificate.

Use the following procedure to log on to the Access Anywhere server through a Secure Gateway.

To log on through a Secure Gateway

1. Using your web browser, go to

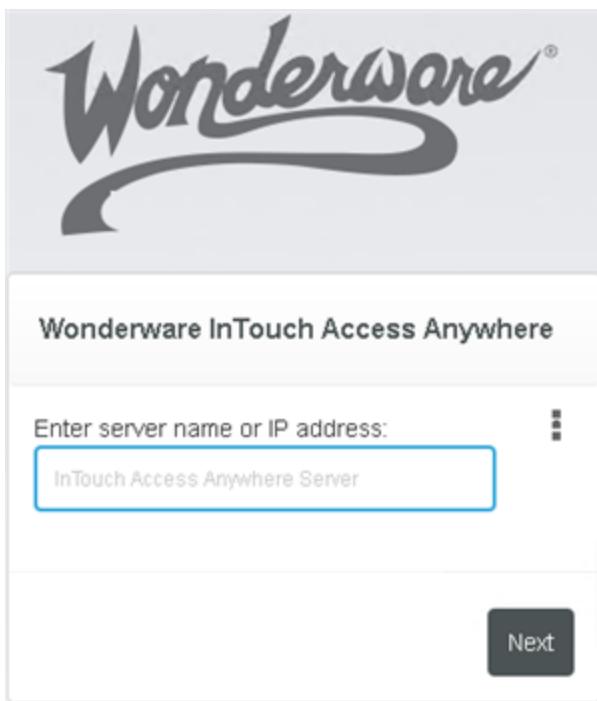
https://Secure_Gateway_Node_Name:443/

or

https://Secure_Gateway_IP_Address:443/

Note: You may need to enter a different port number if your company changed the default port number assigned to the Secure Gateway.

The Secure Gateway start page appears with a field to enter the domain name or IP address of the Access Anywhere server.



2. Enter the IP address or the domain name of the RDP host computer where the InTouch Access Anywhere server is installed.
3. Click or tap **Next**.

The InTouch Access Anywhere server logon page appears with fields to enter your user name, password, and select a ViewApp you want to view.

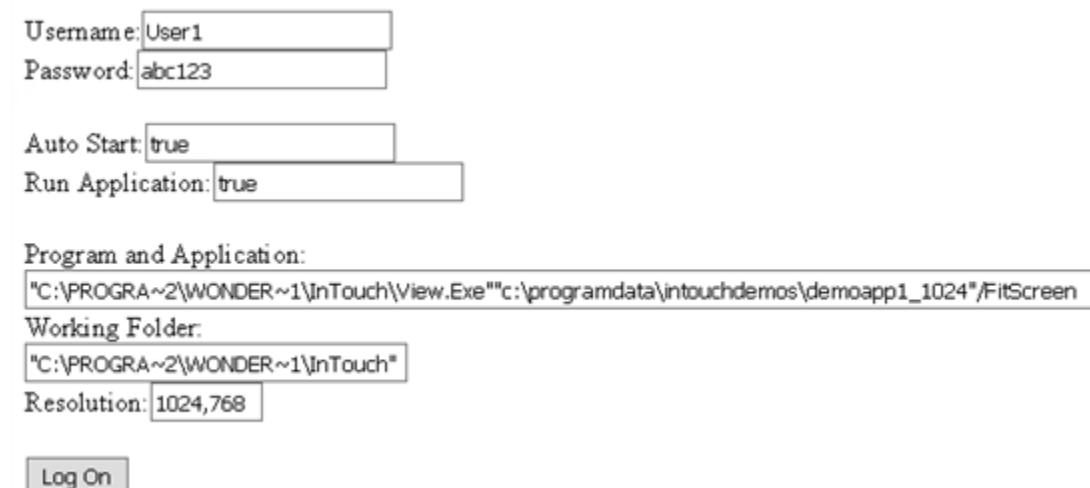
4. Enter your user name and password.
Enter the domain name of your RDS host account in front of your user name if not specified as an Advanced Setting.
5. Select the ViewApp you want to view from the drop-down list.
6. Tap or click **Connect** to initiate an InTouch Access Anywhere browser session.

Sign on with single sign-on

InTouch Access Anywhere supports Single Sign-On (SSO). When using a third-party authentication entity (such as an SSL VPN) that supports SSO Form Post, you only need to sign on once to an InTouch Access Anywhere session using your authentication credentials to begin viewing an InTouch application.

Typically, your IT administrator will send you an HTML file, which is an SSO form that you place on your mobile device or desktop computer. You view the file with an HTML5 web browser.

The following illustration shows a sample SSO log on form with two mandatory fields to enter your username and password. The form may contain other optional fields too based upon the viewing options that your IT administrator wants to grant users to view an application.



The image shows a screenshot of an SSO log on form. It consists of several input fields and a button. The fields are labeled with their respective names and values:

- Username: User1
- Password: abc123
- Auto Start: true
- Run Application: true
- Program and Application: "C:\PROGRA~2\WONDER~1\InTouch\View.Exe""c:\programdata\intouchdemos\demoapp1_1024"/FitScreen
- Working Folder: "C:\PROGRA~2\WONDER~1\InTouch"
- Resolution: 1024,768

Below the fields is a blue rectangular button labeled "Log On".

After clicking **Log On** to submit the SSO form, your web browser should appear with a running ViewApp.

Troubleshoot sign on problems

If you have trouble remotely connecting to your InTouch Access Anywhere server, verify network connectivity and your web browser version by viewing the InTouch Access Anywhere demonstration site:

<http://intouchaccessanywhere.com>

Unable to reach the InTouch demonstration site

If you cannot access the InTouch demonstration site, verify your network connectivity:

- From a desktop computer, use the Ping command to verify the connection to the Secure Gateway or Access Anywhere servers. Third party tools exist for some mobile devices to provide equivalent functionality.
- From a Windows computer, try logging on to the InTouch Access Anywhere server directly using a standard Remote Desktop Connection client. You must be able to log on using standard Remote Desktop Connection client before you can log on to an InTouch Access Anywhere server.

Unable to view the InTouch demonstration application correctly

If you can access the demonstration site but not view the InTouch application correctly, verify your web browser.

- Are you using a supported InTouch Access Anywhere web browser? For a list of supported browsers, see [About InTouch Access Anywhere Supported Browsers](#).
- Browser extensions or add-ons may inject JavaScript code into web pages. If the InTouch application is inaccessible or not working properly, disable or uninstall any active browser extensions or add-ons. Then, restart your web browser to ensure the extensions or add-ons are no longer active.

Able to view the InTouch demonstration application

If the demonstration site appears and you can successfully launch the InTouch demonstration application, you have network connectivity and your browser is compatible with InTouch Access Anywhere.

- If you cannot reach the InTouch Secure Gateway by its domain name, try using its IP address.
- If you cannot reach your InTouch Access Anywhere server by its domain name, try using its IP address.

Assistance from your IT department

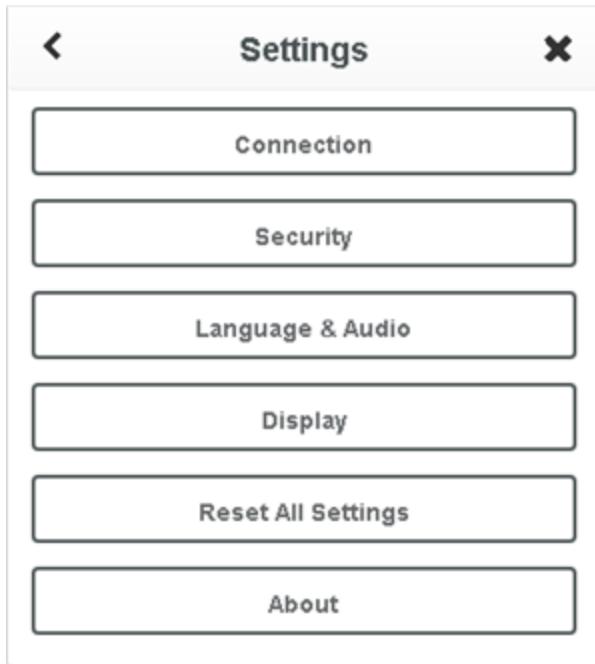
If you are still experiencing logon problems, contact your IT department to get more information about the following:

- Ask your IT department if the port numbers you entered to log on to the Secure Gateway and Access Anywhere servers are correct. The default ports are 443 for the Secure Gateway and 8080 for the Access Anywhere server.
- Ask your IT department to verify the InTouch Access Anywhere service is running.
- Ask your IT department if you are a member of the Remote Desktop Users group on the InTouch Access Anywhere server? You must be a member to access the InTouch Access Anywhere server with a Remote Desktop Connection client.
- Ask your IT department if a trusted certificate is required for the InTouch Access Anywhere Secure Gateway or the InTouch Access Anywhere Server.

Advanced connection options

Click the  **Settings** icon on the logon page to set additional options for your InTouch Access Anywhere connection.

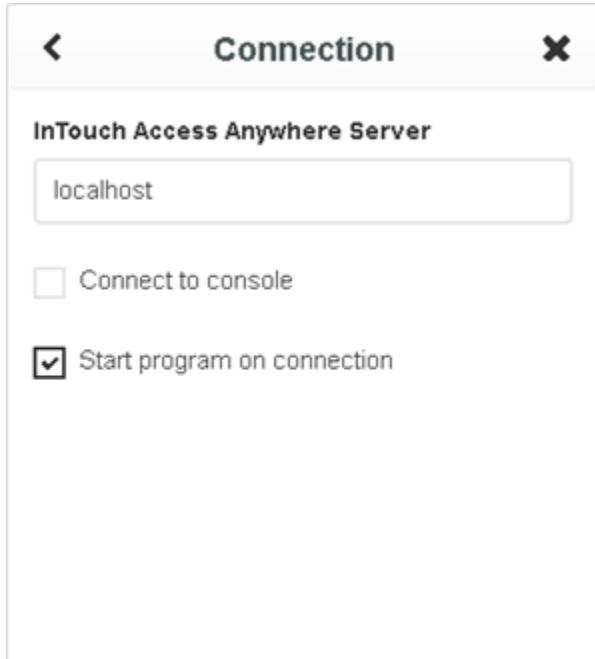
The advanced connection settings are shown on separate web pages by selecting the **Settings** buttons.



Important: You may need information from your IT administrator to specify values for some Advanced Connection settings.

Advanced connection settings: connection page

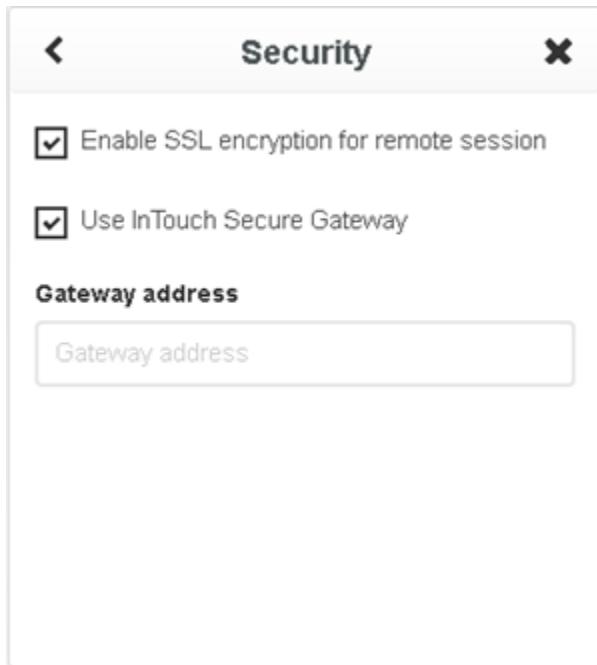
The advanced connections settings on the **Connection** page provide options to specify the name of the InTouch Access Anywhere Server you want to connect to, as well as the option to connect to console.



Settings	Description
InTouch Access Anywhere Server	The name or IP address of the InTouch Access Anywhere Server.
Connect to Console	Connects to a console session of the operating system, instead of a user session.
Start program on connection	This setting cannot be modified. The default value shows the ViewApp you selected running in a web browser.

Advanced connection settings: security page

The Advanced Connection settings on the Security page provide options to enable SSL Encryption and specify the Gateway address.

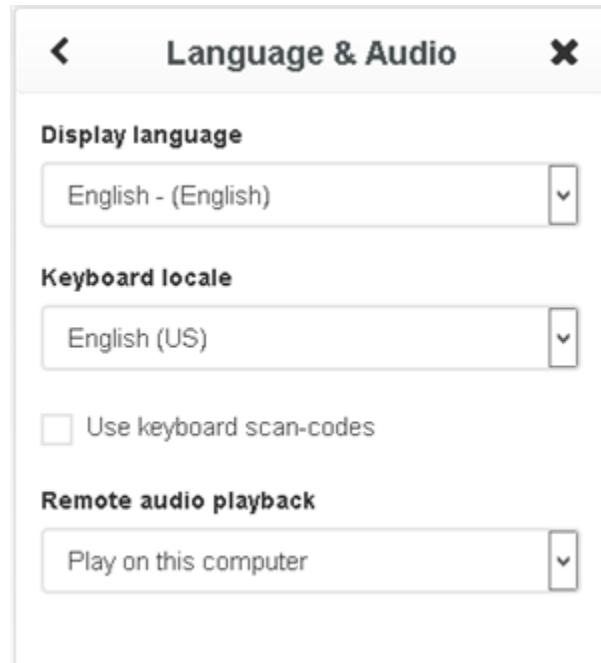


Settings	Description
Enable SSL encryption for remote session	When selected, your web browser uses SSL encrypted websocket communication to the InTouch Access Anywhere Server.
Use InTouch Secure Gateway	When selected, a Secure Gateway connects to the RDP host.
Gateway address	Computer name or IP address of the server hosting the InTouch Secure Gateway. You can enter the address and port for the Secure

Settings	Description
	<p>Gateway in this field. If no port value is specified, 443 is used by default.</p> <p>To specify a custom port, add a colon (:) and the port number to the address. For example, gateway.com:4343.</p> <p>Multiple Secure Gateways can be specified as failover servers. Separate each server address with a comma (,) or semicolon (;).</p> <p>An asterisk (*) means the servers can be specified in any order after the asterisk. For example, if the following is specified: aaa;*;bbb:4433;ccc:4343</p> <p>The Secure Gateway aaa on port 443 is used to initially connect. If aaa is unavailable, then bbb:4433 is used followed by ccc:4343 OR ccc:4343 followed by bbb:4433.</p>

Advanced connection settings: language and audio page

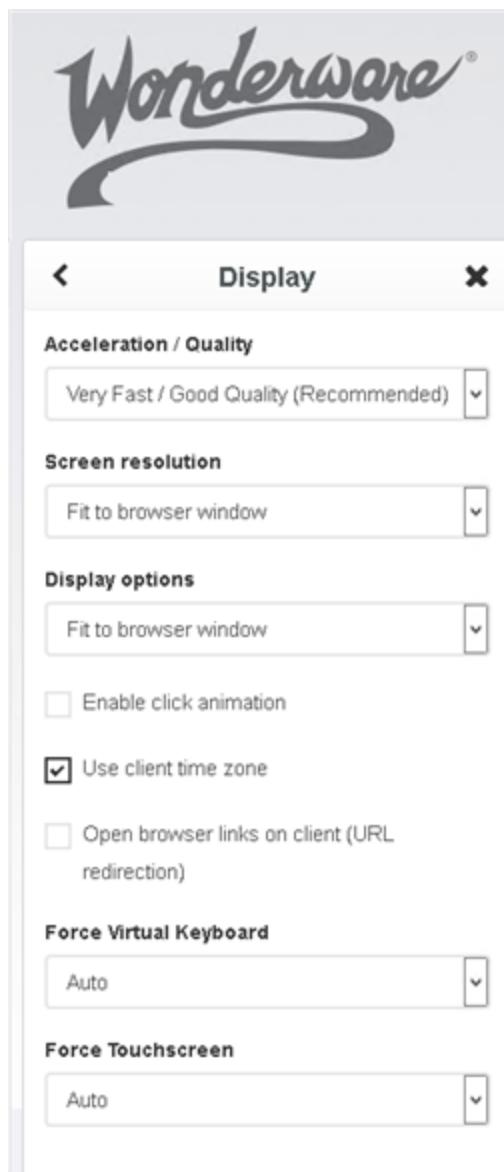
The Advanced Settings Language and Audio Page provides options to set the display language, the keyboard locale and the source of audio playback, if selected.



Settings	Description
Display Language	Changes the language used by the InTouch Access Anywhere Server log on page.
Keyboard locale	Select the keyboard region to be used in an InTouch Access Anywhere session.
Keyboard scan-codes	Enables keyboard scan codes sent by a keyboard to indicate a key has been selected and the alphanumeric character or function associated with the key. Certain applications use scan codes and require this setting to be enabled.
Remote audio playback	Select an option as the source of the session's audio: local computer, remote computer, or do not play audio.

Advanced connection settings: display page

The Advanced Connections settings on the Display Page provide options to set display quality and fit the ViewApp window your browser window or fit to your full screen.



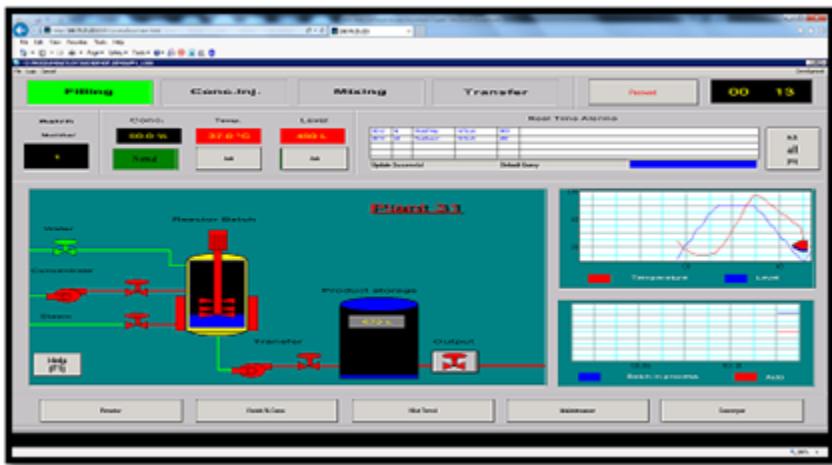
Settings	Description
Acceleration/Quality	Controls the degree of acceleration that is enabled in the session. Faster acceleration results in lower quality application images. Very Fast / Good Quality is the recommended setting.
Screen resolution	The value shown by Screen Resolution matches the selected Display options Fit to browser and Fit to screen values. When Application Resolution is the selected Display options value, the Screen Resolution option will attempt to match the actual application resolution from the drop down list. If a matching resolution is found, the resolution is selected and the Screen

Settings	Description
	<p>Resolution control is disabled. If there is no matching resolution, then the Screen Resolution option is enabled to enter a resolution value.</p>
Display options	<ul style="list-style-type: none"> • Fit to browser window (default) Fit the WindowViewer window within the initial size of your web browser's window. <u>Note:</u> The application's resolution does not change from its initial size if you change the size of your web browser while viewing a ViewApp. <p>When Fit to browser window is selected, the Screen Resolution option is disabled.</p> <ul style="list-style-type: none"> • Fit to screen (Full Screen) Places the entire ViewApp within the current size of the window without showing your web browser's menu bars. • Application Resolution Displays a ViewApp in a browser at its native resolution. If the browser's viewing area does not show the entire application, the user can use the browser's scroll bars to show the rest of application. When this option is selected, InTouch Access Anywhere attempts to match the actual application resolution to a screen resolution shown in the drop down list. If matching resolution is found, then the resolution is selected and the Screen Resolution option is disabled. If there is no matching resolution, then the Screen Resolution control is enabled.
Enable click animation	When enabled, a visual indicator appears on a touch device screen when touch gesture is performed.
Use client time Zone	When selected, this setting enables local time zone redirection. The remote session running on the InTouch Access Anywhere server uses the time of your local computer or mobile device.
Open browser links on client (URL redirection)	When selected, your web browser session is redirected to the URL link launched from the RDS session.
Force Virtual Keyboard	Provides values (Yes , No , Auto) to select a device

Settings	Description
	virtual keyboard while viewing a running InTouch application. Auto is the default, which forces Yes or No based on whether the device displaying the ViewApp has a virtual keyboard or not.
Force Touchscreen	Provides options (Yes , No , Auto) to use touch gestures while viewing a running InTouch application. Auto is the default, which forces Yes or No based on whether the device running the application supports touch gestures or not.

View an InTouch Access Anywhere session

After a successfully logging on, a Remote Desktop Connection (RDC) session is launched and connects you to the selected ViewApp. The ViewApp appears within your browser's window.



While connected, InTouch Access Anywhere intercepts mouse and keyboard events and transmits them to the RDS host. As a result, various keyboard keys and mouse buttons that are usually handled by the browser may behave differently. For example, clicking the F5 button normally causes the browser to reload the current page, but when using InTouch Access Anywhere F5 does not reload the page. Instead, the current page will be transmitted to and handled by the remote InTouch WindowViewer application.

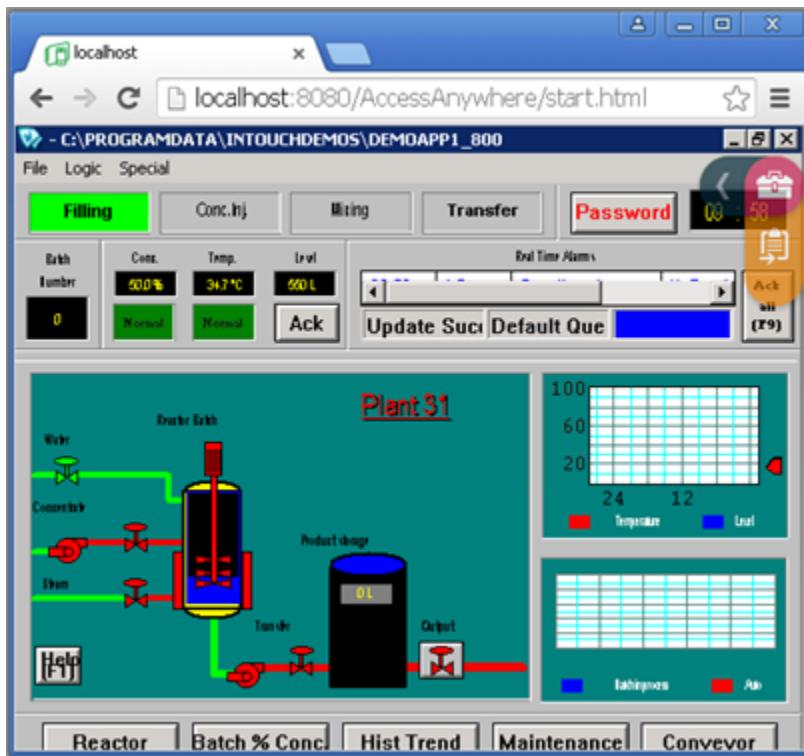
Note: In most browsers, clicking the Back, Forward, or Reload browser buttons will cause InTouch Access Anywhere Server to display a message asking if you wish to leave the current page. If you decide to proceed, the remote session will be disconnected. (Some browsers may not ask for confirmation).

When accessed by InTouch Access Anywhere, a ViewApp cannot be minimized. If the remote InTouch application is shut down, your InTouch Access Anywhere session is automatically terminated in a configurable period set as a parameter of the InTouch Access Anywhere server.

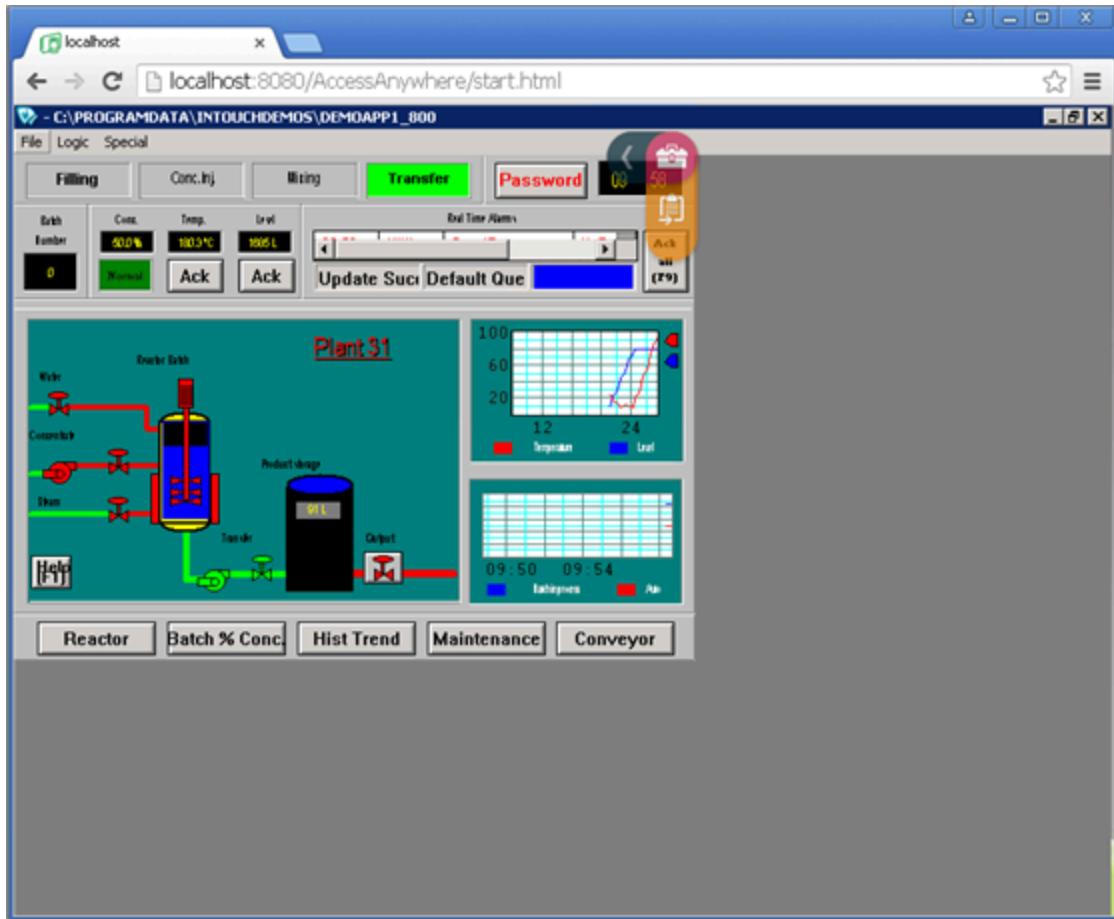
Closing the web browser on your device leaves the RDC session running on the server with the ViewApp still running.

Resize an application window

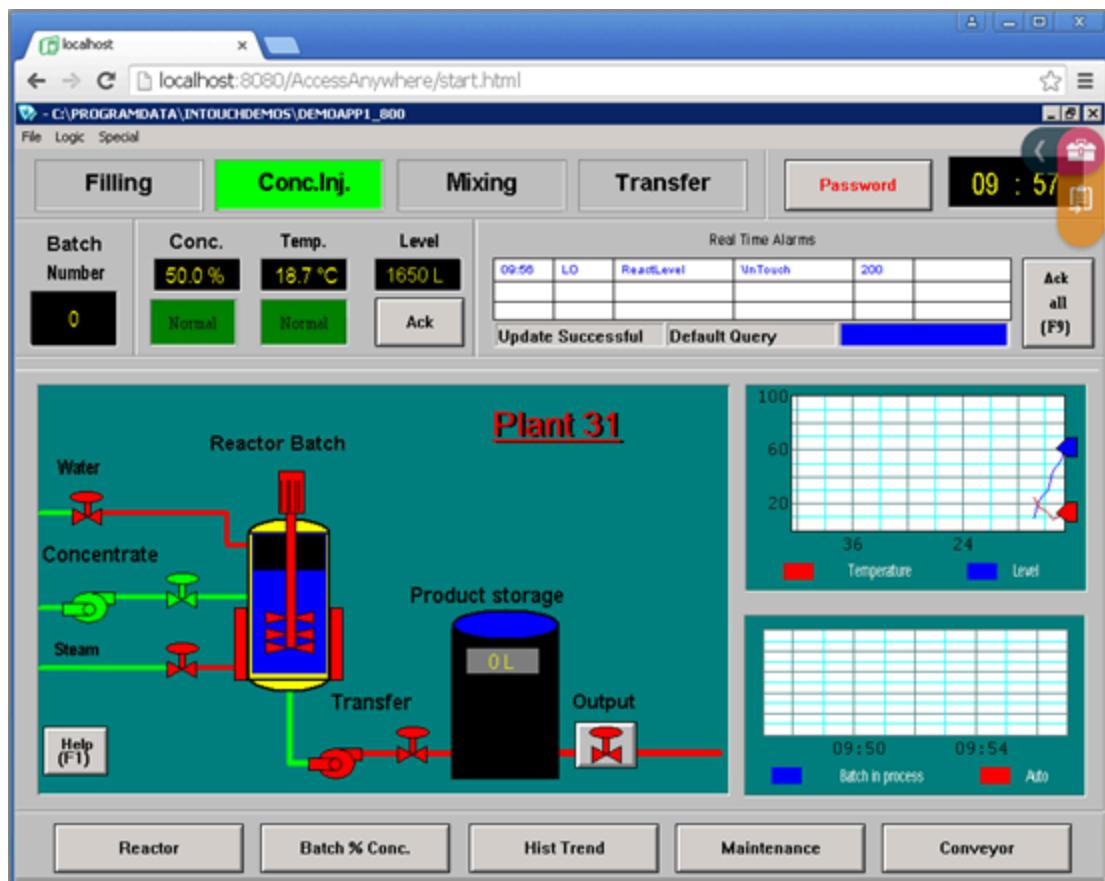
You can change the size of an InTouch application window to fit within your current web browser if you select the **Fit to browser** Display Options setting. The size of the ViewApp window is set to the initial size of your web browser's window when you log on to an InTouch Access Anywhere session.



If you change the size of your web browser window during an InTouch Access Anywhere session, the size of the InTouch application window remains the same as its initial size.



To change the size of a ViewApp window, change the size of your web browser window, close the ViewApp, and disconnect from your InTouch Access Anywhere session. Then, log back on to show your ViewApp at its new size.



Use the Access Anywhere toolbar

tInTouch Access Anywhere provides a toolbar to access a set of features and system keys while a ViewApp is running. The toolbar appears in the browser window and can be moved by dragging and dropping it to another window location.

Tap or click the  toolbar icon once to show a blue arrow to expand the blue bar. The blue bar includes icons that represent system keys and additional functions.

	The gestures help icon appears only when running InTouch Access Anywhere on a touch screen device. Click or tap the help icon to see figures and short explanations of supported finger gestures.
	Click or tap the ESC icon to cancel a value entered in a data entry field.
	Click or tap the TAB icon to shift the window focus to the next data entry field of an application or the next cell of a table.
	Click or tap the Disconnect icon to end the InTouch application browser session and return to the logon

page.



Click or tap the About icon to see information about the specific version of the InTouch Access Anywhere server installed on the RDS host computer.

Tap or click once on the tool box icon to expand the orange bar to show icons that represent keyboard or clipboard functions. The functions shown in the orange bar vary based on whether you are using a computer with a keyboard and mouse or a touch screen device.

Clipboard support

The InTouch Access Anywhere Toolbar provides a feature to copy and paste text or numbers between your local desktop or laptop and a remote InTouch Access Anywhere session using a built-in clipboard.

Note: When using Internet Explorer (MSIE) 10, the Clipboard feature is integrated, so there are no AccessNow clipboard icons. Simply copy and paste text between the local device and Access Anywhere session using Windows copy (Ctrl+C) and paste (Ctrl+V) keyboard shortcuts.

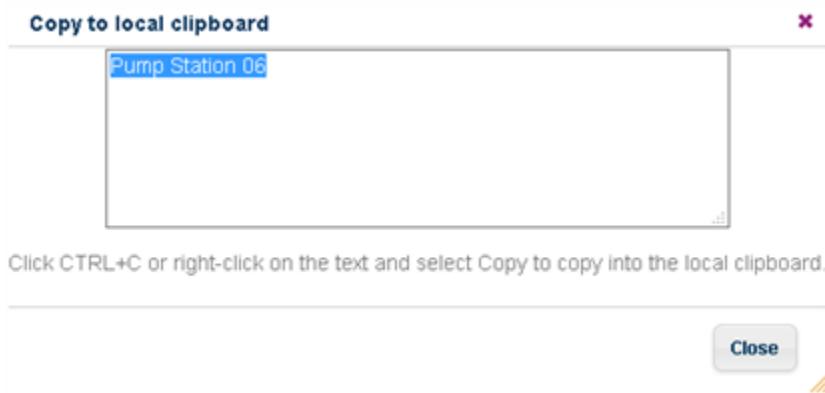
Copy text from a remote InTouch application

The following procedure explains how to copy text from an InTouch application shown in an InTouch Access Anywhere browser session to your local desktop or laptop computer.

To copy text from an InTouch application

1. Select the text or number to copy from an InTouch application running in your InTouch Access Anywhere browser session.
2. Perform a copy keyboard shortcut (Ctrl +C) on the desired text or number.

The **Copy to local clipboard** dialog box appears in your remote browser session.



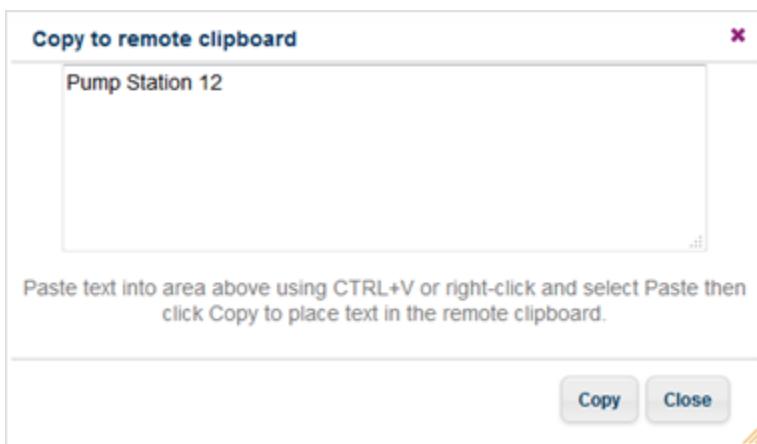
3. Click on the clipboard icon to copy the text to the clipboard.
4. Click **Close**.
5. Open a document or application on your local desktop computer or laptop.
6. Press Ctrl+V to paste the text.

Copy text to a remote InTouch session

The following procedure explains how to copy text from a local desktop or laptop computer to an InTouch application running in a remote Access Anywhere browser session.

To copy text to a remote InTouch application

1. From a local application like Notepad, copy text you want to paste into a remote InTouch application running in an InTouch Access Anywhere browser session.
2. Open the InTouch Access Anywhere browser session to show a running InTouch application.
3. Click the **Open Clipboard** icon on the toolbar.
4. Paste the copied text into the clipboard.



5. Click **Copy**.
6. Set the focus in WindowViewer to a data entry field in the running InTouch application.
7. Press Ctrl+V or right-click and select **Copy** from the shortcut menu to paste the text into the remote InTouch application's data entry field.

Use finger gestures

InTouch Access Anywhere supports control gestures when used on a touch device. The following table shows the different types of gestures you can make on a touch device to emulate mouse events or specific keyboard keys.

Gesture functionality varies across different devices, browsers, and operating systems. Tap the  Gestures help icon in the Toolbar to see the full list of supported gestures. For more information about gesture limitations, see [About device support for InTouch Access Anywhere](#).

Application Behavior or Key	Gesture	Description
Left Mouse Click		One finger tap once
Right Mouse Click		One finger tap and hold
Double Mouse Click		Tap once with two fingers
Double Mouse Click		One finger double tap
Drag Object		One finger tap, hold and drag
Zoom In		Two fingers pinch out
Zoom Out		Two fingers pinch in
Page Up		Three fingers swipe down
Page Down		Three fingers swipe Up

Application Behavior or Key	Gesture	Description
Left Arrow		Three fingers swipe right
Right Arrow		Three fingers swipe Left

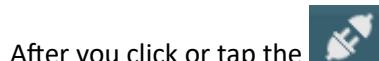
Change an InTouch application shown by WindowViewer

If a ViewApp is currently running in an InTouch Access Anywhere session, another ViewApp cannot be viewed without first logging on to the Access Anywhere server and selecting another ViewApp.

To change an InTouch application shown by WindowViewer

1. Log on to InTouch Access Anywhere that is currently running an InTouch application.
2. Select the **Disconnect** icon from the Access Anywhere toolbar.
3. Click or tap **Yes** to confirm that you want to disconnect your InTouch Access Anywhere session.
You should return to the logon page.
4. Select the ViewApp you want to view from the **Application name** field of the logon page.
5. Enter your user name and password to start another InTouch Access Anywhere session.
After establishing a session, you should see your browser start and show the ViewApp you selected.

End an InTouch Access Anywhere session



After you click or tap the **Disconnect** icon from the Desktop Toolbar, a message appears requesting that you confirm that you want to disconnect from the Access Anywhere server.

After you click **Yes**, the browser closes and you see the InTouch Access Anywhere logon page. No trace of the InTouch Access Anywhere browser session remains on your device after ending a session. But, the RDP session is still running WindowViewer on the InTouch Access Anywhere server.

Session auto-logoff

A session is logged off when a ViewApp is closed. In some cases, a session is not closed immediately or is frozen. InTouch Access Anywhere Server includes an auto-logoff feature that automatically logs off a session if nothing is displayed on the screen for a specified period.

Your InTouch Access Anywhere administrator can set the length of the auto-logoff period with a RDP server or InTouch Access Anywhere configuration parameter.

Automatic reconnect

Active InTouch Access Anywhere sessions automatically attempt to recover from temporary network outages by reconnecting your session. You may experience a slight delay during the reconnect attempt, but after the session is re-established, you can continue working without having to log back on to the session.

About device support for InTouch Access Anywhere

This set of topics describes the runtime behavior of InTouch Access Anywhere on different mobile devices, web browsers, and operating systems.

InTouch Access Anywhere can operate on mobile devices with an HTML5 compliant browser to remotely access and interact with InTouch applications. Many mobile devices use touch gestures to perform the tasks normally done by a mouse using a desktop or laptop computer. Because there is no mouse, touch devices require finger gestures to emulate mouse behaviors.

Also, some mobile devices use built-in software keyboards instead of physical keyboards. Typically, mobile device software keyboards do not have the standard Windows F1-F12, Ctrl, or Alt keys.

About running InTouch Access Anywhere on different operating systems

This set of topics describes specific behaviors of InTouch Access Anywhere by specific combinations of device operating systems and browser combinations. The behaviors are grouped by operating systems. Alternative solutions to issues are provided when available.

Running InTouch Access Anywhere on Android

This table shows the behavior of different browsers on Android devices. For more common behaviors on the Android operating system see [Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations](#) and [Running InTouch Access Anywhere: common behaviors](#).

Behavior Type	Browser(s)	Description
Drag and drop	All	<p>You may experience difficulty dragging and dropping a ViewApp pop-up window to another location.</p> <p>Alternative Solution</p> <p>Use a device stylus to perform drag and drop operations.</p>
Scroll bars	Opera and Chrome	<p>Scroll bars cannot be scrolled by tapping or pressing and dragging the scroll button.</p> <p>Alternative Solution</p> <p>Tap on the gray area of the scroll bar to move a window.</p>

Behavior Type	Browser(s)	Description
Tooltip animations	Opera, Firefox, and Chrome	<p>Tooltip animations configured on Industrial Graphics may not appear with a mouse over action.</p> <p>Alternative Solution</p> <p>Substitute another gesture to implement a mouse-over action for your device and operating system.</p>
Opening an InTouch application	Firefox	<p>Selecting a ViewApp from the Access Anywhere Application Name list box can immediately open the ViewApp without clicking OK.</p> <p>Alternative Solution</p> <p>There is no alternative solution.</p>
Keystrokes over Bluetooth	Firefox	<p>When using a Bluetooth keyboard, a single keystroke may produce two characters.</p> <p>Alternative Solution</p> <p>Use the mobile device's built-in keyboard.</p>
Text display	Chrome	<p>When an Android mobile device runs low on memory, text in a window can appear blurred.</p> <p>Alternative Solution</p> <p>Close all running applications and restart your mobile device before using InTouch Access Anywhere.</p>
Text input	Opera, Firefox, and Chrome	<p>Typing double-byte language characters using the native Android software keypad appear as question marks (?) in a data entry field.</p> <p>Alternative Solution</p> <p>Configure the InTouch Access Anywhere server to use the Windows Keyboard option. Double-byte language characters can be entered correctly from the Android keypad or by using a paired Bluetooth hardware</p>

Behavior Type	Browser(s)	Description
		keyboard.

Running InTouch Access Anywhere on Google Chrome

InTouch Access Anywhere can operate on a Google Chromebook running the Chrome operating system. The following table lists tips when using InTouch Access Anywhere with a Chromebook running the Chrome operating system and web browser.

Function	How to Perform
Mouse Left-click	Click the Chromebook trackpad with one finger.
Mouse Right-click	Click the Chromebook trackpad with two fingers.
Scrolling a document or website	Drag two fingers on the Chromebook trackpad up or down to scroll.
Configure Chromebook	In the address field, enter: chrome://settings.

About Chromebook keyboard commands and InTouch Access Anywhere

The Chromebook keyboard lacks several keys that are used by Windows. ChromeOS provides standard Windows keyboard mappings that use existing keys with the Alt button to represent certain missing keys. InTouch Access Anywhere supports these key combinations:

Command	Chromebook Key Combination
Delete	Alt+Backspace
Page Up	Alt+Up
Page Down	Alt+Down
Home	Ctrl+Alt+Up

Command	Chromebook Key Combination
End	Ctrl+Alt+Down

In addition, InTouch Access Anywhere provides special non-standard mappings for additional key combinations on a Chromebook.

Command	Chromebook Key Combination
F1	Ctrl+1
F2, ..., F12	Ctrl+2, ..., 12
Alt+Tab	Alt+"
Alt+Shift+Tab	Alt+Shift+'
Ctrl+Home	Ctrl+Alt+Left
Ctrl+End	Ctrl+Alt+Right

Running InTouch Access Anywhere on iOS

The following table shows the specific behavior of different browsers running on the Apple iOS operating system. For more common behaviors on the iOS operating system see [Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations](#) and [Running InTouch Access Anywhere: common behaviors](#).

Behavior Type	Browser(s)	Description
Accessing InTouch Access Anywhere	Safari	<p>You may experience difficulty accessing InTouch Access Anywhere if the Safari browser's Private Browsing option is enabled. Private browsing is enabled if Safari bars appear black or dark instead of blue or gray.</p> <p>Alternative Solution</p> <ol style="list-style-type: none"> 1. Open Safari. 2. Select the Pages icon that is shaped as two squares. 3. Select the Private setting and turn it Off.
Accessing InTouch Access Anywhere	Opera Mini on an iOS device	This browser is not a supported. You cannot reach the InTouch Access Anywhere web server. You

Behavior Type	Browser(s)	Description
		can use a supported browser on an iOS device.
Refresh current page	Safari and Chrome	<p>When you click to refresh or navigate away from the current page while connected to the InTouch Access Anywhere server, you will not be prompted to confirm your action, which may result in unwanted page navigation.</p> <p>Alternative Solution</p> <p>There is no alternative solution.</p>
Invoking Direct Push Button	All browsers on iPad or Android tablets	<p>Using a single tap to toggle the state of a push button does not work consistently. A push button can be toggled consistently if the tap gesture is held for three or four seconds.</p> <p>Alternative Solution</p> <p>Modify the ViewApp button to change state by User Inputs>Discrete animation.</p>
Moving between text boxes	Safari and Chrome on iPad	<p>The Next and Previous buttons are not enabled to move between text boxes on dialog boxes with multiple text boxes.</p> <p>Alternative Solution</p> <p>Use a touch gesture to transfer focus to a specific window element.</p>
Text input	Safari on iPad	Typed double-byte language characters do not appear in an InTouch data entry field when using an iPad software keypad.

Running InTouch Access Anywhere on Windows RT

This table shows the behaviors of different browsers on the Windows RT operating system. For more common behaviors on the Windows RT operating system see [Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations](#) and [Running InTouch Access Anywhere: common behaviors](#).

Behavior Type	Browser(s)	Description
Touch gestures	Internet Explorer 10 in a touch environment	<p>You cannot use a touch gesture to open a right-click shortcut menu on the AlarmViewer and AlarmDBView controls.</p> <p>Alternative Solution</p> <p>Access menu options through scripts.</p>
Mouse pointer	Internet Explorer 10 in a touch environment	<p>The mouse pointer may disappear after using the Surface keyboard to enter data.</p> <p>Alternative Solution</p> <p>Use touch gestures in place of mouse inputs.</p>
Pop-up dialog display	Internet Explorer 10 in a touch environment	<p>A pop-up dialog can appear blurred after dragging it to a new screen location.</p> <p>Alternative Solution</p> <p>Refresh the session.</p>
External data subscriptions	Internet Explorer 10	<p>External data subscriptions intermittently do not refresh on starting WindowViewer without user interaction or input. A single touch on the screen will start the data subscription refreshes, except on read-only kiosks and other read-only devices.</p> <p>Alternative Solution</p> <p>Use a timer or clock control to enable user interaction that does not interfere with the InTouch application itself.</p>
Delta deployment	Internet Explorer 10	<p>Window Viewer can stop responding after a delta deploy, especially with applications comprised of a large number of windows.</p> <p>Alternative Solution</p> <p>Close and restart the browser session.</p>

Behavior Type	Browser(s)	Description
Shortcut keys	Internet Explorer 10	<p>Shortcut keys Shift+ and alphabetic characters configured for scripts do not execute the script functions.</p> <p>Alternative Solution</p> <p>Use different combinations of keyboard shortcuts, such as Shift+Ctrl+(alphabetic character).</p>
Unlocking a tablet device	Internet Explorer 10	<p>The browser session may occasionally stop responding when unlocking the tablet on which InTouch Access Anywhere is running.</p> <p>Alternative Solution</p> <p>Close and reopen the browser.</p>
Text input	Internet Explorer	<p>Double-byte language characters cannot be entered into an InTouch data entry field using the Surface software keypad.</p> <p>Alternative Solution</p> <p>Configure WindowViewer on the InTouch Access Anywhere server to use the Windows Keyboard option. Also, double-byte language characters can be entered correctly from the Surface hardware keypad.</p>

Running InTouch Access Anywhere on various operating system, browser, and mobile device combinations

This table shows the behavior of several combinations of operating systems, browsers, and mobile devices.

Behavior Type	Combinations	Description
Arrow keys display	Android, iOS, and WinRT	<p>The background of Up and down arrow keys turn black when you select them in the AlarmDBViewer control. This behavior causes difficulty in using Alarm Viewer controls.</p> <p>Alternative Solution</p> <p>There is no alternative solution.</p>
Selecting an Alarm Viewer Control entry	Android, iOS, and WinRT	<p>A black icon appears when you select any entry in the Alarm Viewer control box and right click to perform a freeze action. This behavior causes difficulty in using Alarm Viewer Controls.</p> <p>Alternative Solution</p> <p>There is no alternative solution.</p>
Key script execution	Safari on iOS or Opera on Android tablets	<p>Key scripts with Ctrl and Alt key combinations do not execute the associated key scripts. The built-in software keyboards in these devices do not have Ctrl nor Alt keys.</p> <p>Alternative Solution</p> <p>Modify your scripted shortcut key combinations to account for this known limitation of tablet device keyboards.</p>
Invoking InTouch keyboard	Safari on iPad or Opera on Android tablets	<p>User input animations do not consistently start the Windows keyboard.</p> <p>Alternative Solution</p> <p>Configure the InTouch application to use the InTouch keyboard if this is an existing application. For new applications, the recommended solution is to use the device's built-in keyboard.</p>

Behavior Type	Combinations	Description
Browser address bar	Chrome on Android or Chrome on iPad tablets	A browser address bar blocks the WindowViewer title bar and menu bar. Alternative Solution Upgrade to latest version of Google Chrome and use browser in Full Screen mode. Older versions of Google Chrome do not support Full Screen mode. Adjusting the browser to use a higher resolution may help to make the menu bar viewable.
Function keys	Safari on iOS Opera on Android tablets	iOS and Android keyboards do not support function keys F1 to F12, or Ctrl and Alt key combinations. Alternative Solution Use different keyboard shortcuts.
Invoking InTouch keyboard	Safari on iPad or Opera, Firefox, and Chrome on Android tablets	The InTouch keyboard shows at the center of the InTouch window, which can cause the keyboard to appear off of the screen. Alternative Solution Scroll to the center of the InTouch window using the browser scroll bars to access the InTouch keyboard if this happens. For new InTouch applications, the recommended solution is to use the device's built-in keyboard.
Playing sound files	Safari on iOS or Internet Explorer version 11.0	Sound WAV files cannot be heard using the PlaySound() method in an InTouch script when the application is viewed with the Safari or Internet Explorer 11.0 web browsers. Alternative Solution Use a different web browser to view an InTouch application.
Slider animation	Safari on iPad or Opera, Firefox, and Chrome on Android tablets	ArchesrA slider graphic animation does not move smoothly. Slider movement is impaired but windows can be scrolled.

Behavior Type	Combinations	Description
Slider animation	Safari on iPad or Opera on Android tablets	The slider control of ArchestrA slider graphics cannot be moved. Alternative Solution There is no alternative solution.
Display	Safari and Chrome on iPad or Opera on Android tablets	The InTouch Access Anywhere keyboard icon overlays the WindowViewer menu bar. Alternative Solution Zoom in and pan or set your browser to a different resolution.
Numeric input	Safari on iPad or Chrome and Opera on a desktop computer	Typing numbers very rapidly from the InTouch numeric pop-up keypad can result in incorrect values appearing in the display.
Key combinations	All browsers on mobile devices except Internet Explorer 10 on WinRT	Ctrl+Shift key combinations do not work for script shortcuts and key scripts. Alternative Solution Use different key combinations for keyboard shortcuts.

Behavior Type	Combinations	Description
Key combinations	All browsers except Internet Explorer 10 on WinRT	<p>Certain mouse events or key combinations do not produce the expected result</p> <ul style="list-style-type: none"> • Working, all browsers <ul style="list-style-type: none"> On Right Key Up • Not working, all browsers <ul style="list-style-type: none"> While Left Key Down Right Click While Right Key Down On Right Key Double Click • Behavior of a Chrome browser on an iPad <ul style="list-style-type: none"> While Left Key Down: The cursor bubble appears on a long press, which means an extended left click is not possible. On Right Key Double Click: The cursor bubble must appear in consecutive selections, which does not result in a double click. On Right Click: Right click functions only when unselecting; function is closer to On Right Up. • While Mouse Over: This behavior works, but if you move your stylus or finger off of the screen, the mouse-over behavior persists.

Running InTouch Access Anywhere: common behaviors

This table describes behaviors that are common to all operating systems and all browsers.

Behavior Type	Description
Lost network connection	<p>When a network connection is lost between a ViewApp in a browser and InTouch Access Anywhere, no notification appears to indicate the loss of the network connection.</p> <p>Alternative Solution</p> <p>Add a timer or clock display that shows the current time to provide an indication of the status of the network connection between the application and the InTouch Access Anywhere Server.</p>
TSE script functions return values	<p>TSE script functions such as TseGetClientId() and TseGetClientNodeName() always return 'AccessAnywhere' as the Client ID and Node Name. This is because all RDP sessions are opened at the Terminal Server node by InTouch Access Anywhere on behalf of the clients.</p> <p>Alternative Solution</p> <p>There is no alternative solution.</p>
Updating applications using NAD	<p>When you update an application from a Network Application Development (NAD) primary computer, a message appears in the NAD client computer but not on the host computer. You can make a change on the NAD primary computer, but because notification does not work, that change will not load into the client copy.</p> <p>Alternative Solution</p> <p>Refresh your session to see the changes. When you open a new session the latest version of the application will appear.</p>
I/O servers	<p>When InTouch requests to start I/O servers, if you click No, the TSE session logs off.</p> <p>Alternative Solution</p> <p>Click Yes when prompted to launch the I/O servers.</p>
Secure access - certificates	<p>If a trusted certificate is not installed when connecting to the Secure Gateway, you may see a warning message containing information that the site's security certificate is untrusted based on your browser's security settings.</p> <p>Alternative Solution</p> <p>Administrators should install a trusted certificate on the InTouch Access Anywhere server or Secure</p>

Behavior Type	Description
	Gateway.
Mouse events	<p>For all browsers: Tablet devices do not support mouse center click events. Script animations using the center button trigger type will not execute.</p> <p>Alternative Solution</p> <p>Modify your applications to use a script trigger other than a mouse center click event.</p>
Dynamic Resolution Changes (DRC)	<p>You will experience problems changing the size of an application window while running InTouch Access Anywhere with the Fit to browser and Fit To Window Advanced Options if DRC is not properly enabled and configured.</p> <p>Solution</p> <p>Ask your InTouch Access Anywhere administrator to complete this procedure, which enables the InTouch Dynamic Resolution Changes (DRC) feature.</p> <ol style="list-style-type: none">1. Connect to the InTouch Access Anywhere server in a remote desktop session (RDP).2. From InTouch Application Manager, select Tools from the menu bar and select Node Properties.3. Select the Resolution tab.4. Select the Allow WindowViewer to dynamically change resolution and Convert to screen video resolution options. <hr/> <p>Note: The <i>InTouch Access Anywhere Administration Guide</i> includes information to enable DRC on the host computer running WindowViewer.</p>

Behavior Type	Description
Display on Devices with Resolution Below 800x600	<p>InTouch Access Anywhere uses a minimum resolution of 800x600, which may not display as expected or desired on devices with a resolution below 800x600.</p> <p>Alternative Solution</p> <p>Ask your InTouch Access Anywhere administrator to modify the minDesktopWidth and minDesktopHeight settings appropriately.</p>
InTouch Access Anywhere Reset	<p>For all browsers: If you attempt to connect to the InTouch Access Anywhere server after clicking Reset on the landing page, the connection times out and fails, and a "Session disconnected" message appears.</p> <p>Alternative Solution</p> <p>Refresh the page after clicking Reset. The connection succeeds.</p>

About deployment

The deployment process instantiates objects to the runtime environment. Deployment includes installing the necessary software on a target computer to support objects. You must deploy the objects (AppObjects, Area and Platform objects, etc.) before you can run a ViewApp.

You can deploy and test your objects at any time during development. When you are ready to test or run a ViewApp in production, you deploy the Galaxy. You can see what your application looks like in a Deployment view or a Model view. Both views show you the structure of your application.

Deployment	Prepare to deploy a ViewApp	About deploying a ViewApp	Deploy an updated ViewApp	About undeploying a ViewApp
				

Prepare to deploy a ViewApp

Deploying a Galaxy copies objects from the computer hosting your IDE to another computer that provides an environment to run a ViewApp. Deployment makes your objects functional. Until you deploy your IDE configuration environment, changes you make to a ViewApp do not appear in the running application. This section describes a set of preparation tasks you need to complete before deploying objects to run your ViewApp.

Prepare to deploy a ViewApp: security

Galaxy security includes a set of deployment permissions that can be assigned to user roles based on the different deployment tasks they need to perform. Restricting deployment permissions to only those user roles that need them to complete their work protects your ViewApps.

Some typical user roles that interact with a ViewApp are:

User Role	ViewApp Work Responsibilities	Expected Deployment Permissions
Designer	Designs and builds the component parts of a ViewApp.	Granted all deployment permissions.
Developer	Integrates objects, graphics, and scripts to create and test a ViewApp	Granted all deployment permissions.
Maintenance	Updates a ViewApp to accurately reflect changes in physical plant processes or equipment.	Can Deploy/Undeploy Application Objects. Can Deploy/Undeploy Device Integration Objects. Can Mark an Object as Undeployed.

Operator	Interacts with a running ViewApp to monitor and manage plant processes.	None
----------	---	------

Before assigning deployment permissions to the different user roles that will interact with a ViewApp, you should have defined users or security groups required by the security authentication mode you selected.

To set deployment security

1. Open the System Platform IDE.
2. From the ribbon, select **Galaxy**, then **Configure**, and then **Security**.
3. The **Security** page appears with the **Authentication mode** tab showing
4. Select the **Roles** tab and expand the **Deployment Permissions** item from the **General permissions** list.
5. For each user role you want to assign deployment permissions to or remove deployment permissions from:
 - a. Select the role in the **Define security roles** list.
 - b. Select or clear the **Deployment Permissions** to fit the role's needs.
6. Select **Save** to save your changes.

Prepare to deploy a ViewApp: galaxy status

You should view the status of your Galaxy's objects before you deploy a ViewApp to determine if there are any objects that are in warning or error state.

A red icon appears to the left of objects containing errors or warnings. To see an object's error or warning messages, right-click on it and select the **Properties** option. The **Errors/Warnings** tab lists any messages.

To determine the status of a Galaxy

1. Connect to the Galaxy hosting the Viewapp that you want to deploy.
2. From the ribbon, select **Galaxy**.
3. At the bottom left of the window, select **License**. The **About** page appears.
4. The **Galaxy Status** area shows the number of instances, templates, deployed instances with pending changes, undeployed instances with changes, objects in an error or warning state, objects that are checked out, and objects you have checked out.
5. Click **OK** to close the **Galaxy Status** dialog box.

Assign an engine to a ViewApp

Before deploying ViewApps, you need to assign each one to the ViewEngine (for access from the Application Manager) or WebViewEngine (for browser access) that it will run on. As you assign ViewApps, consider the I/O, memory, and CPU needs of each one. Assign your ViewApps to distribute these loads evenly across the platforms in the Galaxy environment at runtime.

The operating system determines how engines are assigned to individual CPU cores and attempts to load balance across CPU and other computing resources on the platform. If you need to change which platforms and engines ViewApps are assigned to in order to improve load balancing, undeploy the ViewApps and reassign them to different ViewEngines and WebViewEngines.

If you are using redundant AppEngines, you can control how objects are loaded at runtime.

Prepare to deploy a ViewApp: assign a ViewApp to an engine

Once you are ready to deploy a ViewApp, you must assign the derived ViewApp template to the ViewEngine or WebViewEngine that it will run on. This creates an instance from the derived template and assigns the instance to the ViewEngine or WebViewEngine.

To make a ViewApp available to users from the desktop client, assign it to a ViewEngine. These ViewApps are started from the Application Manager.

To make a ViewApp available to users using the OMI web client in a web browser, assign it to a WebViewEngine.

If you want a ViewApp to be available from both the Application Manager and the OMI web client, assign the derived template to both a ViewEngine and a WebViewEngine. This creates two instances of the ViewApp, one running on each engine.

Assign a ViewApp to an engine

To assign a ViewApp to a ViewEngine or WebViewEngine:

1. Open the IDE and display both the **Templates** pane and the **Deployment** pane.
2. In the **Templates** pane, navigate to the folder containing the ViewApp derived template.
3. Right-click the desired ViewApp template and select **New**, then **Instance** from the context menu.
This creates an instance of the ViewApp in the **Unassigned Host** section of the **Deployment** pane.
4. Drag the new instance to the desired ViewEngine or WebViewEngine.

Note: You can also drag the derived ViewApp Template to the desired engine in the **Deployment** pane. This creates a new instance and places it in the **Deployment** pane.

If you want the ViewApp to be available to users from both the Application Manager and the OMI web client, repeat the procedure, assigning the ViewApp to both a ViewEngine and a WebViewEngine.

About deploying a ViewApp

You deploy object instances for three reasons:

- Testing.
- Placing the application into production to process field data.
- Updating an existing application with changes.

When you are ready to deploy, make sure the following conditions are met:

- Bootstrap software is installed on the target computers.
- The objects being deployed are not in an error state in the Galaxy database.
- You created, configured, and checked in objects to the Galaxy.
- Objects are assigned to a host.

- The object's host is already deployed. A cascade deploy operation, which deploys a hierarchy of objects, deploys all objects in the correct order. A cascade operation deploys an object's host before the object is deployed.
- Any associated script libraries are ready for use on the target computer.

Deploy a ViewApp

Video Tutorial: Deploy a ViewApp in AVEVA OMI

https://player.vimeo.com/video/992311905?badge=0&autoplay=0&player_id=0&app_id=58479

To deploy a ViewApp

1. Select the object or objects that you want to deploy in an Application view.

Note: Deployment view is often best for this purpose, as it shows which objects, if any, are beneath the selected object in the deployment order.

2. On the **Home** ribbon, in the **Deploy** area, select **Deploy**.

You can also deploy a selected object by right-clicking and selecting **Deploy** from the shortcut menu.

The **Deploy** dialog box appears.

Deploy 139 objects

General

InTouchViewApp

Deployment defaults

 Cascade deploy Include redundant partner

Deployed objects

 Skip Force off scan Deploy changes Preserve runtime changes Redeploy

Undeployed objects

 Deploy new objects

Initial scan state

 On scan

Deploy status mismatch

 Off scan Mark as deployed

Note:

Deploying a host object will force a redeploy of all hosted objects.

CancelDeploy

3. Select one or more of the following:

- **Cascade deploy:** Select this check box to deploy the object selected for deployment as well as any objects it hosts. This option is selected by default if the object is a host. If you are deploying an individual host object, clear the check box. Objects being deployed across multiple platforms are deployed in parallel.
- **Include redundant partner:** Select this check box to also deploy an AppEngine's redundancy partner object. This option is selected and unavailable when the redundant engine has pending configuration changes or software updates.

4. In the **Deployed objects** area, select one or more of the following options. These options are not available if the selected object has not been deployed before.

- **Skip:** If one of the objects you are deploying is currently deployed, selecting **Skip** makes no changes to the already-deployed object.
 - **Deploy changes:** If one of the objects you are deploying is currently deployed, this option updates the object in question with new configuration data. The runtime state from the runtime file is preserved and the state is modified with any changes.
 - **Redeploy:** If one of the objects you are deploying is currently deployed, this option deploys the same version as previously deployed. For example, use this option to redeploy an object that is corrupted on the target computer.
 - **Force off scan:** If one of the objects you are deploying is currently deployed, this option sets the target object to off scan before deployment occurs.
 - **Preserve runtime changes:** Selecting this option will produce different results, depending on whether you have chosen **Deploy changes** or **Deploy original**, if values were changed at runtime, and if configuration values were changed in the IDE.
 - If a value was changed in runtime and **Deploy changes** and this option are checked, the object is deployed with the last runtime value, even if you change the configured value in the IDE.
 - If a value was not changed during runtime and **Deploy changes** and this option are checked, the object is deployed with the last configured value from the IDE.
 - If a value was changed in runtime and **Redeploy original** and this option are checked, the object is deployed with the last runtime value, even if you change the configured value in the IDE.
 - If a value was not changed during runtime and **Redeploy original** and this option are checked, the object is deployed with the original configured value from the IDE. If you configured a new value in the IDE, the prior configured value is used.
5. In the **Undeployed objects** area, select **Deploy new objects** to start a normal deployment.
 6. In the **Deploy status mismatch** area, select **Mark as deployed** to mark the object as deployed in the Galaxy. A mismatch happens when the object is previously deployed to a target node, but the Galaxy shows the object is undeployed. Clear this option to redeploy the object to the target node.
 7. In the **Initial scan state** area, select one of the following:
 - **On Scan:** Sets the initial scan state to on scan for the objects you are deploying. If the host of the object you are deploying is currently off scan, this setting is ignored and the object is deployed off scan. When you are deploying multiple objects, the deploy operation deploys all of the selected objects "off-scan." After all of the objects are deployed, the system sets the scan-state to "on-scan."

Objects can run only when both the host/engine is "on scan" and the object is "on scan." If either the host/engine or the object is "off scan," the object cannot run.

Always deploy Areas to their host AppEngines on scan. Because Areas are the primary providers to alarm clients, deploying Areas off scan results in alarms and events not being reported until they are placed on scan.
 - **Off Scan:** Sets the initial scan state to off scan for the objects you are deploying. If you deploy objects off scan, you must use the ArchestrA System Management Console Platform Manager utility to put those objects on scan and to function properly in the runtime environment.
 - The default scan setting is set in the **User Default** settings in the **Configure User Information** dialog box. For more information, see [Configure user preferences](#).
 - Click **Deploy** to deploy the objects. The **Deploy** progress box appears. When the deploy is complete, click

[Close.](#)

Determine deployment status

You can tell if you have objects that need to be deployed by looking at the icons next to the objects listed in the right-most column of the **Model** or **Deployment** views of the IDE.

Deployment status icons for most objects, including ViewApps, are:

- Deployed
-  Not deployed
-  Pending update
-  Pending software update

These additional icons may appear for redundant AppEngines:

	Both redundant AppEngines are not deployed
	This AppEngine is not deployed, but its redundant partner is deployed.
	This AppEngine is deployed, but its redundant partner is not deployed.
	Both redundant AppEngines have pending configuration updates.
	This AppEngine has a pending configuration update, but its redundant partner is up to date.
	This AppEngine is up to date, but its redundant partner has a pending configuration update.
	Both redundant AppEngines have pending software updates.
	This AppEngine has a pending software update, but its redundant partner is up to date.
	This AppEngine is up to date, but its redundant partner has a pending software update.

For directions on deploying view apps, see [Deploying a ViewApp](#).

Deploy an updated ViewApp

When deploying a ViewApp for the first time, the Galaxy Repository compiles all ViewApp configuration data, the display module, and binaries and places them in a destination folder with the server's time stamp file. When you

deploy the ViewApp again, only the files changed after the time stamp from last deploy operation are deployed again.

The following list summarizes the steps to deploy an updated ViewApp.

1. The Galaxy Repository verifies if any files in the ViewApp's destination folder contain any files that have changed since the previous deployment.
2. If no files have changed since the previous deployment, then the client will receive all files from CVR and deploy them to the destination folder, and also set the time stamp (server time) into the application folder.
3. If files have been changed since the previous deployment, then read the time stamp from destination folder, send it to the server, get delta changes, along with deleted files information, and stage them to the destination folder. Remove the deleted graphicl files
4. The Folder Synchronizer synchronizes the newly staged files to the computer running the ViewApp.

About undeploying a ViewApp

You may need to undeploy one of more objects. Undeploying removes the ViewApp instance and its associated data from the target node.

Before you delete or restore a Galaxy, undeploy all objects in the Galaxy.

Undeploying can fail if the target object has objects assigned to it. Make sure you select **Cascade Undeploy** in the **Undeploy** dialog box.

Undeploy a ViewApp

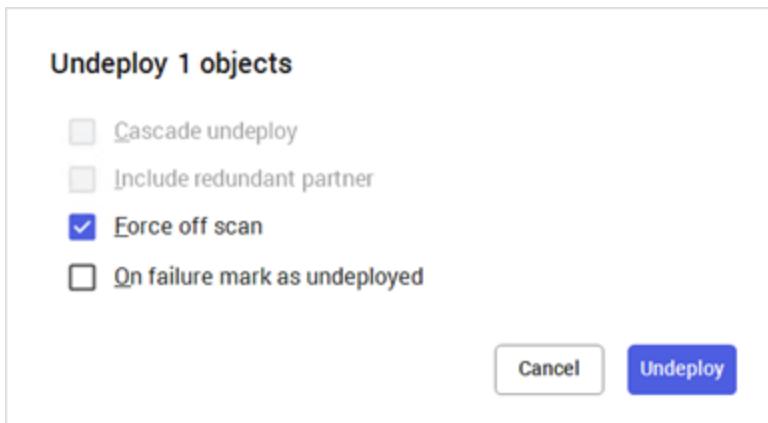
You may need to undeploy one of more objects. Undeploying removes the ViewApp's files and its associated data from the target node.

Undeploying can fail if the target object has objects assigned to it.

Before you delete or restore a Galaxy, undeploy all objects in the Galaxy. Make sure you select **Cascade undeploy** in the **Undeploy** dialog box.

To undeploy a ViewApp

1. Make sure the ViewApp you want to undeploy is not running.
2. From the application view area of the IDE, select the object or objects to undeploy.
3. From the **Home** ribbon, in the **Deploy** area, select **Undeploy**. The **Undeploy** dialog box appears.



The top of the dialog box shows the total number of objects being undeployed. This includes the objects you selected and, if you select the **Cascade undeploy** option, all objects that are beneath a selected object in the hierarchy and will be undeployed as a result.

4. Select one or more of the following options. Some of these options may not be available, depending on the kinds of objects you selected to undeploy.
 - **Cascade undeploy:** Select to undeploy the selected object and any objects it hosts.
 - **Include redundant partner:** Select to also undeploy an AppEngine's redundancy partner object.

Note: The Primary AppEngine in a redundant pair can be undeployed alone because objects hosted by it run on the deployed Backup AppEngine, which becomes Active.

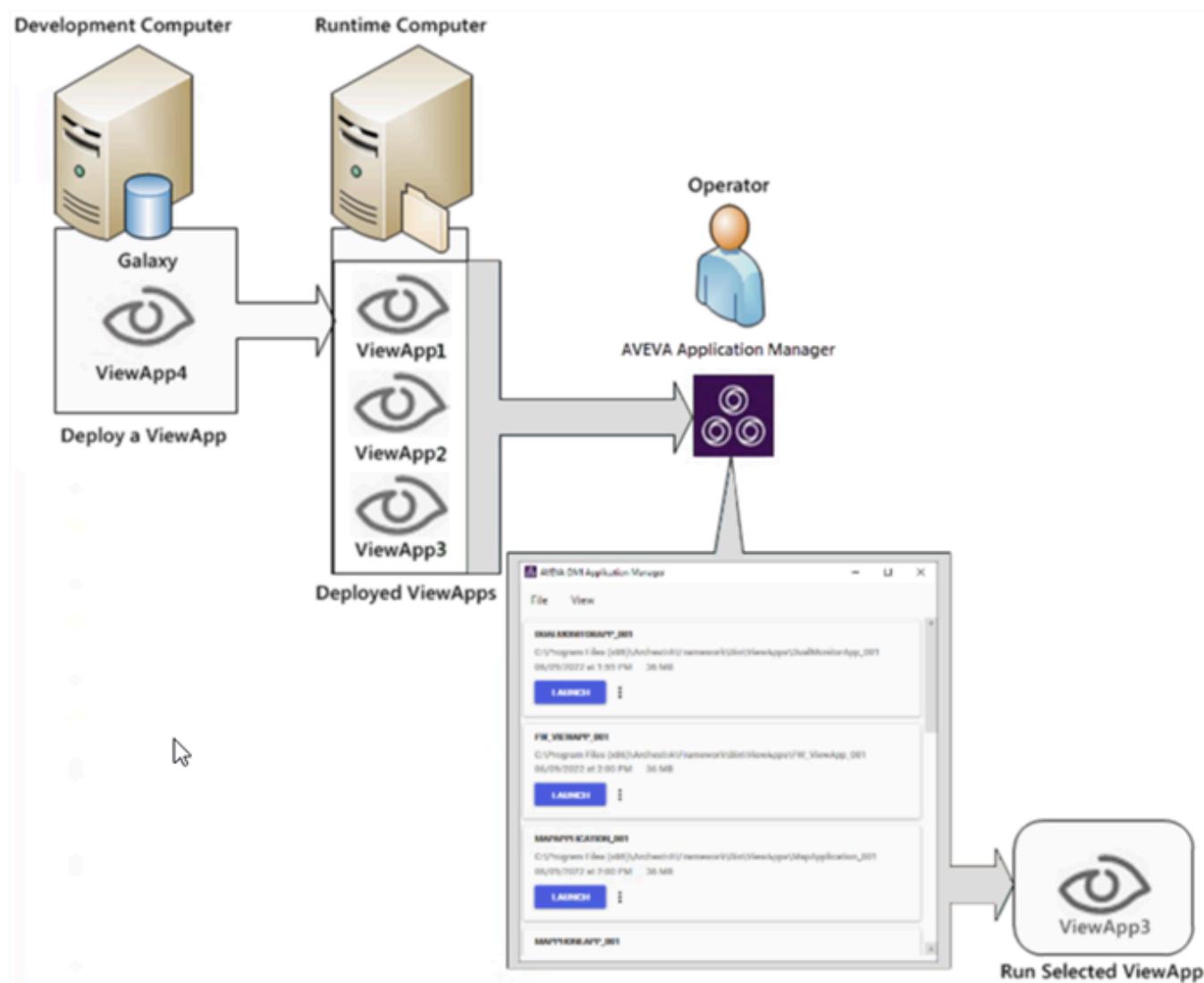
 - **Force off scan:** If one of the objects you are undeploying is currently on scan, selecting **Force off scan** sets the target object to off scan before undeployment. If you do not select **Force off scan** and the target object is on scan, the undeployment operation fails.
 - **On failure mark as undeployed:** Marks the object as undeployed in the Galaxy if the object targeted for undeployment is not found.
5. Click **Undeploy** to undeploy the objects.

Run a ViewApp

Run a ViewApp 	View deployed ViewApps	About the InTouch web client	Launch a ViewApp in a web browser
	Launch ViewApp in the OMI Web Client using connected experience	Log in with AVEVA Identity Manager	Open a ViewApp
	Search a ViewApp NavTree	Process late data at runtime	Enable historical playback

View deployed ViewApps

The AVEVA OMI Application Manager lists the ViewApps deployed to a runtime computer or terminal server. When a user selects the **LAUNCH** button of a listed ViewApp, the ViewApp starts running on the computer.



Launch a ViewApp

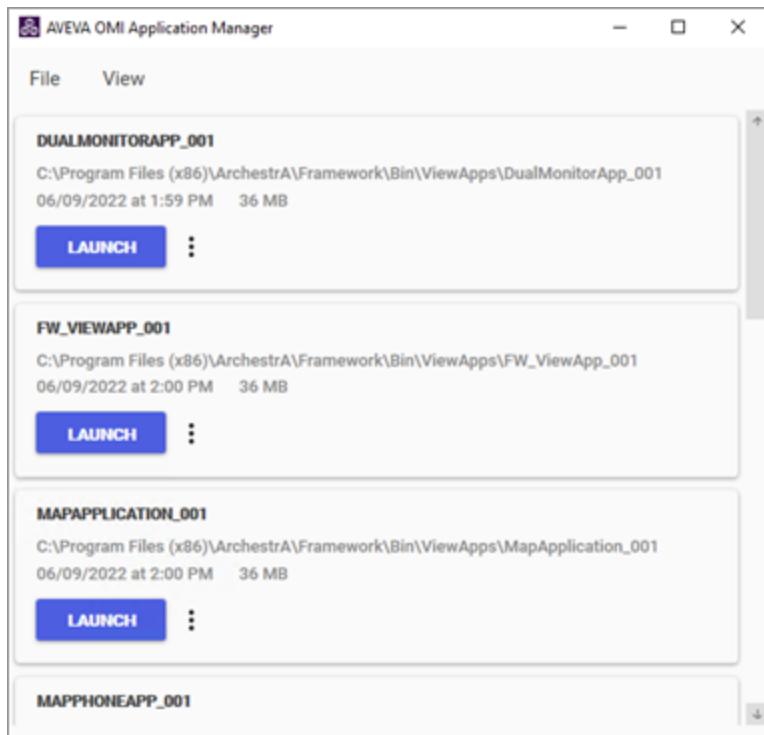
You launch a deployed ViewApp from the AVEVA OMI Application Manager. You start the Application Manager by selecting its shortcut icon on the Windows desktop or as a Start Menu entry. To see a video of this process, see [Launch a ViewApp](#).



To launch a ViewApp

1. Show the desktop of the computer where the ViewApp has been deployed.
2. Select the AVEVA OMI Application Manager desktop shortcut.

AVEVA OMI Application Manager appears with a list of ViewApps currently deployed on the computer. ViewApps are listed in descending alphabetic order by ViewApp name.



For more information about the fields of the AVEVA OMI Application Manager, see [About the ViewApps Application Manager](#).

3. Find the ViewApp you want to run.
4. Select the ViewApp's **LAUNCH** button.

The word **LAUNCHING** appears briefly followed immediately by an AVEVA OMI splash screen.

The ViewApp appears on the screen and as an item on the computer's Windows task bar.

About the ViewApps Application Manager

The AVEVA OMI Application Manager lists all current ViewApp instances deployed to a computer and includes information about each ViewApp. Each deployed ViewApp appears as a separate item in the list.

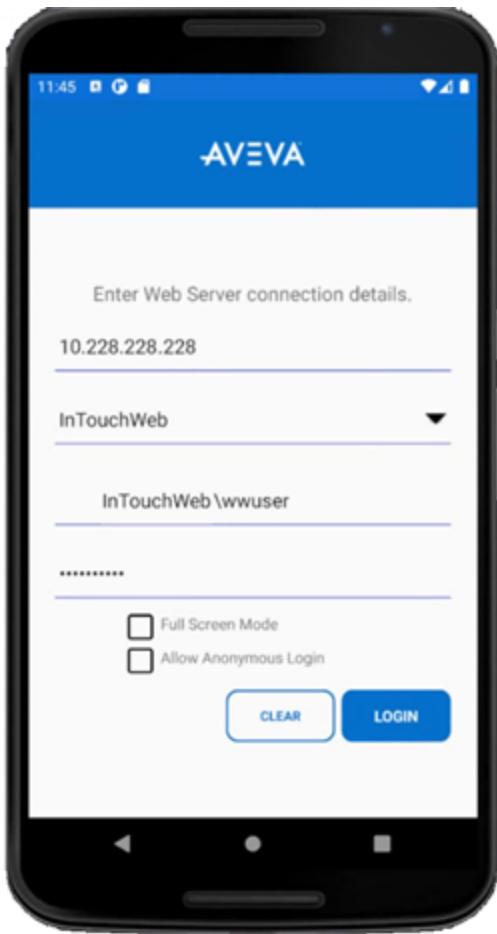


1	ViewApp name	Name of the ViewApp deployed to the computer.
2	Deployed folder path	Path to the folder of the ViewApp deployed on the computer. The Galaxy name and ViewApp name are assigned as the folder name containing the ViewApp's files in the form <i>Galaxy_Name-ViewApp_Name</i> .
3	Date and time	The most recent date and time the ViewApp's derived template was edited. The date and time is displayed in the regional format of the logged in Windows user at the time the AVEVA OMI Application Manager is launched.
4	LAUNCH button	Button to run the selected ViewApp. Click or tap the button to start the ViewApp running on the computer. The word LAUNCHING appears briefly and fades out after several seconds when the ViewApp starts running.
5	Operating Mode	Drop-down list to select the operating mode of the ViewApp as either Read/Write (default) or Read Only.
6	Disk space usage	Amount of disk space used to store the deployed ViewApp files on the computer.

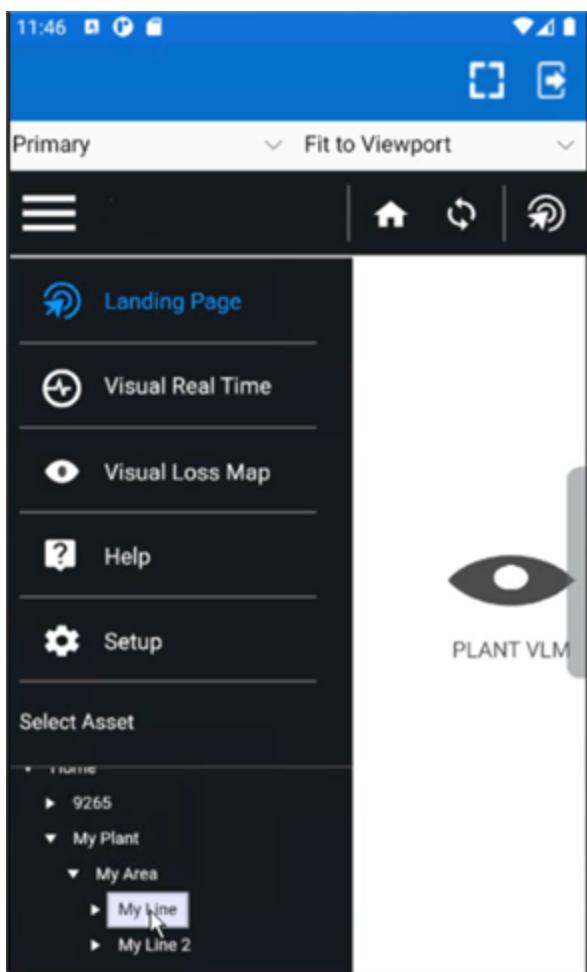
Display a web ViewApp on a mobile device

As of the System Platform 2023 release, the AVEVA Mobile Operations application has been updated to show OMI web ViewApps on a mobile device.

From your mobile device, log in to your web server.



After logging in, the header area of screen shows the primary screen of the ViewApp and the view mode. Selecting tabs at the left and right sides of the device screen show slide in panes. The following example shows the navigation tree of the ViewApp within the slide-in pane.



Responsive layouts work well with ViewApps intended to be displayed on mobile devices. The app also includes options to change screens and view modes.

About the InTouch web client

You can view AVEVA InTouch HMI applications by using the InTouch Web Client in a supported web browser. The Web Client is also available on mobile platforms. A built-in web server provides web browsers access to InTouch HMI applications from any Microsoft Windows client or server operating system without the use of Remote Desktop Protocol (RDP) or Internet Information Services (IIS) for Microsoft Windows® Server.

The InTouch Web Client also includes workspaces. Workspaces enable users to configure ViewApps while viewing them in the web browser. Workspaces are individual, so each user can configure their own ViewApps to meet their individual needs.

Launch a ViewApp in a web browser

If an OMI ViewApp has been deployed to a WebViewEngine, you can open the ViewApp in your web browser. To do so:

Notes: The first time you try to open the OMI web client on a remote device, you may need to install a security certificate to enable it. See [Install a security certificate on Windows](#) for more information.

1. In a supported web browser, go to the URL for the OMI web client:
 - On the machine where the WebViewEngine is deployed, the URL is `https://localhost/omi`. You can also use either of the two formats below.
 - On a machine on the same domain as the one where the WebViewEngine is deployed, the URL is `https://<node_name>/omi`, where `<node_name>` is the name of the platform where the WebViewEngine is deployed.
 - On a machine that is not on the same domain as the WebViewEngine, the URL is `https://<fqdn>/omi`, where `<fqdn>` is the fully-qualified domain name of the platform where the WebViewEngine is deployed.

The OMI web client sign in page opens.

Note: If your installation uses AVEVA Operations Control and the connected experience, you may or may not be asked to log in before you can open the OMI web client. see [Launch using the connected experience](#) for more information.

2. Enter your user name and password.

The OMI web client home page opens, listing all of the OMI ViewApps available through the web client.

3. Select the ViewApp to run.

When the ViewApp opens, the user name that you are logged in as shows in the top-right corner of the page.

Note: Unlike the OMI desktop client, the OMI web client does not notify you if there is an available updated version of the ViewApp that you are running. If you refresh the ViewApp web page, or return to the OMI web client home page and then re-open the ViewApp, the software checks for a newer version of the ViewApp and downloads any updates that have been made to it. We recommend you do this regularly to make sure you have the latest version.

ViewApp behavior in the web client

In most ways, an OMI ViewApp behaves the same in the OMI web client as it does in the OMI desktop client. However, there is an icon at the top right of the browser page which lets you use a few features which do not exist or are different in the desktop client.



Selecting this icon lets you choose from these features:

- **Full screen:** Changes the OMI web client page to full screen mode. To leave full screen mode, press `Esc`.
- **Log out:** Closes the ViewApp and logs you out of the web client. It opens the AVEVA Identity Manager login page so you or someone else can log back in as a different user, if desired. When you log back in, you see the web client home page so you can start any ViewApp.
- **Select screen:** The web client can show only one screen of a multi-screen layout at a time. This area shows the current screen name. If your ViewApp uses a multi-screen layout, you can select the screen name or the down-arrow next to it, then select the screen to view.

Pan and zoom in the OMI web client

Panning and zooming using the mouse works differently in the OMI web client than in the OMI desktop client. It

also works differently than panning and zooming in most web applications. The table below shows how various mouse actions work in the two OMI clients.

Mouse action	OMI desktop client	OMI web client
Scroll up and down	Zooms in and out	Pans the current pane up and down
SHIFT plus scroll up and down	Zooms in and out	Pans the current pane left and right
CTRL plus scroll up and down	Zooms in and out	Zooms the current pane in and out
Drag with center button down	Pans	No effect

Notes:

- The maximum zoom level is 500%.
- The current zoom level is shown at the bottom left of the layout.
- Using the keyboard to zoom (**CTRL SHIFT +** or **CTRL -**), zooms the entire browser window, not just the ViewApp display.
- You can also pan using the scroll bars that appear if the content does not fit in the pane at the current zoom level.

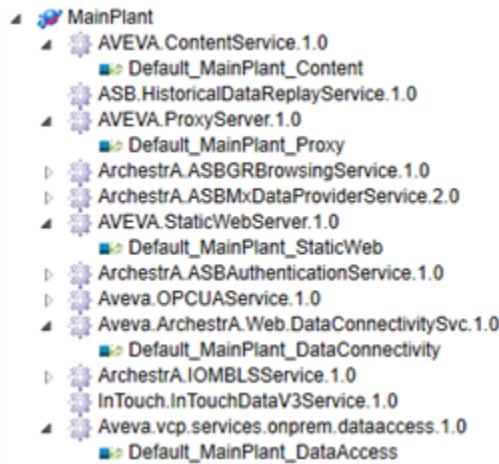
Troubleshoot OMI web client connection issues

If you have a problem starting or working with the OMI web client, following the steps below may help you find and resolve the issue.

1. Check [Known Issues in the OMI web client](#). If your symptom is listed there, follow the recommended workaround.
2. On the System Management Server (SMS) host, start the Configurator and make sure the settings for the web client are correct:
 - a. Under **Common Platform**, make sure there is a green checkmark next to **System Management Server**. If there isn't, select **System Management Server** and select the option to make this host the SMS.
 - b. Select **System Management Server**, then select **Advanced**.
 - c. Select the **Ports** tab, and make sure the **HTTPS Port** is set to 443. Select **OK**.
 - d. Under **AVEVA System Platform**, there should be a green checkmark next to **Identity Manager Registration**. If there isn't:
 - i. Select **Identity Manager Registration**.
 - ii. Select **Configure**.
 - iii. Select **Close**.
- **Note:** If the SMS was not already defined, after performing the steps above, go to the other nodes that have System Platform installed and use the Configurator to connect them to the SMS.
3. On the device that is having the problem, clear the browser cache (select the option to clear the cache for all time). Then close and re-open the browser and try to open the web client again.
4. On the platform that hosts the WebViewEngine, make sure that all five services that are supposed to be deployed and started when the WebViewEngine is deployed are running. These services are:

- ArchestrA.Web.DataConnectivitySvc.exe
- vcp.services.onprem.contentserver.exe
- vcp.services.onprem.dataaccess.exe
- vcp.services.onprem.frontdoor.exe
- vcp.services.onprem..webserver.exe

You can check for these services either on the **Details** tab of the Windows Task Manager or from the IDE by selecting **Galaxy > Configure > System > Services**. When you expand the tree view nodes on the **Configure Services** dialog box, each service name should show a green box to indicate that it is deployed and running. In this sample, the Galaxy name is MainPlant:



5. If the services are not all deployed, undeploy and redeploy the WebViewEngine. Then, repeat step 4 to confirm that the services have started.

If you cannot undeploy an OMI ViewApp

If you are unable to undeploy a ViewApp that is assigned to the WebViewEngine, follow these steps to fix the issue:

1. Follow the instructions in step 4, above, to see if the WebViewEngine services are deployed and running. If they are, go to the next step. If they aren't:
 - a. If it isn't already open, open the **Configure Services** dialog box from the IDE by selecting **Galaxy > Configure > System > Services**.
 - b. Expand the tree view nodes to see the service names, as shown in the image above.
 - c. Undeploy and redeploy the five WebViewEngine services. (The ones that are expanded in the image.)
2. Check to see if you can now undeploy the ViewApp.
3. If you still can't undeploy the ViewApp, restart the node hosting the WebViewEngine.
4. If you still cannot undeploy a ViewApp, please contact AVEVA Global Customer Support.

Launch ViewApp in the OMI Web Client using connected experience

If your installation uses AVEVA Operations Control connected experience for user authorization and security, follow these steps to start an OMI ViewApp.

To launch a ViewApp in the OMI web client using the connected experience

1. In a supported web browser, go to the URL for the OMI web client:
 - On the machine where the WebViewEngine is deployed, the URL is <https://localhost/omi>. You can also use either of the two formats below.
 - On a machine on the same domain as the one where the WebViewEngine is deployed, the URL is https://<node_name>/omi, where <node_name> is the name of the platform where the WebViewEngine is deployed.
 - On a machine that is not on the same domain as the WebViewEngine, the URL is <https://<fqdn>/omi>, where <fqdn> is the fully-qualified domain name of the platform where the WebViewEngine is deployed.

If you are already logged in to another AVEVA product that uses Operation Control connected experience, you are logged in to the OMI web client automatically using Single Sign-on. Skip to step 3.

2. If you are not already logged in to a product that uses the connected experience, a sign-in window opens for you to enter your user name and password.

After you have entered your user name and password, the OMI web client page opens.

3. The OMI web client page lists all of the OMI ViewApps available through the web client. Select the ViewApp to run.

When the ViewApp opens, the user name that you are logged in as shows in the top-right corner of the page.

If you have a problem logging in (authentication or entitlement failure)

- Make sure your internet connection is working.
- If user authentication fails, the ViewApp opens in read-only mode after you click OK.
- If entitlement fails, you will see a message that a valid subscription could not be found. After you click OK, the ViewApp opens in read-only mode.

Corrective action

If you receive an authentication error message, repeat the log in procedure, being sure to enter your user name and password correctly. If you still cannot sign in, contact your CONNECT administrator.

If you receive an entitlement error message, there may be a problem with your AVEVA subscription. Contact your CONNECT administrator.

Log in with AVEVA Identity Manager

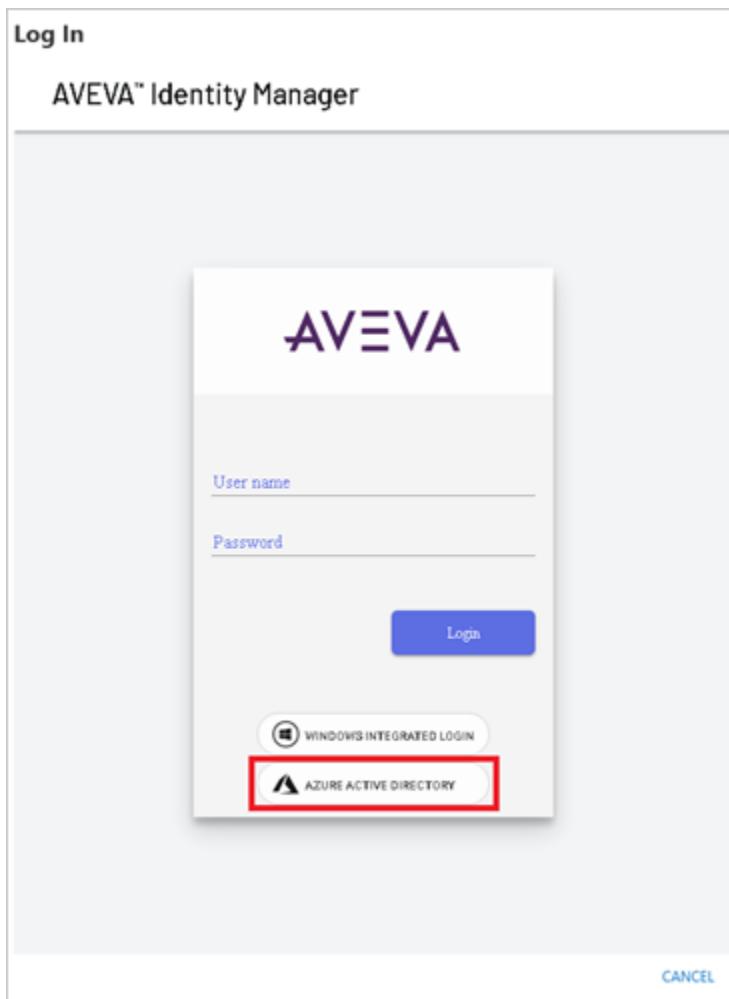
When Azure AD is configured as the identity provider, all required user log-ins will use the AVEVA Identity Manager. This applies to all System Platform components and secured operations, including:

- System Platform IDE
- Operation Control Management Console
- AVEVA OMI ViewApps
- Secured Writes

- Verified Writes

Use Azure Active Directory credentials to sign on

A logon dialog opens when you click **Login** button in the Title Bar, or when the ShowLoginDialog() script function is invoked.



To use Azure Active Directory credentials for log in

1. Select the **Azure Active Directory** button. This launches the Azure AD log-in screen.

Note: Windows Integrated Login is not supported in Controlled Release 2.

2. Enter your credentials and follow the on-screen instructions.

You are logged into the ViewApp when your credentials are validated. A successful log in requires that:

- At least one security role is assigned to you.
- Group membership for Azure AD is valid.

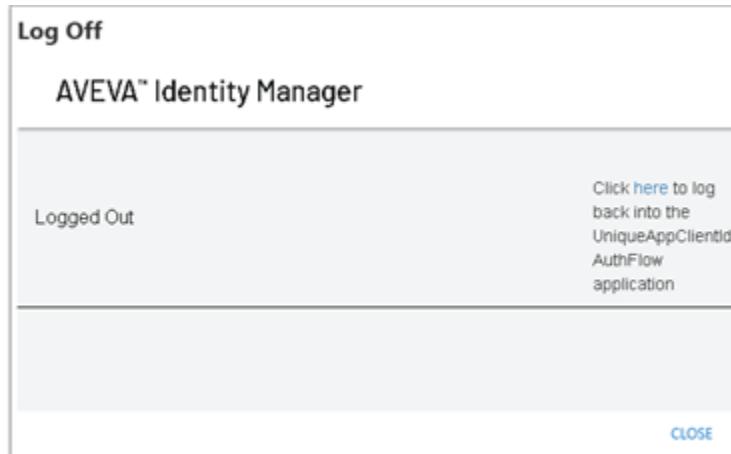
Authentication fails if the group was not added for your user role, and the following error is displayed:

The login attempt failed.

The user must have at least one role assigned to it to be able to log in.

Log Off via Azure AD

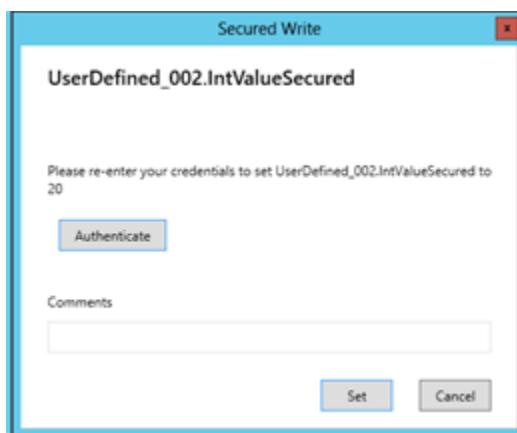
To log out of the ViewApp after signing with Azure AD credentials, click the Logoff button in the Title Bar to display the Logoff dialog. This signifies that the user has ended their session. The dialog is also displayed when the Logoff() script function is invoked.



Use Azure Active Directory credentials for secured write authentication

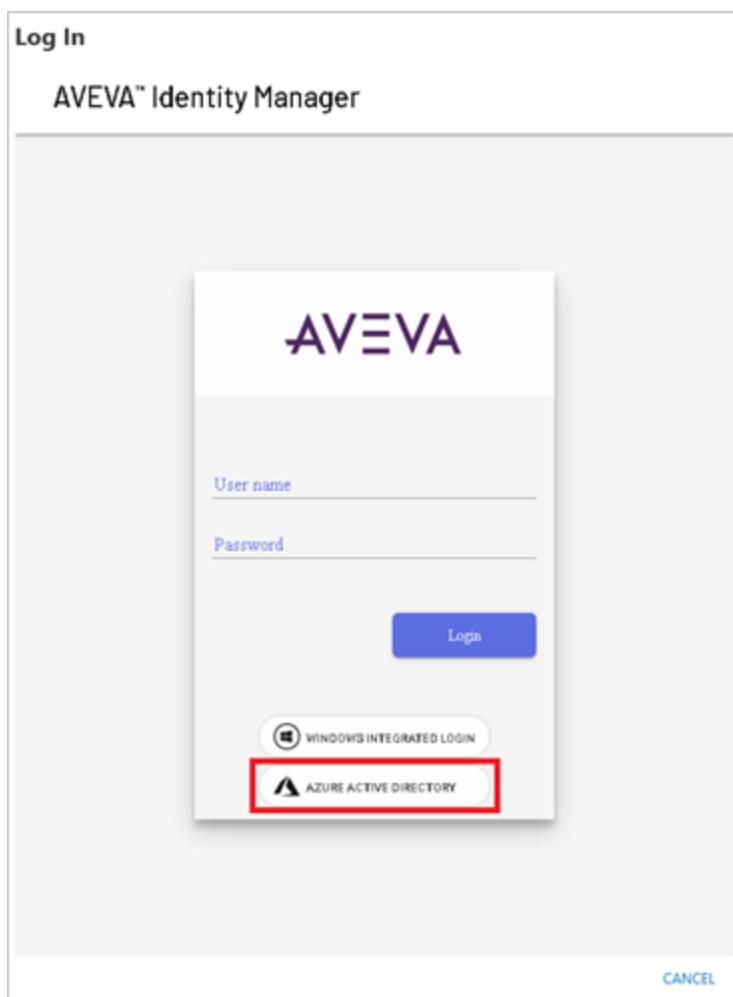
When security is configured with Azure AD as an authentication provider, users can use their Azure AD credentials to authenticate secured write operations. For general information about secured writes, see [Working with the SignedWrite\(\) function for secured and verified writes](#).

An authentication dialog opens when you attempt to write to a Secured-Write attribute, or the SignedWrite() attribute is invoked.



To use Azure Active Directory credentials for Secured Write authentication

1. Click the **Authenticate** button. The AVEVA Identity Manager window opens.



2. Select the **Azure Active Directory** button. This launches Azure AD log-in screen.

Note: Windows Integrated Login is not supported in Controlled Release 2.

3. Enter your credentials and follow the on-screen instructions.
4. To complete the secured write:

- At least one security role assigned must be assigned to you.
- Group memberships for Azure AD must be valid.
- You must have at least one role assigned that allows you to write to attributes.

Authentication fails if the group was not added for your user role, and the following error is displayed:

The login attempt failed.

The user must have at least one role assigned to it to be able to log in.

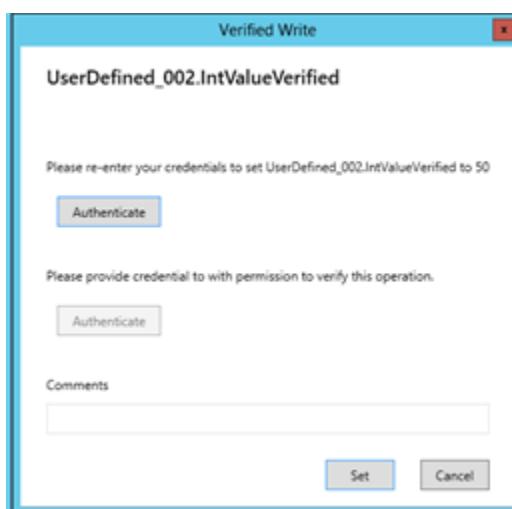
5. In the authentication dialog, add comments and select **Set** to verify the attribute changes.

Use Azure Active Directory credentials for verified write authentication

When security is configured with Azure AD as an authentication provider, users can use their Azure AD credentials to authenticate verified write operations. For general information about verified writes, see [Working with the SignedWrite\(\) function for secured and verified writes](#).

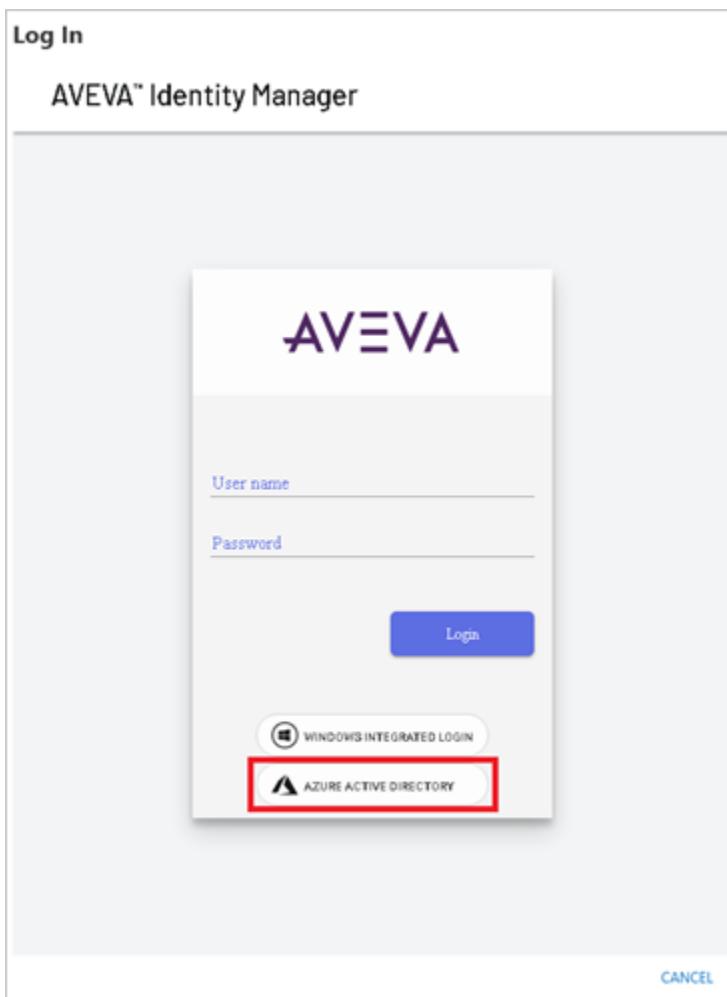
Two users with valid Azure AD permissions are required to write to a verified-write attribute.

An authentication dialog opens when you attempt to write to a Verified-Write attribute.



To use Azure Active Directory credentials for Verified Write authentication

1. User 1 clicks the **Authenticate** button in the authentication dialog. The AVEVA Identity Manager window opens.



2. User 1 selects the **Azure Active Directory** button. This launches Azure AD log-in screen.

Note: Windows Integrated Login is not supported in Controlled Release 2.

3. User 1 enters their credentials and follows the on-screen instructions.
4. The verified write operation can proceed to the next phase if:
 - User 1 has at least one security role assigned.
 - User 1 group memberships for Azure AD must be valid.
 - User 1 has at least one role assigned that allows them to write to attributes.
5. When User 1 is successfully authenticated, the second **Authenticate** button in the Verified Write dialog is enabled.
6. User 2 clicks the enabled **Authenticate** button. The AVEVA Identity Manager window opens.
7. User 2 selects the **Azure Active Directory** button. This launches Azure AD log-in screen.

Note: Windows Integrated Login is not supported in Controlled Release 2.

8. User 2 enters their credentials and follows the on-screen instructions.
9. The verified write operation can proceed to the verification phase if:
 - User 2 has at least one security role assigned.
 - User 2 group memberships for Azure AD must be valid.
 - User 2 has at least one role assigned that allows them to verify attributes.

When User 2 is successfully authenticated, they can add comments in the authentication dialog. User 2 selects **Set** to verify the attribute changes.

If verification is attempted using the same credentials as user 1, the following error is displayed:

Operator and Verifier have to be two different users

Authentication fails if the group was not added for the user's role, and the following error is displayed:

The login attempt failed.

The user must have at least one role assigned to it to be able to log in.

Open a ViewApp

To open a ViewApp, go to the Windows Start Menu, then select either:

- **AVEVA OMI Application Manager**, located in the AVEVA System Platform folder.
- **Application Manager**, located in the AVEVA InTouch HMI folder.

Search a ViewApp NavTree

During runtime, the content of a ViewApp can be searched by entering a search string to find navigation items by their names, aliases, or paths that appear in the navigation model. A search entry field appears above the NavTree control.

filt

X

While entering a search string, any matching navigation item names, aliases, or paths appear in a search results

list.

A screenshot of a search interface. At the top, there is a search bar with the text 'filt' and a clear button 'x'. Below the search bar, a message says '6 navigation items matching "filt"'. A table lists six items:

Navigation Item	Path
Filters	Home
FiltrationUnits	Home
Filter100	Home ▶ FiltrationUnits
Filter200	Home ▶ FiltrationUnits
Filter300	Home ▶ FiltrationUnits
Filter400	Home ▶ FiltrationUnits

After selecting an item from the search results list, the ViewApp shows the content of the navigation item you selected.

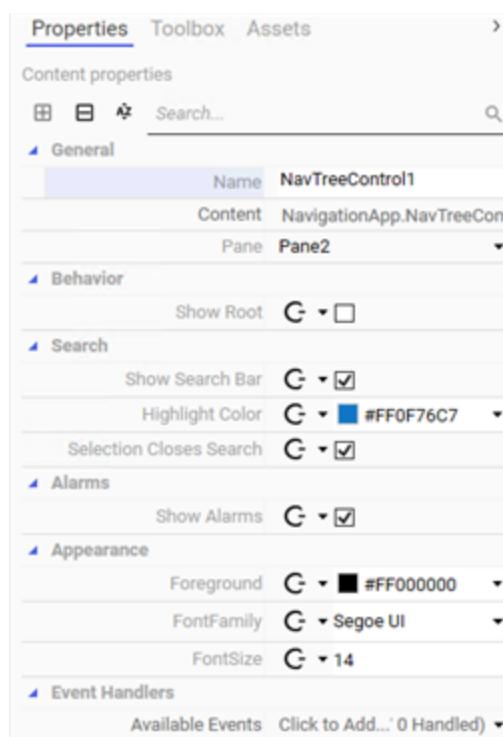
Enable runtime navigation model searches

The search field above a NavTree control can be displayed or hidden during runtime based on the value set to the **Show Search Bar** property. A search field appears when **Show Search Bar** is selected, which is the default value.

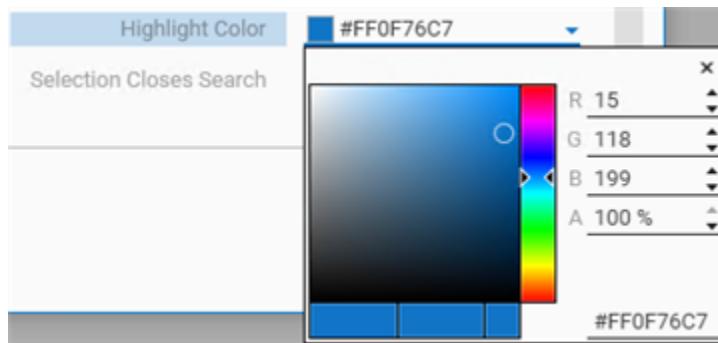
The NavTree control has other properties that affect runtime searches. See [About search results](#) for instructions to show the root of the navigation model.

To enable runtime navigation model searches

1. Open a layout or ViewApp in their respective editors.
2. Drag and drop a NavTree control onto a layout pane.
3. Select the NavTree control placed on the pane.
4. Select the **Properties** grid to show the properties of the NavTree control.



5. Select the **Show Search Bar** property to show the search bar above the navigation tree.
6. Select the **Show Alarms** property if you want to see the alarm status of navigation items that appear in the search results list.
7. Select the **Selection Closes Search** property if you want to hide the search list immediately after selecting an item from the search list.
8. If you want to change the highlight color that appears in the search results list, select the **Highlight Color** property.
 - a. Place the mouse cursor over the data entry field to show a triangle icon at the right.
 - b. Select the triangle icon to show a color picker control.



- c. Select a highlight color from the picker.
9. Save your changes to the NavTree control.

About search results

After you enter a search string, a search is conducted over the entire navigation model of the running ViewApp.

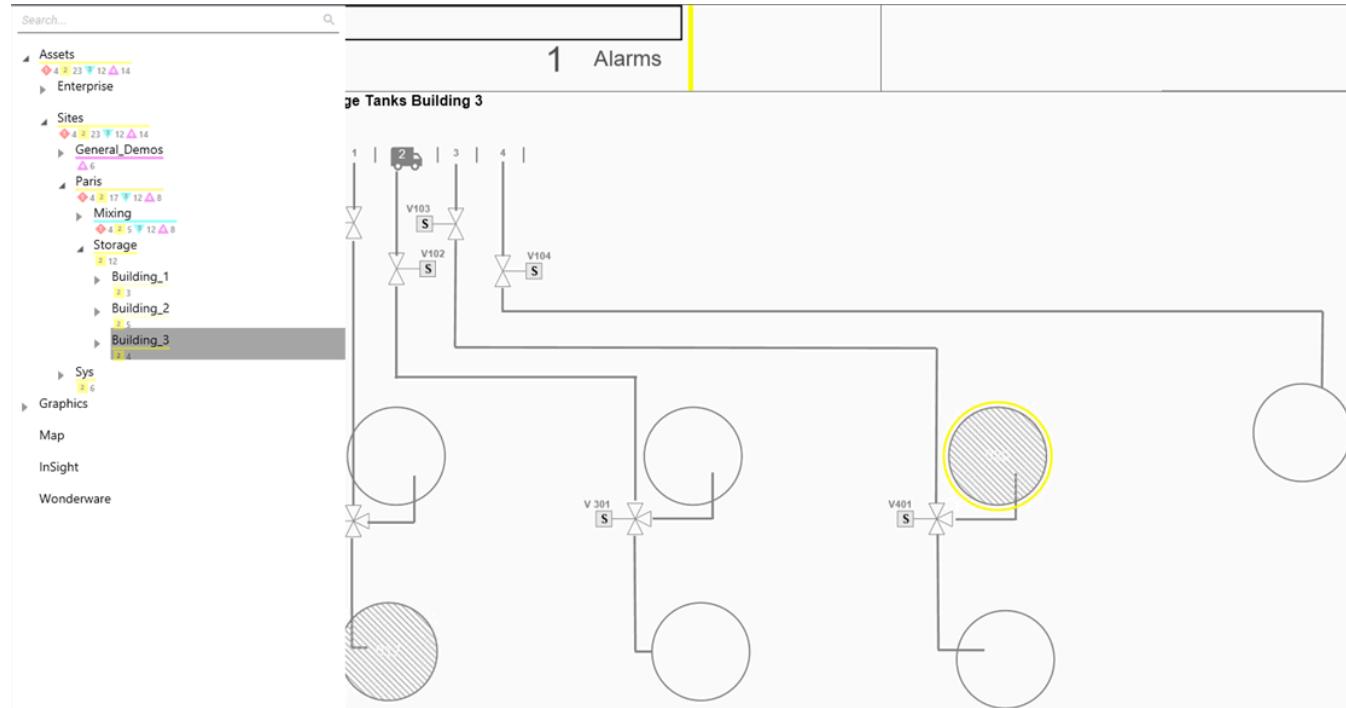
The search results appear in a two column list containing all navigation items that match the search string. The first column displays the navigation item name and Title as <Name (Title)>. The second column displays the path of the item in the navigation model. Highlighting identifies the portion of the item name or path that matches the search string.

If the NavTree control is configured to show alarm propagation states, the aggregate alarm count by priority appears in the search results beneath each navigation item containing active alarms.

fil	
X	
6 navigation items matching 'fil'	
Filters	Home
FiltrationUnits	Home
Filter100	Home ▶ FiltrationUnits
Filter200	Home ▶ FiltrationUnits
Filter300	Home ▶ FiltrationUnits
Filter400	Home ▶ FiltrationUnits

You can change the column widths by dragging the vertical line that separates the columns. Placing the mouse cursor directly over the path shown in the search results list shows the complete path in a tooltip.

Selecting an item from the search results shows its content in the running ViewApp. The NavTree control shows the item you selected.



Alarm Status in the Search Results List

The NavTree control includes the **Show Alarms** property to show or hide aggregated alarm counts by priority for all items of a ViewApp navigation model. The default selected value of the **Show Alarms** property shows an *adorner* or alarm border beneath a navigation item in the search results list whose path has active alarms.



The physical appearance of an alarm adorner in the search results list provides more information about the alarm

- If the color of the alarm border is opaque (solid), the alarm is on the selected (local) object. If the color of the alarm indicator has some transparency, the alarm is on a contained object.
- If enabled, the severity (S1 to S4) and status of the alarm (UNACK_ALM, UNACK_RTN, ACK_ALM) is shown graphically, along with a count for each alarm severity that has been enabled.
- The color of the blinking line beneath the navigation item name indicates the highest-severity (most urgent) alarm.

For more information about the meaning of the physical appearance of an alarm border, see [Aggregated alarm display in OMI ViewApp navigation](#).

Clear Search Results

You can hide the search results list during runtime by clearing the search string from the search entry field.

- Backspace over the search string
- Press the Esc key
- Select the Clear button at the right of the search entry field.

After clearing the search string from the entry field, the search results list disappears and the navigation model appears again.

Process late data at runtime

Data received from the Telemetry Server or other sources may arrive late, especially when using publish/subscribe models such as MQTT, with or without Sparkplug B. "Late data" refers to process data that is received after the time of the actual event. For example, an OMI ViewApp may receive data along with its VTQ timestamp some number of minutes or even hours after the triggering event has occurred.

Application Server allows late data received through a DI Object to backfill historical data in the Historian while presenting live data to OMI ViewApps. Note that value historization is done only for attributes that have the History extension enabled.

When connection to the data source is lost, historical data and alarms are backfilled once the connection is restored. References to ScanGroup items directly in scripts allow late data to be processed. For example, "me.Int1 = DiObject.Scangroup.Item1" assigns late data values to user-defined attribute (UDA) Int1.

Late data received from DI objects referenced through the I/O extension is not backfilled. In the meantime, OMI

ViewApps show the last known good value. The live data feed resumes once the data connection is restored. Historical data is sent directly to the Historian and does not affect the runtime values shown in the ViewApp.

While the Historian is processing the historical data, Application Server uses a Quality and Status animation, "Processing Historical Data," to notify the ViewApp that this is occurring, while the live data feed continues. At the end of the historical data processing, the current data/states/values are determined based on either the live data or, in the absence live data, by using the last-known good values from the historical data. Trend pens, the Trend Client, and any other visualized data that relies on historical data are updated when historization is finished.

The historical data is historized appropriately and can be later queried as if it had been received in order.

Enable historical playback

AVEVA OMI provides the ability for users to review KPI-level information by playing back historical data from an Historian server. Users can switch to playback mode and visualize historical data from historized attributes. Attributes that are not historized, typically static data such as tag names and descriptions, are also shown via a connection to the live data source.

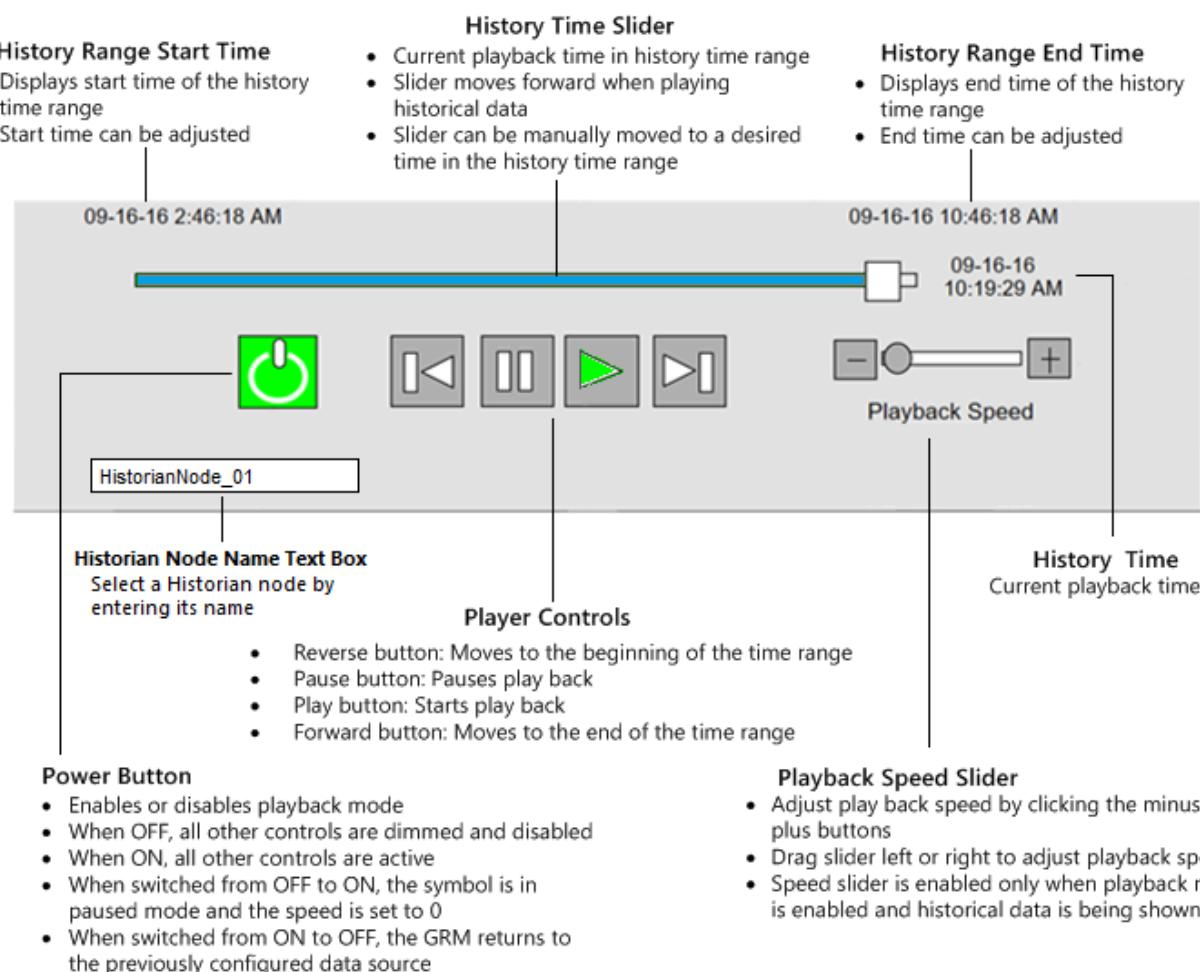
Any non-historized attribute values that are displayed are shown as live data. The appearance of the live data is differentiated from historical data by applying the style associated with the **Live Value Playback** Quality and Status indicator. This "Live Value Playback" Quality and Status indicator is applied to all live values while Historical Playback is active.

Note: Only the highest priority Quality and Style (QS) indicator is shown, and Live Data Playback has the lowest priority of all QS indicators. Therefore, if during playback any other QS indicator is active, live data will not be shown and any non-historized values will be displayed as NaN.

ViewApp Playback attributes can be used with graphic animations or action scripts to provide the functionality to show selected historical data. The graphic's graphic elements incorporate animations or scripts with Playback attributes set as reference values to provide the equivalent functionality of media player controls. For example, you can include a switch graphic element associated with the Boolean **Enabled** attribute that powers on the media player graphic.

To enable historical playback

1. Create a media player graphic in which you associate playback attributes with the various elements of the media player. For example, associate the Boolean **MyViewApp.Playback.Enabled** attribute with the media player power button, and the **MyViewApp.Playback.Playing** attribute with the play graphic. A typical media player graphic looks something like the following illustration, and has similar capabilities, enabled through the attributes provided in the Playback namespace:



2. Add a text box to the media player graphic and link it to the **MyViewApp.Playback.HistorianServer** attribute. Users can enter a node name in the text box to connect to different Historian servers at runtime.
3. Manually deploy the playback service that will play the historical data. See [Manually deploy the playback service](#) for instructions on deploying the playback service.

To configure the Live Playback indicator

1. Select **Galaxy** from the ribbon bar, then select **Configure**.
2. From the list of **Configure** options, select **Galaxy**.
3. Select **Styles** and then select **Quality & Status**.
4. Select the **Live Playback** indicator to configure it. For more information, see the following topics:
 - [Set graphics properties with a galaxy style library](#)
 - [Use quality and status indicators to provide runtime feedback](#)

About playback attributes

Playback attributes are Read/Write and typically used as graphic element animation reference values. When the user interacts with a graphic element's animation or script, Playback attribute values are modified to enable some aspect of showing real time or historical data.

Tags and other static attribute values are generally not historized due to their unchanging nature. When these and any other non-historized static attributes are shown during playback, there are no values that can be pulled from the historical data. Prior to System Platform 2023, non-historized attribute values were displayed as empty or NaN (not a number) in place of the missing data. With System Platform 2023, a live data connection is opened to provide the these missing values. You can configure the appearance of the live data during Playback in the IDE.

If any other Quality and Status indicator is displayed (Communication Error, Security Error, Warning, etc.), live data is not shown for the non-historized values and the missing numerical values are shown as NaN. Other data types (string, date, Bool, etc.) may be shown as empty.

Important: Playback attributes are predefined and available only for AVEVA OMI ViewApps. Playback attributes cannot be included in an InTouch HMI application.

All Playback attributes operate in the MyViewApp.Playback namespace. The names of Playback attributes are specified with the MyViewApp.Playback prefix in the form MyViewApp.Playback.*attribute_name*. To access the Playback namespace, you can reference the SDK assembly ArchestrA.ClientRuntimeData.dll.

Attribute Name	Data Type	Attribute Type	Initial Value	Description
CurrentTime	DateTime	Read/Write	Current playback time	CurrentTime is set to the current playback time when in playback mode. Setting this attribute adjusts the current playback time. The time assigned to CurrentTime can be only within the historical playback period. When the current time is being set and the speed is 0 (pause mode), the client will receive a data update based on the current time of the local system.
Enabled	Boolean	Read/Write	False	When Enabled is set to True, other attributes are initialized to their default values and playback is enabled. When set to False, historical playback is disabled and play real time data.
EndTime	DateTime	Read/Write	Current time	EndTime is

Attribute Name	Data Type	Attribute Type	Initial Value	Description
				<p>initialized to the current time when the Enabled attribute is set to True. Setting this attribute adjusts the ending time of the historical playback range.</p> <p>EndTime can be set to a value regardless of the value set to the Enabled attribute.</p>
HistorianServer	String	Read/Write	Empty string	<p>Node name of the computer hosting the Historian and its historical data. The value assigned to HistorianServer is retentive.</p> <p>If the initial value of HistorianServer is an empty string, the Historian is expected to be installed on the GR node.</p>
Playing	Boolean	Read/Write	False	<p>When Playing is set to True, historical playback becomes active and playback speed is set to 1. The Enabled attribute must be set to True to set Playing to True.</p>
Speed	Integer	Read/Write	0	Historical data playback speed as an integer value. 0 represents playback is stopped or paused. 1

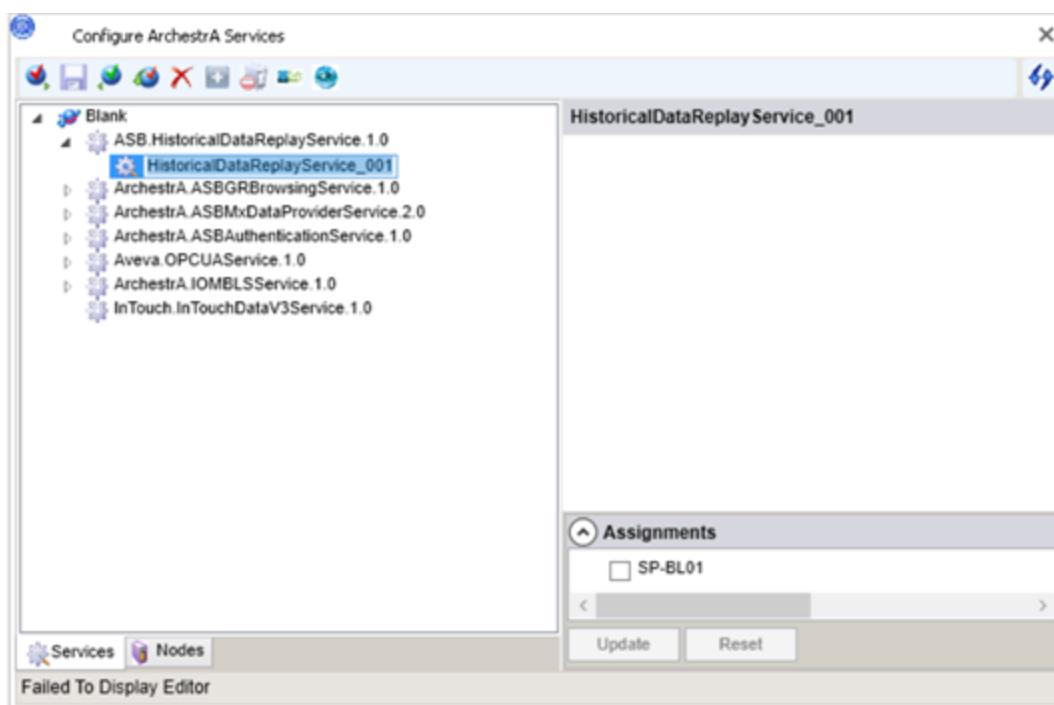
Attribute Name	Data Type	Attribute Type	Initial Value	Description
				represents normal playback speed. Speed values greater than 1 are functionally equivalent to the fast forward function of a media player that plays data faster than it was recorded.
StartTime	DateTime	Read/Write	8 hours prior to the current time	StartTime is initialized to 8 hrs prior to the current time when the Enabled attribute is set to True. Setting this attribute adjusts the starting time of the historical playback range and the current playback time adjusts accordingly. StartTime can be set to a value regardless of the value set to the Enabled attribute.

Manually deploy the playback service

In AVEVA OMI Version 1.0, you must manually deploy the playback service to play historical data from a graphic that incorporates Playback attributes.

To manually deploy the playback service

1. Open the System Platform IDE.
2. Select **Galaxy** from the ribbon bar, then select **Configure**.
3. From the **Configure** options, select **System** and then select **Services**.
4. When the **Services** window opens, select the ASB.HistoricalDataReplayService.1.0 and press **Ctrl + N** to create an instance.



5. Right-click the instance name and select **Rename Node** (or press F2) to rename the instance to the computer name of your Historian Server node.
6. Right-click the instance and select **Deploy** (or press Ctrl + D) to deploy the instance.

Manage OMI applications

This section describes how to manage OMI applications including licensing and entitlements, user authentication and authorization, security, and managing graphics.

This section also describes the behavior of OMI ViewApps and Application Server when running under Operations Control and connected experience. If you do not use Operations Control and connected experience, ViewApps, Application Server, and the remainder of System Platform products will continue to function as they have in previous releases.

Manage product licensing and user authorization

System Platform can run under different types of licensing and security. Originally, System Platform was sold with a perpetual license that allowed customers to use the product for an unlimited length of time. This release of System Platform introduces Operations Control mode, and under Operations Control, additional capabilities for managing user access called the connected experience.

Operations Control uses AVEVA Identity Manager to provide identity service for AVEVA products. The AVEVA Identity Manager helps administrators to implement authentication and allows access tokens to be reused across different AVEVA products.

Connected experience extends Operations Control to enable single sign-on of all your users, and allows centralized user management in CONNECT

The section that follows describes the behavior of System Platform when running under Operations Control and connected experience. If you do not use Operations Control and connected experience, System Platform will continue to function as it has in previous releases.

Connected experience and operations control

Connected experience requires an CONNECT account with a valid Operations Control subscription and user management. Connected experience allows single sign-on to the AVEVA products supported by Operations Control that you are entitled to use. Note that licenses must be installed and activated.

- Unified User Management: Selecting the Connected Experience option enables all Operations Control products on the node to require log in authentication with AVEVA Connect. AVEVA Connect-based authorization is the only security mode available under connected experience.
- Compatibility across nodes: Connected experience must be enabled on all nodes in your system. Applications previously build on nodes not enabled for connected experience must be reconfigured to function in the connected experience environment.

You can disable the connected experience at any time, but if you do, connected experience must be disabled on all nodes in your system. If connected experience is disabled, any applications that were built under connected experience must be reconfigured to function without connected experience, including both authentication methods and product licensing.

Configure operations control and connected experience

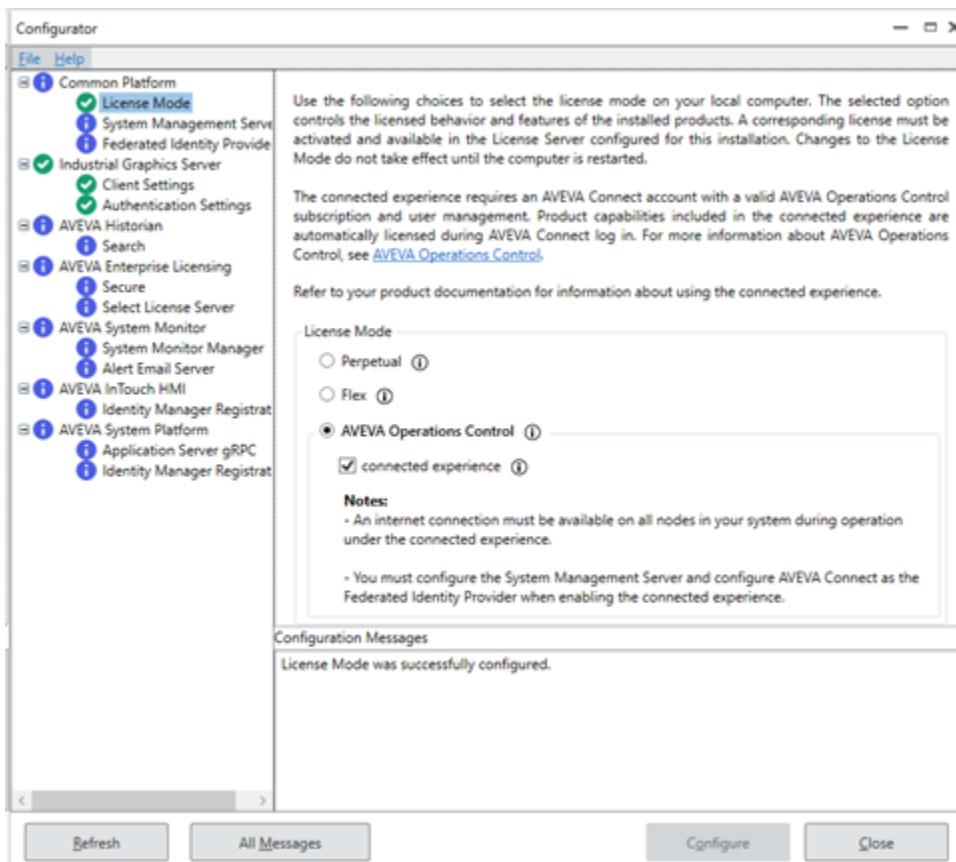
To operate Application Server and other System Platform products under AVEVA Operations Control, either as a product subscription only, or with connected experience, you must configure your installed and licensed products.

For more information refer to:

- [Operations Control Configurator](#)
- [AVEVA Identity Manager Help](#)

To configure Operations Control connected experience mode:

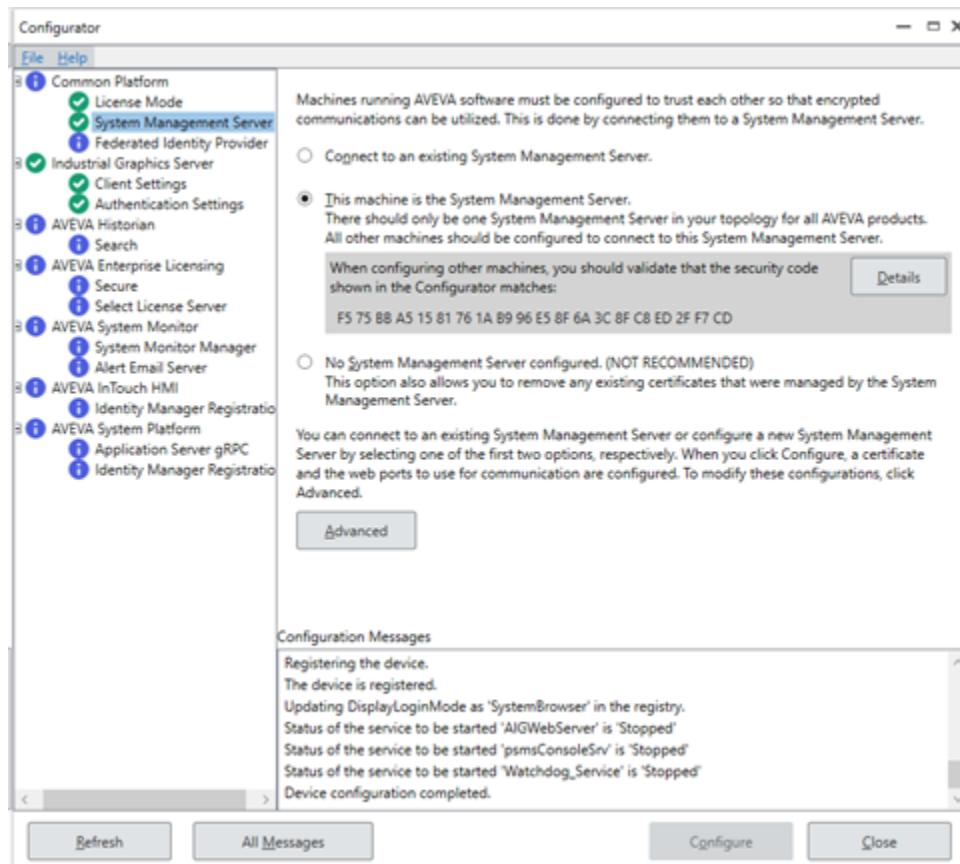
1. Open the Configurator from **Start > AVEVA > Configurator**.
2. Select **License Mode** in the left pane to configure license mode.
 - a. Select the **AVEVA Operations Control** radio button to enable a subscription license for one of the AVEVA Operations Control packages (Edge or Supervisory); that includes unlimited use of all products in the product package for your defined set of users.
 - b. Select the **connected experience** check box to enable a Single Sign-on (SSO) experience across all Operations Control products on this node with AVEVA Connect cloud capabilities, available in the products by default.



3. To configure the System Management Server, in the left pane, under **Common Platform**, select **System Management Server**.

Note: If you are prompted for user credentials for the System Management Server, use the following format to enter the user name: DomainName\UserName. The prompt for user credentials may be displayed if you have domain admin privileges but are not an admin on the local machine. You must be a member of the Administrators or aaAdministrators OS group to configure the System Management Server.

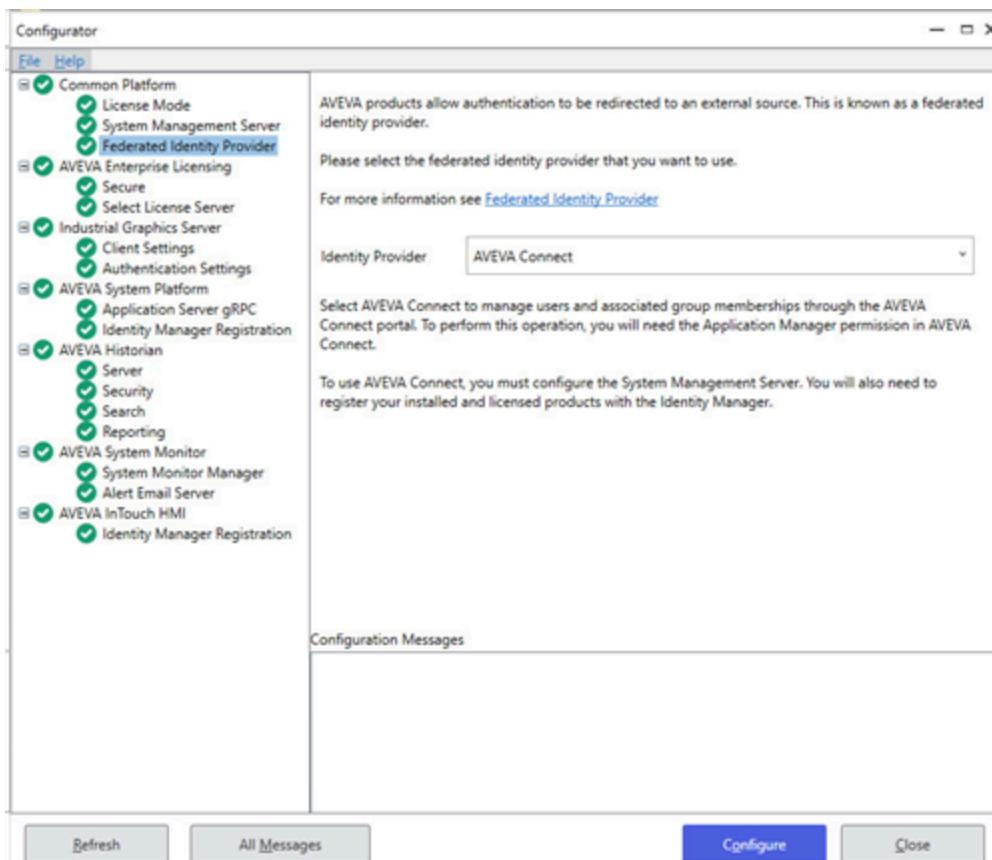
- Connect to a remote server or configure the local machine as the SMS.
- No System Management Server option is **not recommended**. You must configure a System Management Server to use Operations Control and connected experience.
- Only one of the machines in the network is identified and configured as a System Management Server. Machines running AVEVA products can then connect to that single System Management Server to establish trust and configure encrypted communications.
- Configuring a Redundant SSO server does not configure a separate System Management Server. A Redundant SSO server is not required, but provides a backup in case the System Management Server loses access to the external authentication provider.



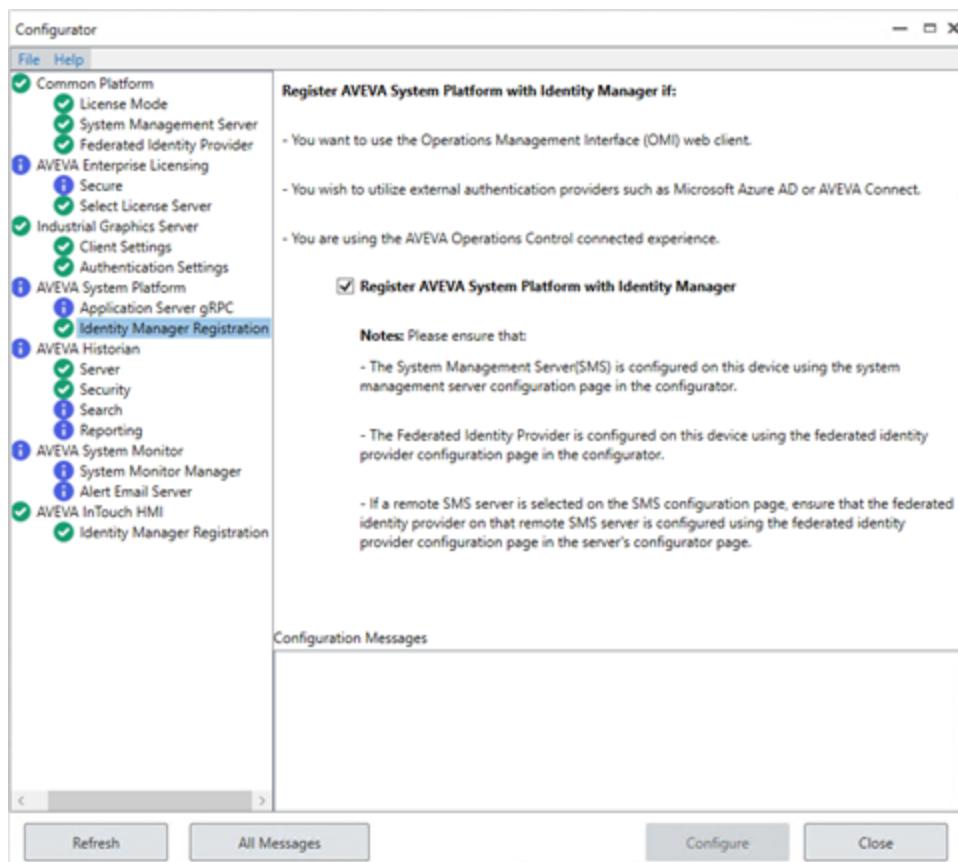
4. Select **Federated Identity Provider** under Common Platform in the left pane to configure a Federated Identity Provider.

The Identity Manager component in the PCS Framework, available on the System Management Server (SMS) and Redundant SSO (RSSO) nodes, can be configured to enable "federated" login with AVEVA Connect. This allows a user to enter the email address associated with their AVEVA Connect account into the Identity Manager login form. Then, they are redirected to AVEVA Connect to complete the log in process.

Note: You must have a valid AVEVA Connect account and be an administrator on that account to configure the Federated Identity Provider.



5. Under **AVEVA System Platform**, select **Identity Manager Registration**. Check that the required configuration requirements are complete. Then, Enable "Register AVEVA System Platform with Identity Manager" and click the **Configure** button. The configuration requirements that must be met before you can register with Identity Manager are:
 - The System Management Server (SMS) is configured.
 - The Federated Identity Provider is configured.
 - If you are connecting to a remote SMS, make sure that the Federated Identity Provider on the remote SMS is configured correctly.

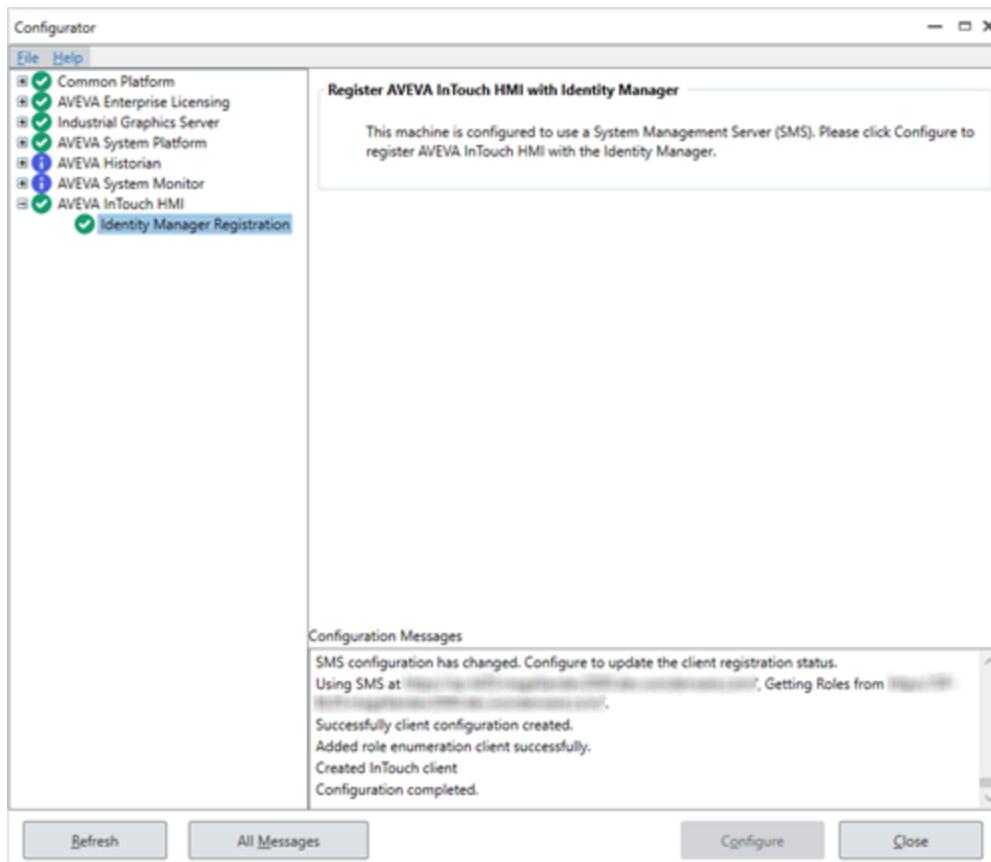


Other Components

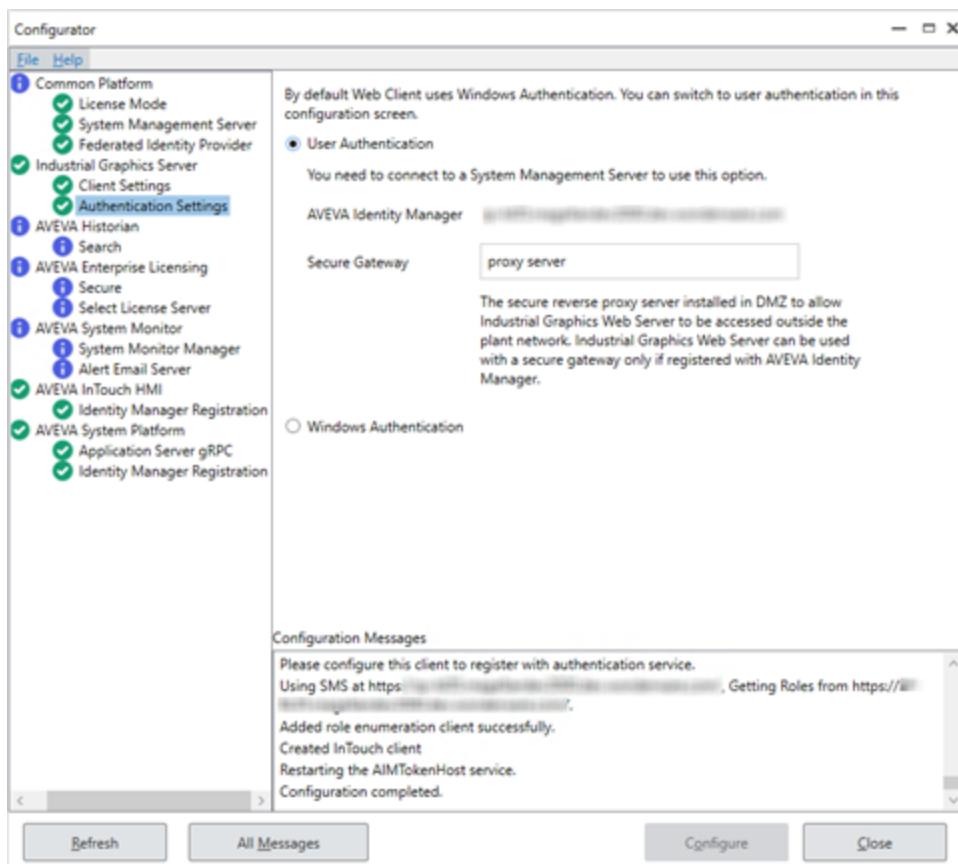
1. To register InTouch with AVEVA Identity Manager, in the left pane, under **AVEVA InTouch HMI**, select **Identity Manager Registration**.

Register InTouch HMI with Identity Manager only when you have completed the preceding steps:

- You have enabled AVEVA Operations Control connected experience as your license mode.
 - You have configured the System Management Server.
 - You have selected AVEVA Connect in the Federated Identity Provider.
2. Click the **Configure** to register with Identity Manager.



3. To Configure the InTouch Web Client to use User Authentication, in the left pane, under **Industrial Graphics Server**, select **Authentication Settings**.



Authentication and entitlement

Authentication and entitlement, along with authorization, are security elements that serve to control user access to the AVEVA products that your organization uses.

- *Authentication* refers to verifying the identity of a user, through the use of a set of unique credentials (user name and password).
- *Entitlement* refers to the products that your organization has subscriptions for, and to which the authenticated user has been granted access. A user, even though they have been authenticated, may not be entitled to use a particular product, if the user does not have a valid Operations Control subscription.
- *Authorization* refers to the level of privileges associated with an authenticated user. This is generally done through the assignment of a user to a user group or through federated identity management.

Federated identity is managed through a *Federated Identity Provider* (FIDP), and allows unified user management and single sign-on for users across the complete portfolio of AVEVA products. This is enabled through the Configurator, and is generally completed at the time of installation but can be reconfigured at any time.

The license mode you select plays an important role in how authentication, entitlement, and authorization are handled within your System Platform environment. See [License mode](#) for more information.

When Operations Control connected experience is enabled, each time a user logs in to a System Platform component, such as the IDE, the Historian, an OMI ViewApp, and so on, the log in is saved as an event in the CONNECT audit log.

View the Data Log

Audit logs let authorized users from both your organization and AVEVA view the log of events or operations that occur in your account from the CONNECT portal. The logs display all operations that have taken place in your organization's account during the selected time period. You must be an Account Administrator, or must have the Report Viewer role to view the audit logs.

Note: Data in the audit logs is available from the time when the Audit functionality is available for your account.

To view the audit logs, go to the [AVEVA Connect](#) web site. On the site navigation menu, select Audit. The Audit log page is displayed.

The audit logs provide the following information:

- Timestamp (date and time) when the operation was performed.
- Details of the operation, such as username, machine name, and the application name (for example, AVEVA OMI or System Platform IDE. The information that is logged includes which product or products were accessed by which authenticated user, as well as their available entitlement access.
- The audit log data can be filtered by using the **Entitlement Check** function in CONNECT. This function provides all data related to product usage through connected experience.

Note: Local time is used while displaying the data on the user interface. However, the UTC time is shown in the operation details and when the data is exported.

See the CONNECT help for more information about [audit logs](#).

Authenticate with the AVEVA Identity Manager

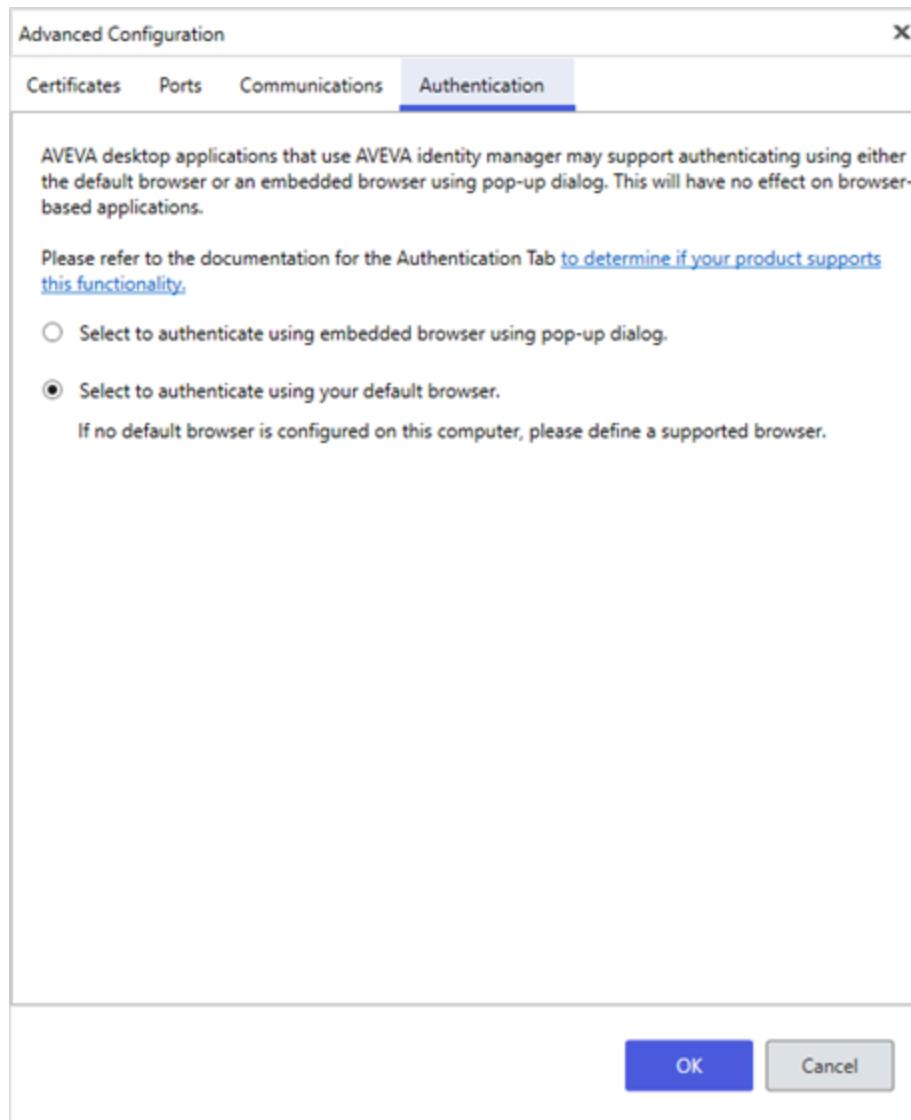
When Operations Control connected experience is enabled in the Configurator, you are prompted to authenticate with the AVEVA Identity Manager when you start your first AVEVA application. Once you are authenticated, single sign-on (SSO) is enabled. With SSO enabled, you can run additional AVEVA applications without having to re-enter your credentials each time.

AVEVA Identity Manager allows two options for signing in, which are set in the Configurator, usually during System Platform installation. This is listed in the Configurator under **Common Platform > System Management Server > Advanced > Authentication**.

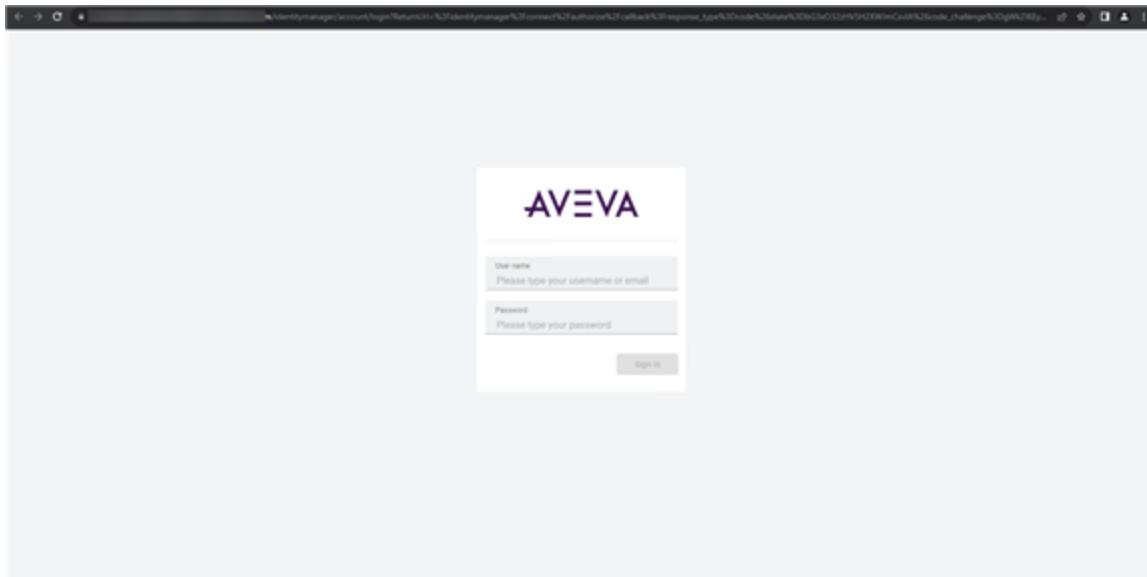
- To use your default browser, choose the **Select to authenticate using your default browser** option.

To authenticate using AVEVA Identity Manager on a web browser:

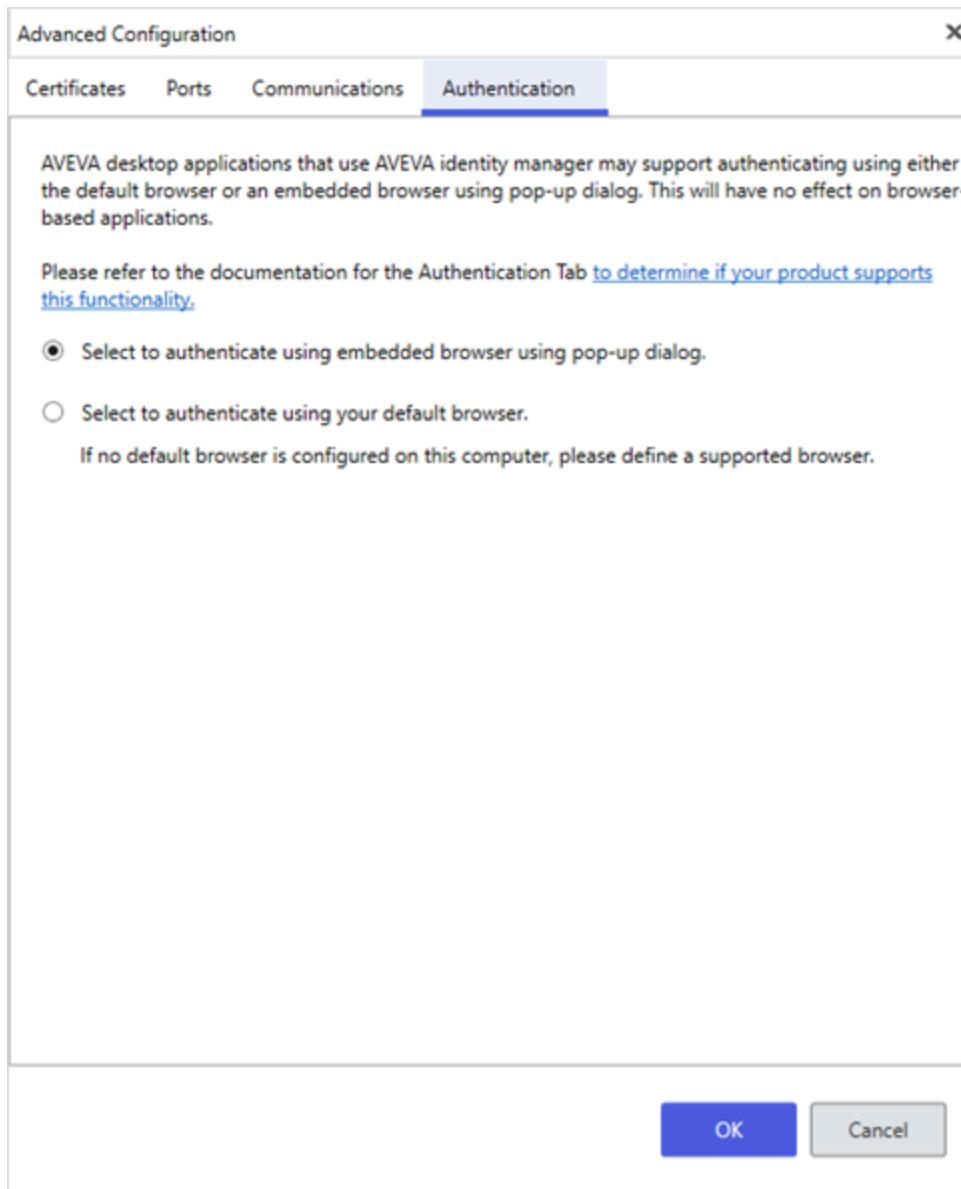
1. Set the option to use the default browser in the Configurator.



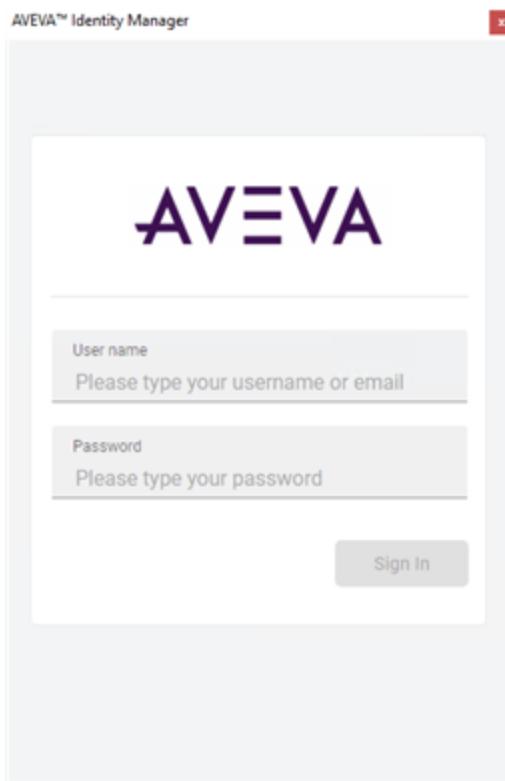
2. When you open an AVEVA Operation Control application, the AVEVA Identity Manager login page opens in the default browser set for your system.
3. In the AVEVA Identity Manager login page, your **User name** is your email address that is registered with an AVEVA Connect account. Enter your email address to login to AVEVA Identity Manager.
4. Next, enter your password in the **Password** field and click **Sign In**.

**To authenticate using AVEVA Identity Manager on an embedded browser:**

1. Set the option to use the embedded browser in the Configurator.



2. When you open an AVEVA Operation Control application, the AVEVA Identity Manager login page opens in the embedded browser.
3. In the AVEVA Identity Manager login page, your **User name** is your email address that is registered with an AVEVA Connect account. Enter your email address to login to AVEVA Identity Manager.
4. Next, enter your password in the **Password** field and click **Sign In**.



Behavior if connection is lost

When you successfully sign in, your user profile which shows your user name and account number, appears in the title bar.

Two different types of connection loss may occur. These are:

- The connection between the application and the AVEVA Identity Manager.
- The connection between the AVEVA Identity Manager and CONNECT.

Lost connection between AVEVA Identity Manager and CONNECT

If, while your application is running, the connection between AVEVA Identity Manager and CONNECT is lost and prevents you from signing in again:

- On-premise capabilities will continue to function.
- Connected experience capabilities will stop working.
- If you try to start another connected experience application on the same node, it may not successfully run until the connection is restored. This applies to another instance of same product or a different Operations Control product.
- You have to sign in again using the Sign in option once the connection is restored.

Lost connection between the application and AVEVA Identity Manager

If, while your application is running, the connection between the AVEVA Identity Manager and the IDE or an OMI ViewApp is lost:

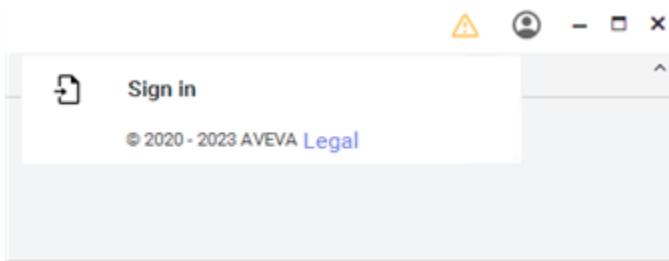
- A warning icon is displayed and provides a notification about the connection state.
- Any open OMI ViewApps will continue to run.
- If you try to open a ViewApp, it will run in read-only mode.
- You will not be able to perform any writes.
- The IDE will continue to function, but you will lose access to any shared Cloud folders.

To address connection loss issues

- If there is loss of connectivity to CONNECT or AVEVA Identity Manager, a warning icon is displayed and provides a notification about the connection state. When you hover the cursor over the warning icon, a tooltip with the connection loss message appears.



- Address any network connection issues. In the IDE, click on the profile icon and sign in again. For OMI ViewApps, you must sign out before you can sign in again.



- When the connection is restored, the warning icon will no longer be visible.

Single Sign-On (SSO)

The connected experience option enables Single Sign-on (SSO) experience across all Operations Control products with CONNECT cloud capabilities.

The connected experience requires an CONNECT account with a valid Operations Control subscription and user management.

Selecting the connected experience option enables all Operations Control products to require log in authentication with CONNECT when starting the first product. Products that are subsequently opened will authenticate using SSO. CONNECT-based authorization is the only security mode available under the connected experience.

Disable connected experience

You can disable the connected experience at any time, but the connected experience must be disabled on all nodes in your system. Galaxies and ViewApps built under the connected experience must be reconfigured to function in an environment that does not have connected experience capabilities, including both authentication methods and product licensing.

License mode

License Mode is set in the Configurator under the **Common Platform** tab. The Configurator lets you select from the following License Mode options:

- **Perpetual:** A specific AVEVA product license purchased for use in perpetuity.
- **AVEVA Operations Control:** A subscription license for at least one of two AVEVA Operations Control packages (Edge, Supervisory); includes unlimited use of all products in the product package for your defined set of users.
 - **Connected experience:** Enables Single Sign-on (SSO) experience across all Operations Control products with CONNECT cloud capabilities. See [About connected experience configuration](#) for more information.

Note that **Operations Control** mode is required in order to enable **connected experience**.

Start and run the IDE

The System Platform IDE in connected experience mode works according to the following steps. These steps assume that the IDE is the first Operations Control application that you are opening. Subsequent applications that you open will not require a separate sign-on (SSO).

1. Open the IDE.
2. You are prompted to authenticate with the AVEVA Identity Manager. For more information, see [Authentication and entitlement](#).
3. Enter your username (email address) and password to sign in to AVEVA Identity Manager. The email address must be registered to an CONNECT account.
 - If the credentials are not recognized, an error message appears. If you made a mistake entering your username and/or password, re-enter your credentials. If you are unable to log in after correctly entering your credentials, contact your administrator for assistance.
 - When you sign successfully, and if you have a current (unexpired) Operations Control subscription available for your account, the IDE opens and your username is shown as the logged-in user.
 - If a valid subscription is not found, the IDE does not open.
4. IDE capabilities are enabled as per the permissions associated with the groups and roles assigned to you as the signed-in user.
5. To connect to a galaxy, see the next section.

Connect to a galaxy

Once you have started the IDE, you can connect to a galaxy. The process for connecting to a galaxy in connected experience mode is similar to the process for connecting to a galaxy where connected experience is not enabled. See [Connect to a Galaxy](#) for more information.

Galaxy migration to support connected experience

When you open an existing galaxy that was configured with the security mode set to Authentication providers (using Azure AD), you must set the security mode to “None” (when in non-connected mode), prior to opening the galaxy in connected experience mode. Connected experience mode is enabled and disabled under **License Mode**, in the System Platform Configurator. See [License mode](#) for details.

Failure to connect to the galaxy

However, the internal system checks are different when connected experience is enabled. You will not be able to connect to the galaxy if:

- Operations Control mode is not enabled on the GR node.
- The galaxy has a security mode that is not compatible with Operations Control connected experience mode. The only compatible security modes are **None** and **Authentication Provider**.
- The credentials you used to log into the IDE do not give you authorization to open the galaxy. In this case, you must restart the IDE and enter the correct credentials.

Successfully connect to the galaxy

The galaxy will open if the above system checks allow you to connect to an existing galaxy or create a new galaxy.

- Your user information will be shown in the IDE.
- AVEVA Connect records that you are using the IDE.

For details about creating a galaxy, see [Create a Galaxy](#).

Connect to a galaxy with no security configured

For compatibility with connected experience, you must set the security mode of the galaxy to either None or Authentication Providers. This applies to both new and existing galaxies.

This section describes what happens after you started the IDE and have connected to a new or existing galaxy. For more information, see:

- [Start and run the IDE](#)
- [Connect to a galaxy](#)

Security mode configuration

1. Log into the IDE as described previously. See [Start and run the IDE](#) for details.
 2. Create a new galaxy, or open a galaxy that does not have security configured.
 3. Go to security configuration. See [Working with Security](#) in the *Application Server User Guide* for details.
 4. Select either **None** or **Authentication Providers**. If connected experience is enabled for System Platform, you are logged in once you start the IDE. You are not prompted to log in again when you connect to a galaxy.
 - If the galaxy security mode is **None**, users will still need to sign in to the IDE before they can connect to the galaxy. Sign in can be explicit or through SSO, if the user has already signed in to another connected experience application. Signing in to IDE is required, regardless of whether or security has been enabled for the galaxy.
- This completes security mode configuration for the galaxy; you can disregard steps 5 through 7.
- If the galaxy security mode is **Authentication Providers**, select the **Roles** tab, and then click the **+** (add new group) button. The list of CONNECT user groups will be shown. Continue to step 5.
5. From the list of groups that are shown, associate IDE permissions to at least one of the groups. These permissions must include:
 - Can Start the IDE
 - User Configuration > Can Create/Modify/Delete Users
 - User Configuration > Can Modify Own User Information
 6. Save the security information.
 7. If prompted, log in again with your username and password.

Connect to a galaxy not using Operations Control connected experience (configuration mismatch)

This workflow describes how to connect to an existing galaxy that is not configured for Operations Control connected experience, when connected experience has been enabled for System Platform.

1. Open the IDE and select a galaxy.

You are notified that the galaxy security mode is not compatible with Operations Control connected experience mode, and that you need to change the security mode to one that is compatible.
2. If you proceed, you are allowed only to change the galaxy security setting to **None** or **Authentication Providers**, as needed.
3. Configure the galaxy security as needed.
4. Save your changes. After the security changes are saved, the galaxy restarts.
5. If prompted, log in again with your username and password.

For additional details, see [Galaxy migration to support connected experience](#).

Sign out of the IDE

The following information applies if you are signed into the IDE in a Operations Control connected experience

environment.

- Once you sign out of the IDE, the IDE closes.
- However, you remain signed into any other connected experience applications, such as OMI ViewApps, that you have signed into.
- You can sign back into the IDE with SSO, provided that you are still signed into other connected experience applications.
- If another user signs into the IDE and you are still signed into other connected experience applications on the same node, you remain as the active user on the other applications.

Start Application Manager

Using Operations Control mode and connected experience has no effect on starting and running Application Manager.

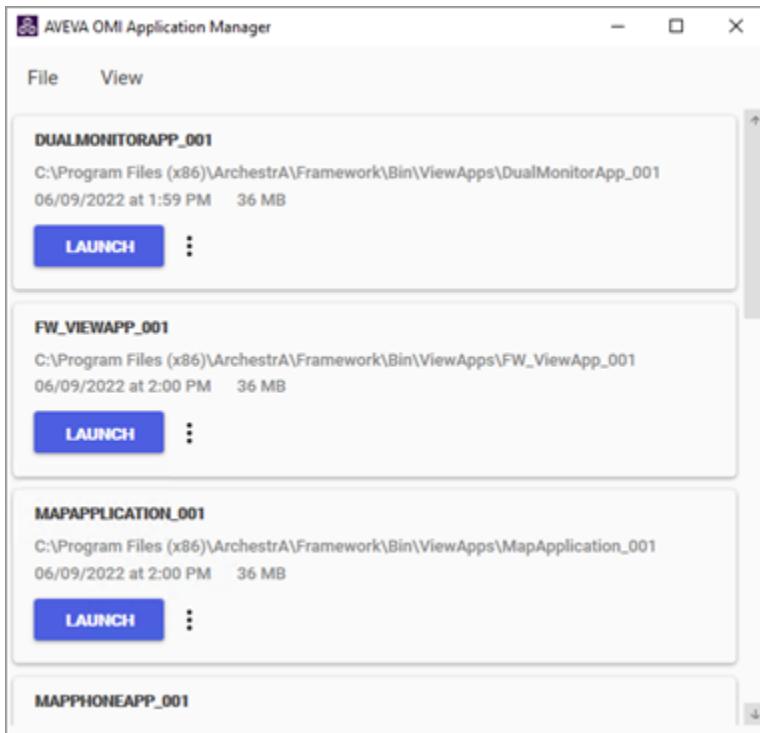
Launch a ViewApp using the connected experience

If your installation uses AVEVA Operations Control and the connected experience for user authorization and security, follow these steps to start an OMI ViewApp.

To launch a ViewApp using the connected experience

- Show the desktop of the computer where the ViewApp is deployed.
- Select the AVEVA OMI Application Manager desktop shortcut.

AVEVA OMI Application Manager appears with an alphabetic list of ViewApps deployed on the computer.



For more information about the fields of the AVEVA OMI Application Manager, see [About the ViewApps](#)

[Application Manager](#).

3. Select **LAUNCH** for the ViewApp you want to run.

- **Already signed in (Single Sign On):** If you are already logged in to another ViewApp or other AVEVA product that uses Operation Control connected experience, you are logged in to the ViewApp automatically using Single Sign-on.
- **Not yet signed in:** If you are not already logged in to a product that uses Operation Control connected experience, a browser window opens for you to enter your user name and password. The browser window is either the System Platform embedded browser or your default browser, depending on what authentication option has been configured for the System Management Server.

After you successfully log in, the ViewApp opens. Your user name is shown near the right side of the titlebar in the ViewApp (but only if the ViewApp includes the OMI TitleBarApp).

If you have a problem logging in (authentication or entitlement failure)

- Make sure your internet connection is working.
- If user authentication fails, the ViewApp opens in read-only mode after you click OK. In read-only mode, you cannot interact with the ViewApp (for example, to change a setpoint or to acknowledge an alarm); you can only monitor what the ViewApp is showing.
- If entitlement fails, you will see a message that a valid subscription could not be found. After you click OK, the ViewApp opens in read-only mode. In this mode, you cannot interact with the ViewApp (for example, to change a setpoint or to acknowledge an alarm); you can only monitor what the ViewApp is showing.

Corrective action

If you receive an authentication error message, repeat the log in procedure, being sure to enter your user name and password correctly. If you still cannot sign in, contact your AVEVA Connect administrator.

If you receive an entitlement error message, there may be a problem with your AVEVA subscription. Contact your AVEVA Connect administrator.

Implementing SSO in your ViewApp

OMI ViewApps leverage the Security.LoggedInUserSsoToken property. OMI Apps consume the user access token via the SDK using security attributes. These attributes are updated whenever the token is refreshed by a user signing in or out.

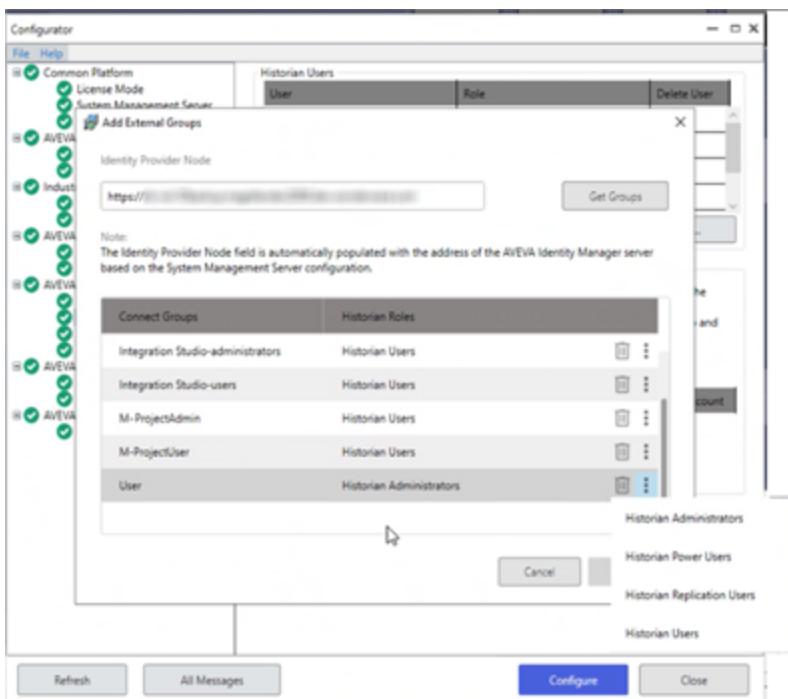
Configure user permissions for connected experience

When connected experience is enabled, ViewApp users will need to be granted additional permissions by granting them either the "Historian Power User" or "Historian Administrator" role.

You can add users to these roles on a Historian node by using the System Platform Configurator.

1. Open the Configurator and select the **Historian Security** plugin.
2. Click the **Add External Groups** button. The **Add External Groups** window opens.

Note: **Add External Groups** is only active if the System Management Server is configured with a federated identity provider, as is the case when .



3. Click the **Get Groups** button in the **Add External Groups** window.
4. Select the group or groups that require additional permissions (roles).
5. Add the appropriate role or roles to the group(s).

Launch ViewApp in the OMI Web Client using connected experience

If your installation uses AVEVA Operations Control connected experience for user authorization and security, follow these steps to start an OMI ViewApp.

To launch a ViewApp in the OMI web client using the connected experience

1. In a supported web browser, go to the URL for the OMI web client:
 - On the machine where the WebViewEngine is deployed, the URL is <https://localhost/omi>. You can also use either of the two formats below.
 - On a machine on the same domain as the one where the WebViewEngine is deployed, the URL is https://<node_name>/omi, where <node_name> is the name of the platform where the WebViewEngine is deployed.
 - On a machine that is not on the same domain as the WebViewEngine, the URL is <https://<fqdn>/omi>, where <fqdn> is the fully-qualified domain name of the platform where the WebViewEngine is deployed.

If you are already logged in to another AVEVA product that uses Operation Control connected experience, you are logged in to the OMI web client automatically using Single Sign-on. Skip to step 3.

2. If you are not already logged in to a product that uses the connected experience, a sign-in window opens for you to enter your user name and password.
After you have entered your user name and password, the OMI web client page opens.
3. The OMI web client page lists all of the OMI ViewApps available through the web client. Select the ViewApp to run.

When the ViewApp opens, the user name that you are logged in as shows in the top-right corner of the page.

If you have a problem logging in (authentication or entitlement failure)

- Make sure your internet connection is working.
- If user authentication fails, the ViewApp opens in read-only mode after you click OK.
- If entitlement fails, you will see a message that a valid subscription could not be found. After you click OK, the ViewApp opens in read-only mode.

Corrective action

If you receive an authentication error message, repeat the log in procedure, being sure to enter your user name and password correctly. If you still cannot sign in, contact your CONNECT administrator.

If you receive an entitlement error message, there may be a problem with your AVEVA subscription. Contact your CONNECT administrator.

Sign out from an OMI ViewApp

You can sign out from an OMI ViewApp by clicking the **Sign Out** option under the **Profile** icon. When you sign out of an ViewApp:

- Operations Control connected experience supports running multiple applications with different users signed in at the same time.
- Sign out closes the user session from AVEVA Connect.
- Inactivity time-out is not supported when connected experience is enabled. Users must actively log out of the OMI ViewApp to end their connection.
- When you sign out of a ViewApp, you are only signed out of that specific ViewApp. Other ViewApps and connected experience applications will continue to run with their current signed in user.
 - The ViewApp from which you signed out continues to run, but in read-only mode. No user information is shown in the title bar.
 - If another user signs into that ViewApp, the AVEVA Connect login dialog is shown. Once the user is authenticated, the ViewApp is loaded again, based on the security mode of the associated galaxy. The information for the new user is shown in the title bar. The information for the new user is now used for SSO.
- When you sign out of a desktop application using the embedded browser, web applications will continue to run.
- When you sign out of a web application, all web applications will close.
- Closing an application does not do anything with the session. Only that instance of the application gets closed. The session is still on the AVEVA Connect server. When you reopen that application, your session is available and you will be authenticated with Single Sign On.
- When you sign out of the IDE, the session is closed and the IDE exits.

Switch OMI ViewApp users

To change the user signed into an OMI ViewApp:

1. The current user signs out from the ViewApp.

This action logs the user out of the ViewApp. However, the ViewApp reloads under the context of the logged-in Windows user, and is now running in read-only mode. The TitleBarApp in the ViewApp does not show any user information.

2. A new user now signs into the ViewApp. The user uses their AVEVA login credentials to sign in.

The new user is authenticated, and a message is logged in AVEVA Connect for this user.

3. The OMI ViewApp reloads, based on the security mode in the associated galaxy.

The TitleBarApp in the ViewApp now shows the new user as the signed-in user.

Secured and verified writes

Operations that require you to log in every time you use them include secured writes, verified writes, and acknowledging alarm states. The behavior of these features does not change, whether or not you are using the connected experience.

To perform a secured write in Operations Control mode

1. Change a runtime value. The secured write dialog is displayed.
2. Enter your AVEVA username and password. The AVEVA Identity Manager authenticates you and the value change is allowed.

To perform a verified write in Operations Control mode

1. Change a runtime value. The verified write dialog is displayed.
2. Enter your AVEVA username and password. The AVEVA Identity Manager authenticates you.
3. The verifier clicks on the authentication button and enters their username and password.
4. Both identities are verified and the value change is allowed.

Deploy scenarios

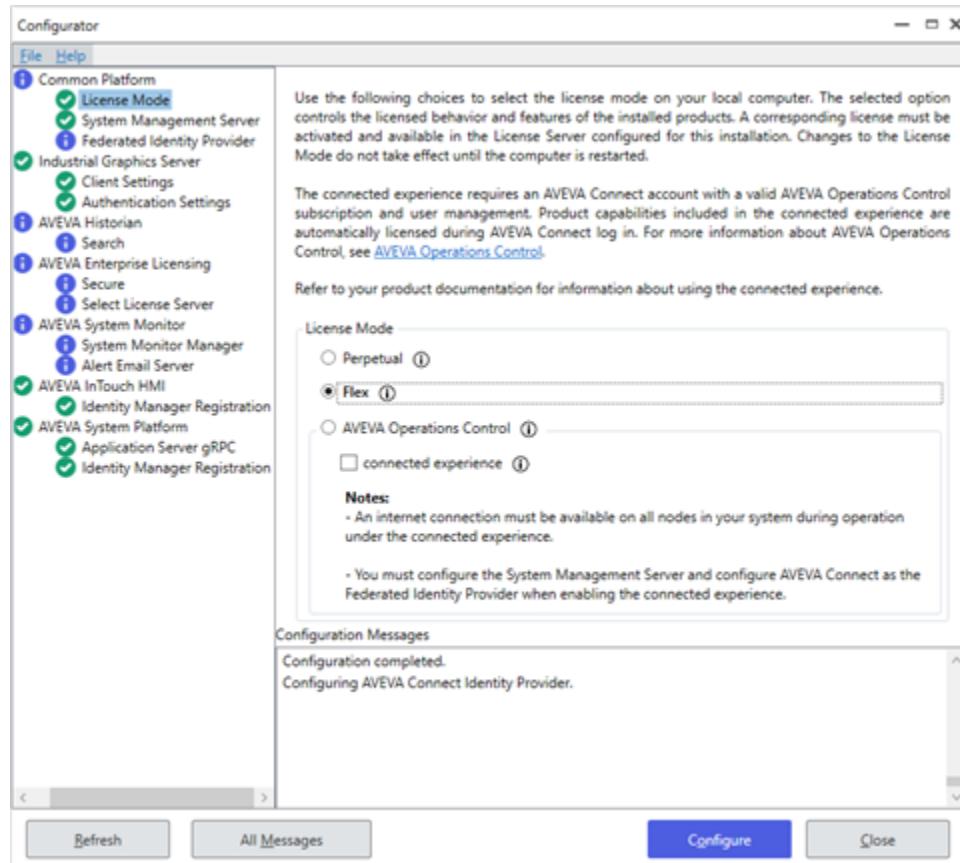
Deployment in an Operations Control connected experience environment

- As long as all nodes are configured for Operations Control connected experience, you can deploy a galaxy that has been enabled for Operations Control connected experience from the GR node to a remote Application Object Server node, provided that the license types match.
- If your GR node is not configured for Operations Control connected experience, but the remote node is, you cannot deploy objects to that node. The opposite case is also true: you cannot deploy from a GR node configured for Operations Control connected experience to a remote node that is not similarly configured.

Change Operations Control connected experience mode

If System Platform is using Operations Control connected experience mode, you can switch out of this mode if necessary. You will also need to change the security settings for any galaxies that use Operations Control connected experience. However, keep in mind that you will have to turn off Operations Control connected experience on all nodes, not just the GR or IDE node.

The license mode change is done through the Configurator. To change the license mode, you can simply uncheck the connected experience option, if you will still be using Operations Control mode. Or, you can change the license mode from AVEVA Operations Control to either **Perpetual** or **Flex**, depending on the type of license you have. Note that a system restart is required before any change to License Mode will take effect.



1. Now that you have changed the license mode, restart the GR.
2. Start the IDE. If the IDE is on a different node, you must change the license mode on this node as well. The IDE is no longer under Operations Control connected experience.
3. Connect to a galaxy. Note that the galaxy is still configured for Operations Control connected experience (using either **None** or the **Authentication providers** security setting).
4. Log into the galaxy as administrator.
5. Optional: Change the galaxy security model to the type of security you want to use.
6. Configure user credentials as needed to ensure that you and other legitimate users can still access the galaxy.
7. Save your changes and redeploy the galaxy.

Using the OCMC with connected experience

When using the Operations Control Management Console (OCMC) with AVEVA Operations Control and the connected experience, you are not required to log in when you start the OCMC itself. Instead, you are asked to log in with your user name and password the first time you open a node under Platform Manager in the tree pane of the OCMC.

If you have previously logged in to another Operations Control application, or to the OCMC, you will be authenticated using Single Sign-on and will not have to log in again.

Galaxy migration to support connected experience

If the galaxy security mode is Authentication providers (using Azure AD), the security mode has to be set to "None" (when in non-connected mode), prior to opening the galaxy in connected experience mode. Use the following procedure to migrate an galaxy to use connected experience if:

- You have a galaxy created or modified under System Platform 2023.
- The galaxy is configured with the **Authentication providers** security mode using Azure AD.
- You want to use Operations Control - connected experience.

This procedure is applicable to both fresh installs of System Platform 2023 R2 and upgrades from System Platform 2023 to System Platform 2023 R2.

To migrate an existing galaxy that uses Authentication providers security

1. Install System Platform 2023 R2 (fresh install) or upgrade your installed System Platform components to System Platform 2023 R2, as applicable.
2. Set the **License Mode** (under Common Platform) to **AVEVA Operations Control**.
3. Leave the **connected experience** option disabled.
4. Use the IDE **Import** function to migrate the existing galaxy to System Platform 2023 R2. The galaxy **Security mode** remains set to **Authentication providers**.
5. Connect to the galaxy.
6. Change the **Security mode** to **None**.
7. Open the Configurator and enable **connected experience**.
8. Connect to the galaxy.
9. Change the **Security mode** to **Authentication providers**.

Note: If you enable connected experience initially (see step 3, above), you will not be able to connect to the galaxy. This is because connected experience uses AVEVA Connect as the federated identity provider, while the galaxy is using Azure AD as the identity provider. In this scenario, when you attempt to connect to the galaxy, you will not be permitted to sign in.

The connected experience requires an AVEVA Connect account with a valid Operations Control subscription and user management.

- Enabling connected experience on a node modifies the behavior of all Operations Control products.
 - User authentication will use AVEVA Connect. After the user authenticated, all subsequent Operations

Control products are enabled for single sign-on.

- AVEVA Connect-based authorization is the only security mode available under the connected experience.
- The connected experience must be enabled on all nodes in your system. Applications previously built on nodes not enabled for the connected experience must be reconfigured to function in the connected experience environment.

About Single Sign-On and the connected experience

- [Single Sign-On \(SSO\)](#)
- [Signing on using the connected experience](#)
- [Signing out using connected experience](#)
- [Connected experience configuration settings](#)

IDE related workflows

- [Start and run the IDE](#)
- [Connect to a galaxy](#)
- [Connect to a galaxy not using Operations Control connected experience \(configuration mismatch\)](#)
- [Connect to a galaxy with no security configured](#)
- [Change Operations Control connected experience mode](#)
- [Sign out of the IDE](#)
- [Deploy scenarios](#)
- [Using the OCMC with connected experience](#)

Runtime workflows

- [Launch a ViewApp using the connected experience](#)
- [Launch ViewApp in the OMI Web Client using connected experience](#)

Signed on using the connected experience

When using AVEVA Operations Control and the connected experience, you may need to log in with your user name and password during several operations when using either the development features of Application Server, such as the IDE, or its runtime applications.

When using the connected experience, the user name you log in with must be a valid AVEVA Connect account name.

The Login window

When you need to log in, Application Server opens a browser window. Depending on an option chosen when configuring Operations Control, this may be a window in your system default browser, or it may be a pop-up dialog displayed by an embedded browser provided with Application Server. In either case, it will prompt you for your user name, then your password.

When you need to log in

In some cases, such as when you start the Application Server IDE or when you open an OMI ViewApp, you only need to log in the first time you open one of these applications. When you then open a second application, or re-open an application you've closed, Application Server uses Single Sign-on (SSO) to automatically sign you back in. However, after a long enough period of inactivity, SSO times out and you will need to log in again.

Other features require you to log in every time you use them. These are generally features which require you to log in whether or not you are using the connected experience. They include secured writes, verified writes, and acknowledging alarm states.

Signing out using connected experience

At times, you may want to log out of System Platform-related product that you are using. This can be especially important if other people can access the same computer, as Single-Sign-on could allow them to access information and perform actions under your user name instead of theirs. Depending on your security procedures, you may need to log out every time you step away from your workstation.

WARNING: When you log out of any AVEVA Operations Control product that uses the connected experience -- the Application Server IDE, an OMI View App, and so on -- you are logged out of **all** products that you are using on that computer. Before logging out, be sure that you have completed your work in all of the products and applications that you are using.

How to log out

Each System Platform application has its own procedure for logging out and, if desired, back in when using the connected experience. here is how to log out of the most common applications:

In the Application Server IDE:

- Select the **Log out** prompt at the top right of the window.

Note: The IDE does not support inactivity timeouts, whether or not connected experience is enabled.

In a desktop OMI ViewApp:

- Select your user name at the top right, then select **Logout**. The **Logout** prompt changes to say **Login**.
- Inactivity time-outs are not supported when using connected experience. Users must actively log out of the OMI ViewApp to end their connection.

In an OMI web client ViewApp:

- Select **Logout** at the top right. You are logged out and the login window opens so you or another user can log back in.
- Inactivity time-outs are not supported when using connected experience. Users must actively log out of the OMI web client ViewApp to end their connection.

Licensing and the OMI web client

AVEVA System Platform licensing controls which products your installation has access to and the number of users who can access them. This topic describes how licensing works with the OMI web client.

Each deployed WebViewEngine consumes one license. This allows an unlimited number of users to run OMI ViewApps deployed on that WebViewEngine. Each user has read/write access in each ViewApp they use. The process of acquiring and renewing the license is as follows:

1. When a WebViewEngine is deployed, it tries to connect to the License Server to acquire a license.
2. Whether it acquired a license or not, it continues to try to connect to the License Server periodically, either to acquire a license or to refresh the license that it has.
3. As long as the WebViewEngine succeeds in acquiring and refreshing its license, all ViewApps deployed to it continue to function normally.
4. If the WebViewEngine fails to acquire or renew a license--for example, if it cannot connect to the License Server--a two-hour countdown begins. During this countdown period:
 - Users can continue to use any open ViewApp deployed on the WebViewEngine. ViewApps will continue to function as normal, with no change in functionality.
 - Users will not see any message or other indication anywhere in the web client that the ViewApp does not have a license.
 - The WebViewEngine continues to try to acquire a license periodically.
 - A warning message is sent to the logger every fifteen minutes about the failure to acquire a license and saying that the OMI web client is in demo mode.

If the WebViewEngine acquires a license before the two-hour period is over, ViewApps return to normal operation. Users will not have seen any difference in operation.

If the WebViewEngine does not acquire a license within the two-hour period, it stops processing new requests for either data or content. This means:

- Any open ViewApp will not receive any new data from external references (User-Defined Attributes).
- Users will not be able to write to any external references. Writes to custom properties and ViewApp namespaces will continue to work.
- Users will see an error message and the action will fail if they try to access the web client home page, open any other ViewApp, or refresh the page of an open ViewApp. The exact error message varies.
- The warning messages sent to the logger change to say that demo mode has expired.

The WebViewEngine continues to try to acquire a license after demo mode has expired. If it is able to contact the license server and acquire one, it resumes normal operation.

Note: If you redeploy the WebViewEngine during or after the demo period, and it cannot acquire a license, it restarts the two-hour demo mode period. For uninterrupted service, you should ensure that the License Server always has enough valid licenses for all deployed WebViewEngines.

About ViewApp security

Security is the responsibility of almost everybody in an organization who works with ViewApps. Designers must

incorporate security standards in objects and graphics. Builders set security requirements for each instance of a ViewApp. Control room operations staff must maintain security for ViewApps running in a production environment. Line operators interact with ViewApp security as part of their daily tasks.

Different roles within an organization perform different security workflows.

ViewApp Designer	ViewApp Builder
Manage Security Certificate Verification	Manage Security Groups
Configure Security	Manage Security Roles
Implement Log in Security	Manage Security Users
Log on with Security Enabled	Security in Graphic Scripts
Client Access Rules and Galaxy Security	Configure Security Attributes
Set Object Security	Configure Navigation Security
Configure Security for Graphics	Set Up a Map Server
Group Locking Security	Configure a Web Control
OPC Communication Driver Security	Configure Custom Navigation Properties
Locking and Security	Working with the SignedWrite() Function for Secured and Verified Writes
	Securing Client Controls
Control Room Operators	Line Operator
Set Deployment Security	Signature Security for Acknowledging Alarms
Configure User Information	User Access and Security During Run Time
Create User Credentials	Gather the Information You Need to Log On to InTouch Access Anywhere
Create Named Credentials	Shelve and Unshelve Alarms at Runtime
Manage Credentials	Acknowledging Alarms with Signature Required
	Gather the Information You Need to Log On to InTouch Access Anywhere

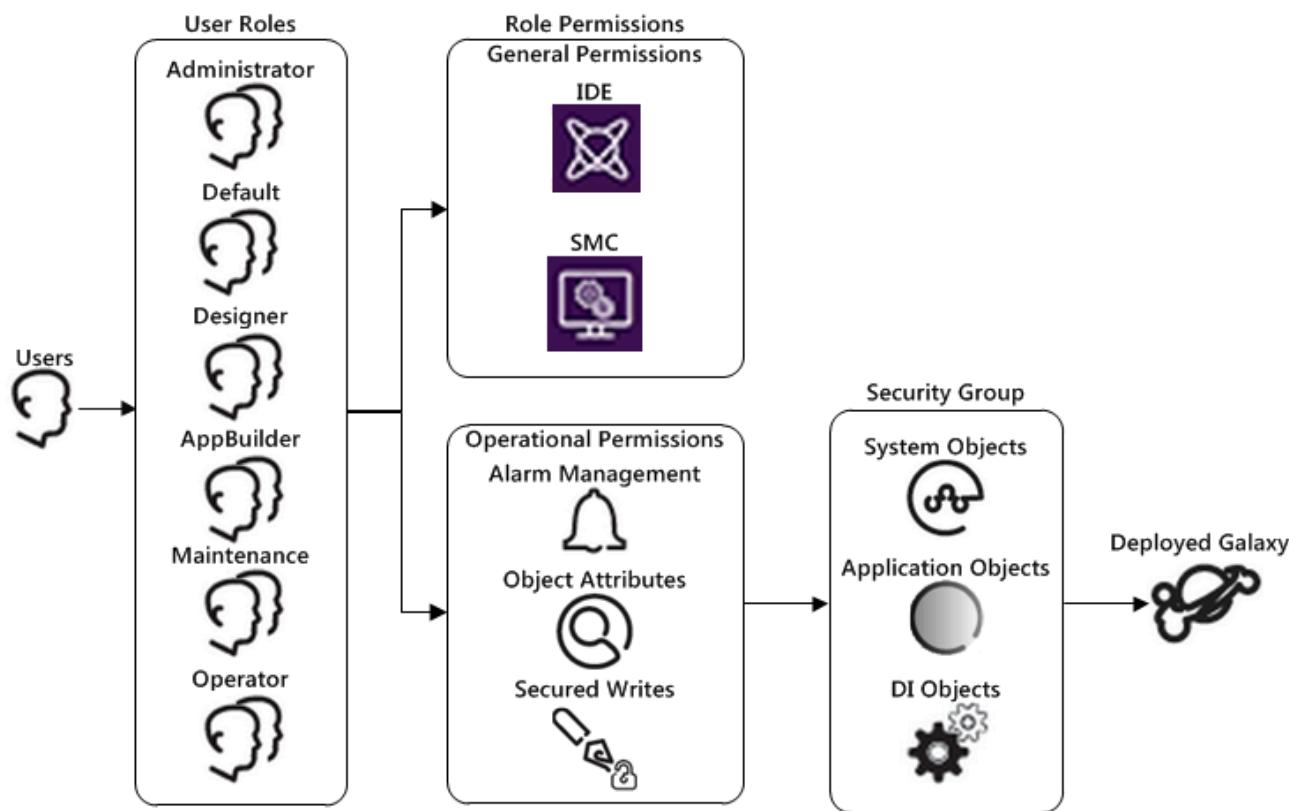
About Galaxy security

Galaxies are created without security. After creating a Galaxy, you can assign security to manage access to:

- IDE for configuring and managing objects and graphics
- Operations Control Management Console for performing maintenance and system administration functions
- Alarms, object data, and secured writes for a running ViewApp
- Credentials, which are linked to specific user groups and allow AVEVA OMI apps to connect to external resources that do not recognize Windows credentials. The security authentication mode must be set to OS Group based or OS User based in order to use credentials.

Each Galaxy in the Galaxy Repository manages its own security model. The security schema managed in a Galaxy is a three-level configuration model to create and maintain the following:

- Users assigned to roles
- Security roles granted system administration, configuration, and runtime (operational) permissions
- Security groups containing specific objects in a Galaxy



The security matrix defines a cascading model of users associated with specific roles that are associated with specific security permissions that are associated with specific objects. User runtime permissions can vary from object to object, action to action, and process to process. The security icons associated with object attributes map directly to control points in the security model.

Configure Galaxy security

Galaxy security uses the local Galaxy configuration to authenticate users. All security for the Galaxy is specified and contained at the specific Galaxy level. When the user logs on, security credentials are checked and access to areas and activities is granted at the Galaxy level.

Regardless of the security system, a single user can have multiple accounts. For example, a user can have an account that provides permissions for working with instances but not templates. The same person can have another supervisory account for working with templates and managing users in the ArchestrA environment. Each account requires a different user name and password. For example:

User Name	Password	Access
bsmith	password	Instances, not templates
bobsmith	super	Instances, templates, not managing users
Robertsmith	admin	Instances, templates, managing users

After you change security for a Galaxy, you see the following behaviors and conditions:

- When you change the authentication mode security, the IDE restarts.
- To switch users, the person must log on as the new user by clicking **Change User** on the **Galaxy** menu.
- If you previously configured security under one authentication mode and then switch authentication modes, only those users created while configuring the new mode are available. Other users are not deleted, just unavailable in the new mode.
- Objects that are reassigned to different security groups are marked as "pending update" and require redeployment for the change in security group to take effect.
- If security was previously configured for an OS-based authentication mode, reconfiguring security synchronizes the user's full name and OS groups if some data in the OS has changed.

About Galaxy security groups

Every object in a Galaxy belongs to only one security group. You create a security group to contain a set of objects in the Galaxy whose attributes and alarms can be managed by users.

By default, all currently used objects are assigned to the Default security group. A user who is a member of the Default security role are granted all Operational permissions.

A user who is a member of a role assigned to the Default security role has permission to:

- Acknowledge alarms
- Change attribute values with "configure" security mode
- Change attribute values with "operate" security mode, including "secured write" and "verified write"
- Change attribute values with "tune" security mode
- Verify writes to Attributes with "verified write" security mode

Manage security groups

Manage security roles

You can create and manage user roles that apply to your organization's processes and work-based authorities. Two roles are defined by default: Administrator and Default.

You can specify General and Operational Permissions for each role.

- General permissions relate to application configuration and administration tasks.

Note: You cannot modify the General permissions of the Administrator role.

- Operational permissions relate to the security groups listed on the Security Groups page. By default, the Administrator has all permissions.

There are five basic Operational permissions that can be granted to a security role

- Manage alarms
 - Can Acknowledge Alarms:** Enables users to manually acknowledge an alarm while a ViewApp is running.
 - Can Shelve Alarms:** Enables users to shelve and unshelve alarms.
 - Can Modify Alarm Modes:** Enables users to modify the mode of an alarm.
 - Can Modify Plant States:** Enables users to modify plant states for state-based alarming.
- Can Modify "Configure" Attributes:** Enables users to configure the attribute's value. Requires that the user first put the object Off scan. Writing to these attributes is considered a significant configuration change, for example, a PLC register that defines a Discrete Device input.
- Can Modify "Operate" Attributes:** Enables users with operational permissions to do certain normal day-to-day tasks like changing setpoint, output and control mode for a PID object, or commanding a Discrete Device object.
- Can Modify "Tune" Attributes:** Allows users to tune an attribute in the runtime environment. Examples of tuning are modifying attribute values that specify alarm setpoints and PID sensitivity.
- Can Verify Writes:** Enables users to provide an authentication signature for attributes configured with Verified Writes security classification. Only users with this permission can verify a task performed by users with the **Can Modify "Operate" Attributes** permission.

Create a security role

Creating a security role consists of two major steps of adding a role to the Galaxy and then assigning permissions to the role.

Notes About General Permissions:

- In the **SMC** permissions list, clearing the **Can Start SMC** check box still enables a user assigned to this role to start the SMC, but not to connect to Platform Manager.
- In the **General permissions** area, clearing the **Can Start/Stop the Engine/Platform** permission still permits the user assigned to this role to set the engine or platform objects On Scan or Off Scan.
- If a role is granted the **Can Modify Deployed Instances** IDE permission, make sure the **Can Create/Modify/Delete...** permissions in the **System Configuration**, **Device Integration Objects**, and **Application**

Configuration groups are also selected. This provides the role with the ability to check in and check out objects.

To create a security role

1. On the **Galaxy** menu, click **Configure** and then click **Security**. The Configure Security dialog box appears.
2. On the **Authentication Mode** tab select **Galaxy**.
3. Click the **Roles** tab.
4. Click the + button to add a security role.

A role is added to the **Roles available** list named **NewRole**.

5. Change the name of the new security role.
6. Select the new security role in **Roles available** list and assign **General** and **Operational** permissions to the role by selecting the checkbox next to each permission you want to add to the role.

Similar IDE permissions are grouped into lists that can be expanded or hidden. Click the small gray box next to the permission groups to expand or hide the permissions.

7. Click **OK** to save your changes and close the **Configure Security** dialog box.

About the default security role

The Administrator and Default users are defined by default when a new Galaxy is created. These users cannot be deleted in an open security setting and are associated with the default Administrator and Default roles.

The Default security role has permission to:

- Acknowledge alarms
- Change attribute values with “configure” security mode
- Change attribute values with “operate” security mode, including “secured write” and “verified write”
- Change attribute values with “tune” security mode
- Verify writes to Attributes with “verified write” security mode

When you create a new user name, it is assigned to the Default role. For more information about creating users and assigning roles, see the IDE help.

About the Credentials tab

The **Credentials** tab lets you add login credentials that some AVEVA OMI ViewApps may need for access to third-party data and applications that do not support standard authentication methods, such as Windows OS credentials, Active Directory, or OpenID Connect.

Initially, a galaxy does not include any credentials. Credentials are created on a galaxy-wide basis, not for individual ViewApps. Although you enter credentials as plain text, the credentials are encrypted when saved, and the credentials are sent to runtime nodes as encrypted data.

Each credential you create is associated with one and only one OS user group. In order to create credentials, OS group-based or OS user-based security must be configured. The credentials you create here are associated with an OS group name. When a user logs into a ViewApp, the user obtains access to credentials that are associated with the same OS group of which the user is a member. For example, if you create credentials that are associated

with the OS group "Operators," a user that is also a member of "Operators" has access to those credentials.

If an app requires credentials to access data, it typically will include a dropdown menu. At runtime, you can select the appropriate credential, that is, a credential assigned to your user group.

Note: Credentials are supported only when **OS User based** or **OS Group based** authentication mode is enabled.

If authentication mode is set to **None** or **Galaxy**, you cannot add or delete credentials, or use them at runtime, even if the credentials were created while the authentication mode was set to OS User or OS Group.

Create a new credential

From the **Configure Security** page, select the **Credentials** tab. The **Credentials** tab is divided in three columns:

- Name
- Type
- Group

To create a new credential

1. Click the **Add Credential**  button. A new credential is added to the **Available Credentials** window.
2. Rename the credential by selecting the default name from the **Name** column and then type the new name.
3. Set the credential **Type** by selecting the down arrow at the right edge of the **Type** column. Type can be one of the following:
 - Username and Password
 - Domain, Username and Password
4. To select the **Group** to associate with the credential:
 - a. Click the ellipses button (...) at the right of **Group** column. The **Select Groups** dialog opens. See [Associate an OS group with a named credential](#) for more information.
 - b. Choose the applicable OS group name from the list of available OS groups, or type the name of an existing OS group.
5. Configure the credential after you have created a credential by entering values for Domain, Username, and Password. See [Configure credentials](#) for additional information.

The screenshot shows the 'Security' section of the AVEVA Operations Management Interface. The 'Credentials' tab is selected. Under 'Named credentials', there is a table with three rows:

Name	Type	Group
Cred1	Username and Password	\Users
Cred2	Domain, Username and Password	DEV\OpsAdministrators

Below this is a 'Details:' section with a 'Change Password' button. It contains a table with two rows:

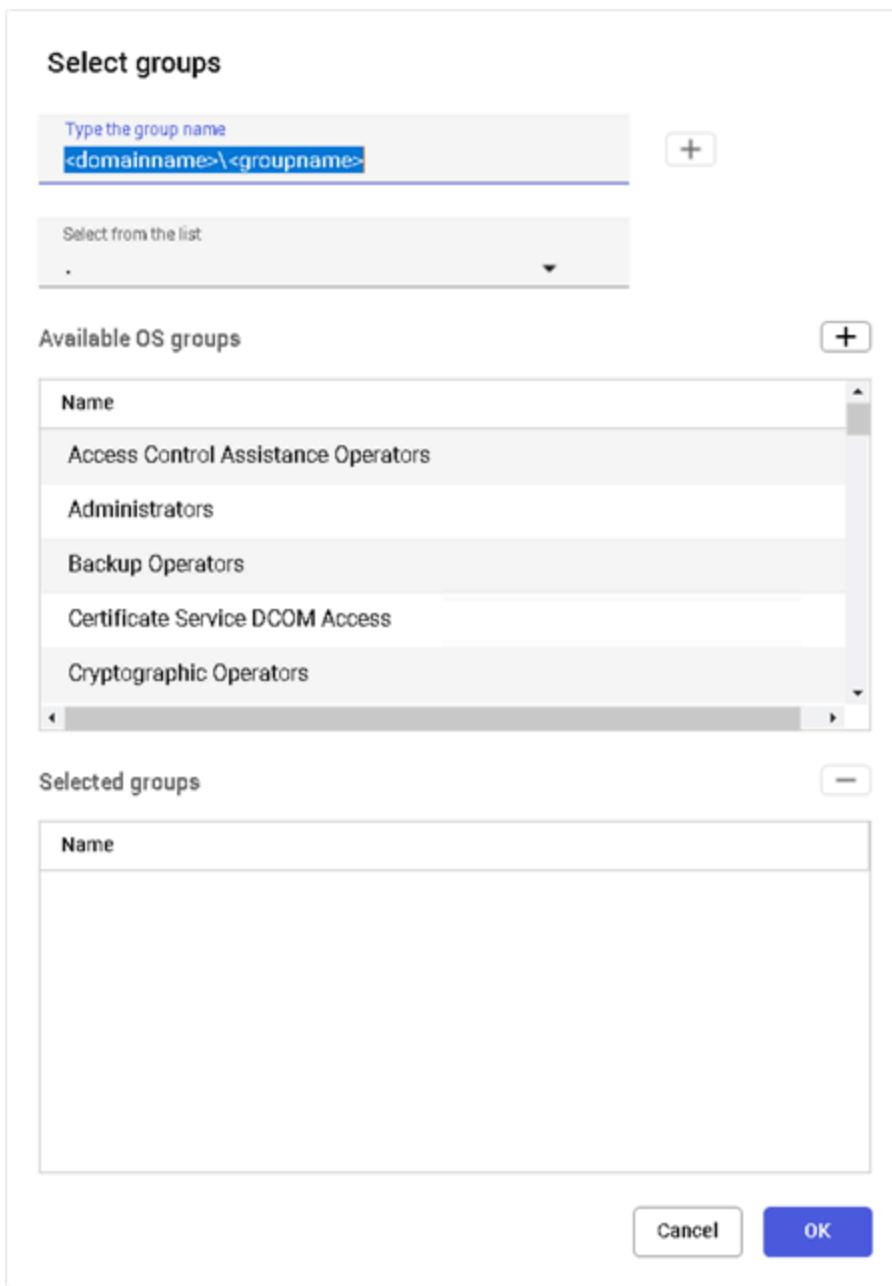
Name	Value
Domain	DEV
Username	Operator1

Associate an OS group with a named credential

To associate an OS group with a credential

1. Select the the ellipses button (...) next to the group name in the **Credentials** tab.

The **Select groups** dialog opens with the local machine selected, and the list of OS groups that exist on the local machine will populate the list of **Available OS groups**.



2. Choose an existing OS group from the **Available OS groups** list. Or, type the domain and OS group name in the **Type the group name** text box.
3. Add the OS group by clicking the + button above the list, on the right side. The OS group is added in the **Selected groups** window.
 - If you are creating a credential that includes domain, select the applicable domain from the **Select from the list** dropdown list, then select the OS group name from the **Available OS groups** window. If the credential type you created does not include domain, you will not be able to select another domain.
 - If you manually enter the domain and/or the OS group name, use a backslash to separate the domain from the group name.
 - If the OS group is on the local machine, you must include ".\\" before the group name.

Note: Only one OS group can be added per credential.

4. Click **OK** to close the **Select groups** dialog and return the **Credentials** tab.

Configure credentials

To configure a credential

1. Select the credential to be configured in the **Named credentials** list.

Name	Type	Group
Cred1	Username and Password	\Users
Cred2	Domain, Username and Password	DEV\OpsAdministrators
Cred3	Username and Password	\Performance Log Users

Details:	
Name	Value
Domain	DEV
Username	Operator1

2. In the **Details** window, enter the **Username** for the credential. If the credential type is Domain, Username, and Password, also enter the **Domain** name.
3. To enter the password for the user, select **Change password**.
4. On the **Change password** dialog box, enter and confirm the password for the user name you entered. Select **Save**.

You can enter null values for the username and password. If a domain value is required, enter the domain name that was used to create the credential.

5. Select **Save** to save the credentials.

Credential definitions

Name: This is the name of the credential. The default credential name is "Credential," and subsequent names are appended by a number that automatically increments by one. To rename a credential, just highlight the name and enter a new one. Credential names must contain at least one character and must be unique.

Configured credentials are typically shown in a dropdown menu in a ViewApp that has been configured to use credentials. The user can then select the applicable credential. In some cases, an app developer may choose to hard-code a credential name. In this case, no user action is required to access the credential at runtime.

Type: Two types of credentials are supported:

- **Username and Password.**
- **Domain, Username and Password.**

Note: Null values are valid for all fields, except the credential name.

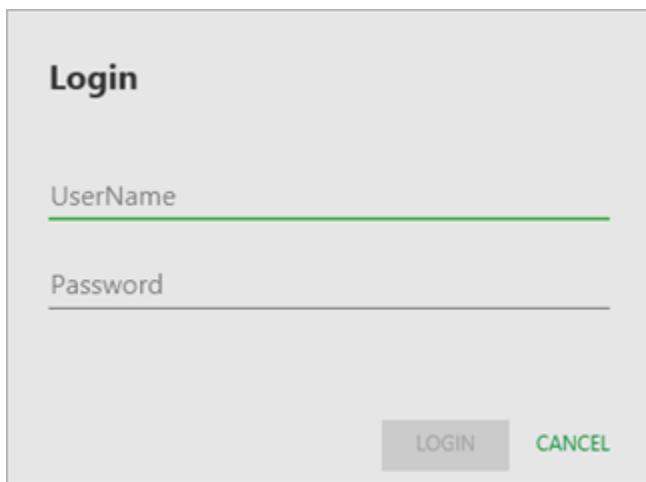
Group: When you add credentials, you must associate each credential with an OS Group name (local or domain) that has been defined previously.

To associate a credential with an OS group, simply select the group name when you add the credential. You can change the group at any time in the **Credentials** tab, but doing so may impact AVEVA OMI apps that have been configured to use the credential. Associations are saved when you click **OK**. See [Associate an OS group with a named credential](#) for more information.

Apply log in security

Log in security can be applied to restrict access to a ViewApp to only authorized users. With security, operators or other users must authenticate themselves with their username and password. Based on how the ViewApp is configured, operators may be able to interact with a ViewApp even though they have not logged in. When an operator logs in they can perform Operate Level writes. Also, operators can perform any actions granted to them on the basis of logging in to a ViewApp..

You configure a ViewApp to show an initial **Login** dialog box when a ViewApp is deployed and started. Typically, an operator will be presented with a button to select and show the **Login** dialog box.



Apply log in security example

This topic describes a simple example of implementing login security interface to permit only authenticated users access to a ViewApp.

Add Users to a Security Authentication System

To implement ViewApp login security, users must be an authenticated member of an AVEVA OMI security system. System Platform AVEVA OMI provides three security modes:

- Galaxy security
- Windows User-Based security
- Windows Group-Based security

Based on the authentication mode you select, ViewApp users must be added to the security system with a user

name and password. The user should be assigned a role in the security authentication system with a defined Access level.

Build a Login Interface

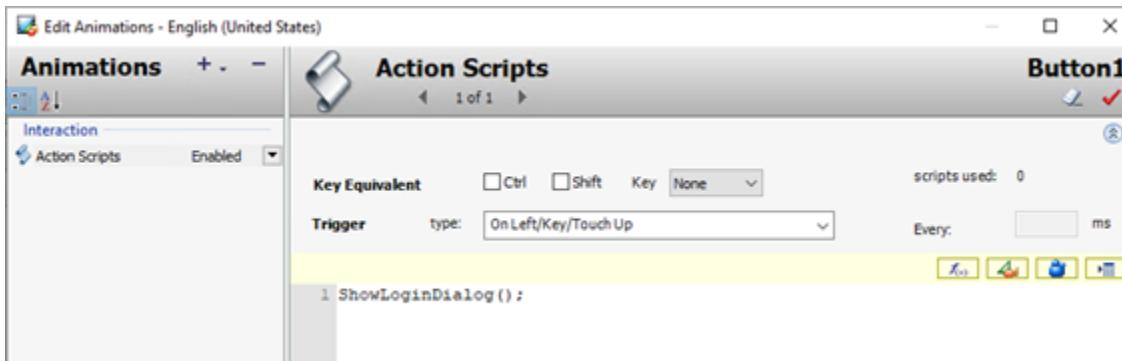
A typical login interface includes a login button that is selected by the user to show the **Login** dialog box with fields to enter a username and password. Also, a log off button can be added for the user to log off from the ViewApp.

Login	MyViewApp.Security.LoggedIn MyViewApp.Security.LoggedInUserName MyViewApp.Security.LastSuccessfulLogin	OFF MM/dd/yyyy hh:mm:ss
Logout		

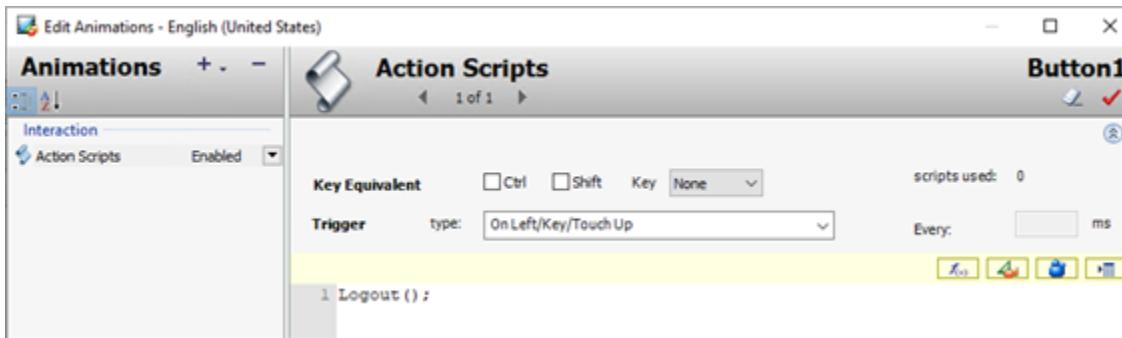
This example of a login interface includes a table that shows the current value of the login security attributes. The attribute status column includes text elements with value display animation mapped to the login security attributes shown in the left column of the table.

Add Action Scripts to the Login and Logout Buttons

- The Login button should include an action script calling the ShowLoginDialog() method, which shows the **Login** dialog box.



- The Logout button should include an action script calling the Logout() method, which logs the user out of the ViewApp.



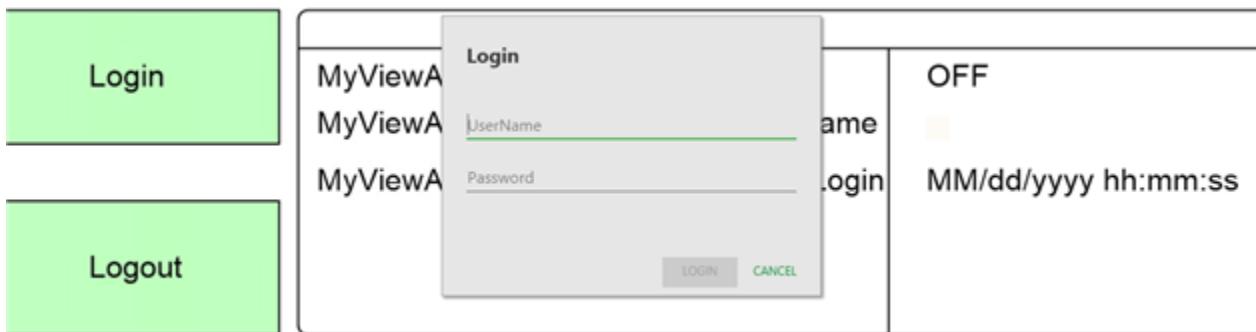
Use Security Attributes with Value Display Animations

Add Value Display animation to a graphic element that you want to use to show security information. Configure the animation to use the Security attributes as reference values.

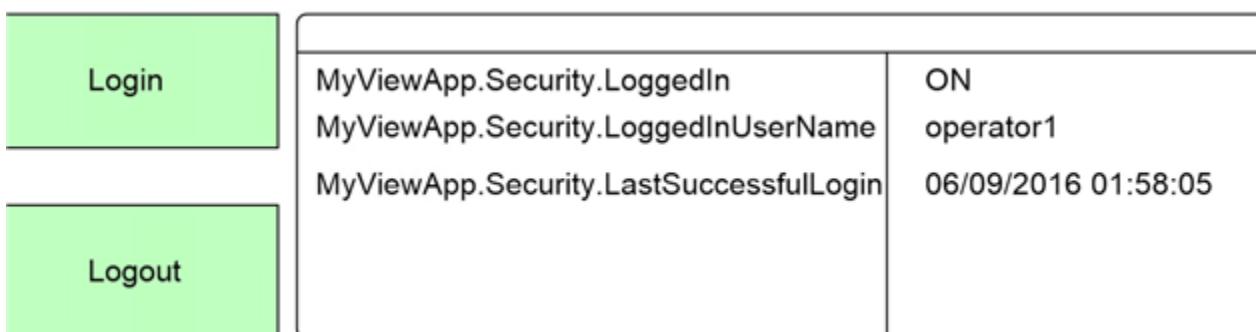
For information about the animation data types to assign Security attributes, see [About security attribute data types](#).

Test the Interface

Run the ViewApp that includes a login security interface. After selecting the Login button, you should see the **Login** dialog box.



After entering a valid user name and password, the **Login** dialog box should disappear and the security attributes should show that a user is logged on, the user name of the logged on user, and the time when the successful log on attempt was made.

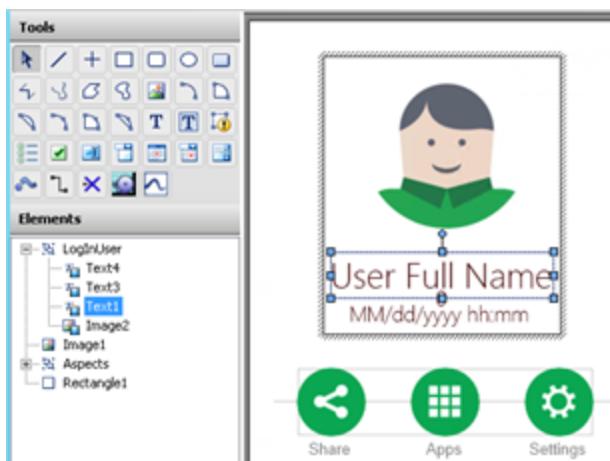


About security attributes

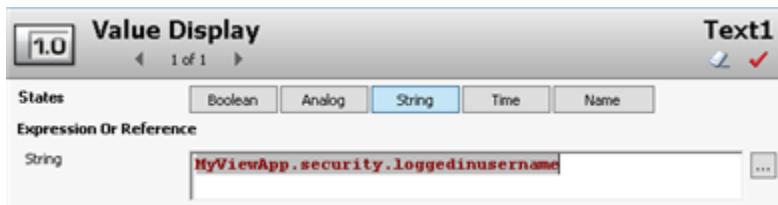
Security attributes are read only attributes that are typically set as reference values of Value Display animation. After a user logs on to a ViewApp, the graphic elements configured with Security attributes show the authorization characteristics of the logged on user.

Security attributes can show the logged on user's user name, and Access Level. Also, the LastSuccessfulLogin and LoggedIn attributes show the date and time of the last successful logon to a ViewApp and if a user is currently logged on to a ViewApp or not.

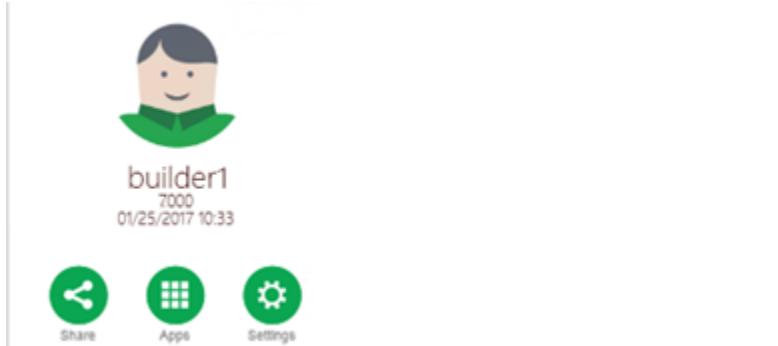
Graphic element to show user name:



Value display animation:



ViewApp (runtime):



Configure security attributes

This topic explains the general sequence of steps to implement a Security attribute in a graphic.

Note: Security must be enabled and users defined in the security system to be able to show user security characteristics as reference values of Security attributes.

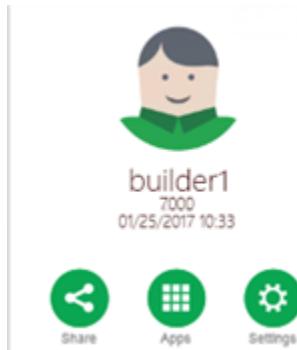
To configure Security attributes

1. Place a text box within a graphic.
2. Using a mouse or touch, select the text element to show the **Animations** dialog box.
3. Select the **Add Animation** button to show the list of possible animations.
4. Select **Value Display** animation as the type of animation to show the current time in the text box added to the graphic.
5. Select the data type and enter the fully qualified Security attribute name.



In this example, **String** is the data type for the **LoggedInUserName** attribute. For more information about the data types associated with each Security attribute, see [More Information About Security Attributes](#).

6. Save your changes to the graphic that contains the text element configured with the Security attribute.
7. Deploy the ViewApp containing the graphic.
8. When the ViewApp is running, you should see the user's log in name in the graphic's text field configured with the security attribute.



About security attribute data types

All security attributes operate in the MyViewApp.Security namespace. The names of security attributes are specified with the prefix MyViewApp.Security in the form MyViewApp.Security.*attribute_name*.

Attribute Name	Data Type	Attribute Type	Initial Value	Retentive	Description
AutoLogon CurrentUser	Boolean	Read/ Write	True	True	Returns true or false to indicate if users can log on automatically to a ViewApp with their Windows credentials.
AutoLogOff	Boolean	Read/ Write	False	True	Returns true or false to indicate if a user will be logged off after a period of inactivity. When set to true, the user is logged off

Attribute Name	Data Type	Attribute Type	Initial Value	Retentive	Description
					from the ViewApp when the inactivity period is exceeded.
AutoLogOff TimeSpan	Integer	Read /Write	600	True	Specifies the length of the inactivity period in seconds a logged in user is allowed.
AutoLogOff Remaining Time	Integer	Read Only	0	False	Returns the remaining length of a user's inactivity period in seconds.
Last Successful Login	DateTime	Read Only	1/1/1601 12:01:01.000 AM	True	UTC timestamp of the most recent successful log in to a ViewApp. When no user is currently logged on to a running ViewApp, LastSuccessfulLogin shows 01/01/0001 00:00:00 as the timestamp.
LoggedIn	Boolean	Read Only	False	False	Returns True or False to indicate if a user is currently logged into a ViewApp or not.
LoggedInAccess Level	Integer	Read Only	No initial value	False	Integer value of the Access Level of the security role assigned to the user logged in to a ViewApp.

Attribute Name	Data Type	Attribute Type	Initial Value	Retentive	Description
					When no user is currently logged in to a running ViewApp, such as after an AutoLogOff, LoggedInAccess Level returns -1.
LoggedInUserName	String	Read Only	No initial value	False	Returns the user name specified by the user to log in to a Viewapp.
LoggedInUserRoles	String	Read Only	No initial value	False	Returns the security roles assigned to the logged-in user in CSV format.

Manage industrial graphics

This section describes Industrial Graphics, how they are stored in the ArchestrA environment, and how they are managed from the IDE.

Industrial Graphics are graphics you use to visualize data in an AVEVA OMI ViewApp.

You manage Industrial Graphics from the IDE to:

- Create a new graphic.
- Edit a graphic with the Industrial Graphic Editor.
- Organize graphics within the Visualization folder.
- Duplicate graphics.
- Import and export graphics.
- Delete a graphics.
- Configure security for a graphic's operations.
- Open the graphic in read-only mode with the Industrial Graphic Editor.

Import and export graphics

You can import graphics to and export them from the Visualization folder.

Import graphics from .aaPKG files

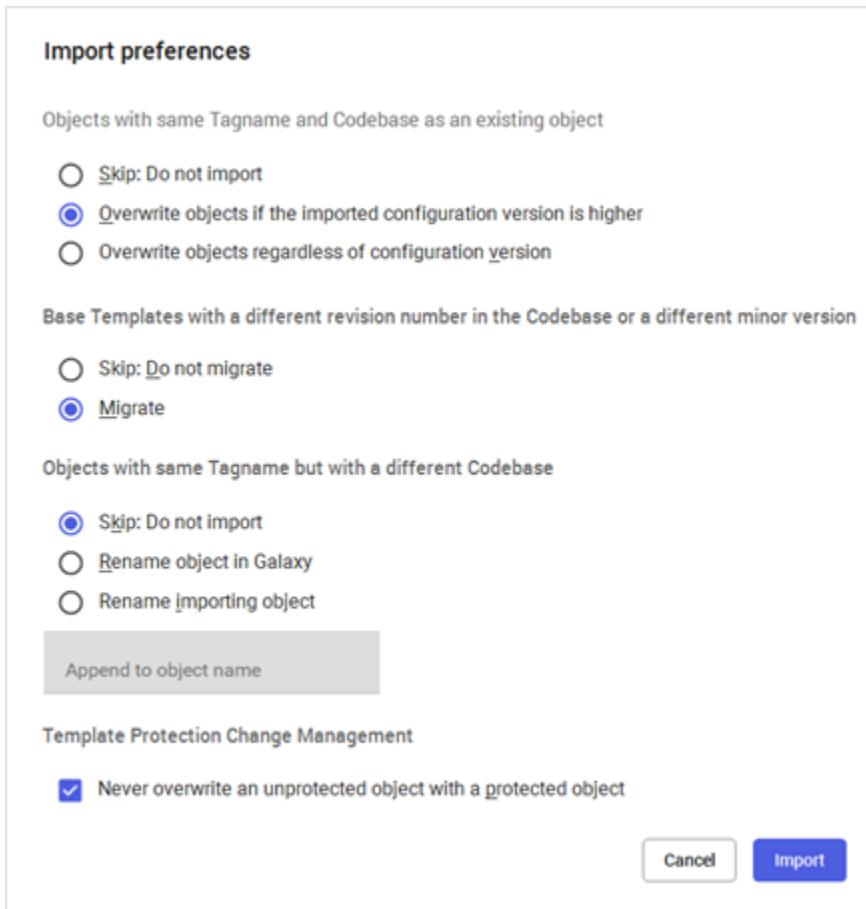
You can import graphics and folders from a graphic .aaPKG file.

When you import templates or instances that contain graphics, the graphics are imported with the template or instance.

When you import all AutomationObjects, the contained graphics and the graphics in the Visualization folder are also imported..

To import graphics from a graphic .aaPKG file

1. On the ribbon, select **Galaxy**, then **Import**.
2. Select **Objects**, then **From package**. The **Import Objects from package** dialog box opens.
3. Select one or more graphic .aaPKG files you want to import and click **Open**. The **Import Preferences** dialog box appears.



4. Select the appropriate options for the import and click **Import**. The graphics and folders are imported.
5. Click **Close**.

Export graphics

You can export graphics to a graphic.aaPKG file.

When you export templates or instances that contain graphics, the graphics are exported with the template or

instance.

When you export all AutomationObjects, the contained graphics and the graphics in the Visualization folder are also exported.

When you export an AutomationObject that contains graphics, if these graphics contain embedded Industrial Graphics from the Visualization folder, they are exported along with the graphics associated with the AutomationObject.

If the graphics contain other AutomationObjects, the graphics, including any embedded graphics from the Visualization folder contained in them, and their parent AutomationObject are not exported with the AutomationObject. They remain as references. Upon import, the system tries to reestablish the connection with those exact references in the target Galaxy.

To export graphics to a graphic .aaPKG file

1. In the Visualization folder, select the graphics to export.
2. On the **Home** ribbon, in the **Export** area, select **Selected**, then **As package**.

If you are exporting a graphic containing embedded graphics associated with an object, you see the **Export Selected objects** dialog box with an option to include the associated automation objects hosting the graphics to be exported.

3. Select whether to include the hosting objects or just the selected graphics, then click **Export**.
The **Export Selected objects** dialog box appears.
4. Browse to the save location and type a name for the graphic .aaPKG file.
5. Click **Save**. The graphics and their folders are exported.
6. Click **Close**.

Programmatically import and export industrial graphics

You can use the Industrial GraphicAccess application programming interface (API) to programmatically export a graphic from the Visualization folder to an XML file. You can use the same API to import a graphic from an XML file to create Industrial Graphics in another galaxy or overwrite an existing graphics. You can also import a graphic XML file created by another application to an ArchestrA Galaxy.

The programmatic API exports or imports an extensive set of properties of an Industrial graphic. For standard Industrial Graphics, an exported or imported graphic contains the following:

- Custom properties
- Graphic elements
- Connector lines
- Graphic groups
- Graphic animations
- Element styles
- Named scripts
- Predefined scripts
- Overridden text strings
- Numeric format styles

- DataStatus elements
- Trend Pen
- Alarm Client
- Trend Client

In addition to the properties of a standard Industrial graphic, an exported or imported Symbol Wizard contains the following:

- Wizard Options
- Choice groups
- Choices
- Layers
- Rules

Implement and use the GraphicAccess API

The set of GraphicAccess application programming interfaces (API) allow users to export graphics to an XML file programmatically. Use the same set of APIs to create an Industrial Graphic in another galaxy or to overwrite an existing graphic(s) by importing the graphic definition from an XML file. The APIs can also be used to export the configured references or GPI for a graphic and importing multiple XML files from a folder location.

Before using the set of GraphicAccess APIs, you must use the GAccess Toolkit to log in and connect to an Galaxy repository.

The APIs are available through the following interfaces:

IGraphicAccess – Export and import an Industrial Graphic

IGraphicAccess2 – Export the substitutable strings and configured references for an Industrial Graphic

IGraphicAccess3 – Export the GPI of an Industrial Graphic

IGraphicAccess4 – Import Industrial Graphics from a folder location

For details about how to use these APIs, see [IGraphicAccess](#) in the System Platform help file.

Overridden text string import and export

The ArchestrA programmatic API supports overridden text strings in graphics. Typically, a string substitution is performed on a text string from the Industrial Graphic Editor before exporting the graphic with the programmatic API. The exported XML file shows the content of overridden and substituted text strings as elements and attributes of the SubstitutedStrings element.

When a graphic is imported using the programmatic API, the graphic shows the text string assigned to the New attribute of the XML file's String element. Users can substitute text strings before importing a graphic by editing the contents of the String element in the exported XML file.

Export overridden text strings

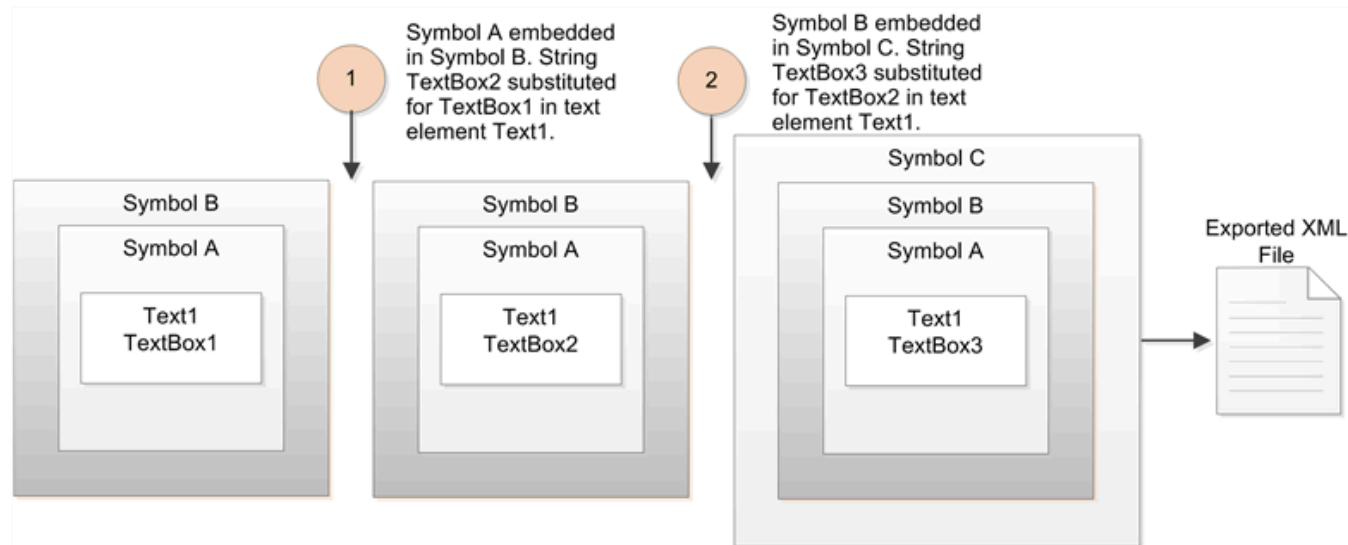
After exporting a graphic, the XML file's String element specifies the original string and the new substituted string with the Old and New attributes. The ElementID attribute specifies the specific text element to be

overridden.

```
<SubstituteStrings>
<String Old="TextBox1" New="TextBox2" ElementID="A.Text1"/>
<SubstituteStrings>
```

In this example, the TextBox1 text string was overridden by the new substituted text string TextBox2. The string substitution occurred in Symbol B that substituted the text of the text element in Symbol A.

Text strings can be overridden within graphics that are embedded within other graphics. Consider an example that shows Symbol A embedded in Symbol B, which is embedded in Symbol C. Two string substitutions are made to a single text element.

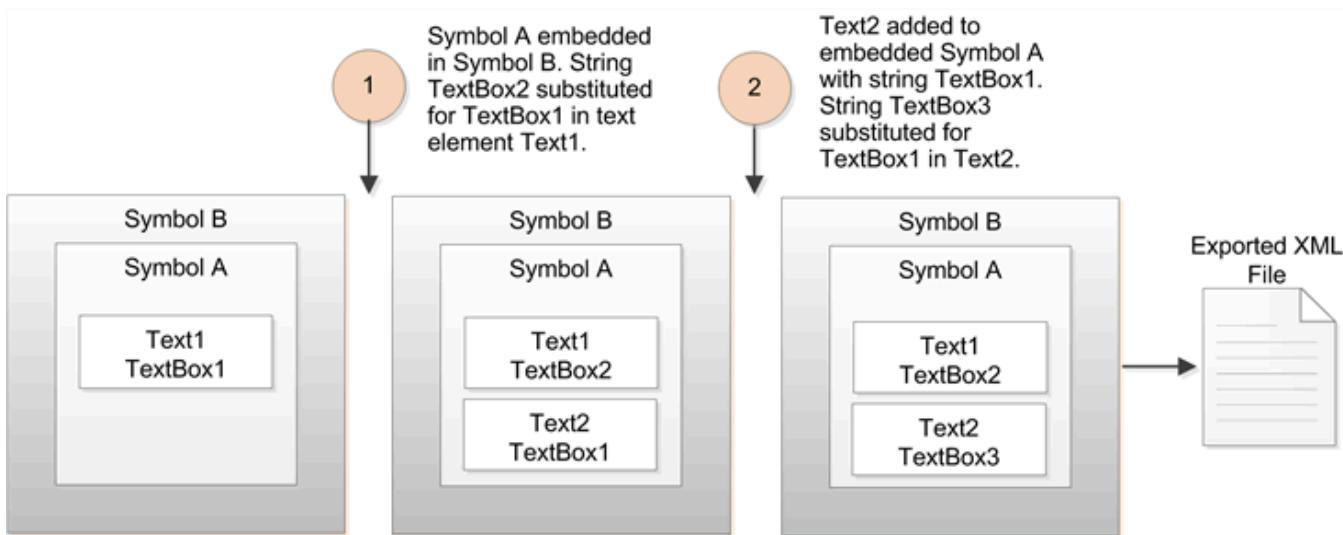


The export XML file's **SubstituteStrings** element shows the text string override information.

```
<SubstituteStrings>
<String Old="TextBox2" New="TextBox3" ElementID="B.A.Text1"/>
<SubstituteStrings>
```

Notice the **Old** and **New** attributes show the text strings of the most recent string substitution before Symbol C was exported. Also, the **ElementID** attribute indicates Symbol A containing text element Text1 is embedded in Symbol B in the form **ElementID="B.A.Text1"**.

The programmatic API can export an embedded graphic containing multiple text elements with the same text string that is overridden by different string substitutions. Consider an example that shows Symbol A embedded in Symbol B. Two string substitutions are made to a single text string in two text elements of embedded Symbol A.



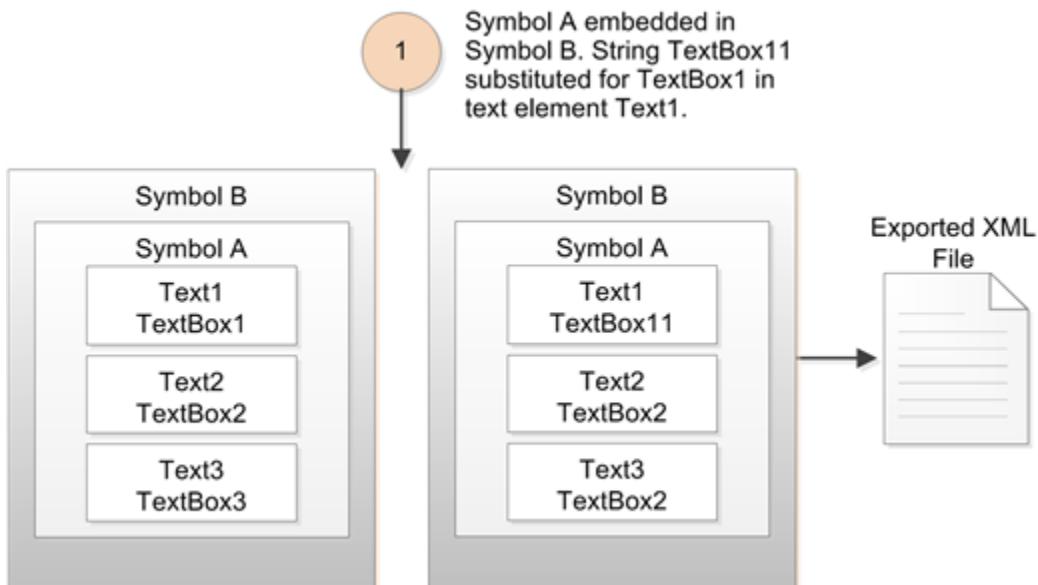
The export XML file's **SubstituteStrings** element shows the text string override information with a **String** element for each text override.

```
<SubstituteStrings>
<String Old="TextBox1" New="TextBox2" ElementID="B.A.Text1"/>
<String Old="TextBox1" New="TextBox3" ElementID="B.A.Text2"/>
<SubstituteStrings>
```

Import overridden text strings

An exported graphic's text strings can be overridden by editing the **SubstituteString** elements of the export XML file before importing the graphic.

The following illustration shows **Symbol A** embedded in **Symbol B**. **Symbol A** contains three text elements with text strings. The **TextBox1** text string of the **Text1** element was overridden to **Textbox11**. Then, **Symbol B** was exported using the programmatic API.



The expected **SubstituteStrings** element in the export XML file should be similar to the following:

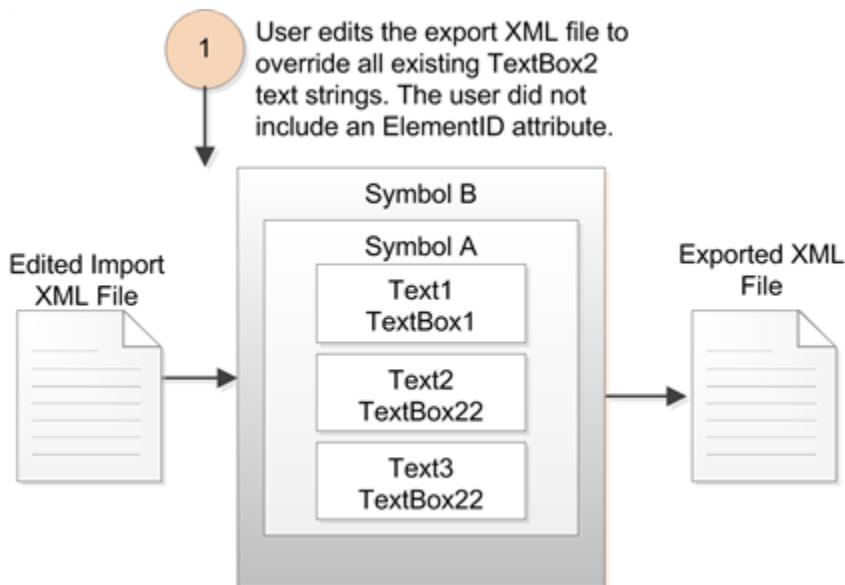
```
<SubstituteStrings>
```

```
<String Old="TextBox1" New="TextBox11" ElementID="A.Text1"/>
</SubstituteStrings>
```

The user edits the export XML file and adds an entry to override a text string without using an ElementID attribute to identify the text element.

```
<SubstituteStrings>
<String Old="TextBox1" New="TextBox11" ElementID="A.Text1"/>
<String Old="TextBox2" New="TextBox22">
</SubstituteStrings>
```

After importing the graphic, the text strings for the Text2 and Text3 elements are overridden to TextBox22. Without an ElementID to identify the text element, the string override replaces all text strings that match the text string specified by the Old attribute.



If the graphic was exported again using the programmatic API, the exported XML file shows that ElementID attributes have been added to each String element.

```
<SubstituteStrings>
<String Old="TextBox1" New="TextBox11" ElementID="A.Text1"/>
<String Old="TextBox2" New="TextBox22" ElementID="A.Text2"/>
<String Old="TextBox2" New="TextBox22" ElementID="A.Text3"/>
</SubstituteStrings>
```

Configure security for graphics

You can set IDE security permissions so that at design time, the user cannot:

- Import or export graphics.
- Create, modify, or delete graphics in the Visualization folder.
- Create, modify, or delete graphics in any AutomationObject template.
- Create, modify, or delete graphics in any AutomationObject instance.
- Create, modify, or delete View Applications, such as InTouchView Applications.
- Deploy or undeploy View Applications, such as InTouchView Applications.
- Edit the configuration of the quality and status display.

To restrict a user, the user must be assigned to a role and the permissions must be assigned to that role, and security must be enabled. For more information on how to configure security, users, and roles, see the Application Server User's Guide.

If the user attempts to export a graphic without appropriate permissions, the message "User doesn't have permission to export graphics object" appears.

If the user attempts to import a graphic without appropriate permissions, the message "User doesn't have permission to import graphics object" appears.

To configure security for graphics

1. On the ribbon, select **Galaxy**, then **Configure**, and then **Security**. The **Security** page appears.
2. Click the **Roles** tab.
3. In the **Roles available** list, click the role you want to assign permissions to.
4. In the **General Permissions** list, expand the **Graphic management permissions**.
5. Select the permissions that you want uses assigned this role to have. Clear the check box for each action that the users *should not* be able to perform.
6. Click **Save**.



AVEVA Group plc
High Cross
Madingley Road
Cambridge
CB3 0HB
UK

Tel +44 (0)1223 556655

www.aveva.com

To find your local AVEVA office, visit **www.aveva.com/offices**

AVEVA believes the information in this publication is correct as of its publication date. As part of continued product development, such information is subject to change without prior notice and is related to the current software release. AVEVA is not responsible for any inadvertent errors. All product names mentioned are the trademarks of their respective holders.