



# FactoryTalk View Site Edition User's Guide

Version 15.00.00



# Important User Information

Read this document and the documents listed in the additional resources section about installation, configuration, and operation of this equipment before you install, configure, operate, or maintain this product. Users are required to familiarize themselves with installation and wiring instructions in addition to requirements of all applicable codes, laws, and standards.

Activities including installation, adjustments, putting into service, use, assembly, disassembly, and maintenance are required to be carried out by suitably trained personnel in accordance with applicable code of practice.

If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

In no event will Rockwell Automation, Inc. be responsible or liable for indirect or consequential damages resulting from the use or application of this equipment.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular installation, Rockwell Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams.

No patent liability is assumed by Rockwell Automation, Inc. with respect to use of information, circuits, equipment, or software described in this manual.

Reproduction of the contents of this manual, in whole or in part, without written permission of Rockwell Automation, Inc., is prohibited.

Throughout this manual, when necessary, we use notes to make you aware of safety considerations.



**WARNING:** Identifies information about practices or circumstances that can cause an explosion in a hazardous environment, which may lead to personal injury or death, property damage, or economic loss.



**ATTENTION:** Identifies information about practices or circumstances that can lead to personal injury or death, property damage, or economic loss.

Attentions help you identify a hazard, avoid a hazard, and recognize the consequence.

**IMPORTANT:** Identifies information that is critical for successful application and understanding of the product.

These labels may also be on or inside the equipment to provide specific precautions.



**SHOCK HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that dangerous voltage may be present.



**BURN HAZARD:** Labels may be on or inside the equipment, for example, a drive or motor, to alert people that surfaces may reach dangerous temperatures.



**ARC FLASH HAZARD:** Labels may be on or inside the equipment, for example, a motor control center, to alert people to potential Arc Flash. Arc Flash will cause severe injury or death. Wear proper Personal Protective Equipment (PPE). Follow ALL Regulatory requirements for safe work practices and for Personal Protective Equipment (PPE).

The following icon may appear in the text of this document.



**Tip:** Identifies information that is useful and can help to make a process easier to do or easier to understand.

Rockwell Automation recognizes that some of the terms that are currently used in our industry and in this publication are not in alignment with the movement toward inclusive language in technology. We are proactively collaborating with industry peers to find alternatives to such terms and making changes to our products and content. Please excuse the use of such terms in our content while we implement these changes.

# Contents

<b>Getting started with FactoryTalk View SE.....</b>	<b>31</b>
Welcome to FactoryTalk View SE.....	31
FactoryTalk systems.....	31
FactoryTalk Services Platform.....	31
Finding more information about FactoryTalk services.....	32
FactoryTalk View Site Edition software.....	32
FactoryTalk View SE features.....	33
Quick start: setting up the software you need.....	34
Step 1: Plan the layout of the network.....	35
Step 2: Install FactoryTalk Services Platform.....	35
Step 3: Install FactoryTalk View SE.....	35
Step 4: Select the connectivity options.....	35
Step 5: Install the necessary activation keys.....	36
Step 6: Set up the FactoryTalk Directory.....	37
Exploring the InstantFizz application.....	37
Creating a FactoryTalk View SE application.....	39
Step 1: Create the application.....	39
Step 2: Create a graphic display.....	40
Step 3: Test run the application in FactoryTalk View SE Client.....	40
<b>Exploring FactoryTalk View Studio.....</b>	<b>42</b>
About FactoryTalk View Studio.....	42
Start FactoryTalk View Studio using a command.....	42
Command-line parameters and examples.....	42
About FactoryTalk Security permissions.....	43
Opening an application in FactoryTalk View Studio.....	43
HMI projects folder settings.....	44
Opening the InstantFizz application.....	45
Parts of the FactoryTalk View Studio main window.....	45
Workbook mode.....	46
Showing and hiding items in the main window.....	47
Working in the Explorer window.....	47
Moving the Explorer window.....	48
Opening and closing folders.....	48

Organize graphic displays.....	49
Opening component editors.....	49
Creating and modifying components.....	50
Adding components to an application.....	51
Naming components.....	51
Using component names that contain spaces.....	52
Avoiding names that conflict with commands or macros.....	52
Deleting, removing, and renaming components.....	52
Restoring a component.....	53
Techniques for working in editors.....	53
Gaining quick access to common operations.....	53
Browsing lists and components.....	53
Supplying tag names.....	53
Selecting and building commands.....	54
Building expressions.....	54
Printing from FactoryTalk View SE.....	54
Selecting a printer.....	55
Printing at runtime.....	55
<b>Planning an application.....</b>	<b>56</b>
Understanding the process you are automating.....	56
Planning the network layout.....	56
Choosing a Windows domain or workgroup.....	56
Windows workgroups.....	57
Determining computers needed.....	57
Planning communications.....	59
Determining how to access data.....	59
Collecting only the necessary data.....	59
Monitoring and controlling alarms.....	59
Planning an alarm monitoring and control system.....	60
Advantages of using FactoryTalk Alarms and Events services.....	60
Choosing device-based alarms.....	60
Choosing server tag-based alarms.....	60
Deciding when to use HMI tags.....	61
Designing the HMI tag database.....	61
Organizing HMI tags.....	61

Designing a dependable control system.....	62
Planning how to secure the system.....	62
Planning to use built-in system availability features.....	62
About redundant application servers.....	63
Setting up the runtime application.....	63
Developing a hierarchy of graphic displays.....	63
Creating templates to ensure consistency.....	64
Planning the multi-monitor deployment.....	64
Applying visual design principles.....	64
Planning how to use trends.....	65
Planning runtime language switching.....	65
Designing a system that is easy to deploy and maintain.....	66
Designing the application for multiple users.....	66
Designing applications for multiple monitors.....	66
Integrating with other applications and customizing the system.....	68
<b>Setting up the FactoryTalk Directory.....</b>	<b>70</b>
About FactoryTalk Directory.....	70
FactoryTalk Directory in a networked system.....	70
FactoryTalk Directory in a FactoryTalk View SE application.....	72
Local station applications.....	72
Network applications.....	73
Specifying the FactoryTalk Directory server location.....	73
Setting up FactoryTalk Directory for local station applications.....	75
What happens if the Network Directory server is unavailable.....	75
Monitoring the Network Directory status.....	76
<b>Setting up security.....</b>	<b>77</b>
About FactoryTalk Security services.....	77
About the FactoryTalk Directory.....	78
Finding more information about FactoryTalk Security.....	78
Gaining initial access to a FactoryTalk system.....	78
About the All Users account.....	79
Logging users on to and off from FactoryTalk View SE.....	79
Logging on to the FactoryTalk Directory.....	80
About single sign-on.....	80
Logging on to FactoryTalk View Studio.....	81

Logging on to a FactoryTalk View SE Client.....	81
Logging on when the client starts up.....	81
Changing the current user while the client is running.....	82
Logging on a different user.....	82
Logging off the current user at runtime.....	82
Changing the current user's password at runtime.....	83
Deciding how to secure a FactoryTalk View SE application.....	83
Securing FactoryTalk View SE applications at runtime.....	84
Securing FactoryTalk system resources.....	86
Setting up FactoryTalk accounts in FactoryTalk View SE.....	86
Setting up accounts in the Runtime Security editor.....	86
Example: Setting up runtime access to HMI components.....	88
Removing All Users from the Runtime Security list.....	89
Specifying login and logout macros.....	89
Removing Runtime Security accounts.....	91
Setting up runtime security for HMI project components.....	91
Assigning security codes to commands and macros.....	91
About the Unspecified_Command.....	92
Assigning security codes to graphic displays.....	92
Assigning security codes to OLE objects.....	92
Assigning security codes to HMI tags.....	93
Other ways to control runtime access to an application.....	94
Lock operators into the runtime environment.....	94
Use electronic signature to control user actions.....	95
About FactoryTalk Security accounts.....	95
Remove the All Users account.....	95
Specifying which users can set up security.....	96
Choosing the types of accounts to use.....	96
Using accounts that originate at the FactoryTalk Directory.....	96
Using Windows-linked accounts.....	97
Using both types of user account.....	97
Creating group accounts.....	97
Setting up user and computer accounts.....	98
Creating computer accounts for network applications.....	99
Setting up system-wide policies.....	100

Product policies.....	100
Using HTTPS protocol.....	100
Add an HTTPS binding for a site.....	102
System policies.....	103
Setting up security for FactoryTalk system resources.....	104
Modifying FactoryTalk Security settings.....	106
Specifying which actions users can perform.....	106
About explicit and implicit permission.....	107
Organizing actions into groups.....	107
Understanding inherited permissions.....	108
Breaking the chain of inheritance.....	108
Using explicit permissions to override inheritance.....	109
Performing secured tasks in FactoryTalk View SE.....	110
Example: Using the Common actions to set up security for user groups.....	111
<b>Working with network distributed applications.....</b>	<b>114</b>
About network distributed applications.....	114
Parts of a network distributed application.....	115
About FactoryTalk systems.....	116
Finding more information about FactoryTalk services.....	116
Key network distributed application concepts.....	116
FactoryTalk Network Directory.....	116
FactoryTalk Security.....	116
About FactoryTalk Security permissions.....	117
HMI servers (network distributed).....	117
HMI projects (network distributed).....	117
HMI clients.....	118
Areas (network distributed).....	118
Absolute and relative references.....	118
Absolute reference syntax.....	119
System availability.....	120
About HMI server redundancy.....	120
Language switching.....	121
Creating network distributed applications.....	121
Adding areas and servers to a network distributed application.....	121
Adding and deleting areas.....	121

Adding an HMI server (network distributed).....	121
Adding a data server.....	122
Adding a Tag Alarm and Event Server.....	123
Deciding when to use multiple servers.....	123
Adding servers for redundant operation.....	123
Adding servers for load balancing.....	123
Running multiple HMI servers.....	123
Providing HMI server names.....	123
Monitoring disk space on HMI servers.....	124
Setting up HMI server properties (network distributed).....	124
Changing the name of the host computer.....	125
Choosing how the server starts.....	125
Starting the HMI server when the first client connects to it.....	125
Starting the HMI server when the operating system initializes.....	126
Setting up HMI server redundancy.....	126
Selecting the secondary server.....	126
Replicating changes to the standby HMI server.....	127
Selecting startup and shutdown components (network distributed).....	127
Starting and stopping HMI server components manually (network distributed).....	128
Monitoring the status of an HMI server.....	129
Deleting HMI servers.....	129
Deleting HMI server project files.....	130
Renaming and deleting network distributed applications.....	130
Backing up and restoring network distributed applications.....	130
<b>Working with network station applications.....</b>	<b>131</b>
About network station applications.....	131
Parts of a network station application.....	132
About FactoryTalk systems.....	133
Finding more information about FactoryTalk services.....	133
Key network station application concepts.....	133
FactoryTalk Network Directory.....	133
FactoryTalk Security.....	134
About FactoryTalk Security permissions.....	134
HMI servers (network station).....	134
HMI projects (network station).....	135

HMI clients.....	135
Areas (network station).....	135
Relative references (network station).....	135
System availability.....	136
Language switching.....	136
Creating network station applications.....	136
Adding areas and servers to a network station application.....	136
Adding and deleting areas.....	136
Adding an HMI server (network station).....	137
Adding a data server.....	137
Adding a Tag Alarm and Event Server.....	138
Providing HMI server names.....	138
Monitoring disk space on HMI servers.....	138
Setting up HMI server properties (network station).....	139
Starting and stopping HMI server components manually (network station).....	139
Monitoring the status of an HMI server.....	139
Deleting HMI servers.....	140
Deleting HMI server project files.....	140
Renaming and deleting network station applications.....	140
Determining where the application is located.....	141
<b>Working with local station applications.....</b>	<b>142</b>
About local station applications.....	142
Parts of a local station application.....	143
About FactoryTalk systems.....	143
Finding more information about FactoryTalk services.....	144
Key local station application concepts.....	144
FactoryTalk Local Directory.....	144
FactoryTalk Security.....	144
About FactoryTalk Security permissions.....	145
HMI servers (local station).....	145
HMI projects (local station).....	145
HMI clients.....	145
Relative references (local station).....	145
Language switching.....	146
Creating local station applications.....	146

Importing a project into a new application.....	146
Adding servers to a local station application.....	146
Adding a Tag Alarm and Event Server.....	147
Setting up HMI server properties (local station).....	147
Selecting startup and shutdown components (local station).....	148
Monitoring the status of an HMI server.....	148
Renaming, deleting, and copying local station applications.....	148
<b>Setting up communications.....</b>	<b>149</b>
About data servers.....	149
About FactoryTalk Live Data.....	149
Using multiple data servers in an application.....	150
Communicating with multiple controllers.....	150
Setting up communications in FactoryTalk View SE.....	150
About FactoryTalk Security permissions.....	151
Adding FactoryTalk Linx data servers to an application.....	151
Setting up FactoryTalk Linx data server general properties.....	152
Setting up FactoryTalk Linx data server redundancy.....	152
Enabling support for FactoryTalk Alarms and Events.....	153
Deleting a FactoryTalk Linx data server.....	153
Setting up communications for FactoryTalk Linx.....	153
About the Primary and Secondary tabs.....	154
Adding OPC data servers to an application.....	154
Setting up OPC data server general properties.....	154
Setting up OPC data server redundancy.....	156
Setting up advanced properties.....	157
Include extended information in the server cache file.....	157
Synchronizing a data server's cache.....	157
Deleting an OPC data server.....	157
<b>Working with tags.....</b>	<b>159</b>
About data server tags, HMI tags, and client tags.....	159
Using direct referencing to eliminate duplication.....	159
Using the extended capabilities of HMI tags.....	161
Securing tag or device values.....	161
Scaling and offsetting values.....	161
Specifying minimum and maximum values.....	161

Storing values in memory.....	162
Reusing HMI tag names.....	162
Using tag data in a FactoryTalk View SE application.....	162
Setting up data server tags.....	162
Setting up HMI tags.....	162
Specifying tag names where tag data is needed.....	163
Logging tag values.....	163
Observing tag-related limits.....	164
About tag references.....	164
Absolute references.....	164
Relative references.....	164
Parts of the Tag Browser.....	165
Viewing tags in folders.....	166
Finding tags in the home area.....	166
Searching for and selecting tags.....	166
Finding a tag or text string.....	166
Finding a list of all tags or text strings.....	167
Finding tags with Tag Browser.....	167
Filtering tags.....	167
Browsing for offline tags.....	168
Browsing from FactoryTalk Linx.....	168
Browsing from other OPC servers.....	168
Working with HMI tags.....	168
HMI tag types.....	168
Data sources for HMI tags.....	169
Managing HMI tags.....	169
Naming HMI tags.....	170
Grouping HMI tags in folders.....	170
Viewing tag statistics.....	170
Parts of the Tags editor.....	171
Accept and Discard buttons (Tags editor).....	171
Form and spreadsheet (Tags editor).....	172
Query box.....	172
Folder hierarchy.....	172
Creating, modifying, and deleting HMI tags.....	172

Selecting data source for HMI tags.....	173
Getting HMI tag data from a device.....	173
Getting HMI tag data from memory.....	174
Creating HMI tags without using the Tags editor.....	174
Creating tags in a third-party application.....	175
Importing and exporting tags.....	175
Importing tags from a PLC database.....	175
Opening the Import PLC Tags dialog box.....	175
Client tags.....	176
Working with client tags.....	176
<b>Setting up FactoryTalk alarms.....</b>	<b>177</b>
About FactoryTalk Alarms and Events.....	177
Advantages in using FactoryTalk Alarms and Events services.....	177
Finding more information about FactoryTalk Alarms and Events.....	178
FactoryTalk Alarms and Events Help.....	179
FactoryTalk Alarms and Events System Configuration Guide.....	179
Key FactoryTalk Alarms and Events concepts.....	179
Alarms and events.....	179
Alarm servers.....	179
Level alarm.....	180
Level alarm limits.....	180
Variable limits and alarm faults.....	181
Deadband.....	181
Deviation alarms.....	182
Digital alarms.....	183
FactoryTalk alarm and event displays.....	183
Alarm and event summary.....	183
Alarm and event banner.....	183
Alarm status explorer.....	183
Alarm and event log viewer.....	184
Automatic diagnostic event summary.....	184
Alarm priority and severity.....	184
Alarm class.....	185
Alarm states.....	185
Alarm tags.....	186

Alarm status tags.....	186
Alarm messages.....	187
Alarm audit, diagnostic, and history logs.....	187
Summary of basic steps for setting up FactoryTalk alarms.....	187
Setting up system-wide alarm and event policies.....	188
To modify system-wide alarm settings.....	188
To modify severity settings for system events.....	189
Securing access to FactoryTalk alarm information.....	189
About FactoryTalk Security permissions.....	189
Working with Rockwell Automation Device Servers.....	189
Setting up support for FactoryTalk Alarms and Events.....	190
Specifying a device-based alarm source.....	191
Working with Tag Alarm and Event Servers.....	191
Setting up alarm priorities and history logging.....	192
Setting up FactoryTalk tag-based alarms.....	192
Setting up status tags for tag-based alarms.....	193
Setting up tag-based alarm messages.....	193
Opening the Alarm Message Editor.....	194
Modifying an existing alarm message.....	194
Adding variables to alarm messages.....	194
Using tag placeholders in alarm messages.....	194
Specifying tag update rates.....	195
Setting up OPC UA alarm monitoring.....	195
Enabling alarm and event in FactoryTalk Linx OPC UA Connector.....	195
Setting up alarm and event history logging.....	196
Defining an alarm and event log database.....	196
Enabling alarm and event history logging.....	197
Setting up alarm and event displays.....	197
Setting up an alarm and event summary.....	198
Setting up an alarm and event banner.....	198
About the alarm and event banner at runtime.....	199
Docking a banner display to the runtime client window.....	199
Setting up an automatic diagnostic event summary.....	200
Working with alarm sources in the Alarm Status Explorer.....	200
Viewing alarm and event history logs.....	201

Using tags to interact with alarms or obtain their status.....	201
Using FactoryTalk alarm functions in expressions.....	203
Retrieving information about the severity of alarms.....	203
Highest severity value of acknowledged alarms.....	203
Highest severity value of unacknowledged alarms.....	204
Retrieving information about the number of alarms.....	204
Number of disabled alarms.....	204
Number of shelved alarms.....	204
Number of shelved alarms returned to normal.....	204
Number of suppressed alarms.....	205
Number of suppressed alarms returned to normal.....	205
Number of acknowledged alarms.....	205
Number of unacknowledged alarms.....	205
Number of unacknowledged alarms returned to normal.....	206
Specifying the FactoryTalk alarm source in an expression.....	206
Using absolute and relative references to alarms.....	206
Using wildcards to specify multiple alarms.....	207
Using tag placeholders to specify the alarm name.....	207
What happens if the alarm source becomes unavailable.....	207
About controller status system alarms.....	208
Importing and exporting alarms.....	208
About import and export formats.....	209
<b>Setting up language switching.....</b>	<b>210</b>
About language switching.....	210
Text strings you can view in different languages at runtime.....	210
Specifying time, date, and numeric formats.....	211
Text strings you cannot view in different languages.....	211
Exported text that does not support language switching.....	211
Text that cannot be exported.....	212
Text that is part of the FactoryTalk View software.....	212
Text shown in FactoryTalk View SE system tags.....	212
Text that is not visible at runtime.....	212
Summary of steps for setting up language switching.....	213
About FactoryTalk Security permissions.....	214
Setting up font support for Windows languages.....	214

Selecting a language for a new FactoryTalk View application.....	214
About the current application language.....	215
Adding languages to an application.....	215
Setting up a default application language.....	216
Showing undefined text strings in the default language.....	216
Exporting application text strings.....	217
Troubleshooting export problems.....	218
Export file formats.....	218
Excel spreadsheet file format.....	218
Unicode text file format.....	218
Working with text strings exported to an Excel spreadsheet.....	219
Maintaining the format of the spreadsheet.....	220
Modifying or translating text strings.....	220
Working with duplicate text strings.....	221
Working with strings exported to a Unicode text file.....	221
File name and format.....	221
Opening a Unicode text file in Microsoft Excel.....	222
Saving a Unicode text file in Microsoft Excel.....	222
Differences in format for Unicode files saved in Excel.....	222
Saving a Unicode text file in Notepad.....	223
File schema.....	223
Working with pairs of double quotes.....	223
Working with backslashes and new-line characters.....	224
Importing translated or modified text strings.....	224
Troubleshooting import problems.....	225
Common errors when importing Unicode text files.....	226
Errors when importing Excel spreadsheet files.....	227
Common errors when importing local messages.....	229
Switch application languages at design time.....	229
Setting up runtime language switching.....	230
Using the Language command to switch languages.....	230
Support for multiple languages in the graphic libraries.....	230
Language support in new graphic libraries.....	231
Using the graphic libraries in a multi-language application.....	231
<b>Setting up FactoryTalk system availability.....</b>	<b>232</b>

FactoryTalk features that maximize system availability.....	232
Finding more information about system availability.....	233
Monitoring the status of application servers.....	233
States for non-redundant and redundant servers.....	234
States for redundant servers only.....	235
What happens if a non-redundant HMI server fails.....	236
Monitoring the status of the Network Directory server.....	236
What happens if the Network Directory server is unavailable.....	237
Redundancy as part of a system availability strategy.....	237
Planning the layout of a redundant system.....	237
About FactoryTalk View SE system limits.....	239
FactoryTalk View SE redundant component limits.....	239
System sizing recommendations.....	239
Activating FactoryTalk View SE in a redundant system.....	240
Setting up redundant servers in FactoryTalk View SE.....	241
Setting up a redundant HMI server pair.....	241
Specify the Network Directory on the secondary computer.....	241
Set up HMI server redundancy options.....	242
Manually replicate changes to the standby HMI server.....	242
Determining the Active HMI server in a redundant pair.....	243
When an HMI server is ready to be active or standby.....	244
Specifying On Active and On Standby macros.....	244
What happens if both servers become active.....	244
Switching the Active and Standby servers manually.....	245
Set up FactoryTalk Security permissions.....	246
What happens when the primary HMI server fails.....	246
Failing over to the Standby secondary server.....	246
Switching back to primary, or staying with the Active server.....	247
Continuing to use the secondary server.....	247
Switching back automatically to the primary server.....	247
Notifying clients when switching back to the primary.....	248
Modifying HMI tag properties at runtime.....	248
Modifying HMI tag properties.....	248
When HMI tag property changes take effect.....	248
Managing HMI data in an online redundant system.....	251

Centralize storage of diagnostic data.....	251
Determine which server will run events.....	251
Synchronize derived tags and data log files.....	251
Monitoring network client and server connections.....	251
About network glitches.....	252
<b>Logging system activity.....</b>	<b>253</b>
About FactoryTalk Diagnostics.....	253
Summary of steps for setting up Diagnostics.....	253
Finding more information about FactoryTalk Diagnostics.....	253
Key FactoryTalk Diagnostics concepts.....	254
Destinations.....	254
Message routing.....	254
Message categories.....	254
Message severity.....	255
Message audience.....	255
How tag writes are categorized.....	255
Setting up FactoryTalk Diagnostics in FactoryTalk View.....	255
Setting up message routing.....	256
Logging to a central database.....	256
Setting up message buffering.....	256
Tracking system events in the Diagnostics List.....	257
Working with the Diagnostics List.....	257
Viewing messages in the Diagnostics List.....	257
Viewing FactoryTalk Diagnostics logs.....	258
<b>Creating graphic displays.....</b>	<b>259</b>
About graphic displays.....	259
Importing and exporting graphic display XML files.....	260
About global object displays.....	260
Techniques for working in graphic displays.....	260
Zooming in and out.....	260
Setting up a display grid.....	261
Techniques for working with graphic objects.....	261
Copying objects.....	261
Copying objects with multiple languages.....	261
Duplicating objects.....	262

Resizing and reshaping objects.....	262
Arranging objects.....	263
Stacking objects.....	263
Aligning objects.....	264
Aligning objects top, middle, and bottom.....	265
Aligning objects left, right, and center.....	266
Spacing objects.....	266
Spacing objects vertically and horizontally.....	267
Flipping objects.....	267
Rotating objects.....	268
Grouping objects.....	268
Ungrouping objects.....	269
Modifying grouped objects.....	270
Changing line properties.....	270
Assigning tags and expressions to objects.....	271
Using tag substitution to replace text strings.....	272
Creating a background for a display.....	272
Testing graphic displays.....	272
Testing the appearance of objects in different states.....	273
Creating global object displays.....	273
Adding standard displays that contain reference objects.....	274
About global object displays at runtime.....	274
Adding Application Libraries to an application.....	274
Working with faceplates in the Graphics editor.....	275
Using objects from the graphic libraries.....	275
Location of library files.....	275
Using placeholders to specify tag values.....	275
Creating a tag placeholder.....	276
Replacing tag placeholders using parameter files.....	276
Loading a parameter file with the initial client display.....	276
Replacing tag placeholders using parameter lists.....	277
Setting up the appearance and behavior of a graphic display.....	277
Specifying the display type.....	277
Allowing multiple running copies.....	278
Caching displays.....	278

Setting up the title bar and other display attributes.....	279
Scaling the graphic display.....	279
Showing the last known values of HMI tags.....	279
Setting the update rate for tags.....	280
Viewing screen statistics of a display.....	280
Specifying the size of the graphic display.....	280
Preventing scroll bars on the main window.....	281
Specifying the display position.....	281
Securing graphic display.....	281
Background color.....	281
Using gradient style.....	281
Tracking screens for navigation.....	282
Setting up the runtime behavior of a graphic display.....	282
Specifying startup and shutdown commands.....	283
Specifying colors for input objects.....	283
Specifying the behavior of interactive objects.....	283
Specifying the behavior of objects with input focus.....	283
Using both types of highlight in the same display.....	284
Showing the on-screen keyboard.....	284
Setting up displays to open more quickly.....	284
Removing displays from the cache.....	285
Changing the default display settings.....	285
Docking displays to the FactoryTalk View SE Client window.....	285
Command parameters for docking displays.....	285
About the appearance and behavior of docked displays.....	286
Docking multiple displays in the same position.....	286
Running other displays in the available client area.....	286
Closing docked displays.....	287
<b>Creating graphic objects.....</b>	<b>288</b>
Types of graphic objects.....	288
About global objects.....	289
Creating different types of drawing objects.....	289
Adding images into graphic displays.....	289
Placing images in graphic displays.....	289
Adding images to an application.....	289

Importing images into graphic displays.....	290
Using bitmaps in a FactoryTalk View application.....	290
When to use a bitmap.....	291
Pasting images into graphic displays.....	291
Techniques for working with objects that use data.....	292
Using the keyboard to select objects that can take focus.....	292
Removing objects from the tab sequence.....	292
Different types of push buttons.....	293
About the runtime error state.....	294
Different types of data display and input objects.....	294
Using input objects to retrieve and send data.....	295
Tag-related limits.....	295
Updating tag values continuously.....	295
Validating operator input.....	296
Shortcut keys for retrieving and sending data.....	297
FactoryTalk View commands for retrieving and sending data.....	297
Parts of the on-screen keyboard.....	297
Different types of indicators.....	298
Setting up states for indicators.....	299
Using the Least Significant bit to trigger states.....	299
Setting up connections for indicators.....	299
Different types of gauges and graphs.....	299
Using gauges to show limits.....	300
Using graphs to compare values.....	300
Showing limits using scales with bar graphs.....	300
Using key objects to simulate keyboard functions.....	301
Creating different types of key objects.....	301
Setting up the auto-repeat function for selected keys.....	302
Using the same set of keys with different graphic objects.....	302
Different types of advanced objects.....	302
Creating piloted control list selectors.....	303
Selecting states in a piloted control list selector.....	303
Creating the Alarm and Event objects.....	303
Creating global objects.....	304
Creating reference objects.....	304

Using placeholders to specify values.....	305
Adding global objects into the graphic libraries.....	305
Setting up the link properties of reference objects.....	305
Setting up tag placeholders for global objects.....	306
Defining tag placeholders for use in reference objects.....	306
Modifying global objects that use tag placeholders.....	306
Working with OLE objects.....	307
Converting OLE objects.....	307
Working with ActiveX objects.....	307
Attributes of ActiveX objects.....	308
Deploying ActiveX components automatically at runtime.....	308
Working with .NET Control objects.....	308
Attributes of .NET Control objects.....	309
Working with Symbol Factory.....	309
Using electronic signatures to authorize runtime changes.....	309
Tracking changes using FactoryTalk Diagnostics.....	312
<b>Animating graphic objects.....</b>	<b>313</b>
About animation in FactoryTalk View.....	313
Setting up animation for FactoryTalk View graphic objects.....	313
Linking animation to tag values.....	314
Using tag placeholders.....	314
Linking animation to expressions.....	314
Linking animation to actions.....	314
Determining start and end points for a range of motion.....	315
Defining the range of motion.....	315
Applying animation to object groups.....	316
Testing animation.....	316
Viewing the animation applied to objects.....	317
Copying or duplicating objects with animation.....	317
About global objects and animation.....	317
Creating effects using the different types of animation.....	318
Showing and hiding objects.....	318
Changing the color of an object.....	318
Example 1: Creating text that blinks.....	319
Example 2: Creating an object that changes color.....	320

Changing the level of fill in an object.....	321
Moving an object horizontally in a display.....	321
Moving an object vertically in a display.....	321
Rotating objects with animation.....	321
Setting up rotation animation.....	322
Changing the width of an object.....	322
Changing the height of an object.....	322
Setting up touch zones.....	323
Creating a horizontal slider.....	323
Creating a vertical slider.....	323
Creating a hyperlink for an object.....	323
Animating OLE verbs.....	323
Using index numbers to navigate to objects in a display.....	324
Checking an object's index number.....	325
How tab index numbers work.....	325
Creating a tab sequence.....	325
Changing index numbers.....	326
Associating objects and displays with keys.....	326
Setting up object keys.....	327
Setting up display keys.....	328
Viewing the key list at runtime.....	328
Setting up object-specific commands using keys.....	329
Animating ActiveX objects.....	330
Connecting tags to an ActiveX object's properties.....	331
About ActiveX object names.....	331
Viewing methods of an object.....	332
Connecting tags to an object's methods.....	332
Using the Invoke command to call a method.....	332
Connecting tags to an ActiveX object's events.....	333
<b>Setting up navigation.....</b>	<b>334</b>
Designing a display hierarchy for an application.....	334
Setting up ways to move among displays.....	335
Using commands to open, close, and switch displays.....	335
Using the navigation menu to switch displays.....	338
Choosing display types with navigation in mind.....	339

Reducing display call-up time.....	339
Setting up keys to run FactoryTalk View commands.....	339
General rules governing precedence.....	340
Precedence and the F1 key.....	340
Precedence and embedded ActiveX objects.....	340
Precedence and embedded OLE objects.....	341
Keyboard shortcuts.....	341
About navigation buttons.....	342
How navigation buttons work.....	343
<b>Creating expressions.....</b>	<b>344</b>
About expressions.....	344
Where you can use expressions in FactoryTalk View SE.....	344
Working in the Expression editor.....	344
Expression components.....	346
Checking the syntax of an expression (Expression editor).....	346
Cutting, copying, and pasting expressions.....	346
Formatting expressions.....	347
Using tag names and tag placeholders in expressions.....	347
Specifying the area with a tag name.....	347
Using tag placeholders to specify tag values.....	348
Using constants in expressions.....	348
Using operators in expressions.....	348
Arithmetic operators.....	348
Relational operators.....	349
Logical operators.....	349
Bitwise operators.....	349
Evaluation order of operators.....	350
Using built-in functions in expressions.....	352
Math functions.....	352
File functions.....	352
Time functions.....	353
Tag functions.....	354
Security functions.....	355
Language function.....	356
FactoryTalk alarm functions.....	356

Automatic Diagnostics functions.....	359
FactoryTalk alarm names and device path in expressions.....	359
Redundancy functions.....	359
ThinManager functions.....	360
DataLogPro functions.....	361
Using if-then-else logic in expressions.....	361
Nested if - then - else.....	362
<b>Creating embedded variables.....</b>	<b>365</b>
About embedded variables.....	365
Inserting embedded variables.....	365
Creating numeric embedded variables.....	366
Syntax for numeric embedded variables that use a tag value.....	366
Syntax for numeric embedded variables that use a literal number.....	367
Creating string embedded variables.....	367
Syntax for string embedded variables that use a tag value.....	368
Syntax for string embedded variables that use a literal string.....	368
Creating time and date embedded variables.....	369
Syntax for time and date embedded variables.....	369
How embedded variables are shown at runtime.....	370
Numeric embedded variables.....	370
String embedded variables.....	371
Time and date embedded variables.....	371
How embedded variables are updated at runtime.....	371
<b>Setting up data logging.....</b>	<b>372</b>
About data logging.....	372
DataLogPro.....	372
DataLogPro editor.....	373
Start and stop logging with command.....	373
Log historical data to an external InfluxDB.....	373
Customize data logging location.....	373
Migrate data log model tags to DataLogPro.....	374
InfluxDB.....	378
InfluxDB storage format and schemas.....	379
Set up InfluxDB user and permission.....	380
Delete InfluxDB data.....	380

Migrate InfluxDB data.....	381
Back up and restore InfluxDB data.....	381
Data Log Models.....	381
Gathering tag data in data log models.....	381
Using multiple data log models.....	381
Data log storage formats.....	381
How log file sets are named.....	382
ODBC database storage format and schemas.....	382
How ODBC tables are named.....	383
Format for ODBC float and string tables.....	383
Format for ODBC tag table.....	384
Creating data log models.....	384
Specifying the storage format.....	385
Selecting the ODBC database format.....	385
Setting up log paths.....	386
Setting up and managing data log files.....	386
Creating file sets.....	386
Deleting file sets and ODBC database records.....	386
Specifying when to log data.....	387
Choosing the data to be logged.....	387
Logging data to ODBC data sources.....	387
Logging data to an existing ODBC data source.....	387
Creating an ODBC data source.....	388
Setting up security to log data to a remote computer.....	388
Creating files at runtime.....	388
Using the DataLogNewFile command.....	388
Switching log paths at runtime.....	389
Setting up switching options.....	389
Switching back manually to the primary path.....	389
Using the DataLogSwitchBack command.....	390
Moving data from the secondary path.....	390
Using the DataLogMergeToPrimary command.....	390
Logging on demand.....	391
Using the DataLogSnapshot command.....	391
Combining logging.....	391

Modifying existing data log models.....	391
Changing log paths.....	392
Making runtime changes without modifying the model.....	392
Starting and stopping data logging.....	392
Ways to start data logging.....	393
Ways to stop data logging.....	393
<b>Setting up trends.....</b>	<b>394</b>
About trends.....	394
TrendPro interface.....	395
Creating TrendPro objects.....	396
Setting up TrendPro properties.....	396
Trace List (trend).....	398
Event list.....	399
Testing a trend.....	400
Changing the trend appearance.....	400
Time Period page (trend).....	400
Chart page (trend).....	401
X-Axis page.....	403
Application page (trend).....	404
Retrieval page (trend).....	405
Using templates for trend objects.....	405
Create a template (trend).....	405
Apply a template (trend).....	406
Working with trends at runtime.....	406
Add or delete traces at runtime (trend).....	406
View data of traces (trend).....	406
Use cursors (trend).....	407
Create time periods (trend).....	407
Use shapes (trend).....	408
Zoom and pan data (trend).....	408
<b>Setting up XY Plot.....</b>	<b>409</b>
About XY Plot.....	409
XY Plot interface.....	409
Creating XY Plot objects.....	410
Setting up XY Plot properties.....	410

Trace List (XY Plot).....	413
Testing XY Plot.....	414
Changing the XY Plot appearance.....	414
Time Period page (XY Plot).....	414
Chart page (XY Plot).....	415
Application page (XY Plot).....	417
Retrieval page (XY Plot).....	418
Using templates for XY Plot objects.....	418
Create a template (XY Plot).....	419
Apply a template (XY Plot).....	419
Working with XY Plot at runtime.....	419
Add or delete traces at runtime (XY Plot).....	419
View data of traces (XY Plot).....	419
Use cursors (XY Plot).....	420
Create time periods (XY Plot).....	420
Use shapes (XY Plot).....	421
Zoom and pan data (XY Plot).....	421
<b>Adding logic and control.....</b>	<b>423</b>
About logic and control.....	423
Creating and using derived tags.....	423
Parts of the Derived Tags editor.....	423
Accept and Discard buttons (Derived Tags editor).....	423
Form and spreadsheet (Derived Tags editor).....	424
Checking the syntax of an expression (Expression editor).....	424
How to use derived tags.....	424
Creating a derived tags component.....	424
About the maximum update rate.....	425
Using multiple derived tag components.....	425
Modifying existing derived tag components.....	425
Starting and stopping derived tags processing.....	425
Starting derived tags in network and local station applications.....	426
Starting derived tags in local station applications only.....	426
Stopping derived tags in network and local station applications.....	426
Stopping derived tags in local station applications only.....	427
Setting up FactoryTalk View SE events.....	427

Parts of the Events editor.....	427
Accept and Discard buttons (Events editor).....	428
Form and spreadsheet (Events editor).....	428
Checking the syntax of an expression (Events editor).....	428
Creating an events component.....	428
About the maximum update rate.....	429
Using multiple events components.....	429
Modifying existing event components.....	429
Starting and stopping events processing.....	429
Ways to start events processing.....	430
Ways to stop events processing.....	430
Creating and using macros.....	430
Macro syntax.....	431
Specifying parameters in a macro.....	432
Typing macro names that contain spaces.....	432
Nesting macros.....	433
Running macros.....	433
Specifying user login and logout macros.....	433
Creating client keys.....	434
Running client key components.....	434
<b>Managing recipes.....</b>	<b>435</b>
About recipes.....	435
About the RecipePro+ editor.....	435
The parts of RecipePro+ editor.....	436
Recipe edit table.....	436
Toolbar commands.....	437
Create, modify, or delete recipes.....	438
Refresh recipes.....	439
Import or export recipes.....	439
Required security permissions.....	440
Migrate RSView32 RecipePro to FactoryTalk View SE RecipePro+.....	441
<b>FactoryTalk View commands.....</b>	<b>442</b>
Using commands in a FactoryTalk View SE application.....	442
How to use commands.....	442
Using tag placeholders in commands.....	443

Where commands run.....	444
Using absolute and relative references.....	445
How relative references are resolved.....	445
Commands that take tags as parameters.....	446
Commands that take HMI project components as parameters.....	446
Creating symbols.....	446
Important guidelines.....	447
Running and building commands.....	447
<b>Scripts.....</b>	<b>449</b>
<b>Creating and deploying .NET controls.....</b>	<b>450</b>
Custom .NET control overview.....	450
Required knowledge.....	450
Environment requirements.....	450
Creating custom Windows Forms controls.....	451
Create a Windows Forms control in Visual Studio.....	451
Creating custom WPF controls.....	459
Create a WPF control in Visual Studio.....	459
Importing and deploying the .NET control assembly.....	467
Import self-created .NET controls or third-party libraries.....	467
Deploy .NET controls on multiple computers.....	468
Update .NET controls.....	468
Delete .NET controls.....	468
JSON support in VBA.....	468
References.....	471
<b>Setting up DDE communications for HMI tags.....</b>	<b>472</b>
Overview of DDE communications.....	472
Setting up an HMI server as a DDE client.....	473
Creating an HMI tag that uses DDE.....	473
Specifying Device as the data source.....	473
Syntax for DDE addresses.....	473
Scanning for new tag values.....	474
<b>Using the SE Client object model and display code.....</b>	<b>475</b>
Customizing applications using VBA with FactoryTalk View.....	475
Summary of basic steps.....	476
About procedures.....	476

How VBA code runs.....	476
Parts of the VBA IDE.....	476
FactoryTalk View SE Client object model.....	478
Finding information about FactoryTalk View SE Client objects.....	478
Finding information about VBA.....	478
<b>Importing and exporting XML files.....</b>	<b>479</b>
About XML.....	479
Exporting graphics data to an XML file.....	479
Modifying exported XML files.....	479
Saving XML files in Notepad.....	480
Testing XML files.....	480
Importing XML files.....	480
Error log file.....	481
Importing graphic display XML files.....	481
Graphic display XML file structure.....	481
<b>FactoryTalk View documentation.....</b>	<b>483</b>
<b>Troubleshoot KEPServer.....</b>	<b>484</b>
Set DCOM configuration for KEPServer Enterprise.....	484
<b>Legal Notices.....</b>	<b>489</b>

# Getting started with FactoryTalk View SE

This chapter describes:

- What FactoryTalk View Site Edition is.
- What a FactoryTalk system is.
- FactoryTalk View Site Edition software.
- How to set up the software you need.
- Running FactoryTalk View SE without activation.
- Exploring the InstantFizz application.
- How to start creating a network distributed application.

## Welcome to FactoryTalk View SE

FactoryTalk View Site Edition (SE) is an integrated software package for developing and running human-machine interface (HMI) applications that can involve multiple users and servers, distributed over a network.

In FactoryTalk View Studio, you can create network distributed, network station, or local station applications that mirror your plant or process.

Use the editors in FactoryTalk View Studio to create and test the application components you need. Then set up the FactoryTalk View SE clients to let operators interact with the application after it is deployed.



### Tip:

The steps or illustrations in this manual may vary depending on your operating systems.

## FactoryTalk systems

FactoryTalk View Site Edition provides all the tools that you need to create powerful, dependable process monitoring and supervisory control applications.

## FactoryTalk Services Platform

FactoryTalk Services Platform provides a set of common services (such as diagnostic messages, health monitoring services, and access to real-time data) for all the FactoryTalk products and applications used in a control system.

Using the FactoryTalk Services Platform, FactoryTalk products can share and gain simultaneous access to resources such as tags and graphic displays that you only need to define once in the system.

The FactoryTalk Services Platform installs behind the scenes during FactoryTalk View SE installation. Following are the services that the platform provides:

- FactoryTalk Directory centralizes access to system resources and names for all FactoryTalk products and components participating in an automated control system. The FactoryTalk Network Directory manages resources for FactoryTalk View SE network applications, and the FactoryTalk Local Directory manages resources for FactoryTalk View SE local applications. For more information, see [Setting up the FactoryTalk Directory on page 70](#).

- FactoryTalk Security centralizes user authentication and authorization at the FactoryTalk Directory. For information about securing FactoryTalk View SE applications, see [Setting up security on page 77](#).
  - FactoryTalk Live Data manages connections between data servers in an application and FactoryTalk Live Data clients such as FactoryTalk View SE clients.
- For information about data communications in FactoryTalk View SE applications, see [Setting up communications on page 149](#).
- FactoryTalk Diagnostics collects and provides access to activity, status, warning, and error messages generated throughout a FactoryTalk system. For information about diagnostics in FactoryTalk View SE, see [Logging system activity on page 253](#).
  - FactoryTalk Administration Console is a standalone tool for developing, managing, and securing multiple FactoryTalk View applications.

---

**NOTE:** You must use the FactoryTalk Administration Console to restore FactoryTalk View SE network applications and to set up computer accounts, if necessary, after upgrading the FactoryTalk Services Platform. Otherwise, FactoryTalk View Studio is the configuration software for developing and testing FactoryTalk View applications.

---

## FactoryTalk services installed with FactoryTalk View SE

The FactoryTalk Services Platform also supports FactoryTalk Alarms and Events services, and software-based product activation. These services are installed separately, during FactoryTalk View SE installation, rather than with the FactoryTalk Services Platform.

## Finding more information about FactoryTalk services

This manual contains information about developing FactoryTalk View SE applications, including information about how FactoryTalk View uses FactoryTalk services.

For additional, detailed information about FactoryTalk systems, services, concepts, and components, see the FactoryTalk Help.

## FactoryTalk View Site Edition software

FactoryTalk View SE consists of several pieces of software that you can use to build network or local HMI applications, customized to your needs.

Depending on the software package you purchased, you will have installed one or more of the following software components, from the FactoryTalk View installation package.

FactoryTalk Services Platform provides a set of common services (such as diagnostic messages, health monitoring services, and access to real-time data) for all the FactoryTalk products and applications used in a control system. For an overview of FactoryTalk Services Platform, see [FactoryTalk Services Platform on page 31](#).

FactoryTalk View Studio is the design-time environment for FactoryTalk View that provides the editors and tools you need to develop and test network distributed, network station, and local station human-machine interface (HMI) applications. It contains editors for creating complete applications, and contains client and server software for testing the applications you create. For information about FactoryTalk View Studio features, see [Exploring FactoryTalk View on page 42](#). Studio

You can also use FactoryTalk View Studio to set up FactoryTalk Security services for the applications you develop. For more information, see [Setting up security on page 77](#).

---

**NOTE:**

FactoryTalk View Studio is also the configuration software for developing FactoryTalk View ME (ME) applications. For information about FactoryTalk View ME, see *FactoryTalk View ME User's Guide* and the ME Help.

---

FactoryTalk View SE Client is software for viewing and interacting with FactoryTalk View SE local station, network station, and network distributed applications at runtime. Use the FactoryTalk View SE Client Wizard to create client configuration files that can be deployed to client host computers.

FactoryTalk View SE Server also called the HMI server, stores HMI project components (for example, graphic displays) and supplies them to clients. The server also contains a database of tags, and performs historical data logging. FactoryTalk View SE Server has no user interface. Once installed, it runs as a set of headless Windows services that supply information to clients as they request it.

FactoryTalk Alarms and Events installs behind the scenes during FactoryTalk View SE installation, and provides system-wide alarm monitoring and control centralized at the FactoryTalk Directory. For runtime clients to receive FactoryTalk device-based and tag-based alarm information, you must set up application servers to support FactoryTalk Alarms and Events services. For details, see [Setting up FactoryTalk alarms on page 177](#).

FactoryTalk Activation services provide a secure, software-based system for activating Rockwell Software products and managing software activation files. For information about activating FactoryTalk View SE software, see the *FactoryTalk View Site Edition Installation Guide*.

FactoryTalk tools and utilities that support the operation of your FactoryTalk View SE software. For an overview, see Chapter 1 of the *FactoryTalk View Site Edition Installation Guide*.

## FactoryTalk View SE features

FactoryTalk View SE provides the flexibility and features that you need to create powerful automation systems for your plant or process. For example, you can:

- Create local station applications for parts of the plant or process that are self-contained, and are not related to other parts of the process.
- Create complex applications that mirror the layout of a plant or process. A FactoryTalk View SE network distributed application can contain several servers running on multiple computers, connected over a network. Multiple client users can connect simultaneously to a network distributed application.
- Open and modify network applications remotely using FactoryTalk View Studio.
- Use the health monitoring and redundancy features built into FactoryTalk and FactoryTalk View SE to support system availability at runtime.
- Use FactoryTalk Security services to centralize the authentication and authorization of system users at the FactoryTalk Directory.
- Create multi-language applications that support switching between up to 40 different languages at runtime. In a network distributed application, multiple clients can run in different languages simultaneously.

- Modify HMI tags at runtime, and have changes take effect at connected clients, without restarting the clients. In a redundant network distributed application, changes made at the active HMI server can be replicated to the standby server.
- Create a complete alarm monitoring system with FactoryTalk Alarms and Events. Using the FactoryTalk Alarms and Events services, FactoryTalk View SE applications can subscribe to and display device-based alarms, programmed directly into Studio 5000® controllers.
- Customize alarm summaries and banners to provide specific alarm data, rather than displaying alarms for the entire system.
- Create global objects in global object displays, and use copies of these objects throughout an application. When you modify the original object, all linked copies are also updated.
- Dock the selected displays to an edge of the FactoryTalk View SE Client window so they can be viewed in a fixed position, at all times.
- Use the FactoryTalk View SE Client Object Model, VBA, and PowerShell scripts to extend the capabilities of FactoryTalk View SE, and to share and interoperate with Windows programs such as Microsoft SQL Server and Microsoft Excel.
- Create trends that show process variables plotted against time. FactoryTalk View SE trends can display real-time or historical data, with up to 100 pens (tags) in each trend.
- Log data simultaneously to the FactoryTalk Diagnostics log and to remote ODBC (Open Database Connectivity) databases to provide various records of production data.

## Quick start: setting up the software you need

The design of the automation and control system you plan to deploy will determine which FactoryTalk View SE software components to install and set up on network computers. Where you install the software also depends on the type of application.

FactoryTalk View Studio Enterprise is the configuration software for developing and testing FactoryTalk View SE network distributed, network station, local station, and FactoryTalk View ME applications.

- To develop or run a Network Distributed application, you can install different combinations of software on each computer, depending on needs.
- To develop or run a Network Station application, you must install all the necessary software components on one computer (except for data servers). Network station applications can connect to data servers that are on different machines.
- To develop or run a Local Station application, you must install all the necessary software components on one computer (except for OPC data servers). Local station applications can connect to OPC data servers that are on different machines.
- To develop or run a View Machine Edition application, see *FactoryTalk View Machine Edition User's Guide* and ME Help.

Steps in this section describe the basic tasks involved in setting up the software needed to develop and run local and network applications.

For detailed installation instructions, and for information about deploying Site Edition network distributed, network station, and local station applications for production, see *FactoryTalk View Site Edition Installation Guide*.

## Step 1: Plan the layout of the network

The layout of the network is particularly important to the design of a network distributed application.

The type and structure of the network can determine which roles participating host computers will play, including whether any of the servers will run as redundant pairs.

If you are planning a network application consisting of more than 10 computers pointing to the same FactoryTalk Directory, a domain controller is strongly recommended. For network distributed applications consisting of 10 computers or fewer pointing to the same FactoryTalk Directory, you can use a Windows workgroup.

---

**NOTE:**

Do not install FactoryTalk Directory, FactoryTalk View SE Server, or any other application software on the same computer as the Windows domain controller. This configuration is not supported.

---

## Step 2: Install FactoryTalk Services Platform

FactoryTalk View applications depend on FactoryTalk software, such as FactoryTalk Directory, to run. During the FactoryTalk Services Platform installation, the FactoryTalk Network Directory and Local Directory are set up automatically on the computer.



**Tip:** If you plan to run only the FactoryTalk Directory server on a computer without any dependent software, install only the FactoryTalk Services Platform on the computer.

---

## Step 3: Install FactoryTalk View SE

When you install the FactoryTalk View SE software, you can choose to install all the components on one computer, or to install individual components on separate computers on the network.

To develop or run a local station application, you must install the FactoryTalk Services Platform and all the FactoryTalk View SE software on one computer.



**Tip:** In a local application, only OPC data servers can be run on a separate computer.

---

To develop or run a network distributed or network station application, you can install all the FactoryTalk View SE software, or just selected components, on participating computers.

For example, you might install only the FactoryTalk View SE Client software on computers run by operators. Similarly, to distribute server loads across the application, you might install only the FactoryTalk View SE Server software on server computers.

## Step 4: Select the connectivity options

During the installation, you are prompted to select the connectivity options.

### When to use FactoryTalk Linx

FactoryTalk Linx is a communication server built around FactoryTalk technology to assist in developing and running your FactoryTalk View SE applications.

For communications with Allen-Bradley local and remote devices, particularly with Studio 5000 controllers, FactoryTalk Linx is the recommended data communications software for FactoryTalk View applications.

FactoryTalk Linx allows you to create redundant controller shortcuts and to perform online tasks such as uploading and downloading Studio 5000 Logix Designer files.

### When to use RSLinx Classic

RSLinx Classic is a 32-bit communication server that links Allen-Bradley networks and devices to Microsoft Windows applications. These range from device programming and configuration applications to HMI applications, to your own data acquisition applications using Microsoft Office, webpages, or Visual Basic.

RSLinx Classic also incorporates advanced data optimization techniques and contains a set of diagnostics. RSLinx Classic is an OPC DA (Data Access) Compliant Server and a DDE server.

For example, install and use RSLinx Classic to serve data through DH+ (Data Highway +) networks, to support complex bridging and routing, and to support unsolicited messaging from a controller to RSLinx.

RSLinx Classic also allows you to create alias topic shortcuts, and to perform online tasks such as uploading and downloading Studio 5000 Logix Designer files.

### Communicating with third-party local and remote devices

For communications with non-AlLEN-BRADLEY local and remote devices, FactoryTalk View SE supports OPC, a protocol used to connect to communication devices via vendor-specific OPC servers.

OPC enables FactoryTalk View to act as a client to other OPC servers. This means that FactoryTalk View can use third-party OPC servers to retrieve tag values from third-party controller devices, such as Siemens or Modicon.

FactoryTalk View SE supports the OPC DA 2.05a, UA 1.02 and 1.03 specifications for exchanging data among automation or control applications, field systems or devices, and business or office applications.

## Step 5: Install the necessary activation keys

The types of product licenses you require depend on the software you have installed, and how you intend to use it.

For information about the types of activation you might need, and how to install activation keys, see the *FactoryTalk View Site Edition Installation Guide*.

### Running FactoryTalk View SE without activation

The FactoryTalk View SE software you install must be licensed for full use.

If for some reason activation is unavailable, the software can run unlicensed for a grace period of up to seven days. This provides time to correct any problems, without disrupting critical applications.

If activation is restored within the seven days, normal operations will resume. If activation remains unavailable when the grace period expires, the FactoryTalk View SE software will run in demo mode.

When running in demo mode, there are some limitations.

- You can run a local station FactoryTalk View application for up to two hours.
- Remote clients cannot connect to a FactoryTalk View server.

Starting from FactoryTalk View version 10.00, some limitations are removed. You can create or load over five graphic displays per HMI server now. You can also fully use the Symbol Factory library without restriction.

## Step 6: Set up the FactoryTalk Directory

The FactoryTalk Directory centralizes access to resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

FactoryTalk Directory centralizes access to application resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

FactoryTalk Local Directory (also called the Local Directory) manages applications that are confined to a single computer, for example, FactoryTalk View SE local station applications. The Local Directory must reside on the same computer as the local station application.

FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed applications, and network station applications. All of the client and server computers participating in a given network distributed application, or network station application must point at the same Network Directory.

Both directories are configured on the computer when you install the FactoryTalk Services Platform. To use the Local Directory as part of a local application, no further setup is required.

To use the Network Directory as part of a network application, you must set up all participating computers to point at the same Network Directory computer. For more information, see [Setting up the FactoryTalk Directory on page 70](#).

## Exploring the InstantFizz application

The InstantFizz application shipped with FactoryTalk View SE is a fully functional example of a network distributed application.

To explore the FactoryTalk View SE development and runtime environments, open the InstantFizz application in FactoryTalk View Studio and launch the SE client.

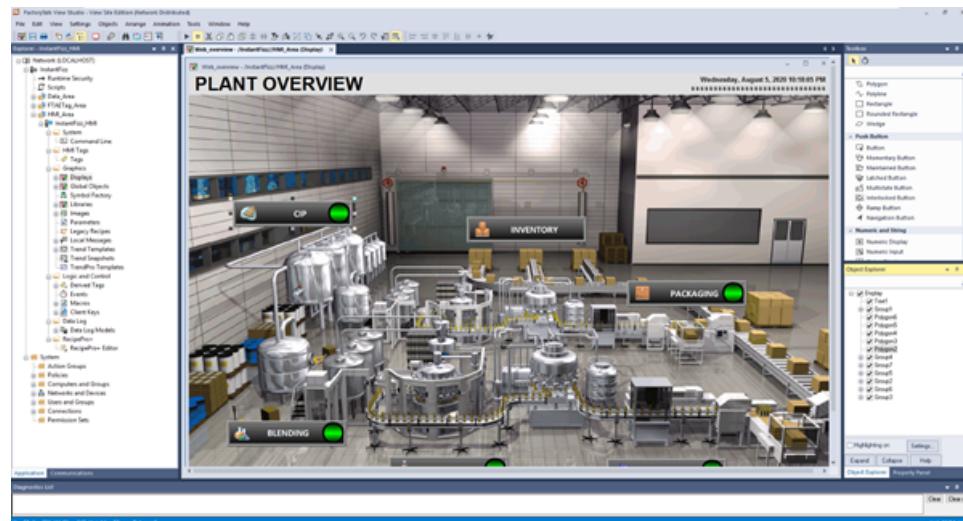


**Tip:** Before running the client, you must download the ControlLogix project IF2\_DEMO.ACD to a programmable controller or a SoftLogix controller. By default, the project is in C:\Users\Public\Documents\RSView Enterprise\SE\HMI projects\InstantFizz\_HMI.

### To open the InstantFizz application:

1. Select **Start > FactoryTalk View Studio**.
2. In the **Application Type Selection** dialog box, select **View Site Edition (Network Distributed)** and click **Continue**.
3. In the **Existing** tab, select **InstantFizz** and click **Open**.

In the following illustration, the plant overview display is open in the Graphics editor. For information about working with FactoryTalk View Studio and the editors, see [Exploring FactoryTalk View Studio](#) on page 42



### To test a display in FactoryTalk View Studio:

1. In FactoryTalk View Studio, in the Explorer window, right-click the display you want to test, and then click **Open**.
2. On the **View** menu, click **Test Display**. To stop testing, on the **View** menu, click **Edit Display**.

### To run the InstantFizz client:

Open C:\Users\Public\Documents\RSView Enterprise\SE\Client, and then double-click the IF\_Client\_1920x1080.cli file to run the application.



You can use the buttons and touch zones provided to navigate through the application, view alarm summaries and so on. There is a **Learning Mode** that highlights features in blue boxes as you navigate throughout the application. To enable these feature highlights, on the menu, select **More... > Learning Mode**. To hide, select **More... > Demo Mode**.



## Creating a FactoryTalk View SE application

You can also create a simple, new application to exercise some of the development and runtime features of FactoryTalk View SE. Following are some steps to get you started.

### Step 1: Create the application

First, create the application itself, and then add at least one HMI server.

FactoryTalk View SE Server also called the HMI server, stores HMI project components (for example, graphic displays) and supplies them to clients. The server also contains a database of tags, and performs historical data logging.

FactoryTalk View SE Server has no user interface. Once installed, it runs as a set of headless Windows services that supply information to clients as they request it.

#### To create a network distributed application:

1. Select **Start > FactoryTalk View Studio**.
2. In the **Application Type Selection** dialog box, select the application type and click **Continue**.
3. In the **New/Open Site Edition Application** dialog box, click the **New** tab.
4. Type the name and description, select a default language, and click **Create**.
5. (optional) To change the default folder of the HMI projects, use the HMI Projects Folder Path Config Utility to specify a new folder.

The default language shown in the New tab is the operating system language. You can accept the default or, in the Language list, select any language that Windows supports. For more information, see [Setting up language switching on page 210](#).

#### To add an area to the application:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application and select **New Area**.
2. In the **New Area** dialog box, specify a name and description and click **OK**.

### To add an HMI server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application or an area name, select **Add New Server**, and then select **HMI Server**.
2. In the **Add HMI Server Wizard**, in the **Select Operation** window, click **Create a new HMI server**.
3. (optional) To add the Application Libraries, click the **Add Application Library** dropdown list, and then select one or multiple libraries.
4. Click **Next**.
5. Type a name and description for the HMI server, specify the host computer if applicable, and click **Finish**.
6. (optional) To change the default folder of the HMI projects, use the HMI Projects Folder Path Config Utility to specify a new folder.

For information about other options in the Select Operation window, see [Adding an HMI server on page 137](#).

## Step 2: Create a graphic display

Adding an HMI server to an application also creates the HMI project, which contains all of the editors and productivity tools you need to create and modify application components.

For a list of editors and their functions, see [Opening component editors on page 49](#). For instructions about using the editors, click Help in the editor dialog boxes.

This example uses the Graphics editor to create a graphic display, and then add a graphic object to that display.

### To create a display

- In the FactoryTalk View Explorer window, expand the HMI server and the Graphics folder, right-click **Displays** and select **New**.

### To add a graphic object to the display

1. In the Toolbox, select **Drawing > Rectangle**. For information about using the Graphics editor, see [Creating graphic displays on page 259](#) and [Creating graphic objects on page 288](#).
2. Drag the mouse to draw the approximate size that the rectangle is to be.

## Step 3: Test run the application in FactoryTalk View SE Client

In FactoryTalk View Studio, you can only test one graphic display at a time. To test navigating among displays, run the application in a FactoryTalk View SE Client.

To run the new application you just created, create a FactoryTalk View SE Client configuration file that specifies:

- The type and name of the application the client will connect to.
- The display to run initially when the client starts. This can be the new display you just created.



**Tip:** While you are developing an application, it is recommended that you test run the application in a FactoryTalk View SE Client at various stages. That way, you can resolve issues that might occur only at runtime.

**To create and run an SE Client configuration file:**

1. Select Start > Rockwell Software > FactoryTalk View Site Edition Client.
2. In the FactoryTalk View SE Client Wizard, click **Create a FactoryTalk View SE Client configuration file**, and follow the on-screen instructions. For details about options in the wizard, click **Help**.
3. When complete, click Run. The graphic display you specified to run initially will open in the FactoryTalk View SE Client window.

When you are finished testing, use the close button on the client's title bar to close the client window.

# Exploring FactoryTalk View Studio

This chapter describes:

- What FactoryTalk View Studio is.
- How to open an application in FactoryTalk View Studio.
- Parts of the FactoryTalk View Studio main window.
- Working in the Explorer window.
- Techniques for working in editors.
- Printing from FactoryTalk View SE.

## About FactoryTalk View Studio

FactoryTalk View Studio is the design-time environment for FactoryTalk View that provides the editors and tools you need to develop and test network distributed, network station, and local station human-machine interface (HMI) applications. It contains editors for creating complete applications, and contains client and server software for testing the applications you create.

You can also use FactoryTalk View Studio to set up FactoryTalk Security services for the applications you develop. For more information, see [Setting up security on page 77](#).



**Tip:** Optionally, you can use the FactoryTalk Administration Console to develop, manage, and secure multiple applications. However, you cannot use the FactoryTalk Administration Console to create or modify HMI servers and HMI project components.

To start FactoryTalk View Studio, do one of the following:

- Select **Start > FactoryTalk View Studio**.
- Use the command-line. For details, see [Start FactoryTalk View Studio using a command on page 42](#).

## Start FactoryTalk View Studio using a command

You can start FactoryTalk View Studio using a command.

### To start FactoryTalk View Studio using a command

1. In the Windows Run command, enter a command with the following syntax:

```
(64-bit OS) "C:\Program Files (x86)\Rockwell Software\RSView Enterprise
\VStudio.exe" /o /type:application_type /app:application_name [/
lang:startup_language]
```

where C: is the drive to which FactoryTalk View is installed.

2. Press **Enter**.

## Command-line parameters and examples

The following table shows the parameters used in the command to start FactoryTalk View Studio. Command-line parameters are case-insensitive. However, if a specified value includes a space, be sure to enclose the value in quotation marks (for example, "value with spaces").

Parameters	Descriptions
/o	Starts FactoryTalk View Studio.
/Type:value	Specifies the application type. The <i>value</i> is one of: <ul style="list-style-type: none"> <li>• <i>sed</i> – SE Network Distributed</li> <li>• <i>sen</i> – SE Network Station</li> <li>• <i>sel</i> – SE Local Station</li> <li>• <i>med</i> – Machine Edition</li> </ul>
/App:value	Specifies the application to be opened.
/Lang:value	Specifies the startup language. The value is a Locale ID in Decimal.

The following examples show how to use the commands.

- To start FactoryTalk View Studio for SE and open the *InstantFizz* application in the default language, type the following command:

```
"C:\Program Files\Rockwell Software\RSView Enterprise\VStudio.exe" /o /
type:sed /app:InstantFizz
```

- To start FactoryTalk View Studio for SE and open the *FTViewDemo* application in French (France), type the following command:

```
"C:\Program Files\Rockwell Software\RSView Enterprise\VStudio.exe" /o /
type:sed /app:FTViewDemo /Lang:1036
```

## About FactoryTalk Security permissions

If FactoryTalk Security services are used to secure parts of an application, to perform certain tasks, users must have the necessary security permissions.

For example, to create or modify the properties of an application, you must at least be allowed the Common actions Read, List Children, Write, and Create Children, at the FactoryTalk Directory that manages the application.

If you receive a FactoryTalk Security message while trying to perform such a task, contact your system administrator about the permissions you might require.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#).

## Opening an application in FactoryTalk View Studio

When you start FactoryTalk View Studio, the Application Type Selection dialog box opens. You can select:

- **View Site Edition (Network Distributed)** to create or open a FactoryTalk View SE network application (also called a network distributed application).
- **View Site Edition (Network Station)** to create or open a FactoryTalk View SE network station application.
- **View Site Edition (Local Station)** to create or open a FactoryTalk View SE local application (also called local station application).
- **View Machine Edition** to create or open a FactoryTalk View Machine Edition application. For information about developing machine-level applications, see *FactoryTalk View Machine Edition User's Guide*.

## To open an existing application:

1. Select **Start > FactoryTalk View Studio**.
2. In the **Application Type Selection** dialog box, select the application type and click **Continue**.
3. On the **Existing** tab, select the application and click **Open**.



**Tip:** If the existing application is not set up to support multiple languages and the <Select a Language> is shown in the Language list, you must select a language before you can open the application. For more information, see [Setting up language switching on page 210](#).

## HMI projects folder settings

By default, SE and ME HMI server projects are stored in C:\Users\Public\Documents\RSView Enterprise\SE\HMI projects and C:\Users\Public\Documents\RSView Enterprise\ME\HMI projects on the HMI server computer. To enhance security and prevent unauthorized modifications to these projects, Windows administrators can use the HMI Projects Folder Path Config Utility to specify a new folder to hold SE and ME HMI projects where only users with permissions can access.

Whoever configures the new HMI projects folder is given full control to the folder. After the configuration, the folder inherits its parent folder's security, and its access is configured for these user groups:

- NT AUTHORITY\IUSR: Full control

---

**NOTE:** This user group is not applicable to ME's HMI projects folder.

---

- NT AUTHORITY\SYSTEM: Full control
- NT AUTHORITY\LOCAL SERVICE: Full control
- NT AUTHORITY\BATCH: All permissions except for two advanced ones "Change permissions" and "Take ownership"
- BUILTIN\Administrators: Full control
- NT AUTHORITY\SERVICE: All permissions except for two advanced ones "Change permissions" and "Take ownership"

Do not change these user groups or their accesses as listed above, which are necessary for FactoryTalk View to work properly. To give more users access to the HMI projects folder, you must grant them full control and ensure that the subfolders and files inherit these permissions.

---

**NOTE:** Be aware of a Windows behavior when configuring the folder security. That is if a folder's access is configured to the Administrators group only, this folder is read-only to users even though they are in the Administrators group. Make sure that the user itself has read and write permissions to the folder.

---

If you choose to use the default folder, Windows administrators can tighten the Windows folder's security settings on the HMI server computer by following these steps:

- Remove the **INTERACTIVE** group from the folder's security properties.
- Add specific users or user groups and assign their permissions to this folder as needed.

## Opening the InstantFizz application

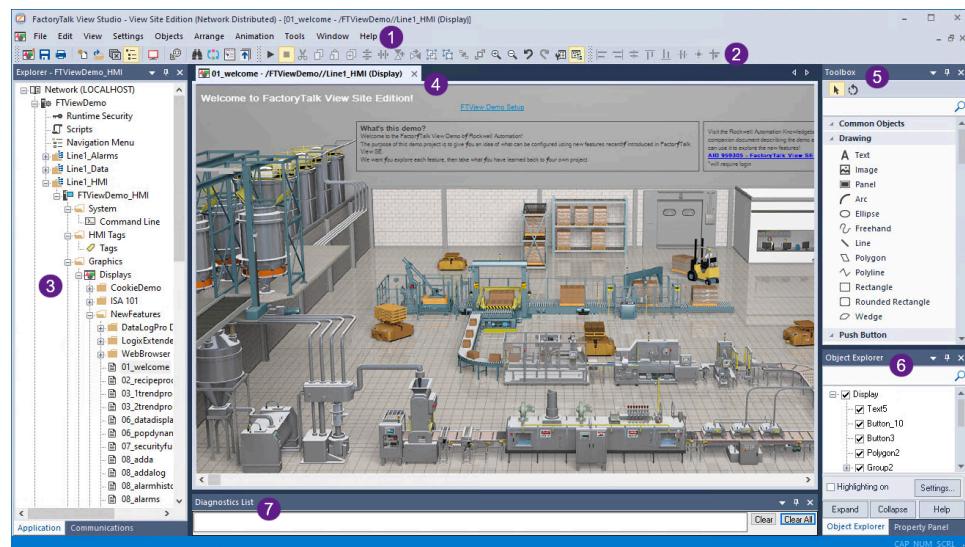
To familiarize yourself with the FactoryTalk View Studio development environment, you can open the InstantFizz application that is installed with FactoryTalk View SE.

### To open the application:

1. Select **Start > FactoryTalk View Studio**.
2. In the **Application Type Selection** dialog box, select **View Site Edition (Network Distributed)** and click **Continue**.
3. In the **Existing** tab, select **InstantFizz** and click **Open**.

## Parts of the FactoryTalk View Studio main window

The following illustration shows an overview of the main FactoryTalk View Studio window.



Many elements are common Windows elements and are not described here. The following are described because of their importance when using FactoryTalk View Studio.

Item	Name	Description
1	Menu bar	Contains the menu items for the active window. Each editor has its own set of menus.
2	Toolbar	Contains commonly used menu items providing quick access to the items without opening the menu.  All FactoryTalk View editors use the standard toolbar. The Graphics editor has additional toolbars that are shown when you open a graphic display, a global object display, or a library.
3	Explorer window	The main tool for working in FactoryTalk View Studio. Provides access to the

Item	Name	Description
		editors that you use to set up an application and create its components.
4	Workspace	The blank area of the FactoryTalk View Studio window. You can drag icons to the workspace from the Explorer to open editors and graphic displays.
5	Toolbox	<p>It is a searchable toolbox containing graphic objects.</p> <p>Right-click objects to add frequently used ones to Common Objects or to show the objects in a list view. Type the object name in the search box, and objects are filtered as you type.</p>
6	Object Explorer	<p>Shows a list of all the objects on the current graphic display. Clearing the checkbox of an object hides it from the graphic display.</p> <p>Grouped objects are listed as expandable items in the tree, with a + icon.</p>
7	Diagnostics list	Shows messages about system activities. You can specify the types of messages to show in the Diagnostics List, move the list, resize it, and clear the messages in it.

## Workbook mode

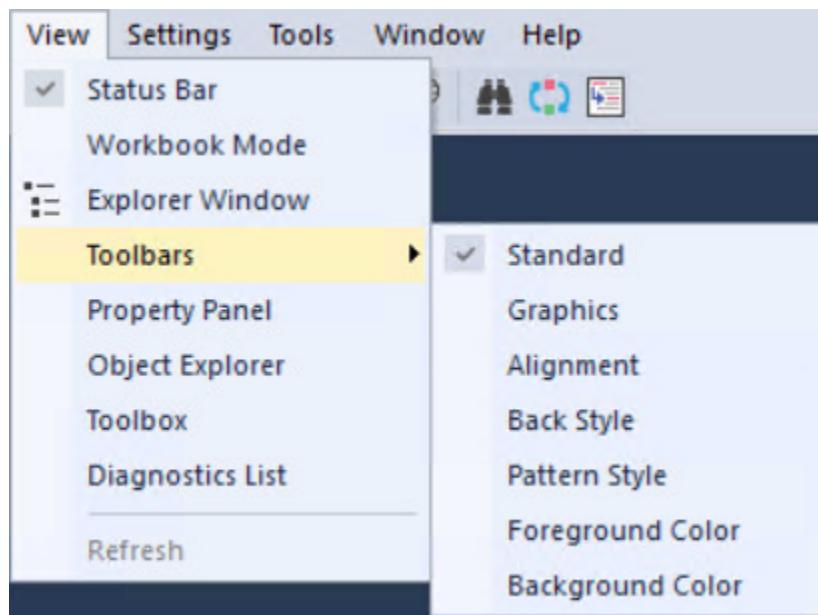
Workbook mode provides a different way to switch between open windows in the FactoryTalk View Studio workspace.

In workbook mode, windows are tabbed, making them easy to identify and select.

To switch between workbook and normal modes, from the **View** menu, choose **Workbook Mode**.

## Showing and hiding items in the main window

To show or hide elements of the main window (except the menu bar), select items on the **View** menu: if there is a check mark beside an item, the item is visible; if there is no check mark, the item is hidden.



In this example, the Status Bar is visible and the Standard toolbar is visible, while the rest is hidden.

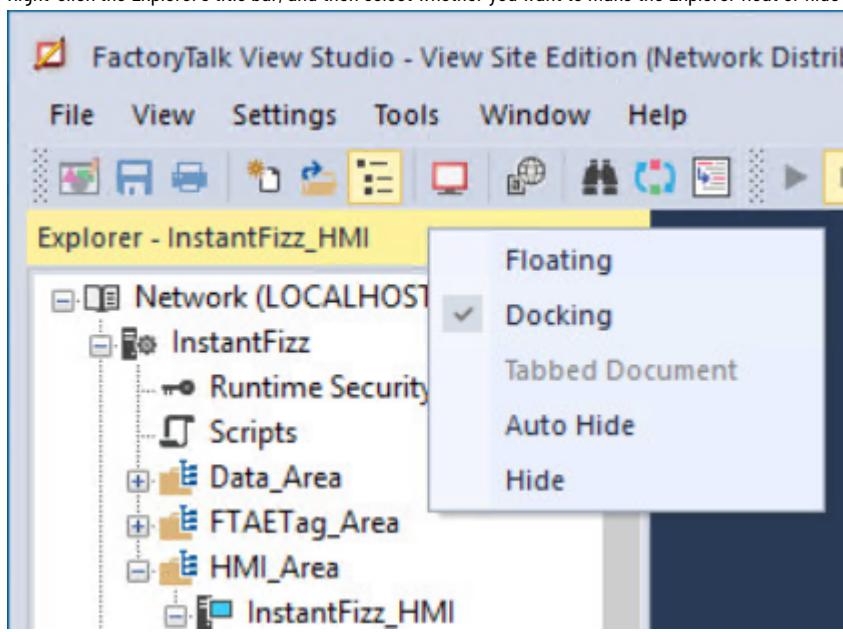
## Working in the Explorer window

The Explorer tree is the main tool for working in FactoryTalk View Studio. It provides access to the editors that you use to set up an application, and to create and modify its components.

## Moving the Explorer window

To undock the Explorer, use one of these methods:

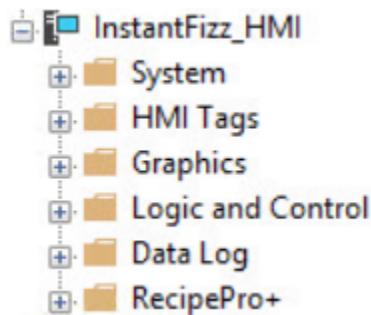
- Click the Explorer's title bar, and then drag the window. To prevent the Explorer from docking automatically, press and hold the **Ctrl** key as you drag.
- Double-click the Explorer's title bar. This makes the Explorer float. You can then drag it to a new position, or double-click the title bar again to automatically dock it.
- Right-click the Explorer's title bar, and then select whether you want to make the Explorer float or hide it.



As long as it is undocked, you can resize the Explorer. To do this, click an edge or corner of the window, and then drag until it's the size you want.

## Opening and closing folders

The Explorer uses folders to organize editors.



To open or close a folder, use one of these methods:

- Click the + or - symbol beside the folder's icon.
- Double-click the folder.
- Click the folder, and then press **Enter** to open or close the folder, depending on its current state.

## Organize graphic displays

Use folders to organize graphic displays in FactoryTalk View Studio. These folders are logical folders and do not represent the actual location of files on your hard drive. They are intended for design time and do not affect the display names at runtime or runtime display navigation. You can create up to two levels of folders.

### To organize graphic displays in folders

1. In the FactoryTalk View Explorer window, expand the HMI server and the **Graphics** folder, right-click **Displays**, and then select **New Folder**.
2. In the **Save** dialog box, enter a name for the folder, and then click **OK**.
3. Create displays in the folder or drag displays to the folder.

## Opening component editors

Use the editors in FactoryTalk View Studio to create or modify application components. Editors are represented by icons in the Explorer window, or by items in menus.

To open an editor, use one of these methods:

- Double-click one of the components in an application (for example, a display).
- Drag the editor's icon from the Explorer to the workspace.
- Right-click the editor's icon, and then click **New** or **Open**.



**Tip:** If the editor can create multiple components, the shortcut menu contains **New**. If the editor can only open a single component, the menu contains **Open**.

- From the **Tools** or **Settings** menu, select the editor you want to open.

Following is a list of FactoryTalk View SE editors and their functions. If you can open an editor from the FactoryTalk View Studio menu bar, the name of the menu is noted.

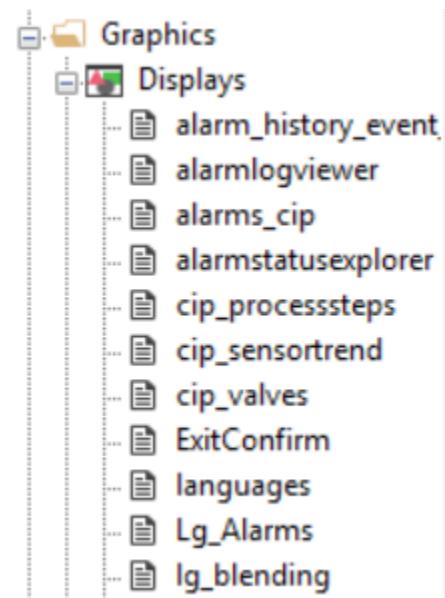
To do this	Use this editor
Set up runtime security accounts for FactoryTalk View, assign security codes to users and groups, and assign login and logout macros.	Runtime Security (Settings menu)
Assign security codes to FactoryTalk View commands and macros.	Runtime Secured Commands (Settings menu)
Specify the type of system activity to be logged and where, when, and how it will be logged.  <b>TIP:</b> The settings you specify in this editor apply <b>only</b> to the computer you are using.	Diagnostics Setup (Tools menu)
Add languages to an application, set up a default language, and export and import text strings.	Languages (Tools menu)
Create server-side script with Windows PowerShell.	Scripts
Open a command line to run FactoryTalk View commands and macros.	Command Line

To do this	Use this editor
Create HMI tags.	Tags
Create graphic displays and global object displays, or open the graphic libraries to use the library objects.	Graphics (Displays, Global Objects, or Libraries icon)
Add images to an application that are to be used repeatedly in graphic displays.	Images
Create parameter files, to specify tags whose values will replace tag placeholders in graphic displays at runtime.	Parameters
Create multiple recipes, with ingredients, tag sets, and data values stored together in a recipe file.	RecipePro+
Create local message files containing trigger values, and the corresponding messages.	Local Messages
Create tags whose values are derived from other tags, or from functions.	Derived Tags
Create events, which are expressions that trigger FactoryTalk View commands or macros.	Events
Create macros to run series of FactoryTalk View commands.	Macros
Remap the FactoryTalk View SE Client keyboard to run FactoryTalk View commands.	Client Keys
Create data log models to specify where, when, and how selected tag values will be logged.	Data Log Models

## Creating and modifying components

The Explorer manages components that represent physical files in folders under the application directory.

Display components are created using the Graphics editor.



## To display components associated with a particular editor:

Use one of these methods:

- Click the + sign beside the editor's icon.
- Double-click the editor's icon.
- Click the editor's icon, and then press **Enter**.

Components created with the editor are shown under the editor's icon in the tree.

## To create a component:

Use one of these methods:

- Drag the editor's icon into the workspace.
- Right-click the editor's icon, and then click **New**.

## To open a component:

Use one of these methods:

- Double-click the component.
- Drag the component into the workspace.
- Right-click the component, and then click **Open**.

## Adding components to an application

1. In FactoryTalk View Studio, in the Explorer window, right-click the editor's icon and select **Add Component into Project**.
2. In the **Add Component into Project** dialog box, locate the existing component you plan to add and select **Open**.



**Tip:** You can also drag components from the Windows Explorer and from the FactoryTalk View Studio window.

When you use this option, you create a copy of the file in the application.

To select several consecutive components, click the first component you want to select, press the **Shift** key, and then click the last component. To select several individual components, press the **Ctrl** key, and then click each component.

## Naming components

When you name a component in FactoryTalk View Studio, the file name is created on disk automatically. The component name shown in the Explorer window becomes part of the file name.

FactoryTalk View supports long file names with a maximum limitation of 260 characters including spaces. When considering a file name, mind the four characters for the extension (.gfx) and all the characters necessary for the path leading to the file. The remainder would be the number of characters, including spaces, you can use for a file name.

Different files will be dependent on the path where they reside.

For example, including spaces, there are 95 characters in the following path and file name, as the limit is 260, leaving 165 additional characters, including spaces, allowable:

```
C:\Users\Public\Documents\RSView Enterprise\SE\HMI projects\InstantFizz_SE\Gfx\cip_overview.gfx
```

Another example path is shown below, which is based on saving a macro:

```
C:\Users\Public\Documents\RSView Enterprise\SE\HMI projects\InstantFizz_SE\Mcr
```

That is 78 total characters just for the path,  $260 - 78 = 182$ . The filename can be up to 182 characters including spaces and allowing 4 characters for the extension.

## Using component names that contain spaces

Component names can contain spaces, like the graphic display name in the previous example. When using component names with spaces in commands, you might need to enclose the component names in double quotes ("").

For example, to load a parameter file with spaces in its name, you would type the following **Display** command:

```
Display Screen1 /P"Bottling Line 1"
```



**Tip:** For information about the command syntax for specific FactoryTalk View commands, see the FactoryTalk View Site Edition Help.

## Avoiding names that conflict with commands or macros

To avoid problems when issuing commands and macros, do not give macros the same names as commands. For example, instead of giving a macro the name **Display**, to avoid confusion with the **Display** command, name the macro **DisplayScreen** instead.

## Deleting, removing, and renaming components

In FactoryTalk View Studio, when you:

- Delete a component, the component *and* the corresponding physical file are deleted.
- Remove a component, the component is removed from the Explorer window, but the physical file is not deleted.
- Rename a component, the component and the physical file are both renamed.

### To delete a component and file:

In FactoryTalk View Studio, in the Explorer window:

- To delete one component, right-click the component, and then select **Delete**.
- To delete multiple components, hold **Ctrl** or **Shift** to select the components, and then press **Delete**.

### To remove a component:

In FactoryTalk View Studio, in the Explorer window, right-click the component you want to remove, and then select **Remove**.

### To rename a component:

1. In FactoryTalk View Studio, in the Explorer window, right-click the component you want to rename, and then select **Rename**.
2. Type the new name, and then click **OK**.

## Restoring a component

If you rename a physical file in Windows Explorer, you will no longer be able to open the corresponding component in FactoryTalk View Studio. You must remove the component, and then add the renamed file back into the application.

### To restore a component after changing the physical file name:

1. In FactoryTalk View Studio, in the Explorer window, right-click the component, and then select **Remove**.
2. Right-click the icon of the editor used to create the component you want to restore, and then select **Add Component into Project**.  
For example, to restore a graphic display, right-click the Displays icon.
3. Find and select the renamed physical file, and then click **Open**.

## Techniques for working in editors

Many of the editors you will use to develop a FactoryTalk View application have similar features, which require similar information. Knowing about these features saves time.

## Gaining quick access to common operations

Shortcut menus provide quick access to actions that you perform frequently in different environments.

In the Explorer, everything in the tree hierarchy has a shortcut menu, except the folders. In the Graphics editor, graphic displays and objects in the displays have shortcut menus.

To open a shortcut menu, position the pointer over an icon, graphic display, or object, and then right-click.

## Browsing lists and components

The Browse button is beside fields that require input. For example, instead of typing information in a text box, you can click the Browse button to open:

- A list containing valid entries for the text box.
- A dialog box or browser, such as the Tag Browser, in which you can search for and select a valid entry.
- An editor, in which you can create a valid entry.

## Supplying tag names

Tags are stored in devices, data servers, and in the HMI tag database.

To supply a tag name, use one of these methods:

- Type the name of the tag. You do not have to create the tag in order to use its name, but be sure to create the tag later, or errors will be reported at runtime.
- Click the **Browse** or **Tags** button (whichever is available) to open the Tag Browser, where you can select a tag.

For more information about tags, see [Working with tags on page 159](#).

## Selecting and building commands

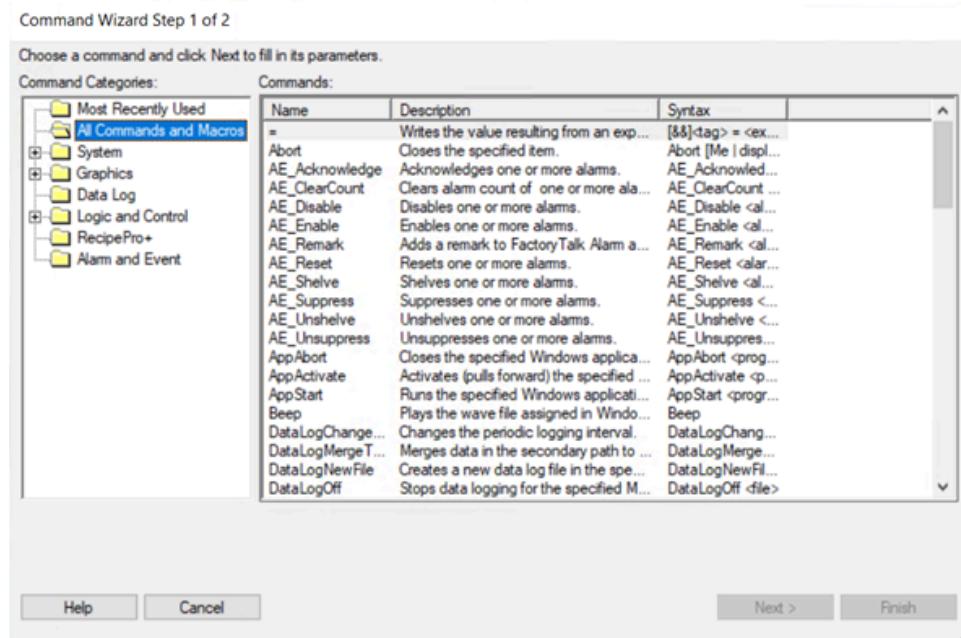
You can use FactoryTalk View commands to interact with and control application components. Most commands accept parameters for added precision and control.

You can set up keys and graphic objects to run commands at runtime. You can also run commands from the HMI server's command line, or create a list of commands in a macro, and run the macro in places where the commands are required.

Use the Command Wizard for assistance with selecting and building commands. For information about specific FactoryTalk View commands, click **Help**.

### To open the Command Wizard, use one of these methods:

- Click the **Browse** button beside a text box where a command is expected, for example, in the Press action text box for a button object, or in the Command Line.
- Double-click in a box where a command is expected or in the Macros editor.
- In the **Command Line**, **Runtime Secured Commands** editor, **Events** editor, and **Macro** editor, from the **Edit** menu, select **Commands**.



## Building expressions

Many editors use expressions to compute values based on tags and functions. Expressions can be complex logical expressions, or they can be tag names.

For more information, see [Creating expressions on page 344](#).

## Printing from FactoryTalk View SE

From each FactoryTalk View SE editor, you can print selected items or the editor's entire contents.

**To print selections:**

1. Select the item that you want to print, for example, a record in an editor's spreadsheet.
2. From the **File** menu, select **Print**.
3. Under Print Range, select **Selection**.

**To print the entire contents of the window:**

1. From the File menu, click **Print**.
2. Under Print Range, click **All**.

## Selecting a printer

You must install a printer before you can select it.

FactoryTalk View can print to a network printer. For information about installing and setting up printers, see your Windows documentation.

**To select a printer:**

1. In any FactoryTalk View editor, from the **File** menu, choose **Print Setup**.
2. If you do not want to use the default printer, specify another printer.
3. Choose the appropriate orientation and paper options.



**Tip:** Print Setup settings apply to the development computer only. If the application will run on a different computer, you must also set up a printer on that computer.

## Printing at runtime

To print graphic displays at runtime, use the **PrintDisplay** command.

To let an operator print specific displays, provide a way to run the command for these displays. For example, create a button object, display key, or client key using **PrintDisplay** as the press action.

When you use the **PrintDisplay** command, FactoryTalk View prints the entire display, including parts that are covered by other displays. To print an image of whatever shows on a specified monitor, use the **ScreenPrint** command.

For more information about the commands, see the FactoryTalk View Site Edition Help.

# Planning an application

This chapter describes:

- Understanding the process you are automating.
- Planning the layout of the network.
- Planning communications.
- Planning how to monitor and control alarms.
- Deciding when to use HMI tags.
- Designing a dependable control system.
- Setting up the runtime application.
- Designing a system that is easy to deploy and maintain.
- Integrating with other applications, and customizing the system.

## Understanding the process you are automating

To design an effective control system using FactoryTalk View Site Edition, it is important to understand the process you are automating.

Gather information from a variety of sources, and pay attention to details that might affect the overall performance of the system. To get planning underway:

- Talk to operators and other experts who will use the system, to get their input about ways to optimize plant operations.
- Talk to management and information systems staff, to find out what they need to support planning and design decisions.
- Break up the process into its constituent parts. This will help you visualize how parts of the automated system might be distributed over a network.
- Identify the process variables that you need, and their locations in the programmable controllers or devices.
- Determine which types of data servers the system will use.
- Determine which processes will generate alarms, and how to monitor alarms.
- Determine which parts of the process should be secured, and the most efficient ways to restrict user access.
- Assess the need for redundancy in different parts of the system.

## Planning the network layout

The layout of the network is particularly important to the design of a network distributed application.

For information about installing the FactoryTalk View SE software, and about system requirements that might affect the network design, see *FactoryTalk View Site Edition Installation Guide*.

## Choosing a Windows domain or workgroup

For network applications consisting of more than 10 computers pointing to the same FactoryTalk Directory, a domain controller is strongly recommended, but not required. The following domain controllers are supported:

- Windows Server 2022 Standard
- Windows Server 2022 Datacenter

- Windows Server 2019 Standard
- Windows Server 2019 Datacenter

**NOTE:**

Do not install FactoryTalk Directory, FactoryTalk View SE Server, or any other application software on the same computer as the Windows domain controller. This configuration is not supported.

## Windows workgroups

For network distributed applications consisting of 10 computers or fewer pointing to the same FactoryTalk Directory, FactoryTalk View SE can be used in a Windows workgroup network environment.

When setting up security for applications in a Windows workgroup, it is recommended that you use the FactoryTalk Directory user accounts. This simplifies account management by centralizing security services at the FactoryTalk Directory.

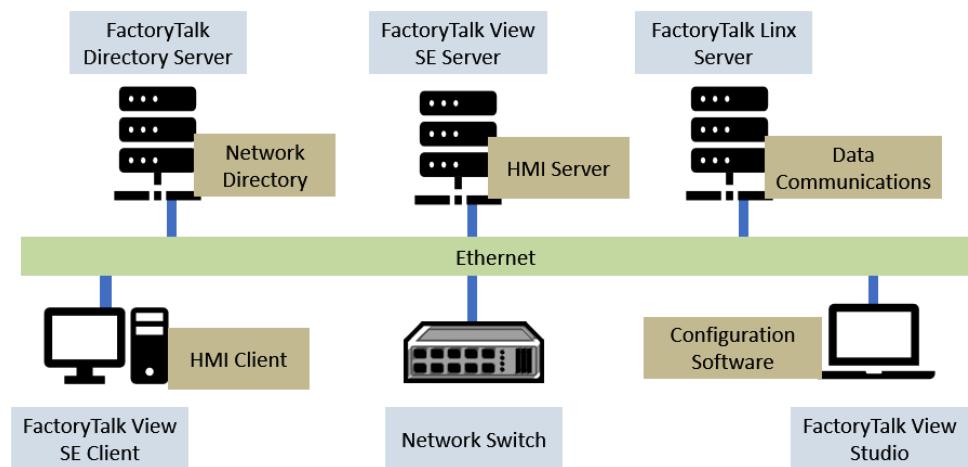
For information about user accounts, see [Setting up security on page 77](#).

## Determining computers needed

You can develop and then test a network distributed application on a single computer. However, network application components are usually distributed over a network, on multiple computers.

For information about installing the FactoryTalk View SE software, and about system requirements that might affect the network design, see *FactoryTalk View Site Edition Installation Guide*.

This example of a distributed system includes a FactoryTalk Directory server, FactoryTalk View SE server and client, FactoryTalk Linx data server, and FactoryTalk View Studio for application development and configuration.



Depending on needs, a network distributed application deployed for production typically involves the following computers:

- **One computer running the FactoryTalk Network Directory server**

FactoryTalk Directory centralizes access to application resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed applications, and network station applications. All of the client and server computers participating in a given network distributed application, or network station application must point at the same Network Directory.

For more information, see [Setting up the FactoryTalk Directory on page 70](#) and [Working with network on page 114](#) distributed applications.

- **One or more redundant pairs of computers running FactoryTalk View SE Server**

The HMI servers in an application store HMI project components, such as graphic displays, and serve the components to application clients.

HMI servers also perform historical data management (logging), and contain a database of tags.

The FactoryTalk View SE Server software must be installed on computers that are to run HMI servers. For redundant HMI servers, the software must be installed on both primary and secondary (or backup) HMI server computers.

For more information about redundant HMI servers, see [Setting up FactoryTalk system availability on page 232](#).

- **One or more redundant pairs of computers running FactoryTalk Alarms and Events Server**

- **One or more redundant pairs of computers running data servers**

Clients communicate through data servers, which provide access to information in devices and other OPC data servers. For information about data servers in a FactoryTalk View SE application, see [Setting up communications on page 149](#) and [Working with network on page 114](#) distributed applications.



**Tip:** If you use an OPC data server instead of FactoryTalk Linx, it is recommended that you run the data server on its own, dedicated host computer.

- **One or more engineering workstations running FactoryTalk View Studio**

FactoryTalk View Studio is configuration software for developing and testing FactoryTalk View SE applications.

For information about FactoryTalk View Studio features, see [Exploring FactoryTalk View Studio on page 42](#).



**Tip:**

FactoryTalk View Studio is also the configuration software for developing FactoryTalk View ME (ME) applications. For information about FactoryTalk View ME, see *FactoryTalk View ME User's Guide* and the ME Help.

- **One or more operator workstations running FactoryTalk View SE Client**

The FactoryTalk View SE Client software provides a runtime environment for operators to interact with FactoryTalk View SE network distributed, network station, and local station applications.

For details about setting up FactoryTalk View SE Clients, see the FactoryTalk View Site Edition Help.

- **One or more maintenance computers running FactoryTalk View SE Administration Console**

The FactoryTalk View SE Administration Console is software for administering FactoryTalk View SE applications after they are deployed.

Use the subset of editors available in the Administration Console to make minor changes to an application without installing FactoryTalk View Studio.

**Tip:**

SE Administration Console will close automatically after being open for two hours.

## Planning communications

To plan communications for the control system, gather information about:

- The types of controllers or devices in the production line, and the software that is available for communicating with them. This will help determine the number of data servers the application needs.
- How the application will access and collect data in the controllers and devices.
- Which systems need to be redundant, to minimize disruptions to clients in the event that communications are interrupted. This will help determine the number of redundant data server pairs the application needs.

For more information, see [Setting up communications on page 149](#).

## Determining how to access data

To access values in programmable controllers or devices, you can use data server tags, HMI tags, or a combination of both.

For many purposes, you can access the values in controllers or devices directly, using a data server in the application. For some purposes, you will need to use tags from an HMI server's tag database.

To determine which method is most appropriate, you must know what kinds of controllers or devices you are using, and how the application will communicate with the controllers or devices.

## Collecting only the necessary data

Design the control system so that only essential data is collected. This will reduce the processing power required for data collection, and help control the amount of traffic on the communication channel or network.

You should also consider data collection requirements when designing the layout of the programmable controller data tables and the tag database.

Ideally, to decrease network traffic and optimize system response, tag addresses should reference contiguous blocks of programmable controller data tables.

## Monitoring and controlling alarms

Alarms are an important part of most plant control applications because they alert operators when something goes wrong.

An alarm can signal that a device or process has ceased operating within acceptable, predefined limits, or it can indicate breakdown, wear, or a process malfunction. Often, it is also important to have a record of alarms and whether they were acknowledged.

In FactoryTalk View SE, you can use FactoryTalk Alarms and Events services, to centralize the distribution of alarm information from various devices in your application, to runtime clients and logs.

For more information about alarm monitoring and control, see [Setting up FactoryTalk alarms on page 177](#).

## Planning an alarm monitoring and control system

Before deciding on the types of alarm monitoring and control to use in your system, plan:

- What conditions will trigger alarms.
- How operators will be notified of those alarms.
- What information alarm messages should contain.
- What actions will occur in response to those alarms.
- Which alarms will require additional actions that can be supplied using FactoryTalk View commands.

## Advantages of using FactoryTalk Alarms and Events services

FactoryTalk Alarms and Events offers a single, integrated set of alarm information, distributed through supporting alarm servers. This means that all clients in a FactoryTalk system can receive a consistent view of device-based, tag-based, and OPC UA alarm activity.

FactoryTalk Alarms and Events services are also integrated with FactoryTalk Security services. This means you can set up FactoryTalk system users and computers with different levels of access to alarm sources and data.

The decisions you make about setting up alarm monitoring and control will depend on the design of your FactoryTalk View SE application, the types of devices the application requires, and the processes you need to monitor for alarms.

## Choosing device-based alarms

Use FactoryTalk device-based alarm monitoring with Studio 5000 controllers that have been programmed using pre-built alarms or alarm instructions.



**Tip:** For a list of Studio 5000 controllers that support FactoryTalk Alarms and Events services, see [FactoryTalk Alarms and Events Help](#).

If you use device-based alarms, you only have to program alarms or alarm instructions once, in the controller: there is no need to create alarm definitions for HMI tags.

Since alarm detection takes place in the controller, processing is faster, time stamps are more accurate since they come from the controller, and alarm states are preserved more reliably.

FactoryTalk View SE Clients receive device-based alarm data by way of Rockwell Automation Device Servers (FactoryTalk Linx) that you add to a FactoryTalk View SE application. For details, see [Working with Rockwell Automation Device Servers on page 189](#).

## Choosing server tag-based alarms

FactoryTalk tag-based alarms are set up by specifying alarm conditions for tags in devices that do not have built-in alarm detection.

Use tag-based alarms to include these devices in an integrated FactoryTalk Alarms and Events system.

You can set up tag-based alarms for tags in older programmable controllers (PLC-5 or SLC 500), for tags in third-party devices communicating through OPC data servers, or for HMI tags in an HMI server's tag database.

If you prefer not to set up built-in alarm detection, you can also set up tag-based alarms for Studio 5000 controllers that do support device-based alarms.

FactoryTalk View SE Clients receive tag-based alarm data by way of Tag Alarm and Event Servers that you add to a FactoryTalk View SE application.

For details, see [Working with Tag Alarm and Event Servers on page 191](#), and [Setting up FactoryTalk tag-based alarms on page 192](#).

## Deciding when to use HMI tags

Use HMI tags in an application to provide extended capabilities, such as:

- Scaling or offsetting a value before sending it to a programmable controller or device.
- Security features that prevent unauthorized changes to a tag's value.
- Flexible addressing. HMI tags do not require hard-coded physical addresses or device-specific variable names. This means you can reuse an application with other devices, simply by changing the physical addresses the tag names are mapped to.
- Descriptive tag names, which you cannot create in some controllers or OPC servers.

For more information about HMI tags, see [Working with tags on page 159](#).

## Designing the HMI tag database

Before creating an HMI tag database, take the time to plan it. A good design helps reduce the maintenance time, and can improve device-to-FactoryTalk View response time.

Collect the following information:

- Process flowcharts (or process and instrument diagrams)
- A list of programmable controller data table or register addresses for the application

## Organizing HMI tags

Before creating the HMI tags, plan how to organize them:

- Develop naming conventions for the HMI tags. Choose names that are familiar and logical to everyone. This makes troubleshooting easier.
- Group related HMI tags in ways that make most sense for the application. For example, group all similar devices, or group related areas of the plant floor.

To group related HMI tags, you can create folders in the Tags editor. For greater organization, create nested folders.



**Tip:** Do not place all HMI tags in the root folder of the database. HMI tags contained in nested folders do not contribute to the total number of tags in the root folder. It is recommended that you limit the number of tags in any folder to less than 1000.

## Designing a dependable control system

Planning which parts of the control system to secure and which to make redundant will help you design a more dependable, available system.

## Planning how to secure the system

When planning how to secure the control system, consider:

- The types of users or groups of users that require access to different areas of the plant, or parts of the control system.



**Tip:** Planning security for groups of users is recommended, to simplify management of users with common security needs.

- Who will have administrative privileges, for example, to set up security for the system.
- Whether and when users must sign in to the system, or change their passwords.
- Which HMI project components to secure, for example, to prevent accidental changes to graphic displays, or to control who can write to certain HMI tags.
- Whether to restrict access to computers in certain areas of the plant.

For more information, see [Setting up security on page 77](#).

## Planning to use built-in system availability features

A complete FactoryTalk system consists of all the networks, devices, and software applications you have deployed to monitor and control your plant or process.

Helping to ensure that the system can provide data in a secure and predictable fashion depends on a number of variables.

To minimize data loss and down time, and to help ensure that critical parts of your system are always available to connected clients, FactoryTalk View SE provides the health monitoring and redundancy features.

- Server status monitoring of non-redundant and redundant application servers.
- Disconnected operation. For example, connected clients can continue to run when the FactoryTalk Directory becomes unavailable.
- Redundant application servers. In a network distributed application, you can set up redundancy for application servers. These include FactoryTalk View SE Servers (also called HMI servers), FactoryTalk Alarms and Events servers, Device Servers (FactoryTalk Linx), and OPC data servers (RSLinx Classic or other data servers that comply with the OPC DA 2.05a, UA 1.02 and 1.03 specification).
- Support for online changes to HMI tags. For information about this feature, see [Modifying HMI tag and alarm properties at runtime on page 248](#).
- Save project online edits to both primary and secondary servers. For information about setting up redundancy options, see [Set up HMI server redundancy options on page 242](#).
- Network connection monitoring on each computer (clients and servers) in the system. For information about this feature, see [Monitoring network client and server connections on page 251](#).

For more information, see [Setting up FactoryTalk system availability on page 232](#).

## About redundant application servers

In theory, the ideal redundant solution includes at least one backup copy of everything—hardware, software, networks, and so on. In practice, this is seldom feasible, or even necessary.

Before setting up redundancy, plan:

- Which components in the system need to have backups—in other words, decide how much redundancy is necessary.
- Where (on which computers) to locate backup systems.
- The network layout, and calculate the processing load expected for each computer. This information can help you plan which parts of an application can share hardware.

In a FactoryTalk View SE network distributed application, you can set up redundant pairs of HMI servers, Alarms servers, and data servers. For more information, see [Setting up FactoryTalk system availability on page 232](#).

In addition, you can set up FactoryTalk View SE to send data log and activity log information to an ODBC-compliant database automatically. You can also set up an HMI server to buffer data locally if the database becomes unavailable.

For information about:

- Data logging, see [Setting up data logging on page 372](#).
- Activity logging, see [Logging system activity on page 253](#).

## Setting up the runtime application

When designing graphic displays, decide on the best ways for users to navigate through the application. To assist with navigation:

- Develop a hierarchy of displays.
- Plan the multi-monitor deployment.
- Create display templates.
- Apply visual design principles that help users interact with the displays.

For more information about working with graphic displays and objects, see [Creating graphic displays on page 259](#) and [Creating graphic objects on page 288](#).

## Developing a hierarchy of graphic displays

Well-organized graphic displays present information clearly and consistently and guide users through the system.

A hierarchy is a series of graphic displays that provide progressively more detail as users move through them. Design the hierarchy to meet the needs of all application users, including managers, supervisors, and operators.

Before designing individual graphic displays, plan:

- An application-wide hierarchy of displays that includes overviews and menus.
- How users will navigate through the hierarchy.
- Which displays to dock inside the FactoryTalk View SE Client window.

For more information, see [Setting up navigation on page 334](#).

## Creating templates to ensure consistency

To maintain consistency in the appearance of graphic displays in an application, present the same information and basic functions in the same places on each display. This makes it easier for users to find similar elements as they navigate from display to display.



**Tip:** Use global objects to make templates easier to distribute and maintain. For information about global objects, see [Creating graphic displays on page 259](#).

To ensure uniformity, develop displays with common elements that act as templates. Each time you develop a new display, start with a copy of the appropriate template. For example, a template might contain:

- The company logo.
- A title.
- The date and time.
- Navigation buttons.

## Planning the multi-monitor deployment

FactoryTalk View SE supports the automatic and dynamic management of multi-monitor deployment for HMI application displays. Multi-monitor provides you with more screens to view more information about process operations from a single workstation.

FactoryTalk View SE multi-monitor deployment offers the following advantages:

- The monitor configuration is defined per client and can be unique per operator station.
- A client configuration can support an individual configuration of up to six monitors per HMI client.
- HMI displays can be displayed on the specific pre-configured monitors or dynamically based on the relative location of the originating request.
- Applications can span multiple monitors. You can easily drag the HMI content from one monitor to another.

For more information about how to design new applications or migrate existing applications for multi-monitor deployment, see [Designing applications for multiple monitors on page 66](#).

## Applying visual design principles

When designing graphic displays, apply visual design principles that help operators and increase their efficiency. Consider employing the following important principles.

### Consistency

- Be consistent in the use of symbols and color.
- Be consistent with button labels and button placement.

When you design several displays, place the same kinds of buttons in the same positions. For example, if there is a Start button in a certain position in one display, don't put a Stop button in the same position in the next display.

## Clarity

- Use symbols that are easily recognizable. For example, use the conventional ISA symbols for tanks and valves, such as the symbols found in the Symbol Factory library.
- Do not overload the display with information.
- Use standard, clear terminology, and avoid abbreviations or acronyms that the user might not understand.
- Use colors with recognizable meanings. For example, in Europe and North America the colors red and green often mean stop and start. Keep color meanings consistent by assigning red only to Stop buttons, and green only to Start buttons.
- Use high-contrast color combinations, such as light text on dark-colored backgrounds.

## Usability

- If you're designing for a touch screen, place important buttons where they will not be blocked by a pop-up window. Users can't press a covered button.  
Also ensure that buttons are large enough and spaced far enough apart for users to touch them easily, even when wearing work gloves.
- Ensure that there is always a clear way to move between displays.

## Planning how to use trends

When designing trends, consider how they will be used in the application. For example, a trend might be used to:

- Analyze process trends.
- Monitor production efficiency.
- Archive process variables to ensure compliance with government regulations.

Based on such considerations, you can determine which tags need to be plotted on the same trend, from a data log model, against time, or against another tag.

For more information, see [Setting up trends on page 394](#).

## Planning runtime language switching

When designing an application that will support multiple languages, consider:

- How operators will switch between languages at runtime.  
For example, you can create buttons that run the **Language** command when pressed for each language the application supports.
- Whether operators will need to switch languages. If so, ensure that they have sufficient security privileges to open graphic displays that contain language switching buttons.
- Displaying the different languages consistently and effectively in graphic displays.  
For example, if you use a caption to identify a button set up to switch to French, the caption will change whenever a language switch occurs. To avoid this, use an image of the French flag on the button, instead.

For more information, see [Setting up language switching on page 210](#).

## Designing a system that is easy to deploy and maintain

FactoryTalk View SE includes tools and features that can help you create applications that are easier to use and maintain. For example, you can:

- Create global objects and distribute copies of the objects throughout the application. When you modify the global object, the changes are applied to all linked copies.

For more information about global objects, see [Creating graphic displays on page 259](#).

- Create special graphic displays to help deployment and maintenance personnel test an application, for example, to troubleshoot communications problems.

- Link parts of an HMI application to logic in programmable controllers, using the **OpenRSLogix5000** command.

For example, you can link actions performed in FactoryTalk View SE to Sequential Function Charts (SFCs) in Studio 5000 Logix Designer, to help operators track the state of processes that are running in the plant.

For details about using the **OpenRSLogix5000** command, see the FactoryTalk View Site Edition Help.

## Designing the application for multiple users

Behavior that is appropriate in a single-user environment might not be appropriate for multiple users.

When designing a network distributed application, consider issues such as the following:

- If a graphic display that is to run on several clients uses a shutdown macro to stop a derived tags file, closing the display on any of the clients will stop derived tags processing.

Since derived tags processing occurs at the HMI server, this would affect all clients that require the derived data. For more information about derived tags, see [Adding logic and control on page 423](#).

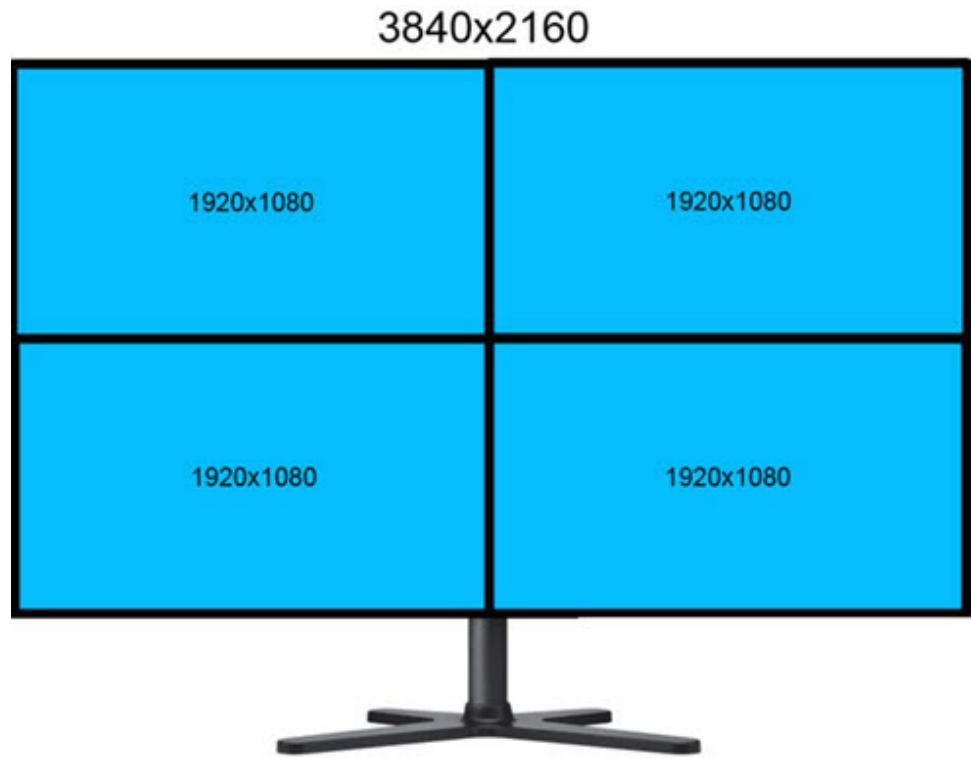
- Since tag values are obtained globally, across a network distributed application, do not use them to store local information. For example, if you use a tag value to indicate the last display shown, all clients connected to the application will receive the same tag value.

## Designing applications for multiple monitors

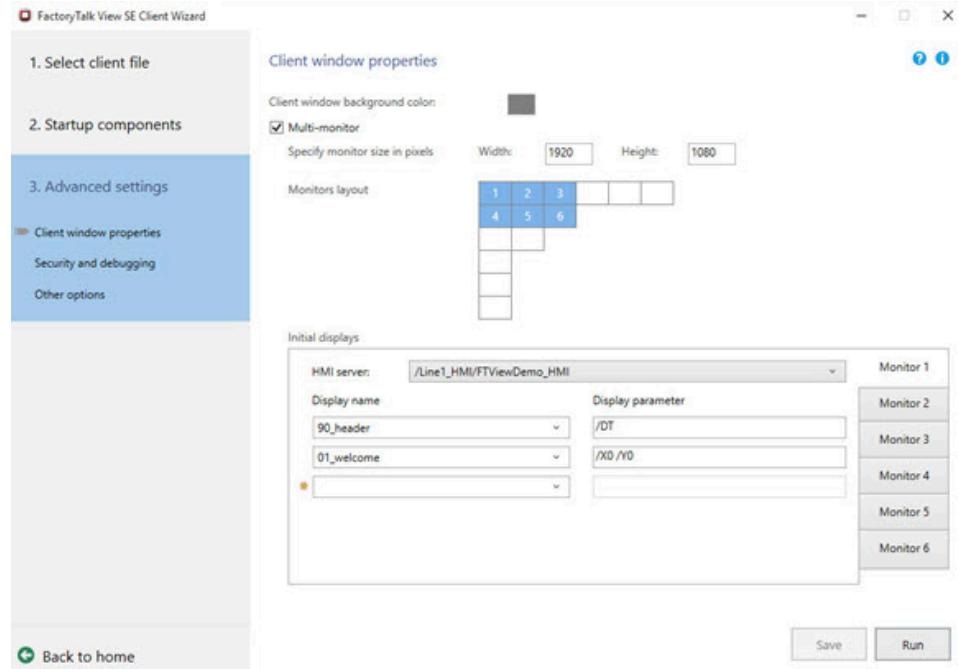
An application designed for a single monitor can be deployed to an operator station with multiple monitors without additional engineering.



A single monitor can be configured to be a multi-display area. This allows you to take advantage of very large resolution monitors without third-party drivers or special hardware.



To design applications for the multi-monitor deployment, in the SE client configuration file, enable the multi-monitor option and configure each monitor accordingly. For more information about FactoryTalk View SE Client Wizard, see FactoryTalk View SE Help.



## Example 1: Migrate existing applications for multiple monitors

1. In the FactoryTalk View SE Wizard, open the existing configuration file.
2. In the **Advanced settings** page, select **Multi-monitor** and configure each monitor accordingly, such as monitor size, monitor layout and initial displays.



**Tip:** If needed, specify a navigation display as the initial display for each monitor. This helps navigate between different displays.

3. Save and test the client configuration file.

## Example 2: Design new applications for multiple monitors

1. In FactoryTalk View Studio, design and develop your applications.
2. If needed, in the **Display** command, use the **/M** and **/position** parameters to specify the monitor and window position where a display is shown. For more information about the **Display** command, see FactoryTalk View SE Help.
3. In the FactoryTalk View SE Wizard, create a client configuration file and specify the startup components.
4. In the **Advanced settings** page, select **Multi-monitor** and configure each monitor accordingly, such as monitor size, monitor layout and initial displays.
5. Save and test the client configuration file.

## Integrating with other applications and customizing the system

If you regularly require data from programs such as Microsoft Excel, or Microsoft SQL Server, consider using the FactoryTalk View SE Client object model and display code with VBA to integrate these applications with FactoryTalk View SE.

You can also use the FactoryTalk View SE Client object model to customize the control system in the following ways:

- **Create custom alarm events.** You can write alarm detection algorithms using PLC logic, and then create events in FactoryTalk View SE to respond to the algorithms.
  - **Validate operator input.** Use VBA logic to validate the operator's input, for example, to ensure that the value an operator enters in a numeric input object falls within 10 percent of the value of another numeric input object.
  - **Create custom forms for operators.** You can use VBA to create custom forms, for example, pop-up dialog boxes that an operator can interact with at runtime.
- You can also use VBA logic to validate operator input, for example, to ensure that the value an operator enters in a numeric input object falls within the correct range of values.
- **Design intuitive graphic displays.** Use the FactoryTalk View SE Client object model to provide ActiveX controls with data, for use in graphic displays.
- For example, use list boxes or combo boxes in graphic displays to allow an operator to select options such as recipe items.
- **Manipulate the FactoryTalk View SE Client window.** Write VBA code to arrange graphic displays based on the size of the FactoryTalk View SE client window. This allows you to adapt an application dynamically to various desktop sizes and resolutions.
  - **Send custom messages to the Diagnostics log.** Send specific messages to the Diagnostics List and Diagnostics log, to record operation conditions and events through VBA code.

- **Access HMI tag database.** The FactoryTalk View SE Client object model allows you to manage HMI tags and tag values through VBA code.
- **Secure the system.** The FactoryTalk View SE Client object model allows you to obtain security information about who is using the system, and to use the security information and events to control access to the system.

For example, you can restrict a user's access to a graphic display by creating code that opens the graphic display only when the user is logged in with a particular security code.

# Setting up the FactoryTalk Directory

This chapter describes:

- What the FactoryTalk Directory is.
- Using the FactoryTalk Directory in a networked system.
- How to specify the location of the FactoryTalk Network Directory server.
- What happens if the Network Directory server is unavailable.

## About FactoryTalk Directory

FactoryTalk Directory centralizes access to application resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

Like a telephone directory or electronic address book, the FactoryTalk Directory provides a lookup service for the parts of an application, so that they can find each other on a single computer or over a network.

Application components can be stored in their original environments and made available to all clients, without the need for duplication.

FactoryTalk View Site Edition applications use two types of FactoryTalk Directory:

- FactoryTalk Local Directory (also called the Local Directory) manages applications that are confined to a single computer, for example, FactoryTalk View SE local station applications. The Local Directory must reside on the same computer as the local station application.
- FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed applications, and network station applications. All of the client and server computers participating in a given network distributed application, or network station application must point at the same Network Directory.

Both the Local and the Network Directory are installed with the FactoryTalk Services Platform. For more information, see the *FactoryTalk View Site Edition Installation Guide*.

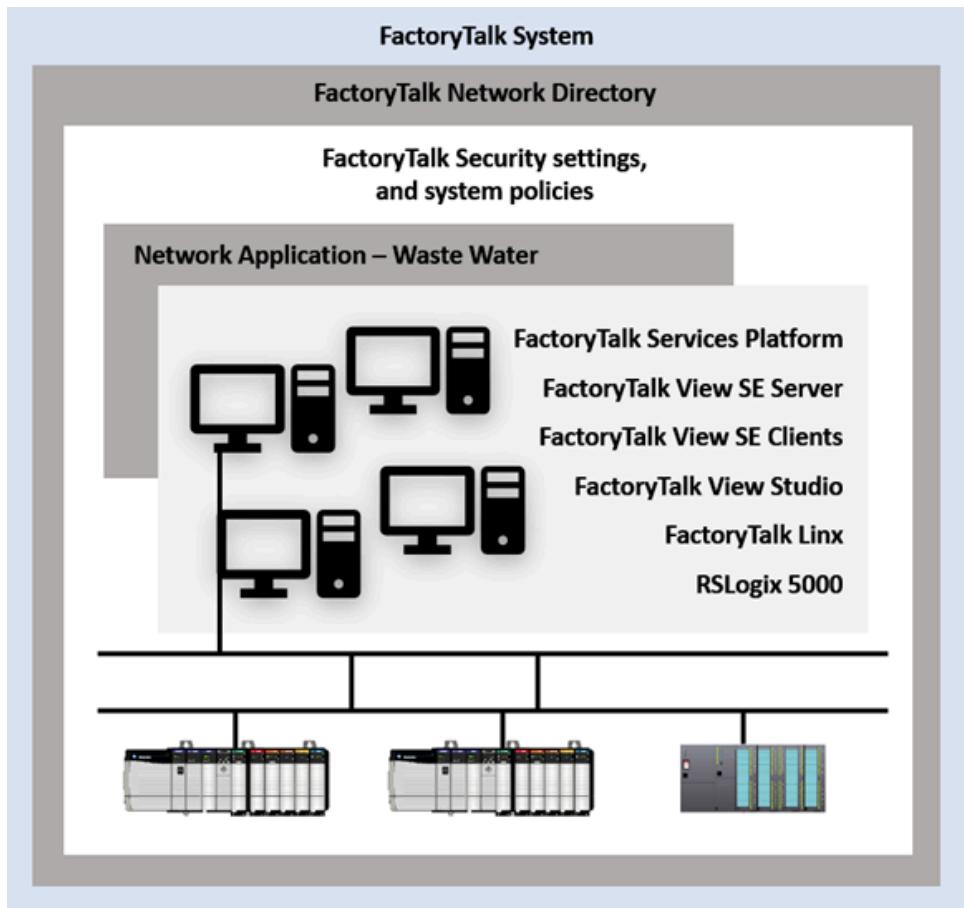
## FactoryTalk Directory in a networked system

FactoryTalk Directory is one of a set of FactoryTalk common services used by FactoryTalk View Site Edition and other Rockwell Automation software products.

An automation and control system that uses FactoryTalk services, and integrates FactoryTalk products and components, is known as a FactoryTalk system. A complete FactoryTalk system consists of all the networks, devices, and software applications you have deployed to monitor and control your plant or process. Helping to ensure that the system can provide data in a secure and predictable fashion depends on a number of variables.

For an overview of FactoryTalk services, see [About FactoryTalk systems on page 116](#).

The following illustration shows how a FactoryTalk View SE application might be deployed in a networked FactoryTalk system, along with the Network Directory, FactoryTalk Linx, and RSLogix software.



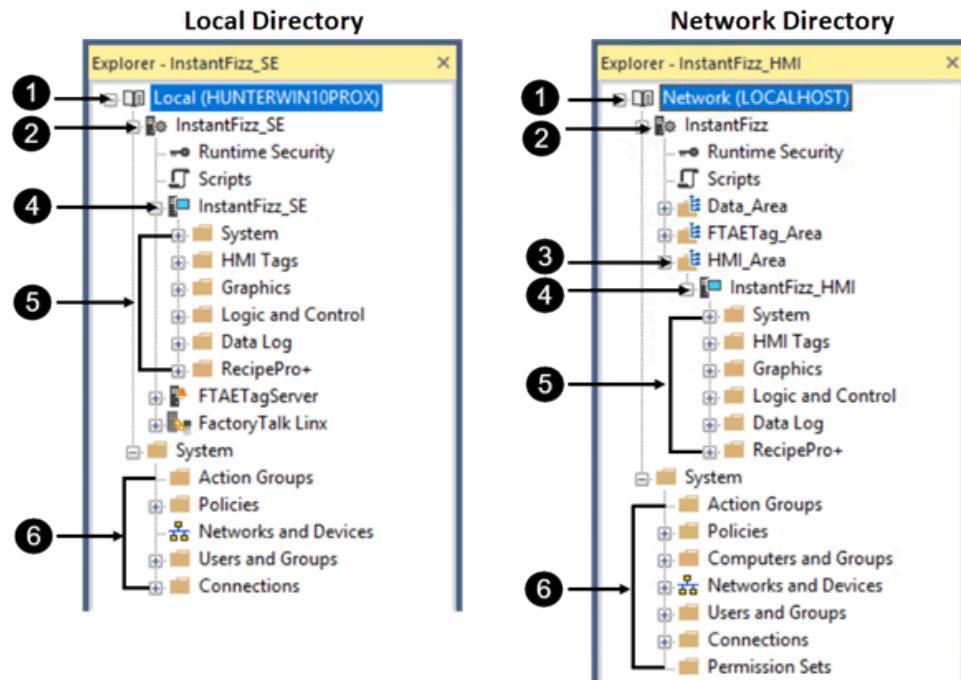
In the illustration, computers hosting FactoryTalk View SE Clients and FactoryTalk View Studio software use the Network Directory to find computers hosting FactoryTalk View SE Servers.

This lets clients gain access to various application services and components, such as FactoryTalk View graphic displays and data log models.

The FactoryTalk View SE Servers in the system use the Network Directory to find FactoryTalk Linx, to gain access to data on network devices (for example, PLC-5 and Studio 5000 devices), and to update the graphic displays.

## FactoryTalk Directory in a FactoryTalk View SE application

In FactoryTalk View Studio, when you create or open an application, the FactoryTalk Directory that manages the application is represented at the top of the Explorer tree, as shown in these illustrations:



Item	Description
1	Host computer
2	Application
3	Area
4	HMI server
5	HMI project components and editors
6	FactoryTalk system settings

For a local station application (shown on the left), the Explorer displays a Local icon and the name of the Local Directory's host computer. For a network distributed application or a network station application, the Explorer displays a Network icon and:

- **LOCALHOST**, if the Network Directory server is on the same computer as FactoryTalk View Studio.
- The host computer name, if the Network Directory is on a different computer.

## Local station applications

In a FactoryTalk View SE local station application, the Local Directory, FactoryTalk View SE Server, and FactoryTalk View SE Client all must be installed and run on the same computer.

If necessary, you can add one FactoryTalk Linx data server, or one FactoryTalk Tag Alarm and Event Server, which must be also installed locally. In a local station application, only OPC data servers can be installed on other computers.

For more information, see [Working with local station applications on page 142](#).

## Network applications

Network applications include:

- Network distributed applications  
For more information, see [Working with network distributed applications on page 114](#).
- Network station applications  
For more information, see [Working with network station applications on page 131](#).

A network application can be one of several managed by the same Network Directory, or it might be the only application using the directory server.



**Tip:** Each computer on the network can connect to only one computer running the Network Directory software. You cannot connect a single computer to multiple FactoryTalk Directory servers.

### Multiple applications, one Network Directory computer

A single Network Directory can manage multiple applications that represent different control systems, all hosted on the same network. This means you can run multiple applications managed by the same Network Directory simultaneously, each application controlling a different facility.

Application users can be members of different domains. If you are not using domains, all of the computers that connect to the network distributed application must be in the same Windows workgroup.



**Tip:** The collective limits of applications (including HMI servers, data servers, components, and so on) cannot exceed the system limits. For more information, see [System sizing recommendations on page 239](#).

### Multiple applications, multiple Network Directory computers

A network can also contain more than one computer running the Network Directory software. This is useful for running multiple network distributed applications that:

- Are separated by a slow network, for example, a wide-area network (WAN).  
To do this, set up one Network Directory at each site on either side of the WAN.
- Do not need to share data with each other.

## Specifying the FactoryTalk Directory server location

For FactoryTalk View SE network applications, you need to decide which computer will host the Network Directory server.

You can run the Network Directory server on the same computer as other FactoryTalk View software components, or you can run it alone, on a dedicated computer.

**NOTE:**

Do not install FactoryTalk Directory, FactoryTalk View SE Server, or any other application software on the same computer as the Windows domain controller. This configuration is not supported.

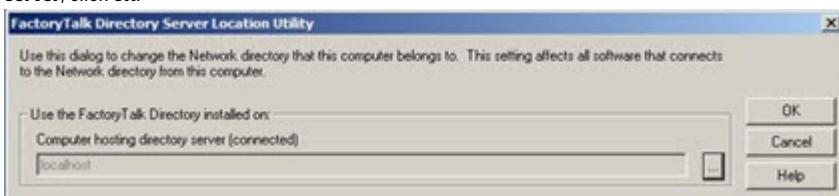
After installing FactoryTalk View SE, and before you run FactoryTalk View Studio, the FactoryTalk View SE Client, or the FactoryTalk View SE Administration Console, you must use the FactoryTalk Directory Server Location Utility to specify:

- **localhost** on the computer running the Network Directory server.
- The name of the computer running the Network Directory server, on every other computer that is to participate in the application.

For information about deploying all the parts of a network distributed or network station application, see the *FactoryTalk View Site Edition Installation Guide*.

### To specify localhost on the Network Directory server computer:

1. On the computer hosting the Network Directory server, select **Start > Rockwell Software > Specify FactoryTalk Directory Location**.
2. You are prompted to sign in. Type your name and password, and then click **OK**.  
To use the FactoryTalk Directory Server Location Utility, you must sign in as a user with administrative privileges at the Network Directory and in Windows, on the computer where the utility is running.
3. In the FactoryTalk Directory Server Location Utility, if **localhost** is shown under **Computer hosting directory server**, click **OK**.



If **localhost** is not shown, click the **Browse** button.

4. In the FactoryTalk Directory Server Configuration dialog box, click **This computer**, and then click **OK**.



5. Click **OK**.

## To specify the directory location on application computers:

1. In the notification area, click the **FactoryTalk Directory** icon.

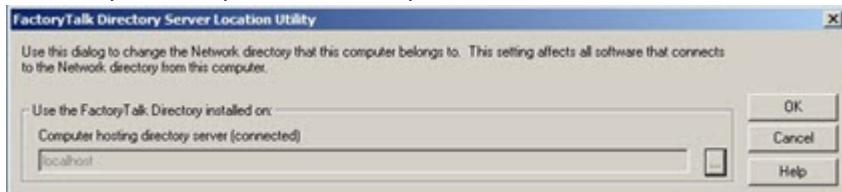


### Tip:

To use the FactoryTalk Directory Server Location Utility, you must sign in as a user with administrative privileges at the Network Directory and in Windows, on the computer where the utility is running.

2. The notification shows the connection state.

3. In the FactoryTalk Directory Server Location Utility, click the **Browse** button.



4. In the FactoryTalk Directory Server Configuration dialog box, click **Remote computer**, type the name of the Network Directory server, and then click **OK**.



To find and select the Network Directory server computer, click the **Browse** button.

5. Click **OK**.



**Tip:** To specify a remote computer as the Network Directory server, you must sign in as a user with administrative privileges at the Network Directory and in Windows, on the remote computer.

6. Repeat steps 1 to 5 on each computer that is to participate in the network distributed application.

## Setting up FactoryTalk Directory for local station applications

For FactoryTalk View SE local station applications, you do not need to do anything to set up the Local Directory, as the location is set to *localhost* automatically, during installation of the software.

For information about deploying all the parts of a local station application, see the *FactoryTalk View Site Edition Installation Guide*.

## What happens if the Network Directory server is unavailable

The Network Directory server cannot be made redundant. Instead, if the Network Directory becomes unavailable while client computers are connected to an application, the clients continue to run, using a local cache of directory information.

There is no need to restart previously connected clients; they will continue to resolve tag addresses, read and write tag values, acknowledge alarms, and open graphic displays, even if the tags and displays the clients require were never used before.

While the Network Directory is unavailable, you cannot modify the structure of any dependent application.

For example, you cannot add areas or servers to the application, create security accounts, or change system security policies.

When the Network Directory is available again, all dependent clients in the system will resume using the directory automatically.

## Monitoring the Network Directory status

FactoryTalk Directory centralizes access to application resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed applications, and network station applications. All of the client and server computers participating in a given network distributed application, or network station application must point at the same Network Directory.

In the **Log On to FactoryTalk** dialog box, you can view the current status of the active Network Directory server:

- **(Connected)** means all FactoryTalk products and components participating in a FactoryTalk system on the current computer are connected to and communicating with the Network Directory server computer.
- **(Read-only)** means FactoryTalk system participants on the current computer are disconnected from the Network Directory server and are retrieving information from a local cache.
- **(Unknown)** means that the connection status is temporarily unknown, for example, because the system is starting up and waiting to determine which server is active, or is unable to determine the current status.

### To check the status of the Network Directory server:

1. In the notification area, click the **FactoryTalk Directory** icon.
2. The notification shows the connection state.

# Setting up security

This chapter describes:

- What FactoryTalk Security services provide.
- Gaining initial access to a FactoryTalk system.
- Logging users on to and off from FactoryTalk View SE.
- Deciding how to secure a FactoryTalk View SE application.
- How to set up FactoryTalk accounts in FactoryTalk View SE.
- How to set up runtime security for HMI project components.
- Other ways to control runtime access to an application.
- Working with FactoryTalk Security accounts.
- Choosing the types of accounts to use.
- How to set up user and computer accounts.
- How to set up system-wide policies.
- Setting up security for FactoryTalk system resources.
- Understanding inherited permissions.
- Performing secured tasks in FactoryTalk View SE.

## About FactoryTalk Security services

For FactoryTalk products like FactoryTalk View SE, the FactoryTalk Directory stores information about which users are allowed access to the parts of a control system.

FactoryTalk Security uses this information to provide two basic services:

- **User authentication** verifies the user's identity, and whether a request for service actually originated with that user.
- **User authorization** verifies the user's request to access a software resource, based on the access rights and privileges defined for that user.

For example, when a FactoryTalk View SE network distributed application user signs in to FactoryTalk View Studio, FactoryTalk Security services verify the user's identity first.

If authentication succeeds, security services check permissions assigned to the user, to authorize actions performed on secured parts of the application.

In a network distributed application, security services also check whether the user is allowed to perform authorized actions on the current computer.

In addition, FactoryTalk Security services manage system-wide policies, such as how often users must change their passwords, or whether users can back up and restore applications. For more information about these policies, see [Setting up system-wide policies on page 100](#).

## About the FactoryTalk Directory

FactoryTalk Directory centralizes access to application resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

- FactoryTalk Local Directory (also called the Local Directory) manages applications that are confined to a single computer, for example, FactoryTalk View SE local station applications. The Local Directory must reside on the same computer as the local station application.
- FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed applications, and network station applications. All of the client and server computers participating in a given network distributed application, or network station application must point at the same Network Directory.

FactoryTalk Security settings are stored separately for a Local Directory and a Network Directory, even if both are in use on the same computer.

This means that you must set up security permissions twice—once for the Local Directory and once for the Network Directory—to give one user access to a local station and a network distributed application on the same computer.

For more information about the FactoryTalk Directory, see [Setting up the FactoryTalk Directory on page 70](#).

## Finding more information about FactoryTalk Security

For information about FactoryTalk Security from a FactoryTalk system perspective, and for detailed setup instructions, see the *FactoryTalk Security System Configuration Guide* or FactoryTalk Services Platform Help.

- In FactoryTalk View Studio, select **Help > Online Books > FactoryTalk Security System Configuration Guide**.
- In FactoryTalk View Studio, select **Help > Contents > FactoryTalk Services Platform Help**.

You can open FactoryTalk Services Platform Help by clicking **Help** in the dialog boxes used to set up security for FactoryTalk resources.

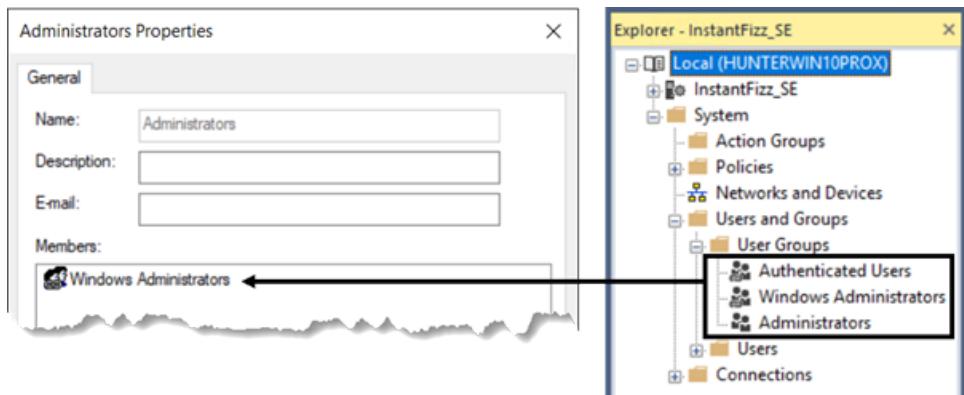
## Gaining initial access to a FactoryTalk system

When you install FactoryTalk View SE, during installation of the FactoryTalk Services Platform, the Local Directory and the Network Directory are set up on the computer.

As part of the directory configuration, these FactoryTalk user groups are created:

- **Windows Administrators** is a group linked to Windows users with administrative privileges on the computer.
- **Administrators** is a group for FactoryTalk users with administrative privileges.

The following illustration shows where to find these groups in FactoryTalk View Studio. The Windows Administrators group belongs to the FactoryTalk Administrators group. Both groups are created during FactoryTalk Directory.



By default, the Windows Administrators group belongs to the FactoryTalk system Administrators group.

This means that Windows users with administrative privileges on the local computer have full access to local and network applications, including the ability to set up security for the FactoryTalk system.

## About the All Users account

During FactoryTalk Services Platform installation, an account named All Users is created automatically at the Local Directory and the Network Directory.

When you create a local or a network distributed or network station application, an All Users account is also added automatically to the Runtime Security list for the application. By default, the account is allowed all runtime security codes.

All of this means that initially, after the FactoryTalk View SE software is installed, any FactoryTalk system user you create will be able to open, create, or modify an application in FactoryTalk View Studio, or run an application in the FactoryTalk View SE Client.

To retain full access for all users, nothing further is required.

To restrict access to the FactoryTalk system for selected users and computers, you must remove the All Users account, create FactoryTalk security accounts for the users and computers you plan to secure, and then give users the appropriate security permissions.

For more information, see [Deciding how to secure a FactoryTalk View SE application on page 83](#).



**Tip:** The All Users account is visible in the Runtime Security editor, but not in the User Groups folder that contains other FactoryTalk group accounts. For information about removing All Users from Runtime Security and from the FactoryTalk Directory, see [Removing All Users from the Runtime Security list on page 89](#) and [About the All Users account on page 79](#) respectively.

## Logging users on to and off from FactoryTalk View SE

After the FactoryTalk View SE software is installed, all users have full initial access to network and local applications on the computer.

There is no need to sign in, to run FactoryTalk View Studio, the FactoryTalk View SE Administration Console, or a FactoryTalk View SE Client. The current Windows user is automatically logged on to FactoryTalk View SE.

However, you do need to sign in and off to change users, or to gain access to *secured* parts of the FactoryTalk system.

The following sections describe how to sign in to and off from the FactoryTalk View SE software components, and the FactoryTalk Directory.

## Logging on to the FactoryTalk Directory

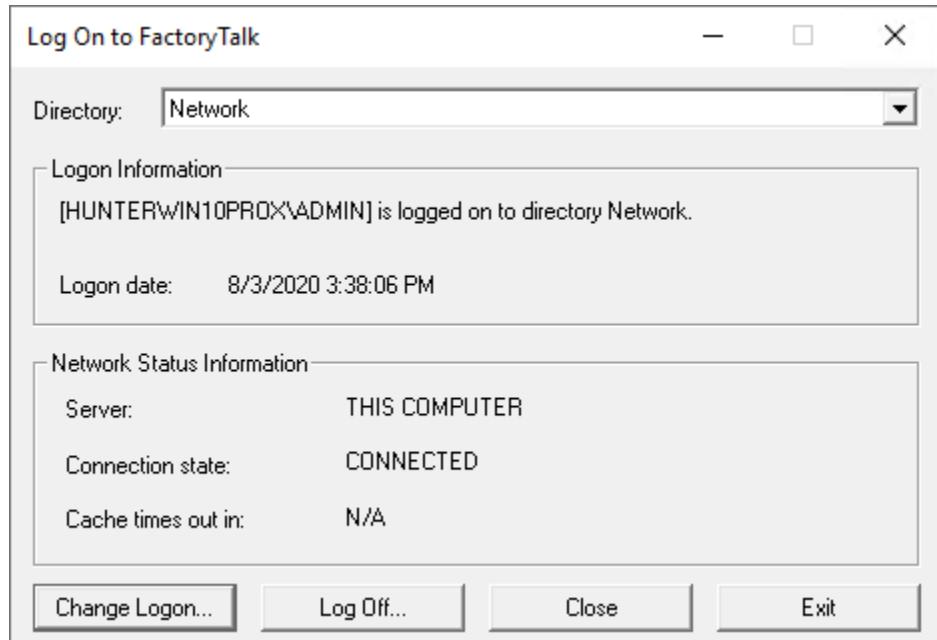
To sign in to the FactoryTalk Directory in the current Windows session, you can:

- Open or run a FactoryTalk View SE application in the FactoryTalk View SE Client, FactoryTalk View Studio, or the FactoryTalk View SE Administration Console.
- Use the Log On to FactoryTalk tool.

### To open the Log On to FactoryTalk tool:

Select **Start > Rockwell Software > Log On to FactoryTalk**.

For more information about logging on to the FactoryTalk Directory, see FactoryTalk Services Platform Help.



## About single sign-on

Single sign-on is a FactoryTalk Security policy that allows one user access to multiple FactoryTalk products without having to sign in to each product separately.

Single sign-on is set up as a system policy, and is enabled by default.

As long as single sign-on is enabled, and there is a user logged on to the FactoryTalk Directory, that user's credentials are checked first when a FactoryTalk product is started.

If FactoryTalk Security services authorize the single sign-on user, there are no further requests to sign in in the current Windows session.



**Tip:** An exception is the Specify FactoryTalk Directory Location tool. To use the tool, or to specify a remote computer as the Network Directory server, you must sign in as a user with administrative privileges. For details, see [Specifying the location of the FactoryTalk Directory server on page 73](#).

For more information about single sign-on and other FactoryTalk system-wide policies, see FactoryTalk Services Platform Help.

## Logging on to FactoryTalk View Studio

To sign out the current user in FactoryTalk View Studio, or in the FactoryTalk View SE Administration Console, from the **File** menu, select **Log Off <UserName>** (where *UserName* is the name of the current user).

When logging off, you must close the currently open application.

After logging off the current user, from the File menu, you can click **Log On** to sign in a different user.



**Tip:**

If single sign-on is enabled, changing the FactoryTalk View Studio user does not change the FactoryTalk Directory user. To change the current directory user, you must use the Log On to FactoryTalk tool. For details, see [Logging on to the FactoryTalk Directory on page 80](#).

## Logging on to a FactoryTalk View SE Client

Following installation of the FactoryTalk View SE software, the All Users account is automatically added to the Runtime Security list and allowed all runtime security codes.

This gives any FactoryTalk View SE Client user permission to run a client, open displays, write to tags, and execute commands and macros.

In a secured FactoryTalk system, you must remove the All Users account, add users to the Runtime Security list, and then give the users the security permissions needed to run an application. For more information, see [Removing All Users from the Runtime Security list on page 89](#), [Setting up accounts in the Runtime Security editor on page 86](#), and [Performing secured tasks in FactoryTalk View SE on page 110](#), respectively.

## Logging on when the client starts up

If single sign-on is enabled, when the FactoryTalk View SE Client starts up, the client will attempt to sign in the current FactoryTalk Directory user.

If there is no user logged on to the FactoryTalk Directory, the client will attempt to sign in the current Windows user if that user is set up as a FactoryTalk Windows-linked user.

If the Windows-linked user is not authorized to run the application, you will be asked whether another user would like to sign in. To open the FactoryTalk View SE Client Login dialog box (shown in the following illustration), click **Retry**.



To force all users to sign in when a FactoryTalk View SE Client starts up, you can disable single sign-on. For details, see FactoryTalk Services Platform Help.

## Changing the current user while the client is running

To allow users to sign in to a FactoryTalk View SE Client at runtime, provide them with a way to run the **Login** (or **Logout**) command.

For example, in a graphic display, include a button that has the **Login** command as the press action. When the current user clicks the button, the FactoryTalk View SE Client Login dialog box will open.



### Tip:

If single sign-on is enabled, changing the FactoryTalk View Studio user does not change the FactoryTalk Directory user. To change the current directory user, you must use the Log On to FactoryTalk tool. For details, see [Logging on to the FactoryTalk Directory on page 80](#).

## Logging on a different user

To sign in a different user, in the FactoryTalk View SE Client Login dialog box, type the user name and password of the user that wants to sign in, and then click **OK**.



**Tip:** FactoryTalk user accounts that have the same name, whether the accounts are in the current domain or from another domain, must sign in using the syntax **domain\user name**.

## Logging off the current user at runtime

To allow the current FactoryTalk View SE Client user to sign out from the client at runtime, provide a way to run the **Logout** command.

For example, in a graphic display, include a button that has the **Logout** command as the press action. When the current user logs off, the following events occur:

1. All displays are closed (including cached displays).
2. The current user's logout macro is executed, and the user is logged out.
3. If the FactoryTalk View SE Client is using an activation key, the key is released and made available for other clients to use.

4. The login dialog box is presented. A user cannot cancel or close the dialog box and it will continue to be presented until a valid user logs in. However, if this is not what you want, you can configure a Login command in the logout macro.



**Tip:** You can set up the FactoryTalk View SE Client to sign out the current user automatically, after a period of inactivity. For details, click **Help** in the FactoryTalk View SE Client Wizard.

## Changing the current user's password at runtime

To allow the current FactoryTalk View SE Client client user to change their password at runtime, provide them with a way to run the **Password** command.

For example, in a graphic display, include a button that has the **Password** command as the press action. When the current user clicks the button, the Change Password dialog box opens.

### To change a password:

In the Change Password dialog box, type the current password followed by the new password (twice), and then click **OK**.



**Tip:** Whether a FactoryTalk user needs to change passwords, and how often, is set up as a system security policy. For details, see FactoryTalk Services Platform Help.

## Deciding how to secure a FactoryTalk View SE application

As part of designing a complete FactoryTalk control system, consider how—and to what extent—you want to secure parts of the system.

You might decide that you only need to secure FactoryTalk View SE applications at runtime, or you might decide that you need to secure all the FactoryTalk system resources your application uses.

Make the decision based on what is appropriate for the application and its users. For example, consider:

- The roles that the different users, software, computers, and network devices are to play in the FactoryTalk View SE application, and in the entire FactoryTalk system.  
Keep in mind that security settings held at the FactoryTalk Network Directory apply to all FactoryTalk products participating in a single network application.
- The types of user groups you want to set up accounts for.  
Setting up group accounts is recommended to simplify management of multiple users with similar needs. For more information, see [Creating group accounts on page 97](#).
- Whether some user groups should have access to resources only from specific computers, or groups of computers.
- Which user groups should be able to create and modify application components.
- Which user groups should be able to set up security for the application. For example, decide who is allowed to create or modify user accounts, or set up system-wide security policies.
- Which system-wide security policies are appropriate for the entire control system.

For example, you might require users to change their passwords periodically, or force users to sign in every time they start FactoryTalk View Studio or a FactoryTalk View SE Client.

- Which user groups should be able to run applications in the FactoryTalk View SE Client and, in a network distributed application, which parts of the application users should have access to.
- Which HMI project components—FactoryTalk View commands and macros, graphic displays, OLE objects, or HMI tags—you need to secure at runtime.

The choices you make will determine the setup tasks you need to perform.

## Securing FactoryTalk View SE applications at runtime

FactoryTalk View SE manages runtime access to applications and to FactoryTalk View commands and macros, graphic displays, OLE objects, and HMI tags.

These are the tasks involved in setting up runtime security for a FactoryTalk View SE application:

- In the Runtime Security editor:
  - Add the FactoryTalk user and group accounts that you want to secure.
  - Assign FactoryTalk View security codes (A–P) to the accounts that you add.
  - Remove ALL USERS from the list of Runtime Security accounts.

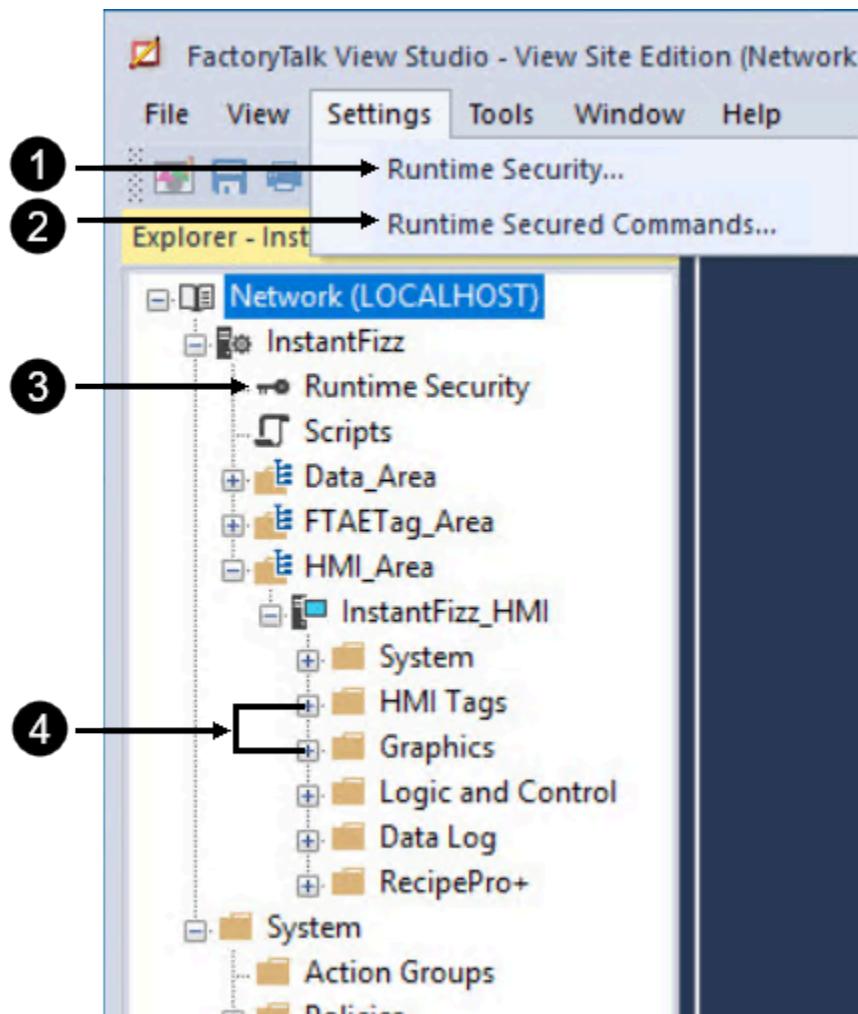
---

**NOTE:** To run an application in the FactoryTalk View SE Client, users and groups in the Runtime Security list must have at least one FactoryTalk View security code (A–P), in addition to the Common actions Read and List Children. For more information about FactoryTalk security permissions, see [Performing secured tasks in FactoryTalk View SE on page 110](#).

---

- In FactoryTalk View SE editors, assign FactoryTalk View security codes (A–P) to the HMI project components you want to secure.

You perform these tasks in FactoryTalk View Studio, as shown in the following illustration.



Item	Description
1	To set up runtime security accounts for an application, from the Settings menu, select <b>Runtime Security</b> .
2	To assign security codes to FactoryTalk View commands and macros, click <b>Runtime Secured Commands</b> .
3	You can also double-click the Runtime Security icon in the Explorer window.
4	In the Tags and Graphics editor, assign security codes to HMI tags, graphic displays, and OLE object verbs.

For details, see the following:

- [Setting up FactoryTalk accounts in FactoryTalk View on page 86 SE](#)
- [Setting up runtime security for HMI project components on page 91](#)
- [Other ways to control runtime access to an application on page 94](#)

## Securing FactoryTalk system resources

FactoryTalk Security services manage access to system resources such as the FactoryTalk Directory, the application itself, areas within the application, and participating users, computers, and devices.

These are the tasks involved in securing access to these FactoryTalk system resources:

- Remove ALL USERS from the FactoryTalk Local or Network Directory.
- Create FactoryTalk accounts for the users, groups, and computers you want to secure.
- Assign security permissions to FactoryTalk user and group accounts.
- Set up system-wide security and product policies.

For more information, see the following:

- [About FactoryTalk Security accounts on page 95](#)
- [Setting up user and computer accounts on page 98](#)
- [Setting up system-wide policies on page 100](#)
- [Setting up security for FactoryTalk system resources on page 104](#)
- [Understanding inherited permissions on page 108](#)
- [Performing secured tasks in FactoryTalk View on page 110 SE](#)

For comprehensive information and setup instructions, see FactoryTalk Services Platform Help.

## Setting up FactoryTalk accounts in FactoryTalk View SE

To secure a FactoryTalk user's access to FactoryTalk View SE application components at runtime, you must set up runtime security for the user by:

- Adding the user's FactoryTalk account to the Runtime Security list.
- Giving the account at least one FactoryTalk View security code (A-P).

Optionally, you can also specify login and logout macros for Runtime Security accounts. For more information, see [Specifying login and logout macros on page 89](#).

After setting up Runtime Security accounts, you can set up the HMI project components you want to secure. For more information, see [Setting up runtime security for HMI project components on page 91](#).

## Setting up accounts in the Runtime Security editor

When you add a user or group account to the Runtime Security list, you are creating a reference only to an existing FactoryTalk security account.

If you want to add a user that does not exist, you must create the FactoryTalk account first, and then add the account in the Runtime Security editor.

For information about creating FactoryTalk accounts, see [Setting up user and computer account on page 98](#).

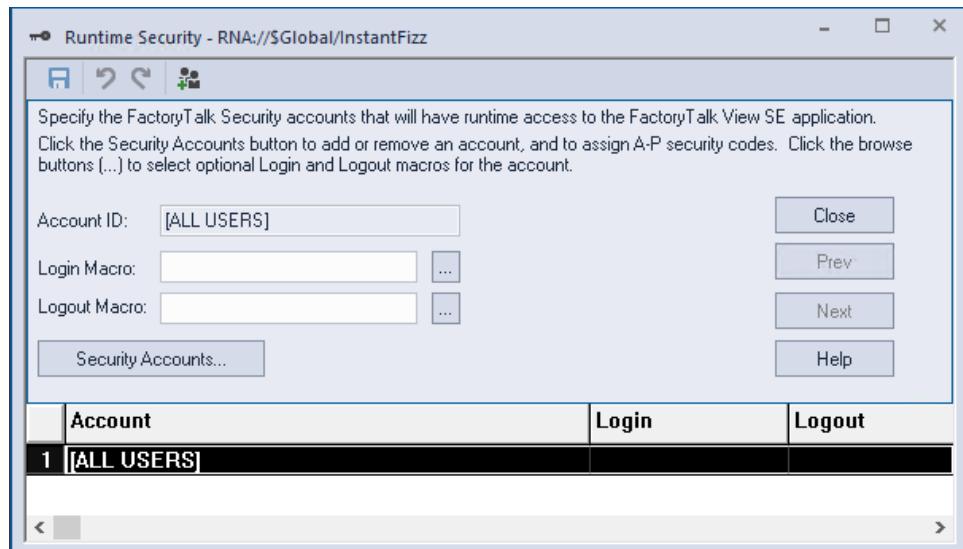
If you are running multiple applications managed by the same Network Directory, you have to add FactoryTalk accounts to each application separately.

In a network distributed application containing multiple areas, all of the areas share the same Runtime Security accounts list.

## To open the Runtime Security editor:

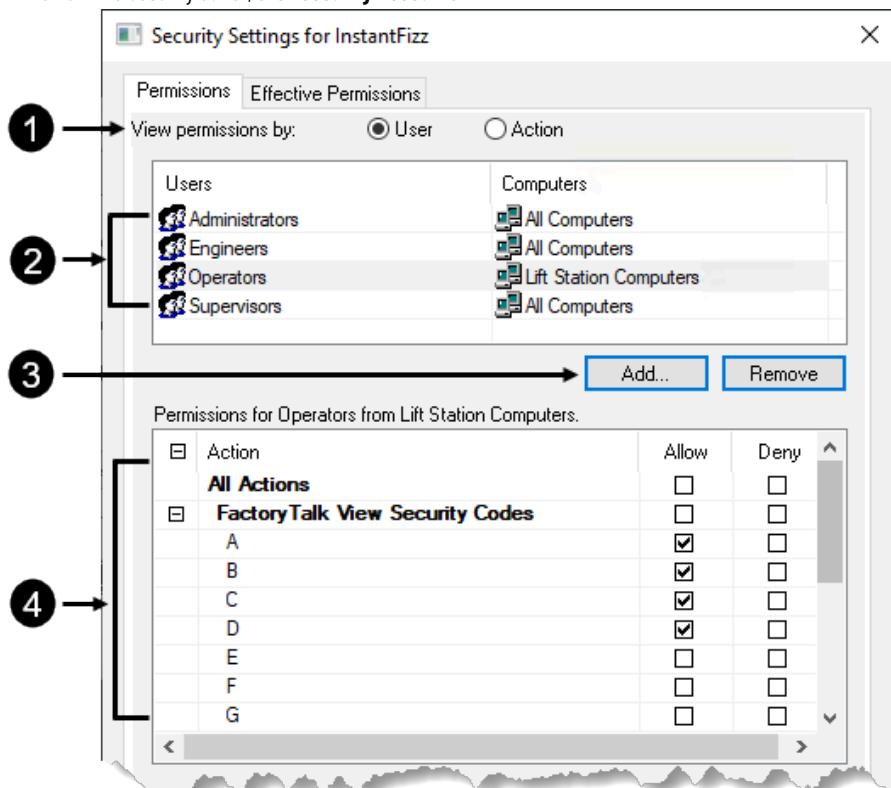
In FactoryTalk View Studio, from the Settings menu, select **Runtime Security**. You can also double-click the Runtime Security icon in the Explorer window.

To add users, or to view or modify the security codes for the selected account, click **Security Accounts**.



## To add users and assign security codes to them:

1. In the Runtime Security editor, click **Security Accounts**.



Items	Description
1	Set up security codes by user or action.

Items	Description
2	List of user groups with levels of access to securable HMI project components.
3	To add FactoryTalk users or groups, click <b>Add</b> .
4	To assign security codes to the selected user or groups, select the <b>Allow</b> checkboxes.

2. In the Security Settings dialog box, in the **Permissions** tab, select the **User** option.  
If you prefer to set up permissions by assigning user accounts to the selected security code, select **Action**.
3. Click **Add**, select the user or group account to add, and then click **OK**.



**Tip:** If you are setting up security for a network application, you must select a computer account with the user account, before you can click **OK**.

4. In the Runtime Security list, select the account you just added.
5. Select the Allow checkbox beside the FactoryTalk View Security Codes that you want to explicitly allow for the selected account.  
To select all of codes A to P, select the Allow checkbox beside All Actions, or the checkbox beside the FactoryTalk View Security Codes heading.
6. Select the Deny checkbox beside the FactoryTalk View Security Codes that you want to explicitly deny for the selected account.



**Tip:** You can also deny a security code by clearing the Allow checkbox. If you do this, keep in mind that an explicit Deny always takes precedence, even over an explicit Allow. This is important if the account you are setting up security for belongs to more than one group.

For example, if John Doe belongs to one group that allows codes ABC, and to another group that explicitly denies code B, then John Doe will only allow codes A and C. For more information, see [Understanding inherited permissions on page 108](#).

7. Repeat steps 3 to 6, for each user or group account you want to set up with Runtime Security.

## Example: Setting up runtime access to HMI components

Only users assigned the necessary FactoryTalk View security codes can run secured commands, open secured graphic displays, or write to secured HMI tags at runtime.

In this example, three user groups are added to the Runtime Security accounts list, and given the following security codes:

- The Supervisors group is allowed security codes A through P.
- The Operators group is allowed security codes B, C, and E.
- The Visitors group is allowed Security B.

Then, the following HMI project components are given security codes:

- A graphic display named Overview has security code B.
- A graphic display named Boiler has security code C.

- The **DisplayClientClose** command has security code D.
- An HMI tag named BoilerTemp has security code E.

This means that members of:

- The Supervisors group have full runtime access.
- The Operators group can open the Overview and Boiler graphic displays, and can write to the BoilerTemp tag. Members cannot run the **DisplayClientClose** command.
- The Visitors group can only open the Overview graphic display.

If a user belongs to more than one group, the security codes allowed or denied each group are combined for that user.

For example, if John Doe belongs to one group that is allowed codes A and B, and to another group that is allowed C and D, then he is allowed A, B, C, and D.

Or, if John belongs to one group that allows A, B, C, and D, and to another that explicitly denies B, then John is allowed only A, C, and D.

## Removing All Users from the Runtime Security list

Following installation of the FactoryTalk View SE software, all FactoryTalk users have full runtime access to network and local applications on the computer.

The All Users account is added automatically to the Runtime Security list, for each network or local application you create. To set up runtime security for individual user or group accounts, you must remove the All Users account.



**Tip:** You cannot remove the All Users account until you add at least one other account to the Runtime Security list.

Removing All Users from the Runtime Security list does not delete the corresponding account from the FactoryTalk Directory.

This means that parts of the control system will remain unsecured for all users, even after the account is removed. For example, all users will still be able to create and modify applications in FactoryTalk View Studio.

To restrict access to FactoryTalk system resources, you must also remove All Users from the FactoryTalk Directory. For details, see [About the All Users account on page 79](#).

### To remove the All Users account from the Runtime Security list:

1. In FactoryTalk View Studio, in the Explorer window, right-click the Runtime Security icon, and then click **Open**.
2. In the Runtime Security editor's accounts list, select **Security Accounts**.
3. In the Security Settings dialog box, select the ALL USERS account, and then click **Remove**.

## Specifying login and logout macros

You can assign a login and logout macro to any user or group account listed in the Runtime Security editor. The macros will run when the user (or users) sign in to or out of the application at runtime.

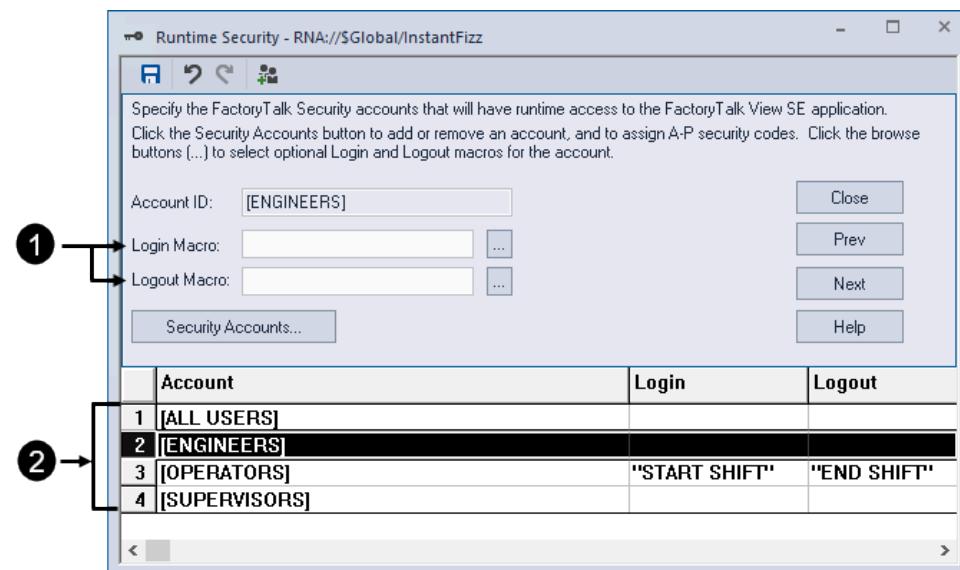
Any macro component containing any FactoryTalk View command can be a login or logout macro. For example, a login macro might contain a command to open a specific graphic display, and a logout macro might contain commands to redefine sensitive keys.

In a network distributed application, a login or logout macro will only run in the area where it's located. For example, if you specify a login macro in the Bottling area, when the user signs in to a FactoryTalk View SE Client, the macro will run only if the Bottling area is the client's home area.



**Tip:** In a network distributed application, when browsing the login or logout macro, only the macros from the first created HMI server are shown. To use macros from other HMI servers, enter the commands contained within them directly in the **Login Macro** or **Logout Macro** box.

You specify the client's home area when you select components for the client configuration file. For more information, click **Help** in the FactoryTalk View SE Client wizard.



Items	Description
1	Type a macro name, or click the Browser button to find and select a macro for the selected account.
2	In the accounts list, square brackets [ ] indicate a user group.

For more information about logging on and off at runtime, see [Logging users on to and off from FactoryTalk View SE on page 79](#).

For information about creating macros, see [Adding logic and control on page 423](#).

### To assign a login or logout macro to a user account:

1. In the Runtime Security editor, click the name of the user or group account you want to set up with a login or logout macro.
2. Enter a Login or Logout Macro name, and then click **Accept**.

To find and select a macro in the **Command Wizard**, click the **Browse** button.

- 
3. Repeat steps 1 and 2 for each account that you want to have a macro, and then click **Close**.
- 



**Tip:** If the macro's name contains spaces, you must enclose the name in quotes.

---

## Removing Runtime Security accounts

Removing a user or group account from the Runtime Security list does not delete the corresponding FactoryTalk account from the Network or Local Directory.

However, if you delete a user or group from the FactoryTalk Directory, and the account exists in the Runtime Security list, it will be removed automatically the next time you save changes in the Runtime Security editor.

You can also remove Runtime Security accounts manually.

### To remove accounts from the Runtime Security list:

1. In the Runtime Security editor, click **Security Accounts**.
2. In the Security Settings dialog box, select the user or group account you want to remove, and then click **Remove**.

## Setting up runtime security for HMI project components

FactoryTalk View SE manages runtime access to FactoryTalk View commands and macros, graphic displays, OLE objects, and HMI tags.

Set up security for these HMI project components in FactoryTalk View Studio in their respective editors.

## Assigning security codes to commands and macros

In the **Runtime Secured Commands** editor, you can create a list of the commands and macros to be secured, and then assign a security code to each.

Only users or groups assigned the same security code as a particular command or macro will be able to run the command or macro at runtime.

---

**NOTE:** Do not restrict access to the **Login** command.

---

### To assign security to commands and macros:

1. In FactoryTalk View Studio, in the **Explorer** window, double-click **Settings > Runtime Secured Commands**.
2. In the **Runtime Secured Commands** editor, in the **Secured Commands** dialog box, click **Next** until the first empty line in the spreadsheet is highlighted and the **Command** field is cleared.
3. In the **Command** field, type the name of the macro or command you are setting up security for, or click the **browse button (...)** to open the **Command** wizard and select one.
4. Type a description to provide further information about the macro or command.
5. Beside the **Security Code** field, click the down arrow to display the list of codes, then click one to select it.
6. Click **Accept** to add the command and its information into the Secured Commands spreadsheet.
7. Repeat steps 2 through 6 to continue setting up security codes for commands. When you are finished, click **Save** to save the changes to the secured commands list and exit from the **Secured Commands** editor.

For details about options in the **Runtime Secured Commands** editor, click **Help**.

## About the Unspecified\_Command

In the **Runtime Secured Commands** editor, the first record in the **Commands** list is called *Unspecified\_Command*.

Any commands or macros that are not listed as secured commands will use the security code selected for the *Unspecified\_Command*.

Initially, the security code is the asterisk (\*), which means that any user that has at least one security code will have access to any unspecified commands or macros.

If you leave the security code for the *Unspecified\_Command* as the asterisk, you have to list all the FactoryTalk View commands and macros you want to secure, in the **Runtime Secured Commands** editor.

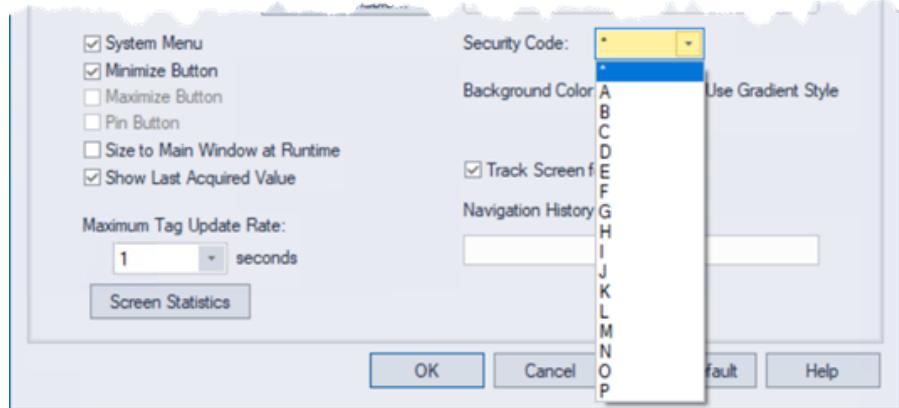
## Assigning security codes to graphic displays

In the Graphics editor, you can set up security for graphic displays. You can assign a security code while you are creating a graphic display, or you can assign it later.

Only users or groups assigned the graphic display's security code will be able to open the display at runtime. Users must have at least one security code assigned to them, to open graphic displays assigned an asterisk (\*).

### To assign security to a graphic display:

1. In FactoryTalk View Studio, in the Explorer window, double-click the graphic display that you want to secure.
2. Right-click in the display, and then select **Display Settings**.
3. In the **Properties** tab of the Display Settings dialog box, in the **Security Code** list, select a security code other than the asterisk (\*), and then click **OK**.



## Assigning security codes to OLE objects

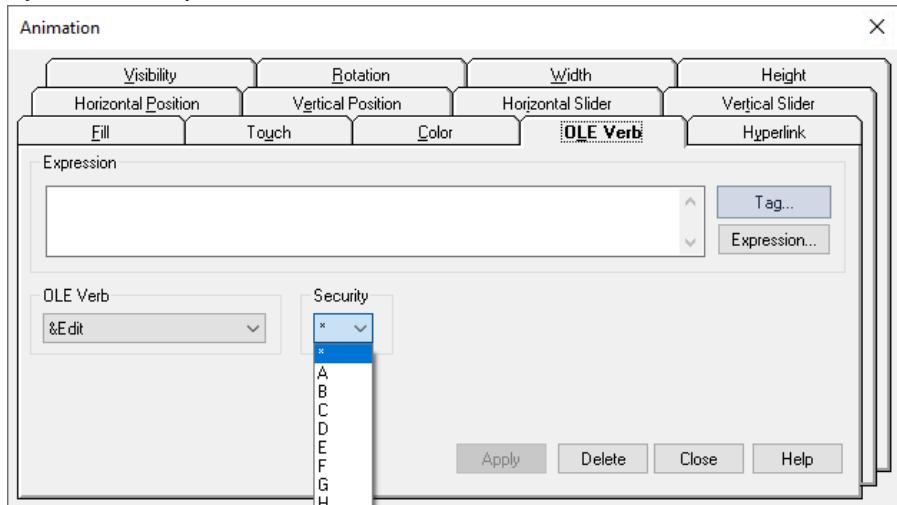
In the Graphics editor, you can set up security for a graphic object with OLE verb animation.

Only users or groups assigned the OLE verb's security code will be able to activate or modify the OLE object at runtime. Users must have at least one security code assigned to activate OLE objects assigned an asterisk (\*).

**NOTE:** Once an OLE object is activated, there is no way to control what an operator can do within the associated program. Therefore, the only way you can secure the program is by assigning security to the OLE object verb in FactoryTalk View.

### To assign security to an OLE object:

1. In FactoryTalk View Studio, in the Explorer window, double-click the graphic display containing the OLE object that you want to secure.
2. Right-click the OLE object, select Animation > OLE Verb.



3. In the OLE Verb tab, select the verb that you want to secure.
4. In the Security list, select a security code other than the asterisk (\*).
5. Click Apply.

## Assigning security codes to HMI tags

In the Tags editor, you can set up security for an HMI tag, to prevent users from writing to the tag.

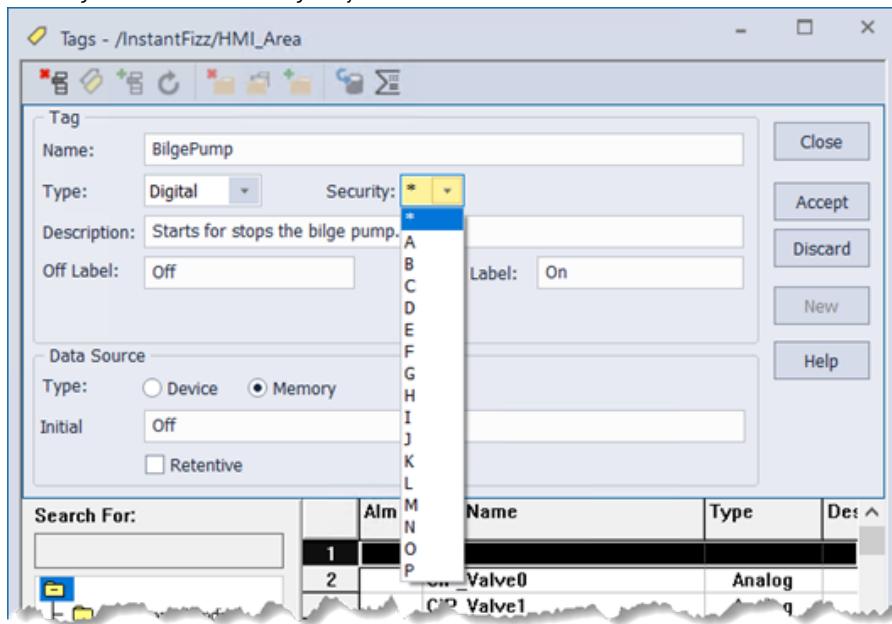
Only users or groups assigned the tag's security code, will be able to change the tag's value at runtime. Users must have at least one security code assigned to change the value of tags assigned an asterisk (\*).



**Tip:** You cannot assign a security code to data server tags. To prevent users from changing the value of a data server tag, map its address to an HMI tag, and then secure the HMI tag.

## To assign security to an HMI tag:

1. In FactoryTalk View Studio, in the Explorer window, right-click the Tags icon, and then click **Open**.
2. In the Tags editor, select the HMI tag that you want to secure.



3. In the Security list, select a security code other than the asterisk (\*).
4. Click **Accept**.

## Other ways to control runtime access to an application

To further restrict access to a FactoryTalk View SE application at runtime, you can prevent users from leaving the FactoryTalk View SE Client environment.

You can also use the FactoryTalk View signature button in an application, to control selected user actions.

## Lock operators into the runtime environment

To lock operators into the runtime environment, you can do one or more of the following:

- **Limit the ability to manipulate graphic displays**, by removing the title bar or minimize and maximize buttons from selected displays.  
To do this, in the Display Settings dialog box, clear the checkboxes of Title Bar, Minimize Button, and Maximize Button. For details, see [Creating graphic displays on page 259](#).
- **Limit the ability to manipulate the client window** by removing the title bar, or minimize and maximize buttons from the client.  
To do this, in the FactoryTalk View SE Client wizard, clear the checkboxes of Show title bar, and Show system menu and close button. For details, click **Help** in the FactoryTalk View SE Client wizard.
- **Prevent switching to other applications**. To do this, in the FactoryTalk View SE Client wizard, select the checkbox, Disable switch to other applications. For details, click **Help** in the FactoryTalk View SE Client wizard.
- **Restrict access to the desktop**, using the Desklock tool.  
To open Desklock, select **Start > Rockwell Software > DeskLock**. For details about using DeskLock, click **Help** within the tool.

## Use electronic signature to control user actions

To further secure commands, graphic objects, and tags, use built-in signature functions on the graphic objects or use the signature button.

With electronic signature, you can control operator actions at runtime, such as:

- Setting the value of a tag.
- Running a command.
- Downloading values to programmable controllers or devices.

For more information about the signature button, see [Creating graphic objects on page 288](#).

## About FactoryTalk Security accounts

Every user and group of users requiring access to any secured part of a FactoryTalk View SE application must have an account set up at the Network Directory or Local Directory that manages the application.

For a network distributed application, you can also set up accounts for computers, or groups of computers, in the application.

Since user and computer accounts are stored at the FactoryTalk Directory, the accounts you create for a FactoryTalk View SE application are available to any FactoryTalk product using the same directory.

FactoryTalk Security settings are stored separately for a Local Directory and a Network Directory, even if both are in use on the same computer.

This means that you must set up security permissions twice—once for the Local Directory and once for the Network Directory—to give one user access to a local station and a network distributed application on the same computer.

## Remove the All Users account

During FactoryTalk Services Platform installation, an account named All Users is created automatically at the Local Directory and the Network Directory.

This means that initially, after the FactoryTalk View SE software is installed, any FactoryTalk system user you create will be able to open, create or modify an application in FactoryTalk View Studio.

To restrict access to FactoryTalk system resources for selected users and computers, you must remove the All Users account, create accounts for the users and computers you want to secure, and then give the accounts the appropriate security permissions.

Removing All Users from the FactoryTalk Directory also removes the account at all levels of the directory hierarchy, including any application the directory manages, and the directory System folder.

This means that all users will no longer have full access to all parts of the control system.

---

**NOTE:** Before removing the All Users account from FactoryTalk Directory, ensure that you have created at least one other account that has permission to set up security for the directory. For more information, see [Specifying which users can set up security on page 96](#) next.

---

## To remove the All Users account from the FactoryTalk Directory:

1. In FactoryTalk View Studio, in the Explorer window, right-click the directory node, and then select **Security**.
2. In the accounts list, select ALL USERS, and then click **Security Accounts**.
3. In the Security Settings dialog box, select the ALL USERS account, click **Remove**, and then click **OK**.



**Tip:** If you remove All Users from the FactoryTalk Directory, the corresponding account in the Runtime Security editor will also be removed. To avoid unexpected results before removing the All Users account, ensure that you have set up the necessary runtime accounts.

## Specifying which users can set up security

When setting up security, one of the first things you should do is ensure that only a group of authorized, trusted users is able to change security settings at the FactoryTalk Directory.

You can do this by allowing only members of the Administrators group permission to perform the Configure Security action on the FactoryTalk Directory.

The Administrators group is created automatically, for both the FactoryTalk Network Directory and Local Directory, during FactoryTalk Services Platform installation. To give individual users administrative permissions, add them to the Administrators group.



**Tip:** A Windows Administrators group is also created and added automatically to the FactoryTalk Administrators group. This means Windows administrators on the computer can also set up security. For more information, see [Gaining initial access to a FactoryTalk system on page 78](#).

For information about setting up security permissions, see [Setting up system-wide policies on page 100](#). For details about adding users to a group, see FactoryTalk Services Platform Help.

## Choosing the types of accounts to use

In FactoryTalk View Studio, you can create user and user group accounts that:

- Originate at the FactoryTalk Directory.
- Refer to Windows users and groups. These are called Windows-linked user accounts.

You can modify the properties of the accounts you create. You can also delete, disable, or reactivate them. For details, see FactoryTalk Services Platform Help.

## Using accounts that originate at the FactoryTalk Directory

When a user that originates at FactoryTalk Directory attempts to access system resources, FactoryTalk Directory determines whether the user's name and password are valid, and whether the account is active or locked out.

Use this type of user or group account when you want to centralize security administration without relying on a Windows domain, or when the security needs of the Windows network differ from those of the control network.

For example, you might consider FactoryTalk accounts when:

- Using a Windows workgroup, instead of a Windows domain.
- All operators share the same Windows account to gain access to a computer.
- The computer is always logged on under a particular Windows account. In this case, separate accounts allow different operators to gain different levels of access to the control system, independent of their access to Windows.
- The computer automatically signs in to the Windows network after restarting, so it can run control programs automatically.

For more information, see FactoryTalk Services Platform Help.

## Using Windows-linked accounts

When a Windows-linked user attempts to access system resources, FactoryTalk Directory relies on Windows to determine whether the user's name and password are valid, and whether the account is active or locked out.

Use this type of user or group account when the security needs of the Windows network match those of the control system. For example, you might consider Windows-linked accounts when:

- The control system is in its own domain, perhaps separate from business systems, and user accounts and passwords can be shared between Windows and FactoryTalk software programs.
- Operators can sign in to and out of computers with their own Windows accounts, and the software programs they use start automatically.

For more information, see FactoryTalk Security Help.

## Using both types of user account

If you decide to use both types of FactoryTalk Security user account in an application, remember that Windows-linked accounts only refer to accounts that already exist at the Windows domain.

This means that you cannot convert an original FactoryTalk account to a Windows-linked account, nor can you add a FactoryTalk account to a Windows-linked user group.

## Creating group accounts

Whichever type of FactoryTalk Security account that you decide to use, when setting up security for a FactoryTalk View SE application, it is recommended that you create group accounts and set up access permissions for them first.

Using group accounts makes it easier to assign and manage permissions for multiple users with the same security needs. Rather than assigning permissions to each user separately, you can create accounts for new users, and then add these users to the appropriate groups.

Later, if you want to change an individual user's permissions, you can move the user's account to a different group. You can also change permissions for several individual users at once, by modifying the group that the users belong to.

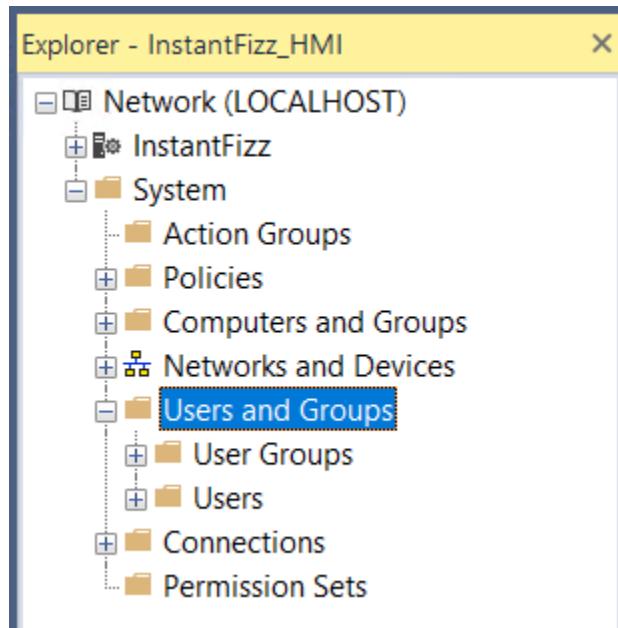


**Tip:** You can add Windows-linked groups to FactoryTalk groups. The Windows Administrators group is added automatically to the FactoryTalk Administrators group during FactoryTalk Directory installation. For more information, see FactoryTalk Services Platform Help.

## Setting up user and computer accounts

In FactoryTalk View Studio, create user accounts in the Users and Groups subfolder of the System folder.

The following illustration shows where to find this folder in the Explorer window.



### To create a user group:

- In the Users and Groups folder, right-click the User Groups folder, select **New > User Group**.  
For details about options in the New User Group dialog box, click **Help**.

### To create a user:

- In the Users and Groups folder, right-click the Users folder, select **New > User**.  
For details about options in the New User dialog box, click **Help**.

### To create a Windows-linked group:

- In the Users and Groups folder, right-click the User Groups folder, select **New > Windows-Linked User**.  
For details about options in the New Windows-Linked User dialog box, click **Help**.

### To create a Windows-linked user:

- In the Users and Groups folder, right-click the Users folder, select **New > Windows-Linked User**.  
For details about options in the New Windows-Linked User dialog box, click **Help**.



**Tip:** Windows-linked accounts refer to existing Windows accounts. If you want to create a Windows user or group, you must do so in Windows. For details, see Windows help.

## Creating computer accounts for network applications

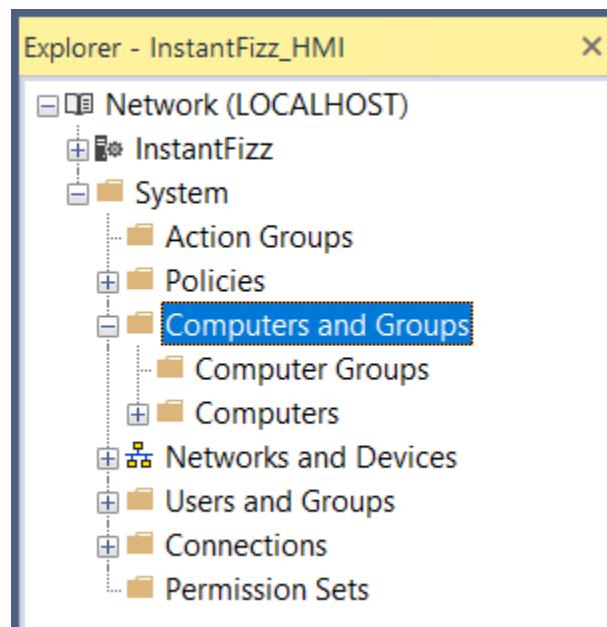
For FactoryTalk View SE network applications, in addition to specifying which users have access to system resources, you can specify the computers where they have access.

You do this by creating computer accounts, and then associating the user accounts with a computer or group of computers, when you set up security permissions.

If you do not want to associate user accounts with specific computers, you can use the All Computers account when setting up permissions. This account is created automatically at the Network Directory.

In FactoryTalk View Studio, create computer accounts in the System folder, in the Computers and Groups folder.

The following illustration shows where to find this folder in the Explorer window.



**Tip:** The Computers and Groups folder does not exist for a local station application. You cannot create computer accounts for applications that are confined to a single computer.

### To create a computer:

- In the Computers and Groups folder, right-click the Computers folder, and then select **New Computer**.  
For details about options in the New Computer dialog box, click **Help**.

### To create a computer group:

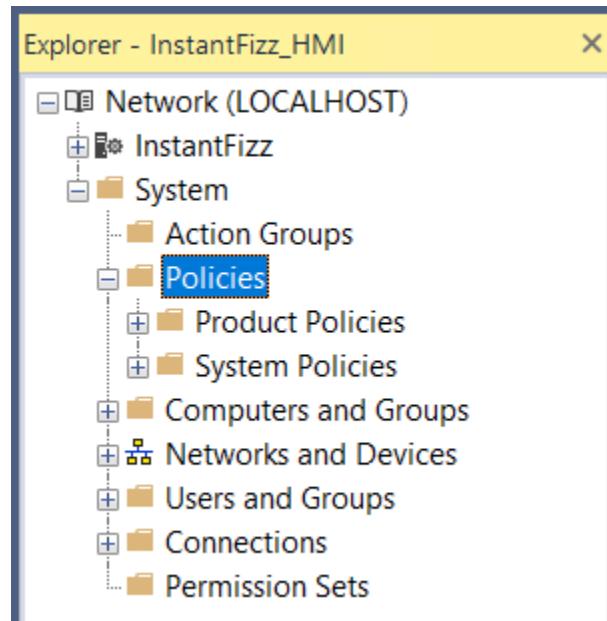
- In the Computers and Groups folder, right-click the Computer Groups folder, and then select **New Computer Group**.  
For details about options in the New Computer Group dialog box, click **Help**.

## Setting up system-wide policies

In FactoryTalk View Studio, you can set up product and system policies that determine the general characteristics of the system.

These specifications are stored at the FactoryTalk Directory and, like all settings in the **System** folder, apply to all FactoryTalk View products managed by a single Local Directory or Network Directory.

The following illustration shows you where to find the **Policies** folder in the Explorer window.



### Product policies

Product policies are sets of features that you can secure for the individual products in a FactoryTalk system.

In FactoryTalk View SE, you can set up product policies for using the HTTPS protocol or custom websites under Internet Information Services. For more information about the custom website, click **Help** in the FactoryTalk View SE Custom Website Setup tool.

For more information about product policies, see FactoryTalk Services Platform Help.

### Using HTTPS protocol

For secure communication, the HTTPS protocol is recommended.

If the HTTPS protocol is enabled, all FactoryTalk View SE client and server components will use HTTPS for communication with the default port number 443.

If you select to use trusted certificates only, make sure that:

- The certificate is issued to the computer name.
- The certificate is trusted on each SE client and server computers.

**NOTE:** Computers of a redundant server pair must trust each other's certificate.

Add the certificate to the local machine certificate store, instead of the current user certificate store. Otherwise, some features such as redundancy may not work correctly. For more information about certificate stores, see Microsoft Docs at <http://docs.microsoft.com/en-us/windows-hardware/drivers/install/local-machine-and-current-user-certificate-stores>.

- The webpage <https://computername/RSViewSE> can be correctly opened, where *computername* is the name of the SE server computer.

For more information about managing trusted certificates, see Microsoft TechNet at [http://technet.microsoft.com/en-us/library/cc754841\(v=ws.11\).aspx](http://technet.microsoft.com/en-us/library/cc754841(v=ws.11).aspx).

Before using HTTPS, on the SE server and SE Studio computers, make sure to configure the HTTPS binding for your website in IIS.



**Tip:** If you change to enable or disable the HTTPS policy, restart all server and client computers to take effect.

The HTTPS Protocol policy takes precedence over the Custom Website policy. If you enable HTTPS, we recommend that you disable the Custom Website policy.

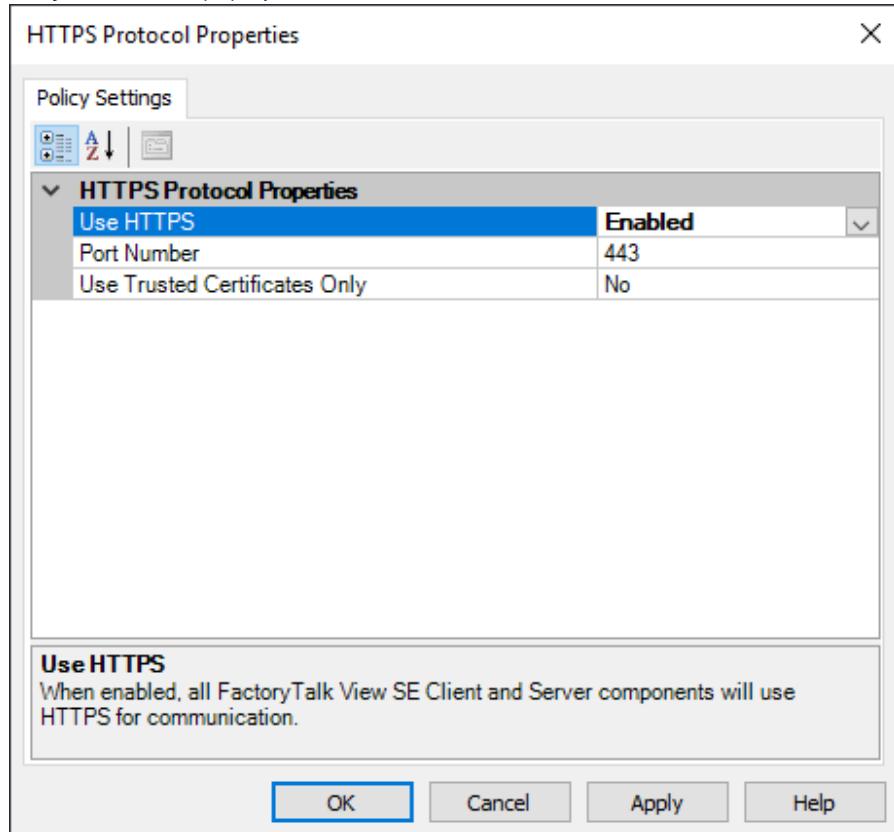
You can enable the HTTPS protocol using the FactoryTalk View SE Website Setup tool or changing the Product Policies for HTTPS in FactoryTalk Services Platform.

### To enable the HTTPS protocol with the Website Setup tool

1. From any computer connected to the FactoryTalk Directory, open the FactoryTalk View SE Website Setup tool, and then in **HTTPS protocol**, click **Enable**.
2. (optional) To use trusted certificate only, select the **Use trusted certificate only** checkbox.
3. Restart all computers connected to that FactoryTalk Directory.

## To enable the HTTPS protocol with FactoryTalk Product Policies

1. From any computer connected to the FactoryTalk Directory, open the FactoryTalk Administration Console.
2. Select **System > Policies > Product Policies > FactoryTalk View SE > HTTPS Protocol**.
3. Change the **Use HTTPS** property to **Enabled**.



After you've enabled the HTTPS protocol, you can enable the **RSViewSE** site's **Require SSL** setting, deactivating the HTTP protocol (with the default port 80):

1. Open Internet Information Services (IIS) Manager.
2. In the **Connections** pane, select **Sites > Default Web Site > RSViewSE**.
3. Double-click **SSL Settings**.
4. Select the **Require SSL** check box.
5. In the **Actions** pane, click **Apply**.

---

**NOTE:** If you decide to disable HTTPS, remember to clear this **Require SSL** setting.

---

## Add an HTTPS binding for a site

Before you begin, make sure that:

- You have the certificate that you want the site to use for Secure Sockets Layer (SSL).
- Microsoft Windows update KB3140245 is installed. For more information about the update, see Microsoft Support at <http://support.microsoft.com/en-in/help/3140245>.

## To add an HTTPS binding for your site

1. On the computer that hosts the website, from **Control Panel**, select **Administrative Tools > Internet Information Services (IIS) Manager**.
2. In the **Connections** pane, click the server node in the tree.
3. On the **Server Home** page, double-click **Server Certificates**.
4. In the **Actions** pane, select an action to manage the certificate.
5. Follow the instructions to finish the configuration. For more information about server certificates, see Microsoft TechNet at <http://technet.microsoft.com/library/831637.aspx>.
6. In the **Connections** pane, click **Default Web Site** in the tree.
7. In the **Actions** pane, click **Bindings**.
8. In the **Site Bindings** dialog box, click **Add**.
9. In the **Add Site Binding** dialog box,
  - From the **Type** list, select **https**.
  - From the **IP address** list, select **All unassigned** or the IP address of the website as needed.
  - In the **Port** field, type another port number if needed.
  - From the **SSL certificate** list, select the certificate that you want the site to use.
10. When finished, click **OK**.

For more information about adding a binding to a site, see Microsoft TechNet at <http://technet.microsoft.com/en-us/library/ac19cc3c-7be9-44ef-8ea6-3af39fe69a7b>.

## System policies

In a FactoryTalk View SE application, you can set up the following system policies. Local station applications do not contain Health monitoring policies or Live Data policies because these apply only to network distributed applications.

- **FactoryTalk Alarms and Events** settings include how to send audit messages to the alarm and event history log, whether to send suppressed alarms to the alarm history log, options for buffering events, and severity ranges associated with each priority category.
- **Application Authorization** settings determine which applications are authorized to access the FactoryTalk Directory.
- **User rights assignment** settings determine which users can back up and restore FactoryTalk Directory contents, manually switch the Active and Standby servers in a redundant server pair, or modify the security authority identifier.
- **Health monitoring policy** settings define system availability parameters. These include how often the system checks network connections to remote computers, and how long a network disruption can last before the system determines that communications have failed.

---

**NOTE:** Changing health monitoring policy settings can result in unexpected behavior. For most networks, the default policy settings provide the best results.

---

- **Live Data policy** settings determine which communications protocol will be used in a FactoryTalk system distributed over a network.

**NOTE:** Changing live data policy settings can result in unexpected behavior. Do not change the settings in a running production system. For the changes to take effect, all computers on the network must be shut down and restarted.

---

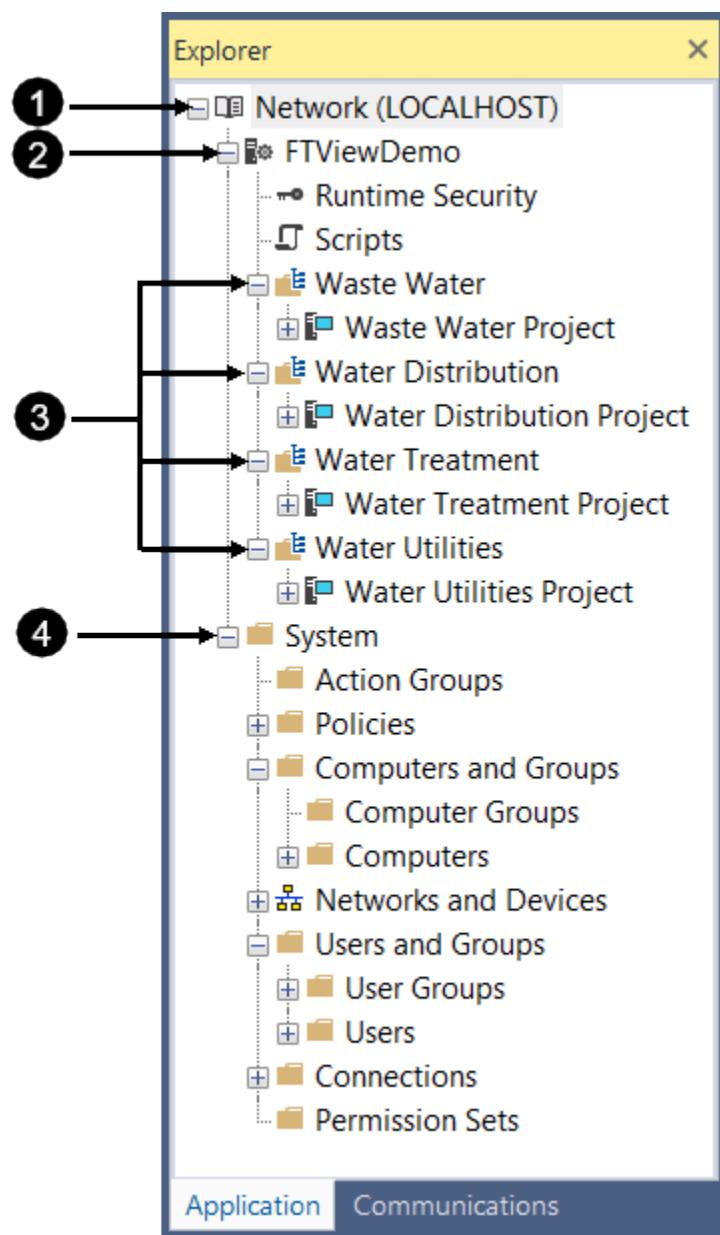
- **Audit policy** settings determine what security information is recorded while the system is in use. This includes whether FactoryTalk Diagnostics logs an audit message when a user attempts an action and is allowed or denied access.
- **Security policy** settings determine the general characteristics of security accounts and passwords. This includes whether single sign-on is enabled, and how many invalid sign-in attempts are allowed before an account is locked out.

For details about setting up system policies, see FactoryTalk Services Platform Help.

## Setting up security for FactoryTalk system resources

FactoryTalk Security manages security for system resources including the FactoryTalk Directory, the application, areas within the application and their contents, and the System folder and its contents.

In FactoryTalk View Studio, set up user access to these resources by right-clicking their folders or icons in the Explorer window, and then selecting **Security**. The following illustration shows which resources you can secure in this way:



Item	Description
1	Set up access to FactoryTalk Directory.
2	Set up access to the application.
3	Set up access to areas in the application. HMI servers always inherit permissions set up for the parent area. You cannot set up access to an HMI server separately.
4	Set up access to the System folder, or any of its subfolders. In this way, you can specify which users can set up system policies, or create user and computer accounts.

## Modifying FactoryTalk Security settings

Use the Security Settings dialog box to add the users or groups of users that are to have access to the resource, and then allow or deny permission to perform the actions that define levels of access.

When setting up permissions for a network distributed application, you can associate users and groups of users with specific computers or groups of computers.

This means you can determine not only who has access to system resources, but from which computers.

### To open the Security Settings dialog box:

In FactoryTalk View Studio, in the Explorer window, right-click the folder or icon for the resource you want to secure, and then select **Security**.

For details about options in the Security Settings dialog box, click **Help**.



**Tip:** The Security Settings dialog box in the illustration is for a network application. The Computers column does not exist in the dialog box for local station applications.

### To assign FactoryTalk Security permissions to users:

1. In the Permissions tab, select the **User** option for viewing permissions.

You can also set up permissions by selecting the **Action** option. This means that you select an action and then specify which users or groups of users can perform the action.

2. Click **Add**, select the user or group of users to add, and then click **OK**.



**Tip:** For a network application, you must associate the user or group of users with a computer, or group of computers, before you can click **OK**.

The user or group of users you added should be selected (highlighted) in the Users list, in the Permissions tab.

3. To explicitly allow permission to perform an action for the selected user or group of users, select the Allow checkbox beside the action.

You can also select the Allow checkbox beside:

- All Actions, to select all the actions that apply to this resource.
- A category of actions, such as Common, to select all the actions in the category.

4. To explicitly deny permission to perform an action, for the selected user or group of users, select the Deny checkbox beside the action.

5. Repeat steps 2 to 4 for each user or group of users that you are setting up with permissions.

6. Click **OK**.

Any users that are not set up with permissions are removed from the list.

For details about assigning permissions, see FactoryTalk Services Platform Help.

## Specifying which actions users can perform

To secure access to a system resource, you specify which users or groups of users have permission to perform actions on the resource.

In the previous illustration of the Security Settings dialog box, the Operators group has permission to perform the Common actions Read and List Children, at the FactoryTalk Network Directory.

This means that members of the Operators group are allowed to run applications managed by the directory, in a FactoryTalk View SE Client. For more information about what the Common actions allow, see the example on [Performing secured tasks in FactoryTalk View SE on page 110](#).

## About explicit and implicit permission

Selecting the Deny checkbox for an action denies permission explicitly. If you do this, keep in mind that an explicit Deny takes precedence over an explicit Allow. This is important if the account you are setting up security for belongs to more than one group.

For example, if a user belongs to one group that is allowed to delete applications, and belongs to another group that is explicitly denied that permission, then the user will not be allowed to delete applications.

Alternatively, you can clear the Allow checkbox for an action, to implicitly deny permission to perform the action.

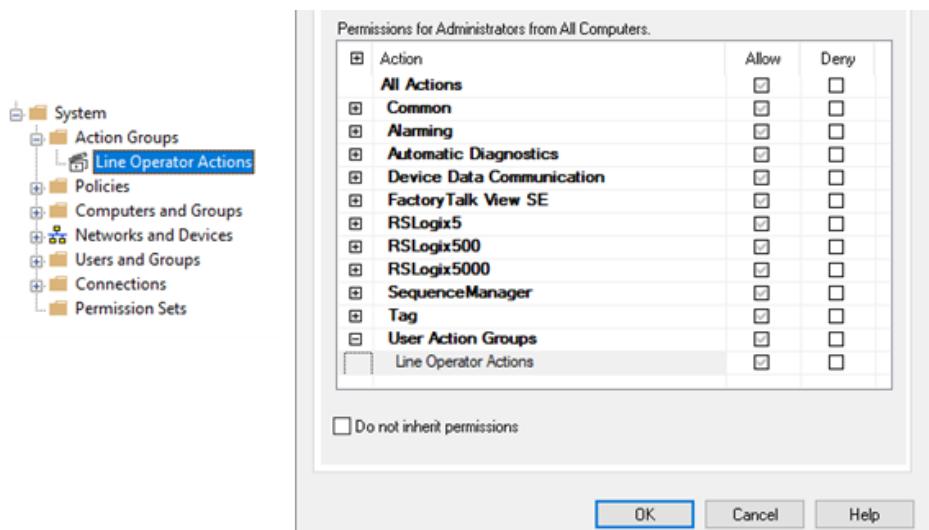
In this case, if a user belongs to one group that is allowed to delete an application, and belongs to another group that is implicitly denied that permission, then the user will be allowed to delete applications.

## Organizing actions into groups

To assign more than one action to a user or group of users in a single step, you can organize actions into groups. For example, you can group actions according to:

- Roles or jobs (operator, supervisor, maintenance engineer, and so on).
- The equipment users have access to (hoppers, mixers, ovens, and so on)

When you add an action group, it is shown in the Explorer window under the Action Groups folder, and in the Security Settings dialog box under User Action Groups, as shown in the following illustration. The newly added group, Line Operator Actions, is shown in the Explorer window and the Security Settings dialog box.



You allow or deny permission to perform an action group, just as you allow or deny permission to perform a single action. For information about assigning permissions, see [Specifying which actions users can perform on page 106](#).

**To add an action group:**

In FactoryTalk View Studio, in the Explorer window, right-click the Action Groups folder, and then click **New Action Group**.

For details about options in the New Action Group dialog box, click **Help**.

**Understanding inherited permissions**

Inheritance means that any security settings you define at the FactoryTalk Directory, extend to all system resources that the directory manages. These resources include the application and areas within the application, plus the System folder and its subfolders.

Inheritance allows you to define basic levels of access for a broad set of users, across a FactoryTalk system. You can then refine security settings for selected users as necessary, by overriding permissions inherited by the lower-level resources.

In a FactoryTalk View SE application, an HMI server always inherits the permissions assigned to the area, in which it resides. You cannot set up access to an HMI server separately.

However, the chain of inheritance that starts at the FactoryTalk Directory does not extend to a user's ability to access certain HMI project components at runtime.

To restrict access to specific FactoryTalk View commands and macros, graphic displays, OLE objects, or HMI tags, you must secure these components separately, within the FactoryTalk View SE application.

For details, see [Setting up runtime security for HMI project components on page 91](#).



**Tip:** To secure individual HMI tags at runtime, assign security codes to them in FactoryTalk View. To control access to tags in general, including HMI and data server tags, allow or deny permission to perform the Tag action Write Value.

For more information about how inheritance works, see FactoryTalk Services Platform Help.

**Breaking the chain of inheritance**

You can override inherited permissions by breaking the chain of inheritance.

**To break the chain of inheritance:**

- In the **Security Settings** dialog box for the resource, select the checkbox, Do not inherit permissions.



You are prompted to choose one of these options, instead of inheriting permissions:

- Copy the inherited permissions and make them explicit for the resource.

- Remove all inherited permission from the resource.



**Tip:** Breaking the chain of inheritance applies to the resource, not to the user or group of users selected in the Security Settings dialog box.

If you choose to copy inherited permissions, the change is applied to all listed users that have any inherited permissions. If you choose to remove all inherited permissions, all listed users that have only inherited permissions are removed from the Security Settings dialog box.

## Using explicit permissions to override inheritance

You can also override inheritance by allowing or denying explicit permission to perform actions on selected resources.

For example, in the table of [Example: Using the Common actions to set up security for user groups on page 111](#), the Supervisors group has permission to perform all Common actions, except for Configure Security, at the FactoryTalk Network Directory.

In the Security Settings dialog box for the application, inherited permissions set up at the FactoryTalk Directory for the Supervisors group show as gray check marks, as in the following illustration.

Action	Allow	Deny
All Actions		
Common		
Configure Security	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Create Children	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Delete	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Execute	<input checked="" type="checkbox"/>	<input type="checkbox"/>
List Children	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Do not inherit permissions

Items	Description
1	Explicit allow
2	Explicit deny
3	Gray check marks indicate inherited permission to perform those actions.

To extend or further restrict this group's access at the application level, you might make the following explicit changes, to override the inherited permissions:

- Select the Allow checkbox beside the Configure Security action. Explicitly allowing the Configure Security action on the application means that Supervisors can set up security for the application, add Runtime Security accounts and assign security codes to those users, and secure HMI project components.

- Select the Deny checkbox beside the Delete action.
- Explicitly denying the Delete action on the application means that Supervisors are prevented from deleting applications.

The changes are reflected in the Effective Permissions tab, as shown in the following illustration.

Effective permissions for Supervisors from All Computers.

Action	Allowed
All Actions	✓
Common	✓
Configure Security	✓
Create Children	✓
<b>1 Delete</b>	✗
Execute	✓
List Children	✓
Read	✓
Write	✓

Items	Description
1	Check mark means Configure Security is allowed.
2	No check mark means Delete is denied.

When setting up permissions, or overriding inherited permissions, keep these rules of precedence in mind:

- Explicit Deny takes precedence over explicit Allow.
- Explicit Allow takes precedence over implicit Deny.



**Tip:** When you use explicit permissions (Allow or Deny) to override inheritance on a particular resource, these permissions become the ones that are inherited by lower-level resources.

## Performing secured tasks in FactoryTalk View SE

To perform certain tasks in FactoryTalk View SE, users must have the security permissions needed for access to the FactoryTalk Directory, application, or area where the task is to be authorized.

For information about:

- How to assign permissions to users, see [Specifying which actions users can perform on page 106](#).
- How inherited permissions work, see [Understanding inherited permissions on page 108](#).

Some of the tasks FactoryTalk View SE users can perform require additional permissions. For information about permissions related to:

- Switching the Active and Secondary servers in a redundant pair, see [What happens if both servers become active on page 244](#).
- Enabling or disabling, suppressing or unsuppressing, or acknowledging FactoryTalk alarms, see [Setting up FactoryTalk alarms on page 177](#).

In the **Security Settings** dialog box, you might also see actions for other FactoryTalk products, for example, FactoryTalk Linx. For details about product-specific actions, see the product documentation.

## Example: Using the Common actions to set up security for user groups

The following table shows how a system administrator might assign the Common actions to four groups of FactoryTalk View users—Administrators, Engineers, Supervisors, and Operators—to give them appropriate levels of access to a network distributed application.



**Tip:** In addition to the Common actions listed, each group in this example is allowed the Tag action Write Value, which governs general access to HMI and data server tags.

You can also set up runtime security for individual HMI tags. For details, see [Assigning security codes to HMI tags on page 93](#).

For information about overriding inherited permissions, see [Understanding inherited permissions on page 108](#).

To set up this level of access	For this group	Set up these permissions at the Network Directory	And then override inherited permissions
<p>Full access.</p> <p>This includes the ability to:</p> <ul style="list-style-type: none"> <li>• Create applications</li> <li>• Add areas and servers</li> <li>• Create HMI project components</li> <li>• Set up permissions for all resources the FactoryTalk Directory manages</li> <li>• Create user accounts</li> <li>• Add Runtime Security accounts</li> <li>• Secure HMI project components</li> </ul> <p>For a complete list of tasks users with full access can perform, see the previous table in this example.</p>	Administrators	<p>Allow Common actions:</p> <ul style="list-style-type: none"> <li>• Configure Security</li> <li>• Create Children</li> <li>• Delete</li> <li>• List Children</li> <li>• Read</li> <li>• Write</li> </ul> <p>Allow Tag action:</p> <ul style="list-style-type: none"> <li>• Write Value</li> </ul>	<p>No changes.</p> <p>Retain inherited permissions at all lower-level resources.</p>

To set up this level of access	For this group	Set up these permissions at the Network Directory	And then override inherited permissions
<p>The same access as Administrators, except <i>members of this group cannot:</i></p> <ul style="list-style-type: none"> <li>• Set up security for the Users and Groups subfolder of the System folder.</li> <li>• Create users at the FactoryTalk Directory.</li> </ul> <p>However, members of this group <i>can</i> add Runtime Security accounts, assign security codes to them, and use the Runtime Secured Commands editor.</p>	Engineers	<p>Allow Common actions:</p> <ul style="list-style-type: none"> <li>• Same as Administrators</li> </ul> <p>Allow Tag action:</p> <ul style="list-style-type: none"> <li>• Write value</li> </ul>	<p>Explicitly deny these Common actions on the Users and Groups folder (in the System folder):</p> <ul style="list-style-type: none"> <li>• Configure Security</li> <li>• Delete</li> <li>• Create Children</li> </ul> <p>Retain all other inherited permissions.</p>
<p>Runtime access (see Operators group, next), plus, <i>members of this group can:</i></p> <ul style="list-style-type: none"> <li>• Modify existing applications.</li> <li>• Modify HMI server properties.</li> <li>• Create HMI project components.</li> </ul>	Supervisors	<p>Allow Common actions:</p> <ul style="list-style-type: none"> <li>• Read</li> <li>• List Children</li> </ul> <p>Tag action:</p> <ul style="list-style-type: none"> <li>• Write Value</li> </ul>	<p>Explicitly allow these Common actions on the application:</p> <ul style="list-style-type: none"> <li>• Delete</li> <li>• Write</li> <li>• Create Children</li> </ul> <p>Retain all other inherited permissions.</p>
<p>Runtime access. <i>Members of this group can only:</i></p> <ul style="list-style-type: none"> <li>• Load existing applications.</li> </ul> <p>However, members of this group cannot modify HMI server properties, nor view HMI project components. HMI servers show in the Explorer window as locked.</p> <ul style="list-style-type: none"> <li>• Run applications in the FactoryTalk View SE Client.</li> <li>• Write to tags at runtime.</li> </ul> <p><b>IMPORTANT:</b> to restrict access to individual HMI tags, in the Runtime Security editor, assign security codes to this group.</p>	Operators	<p>Allow Common actions:</p> <ul style="list-style-type: none"> <li>• Read</li> <li>• List Children</li> </ul> <p>Allow Tag action:</p> <ul style="list-style-type: none"> <li>• Write Value</li> </ul>	<p>No changes.</p> <p>Retain all inherited permissions at lower-level resources.</p>



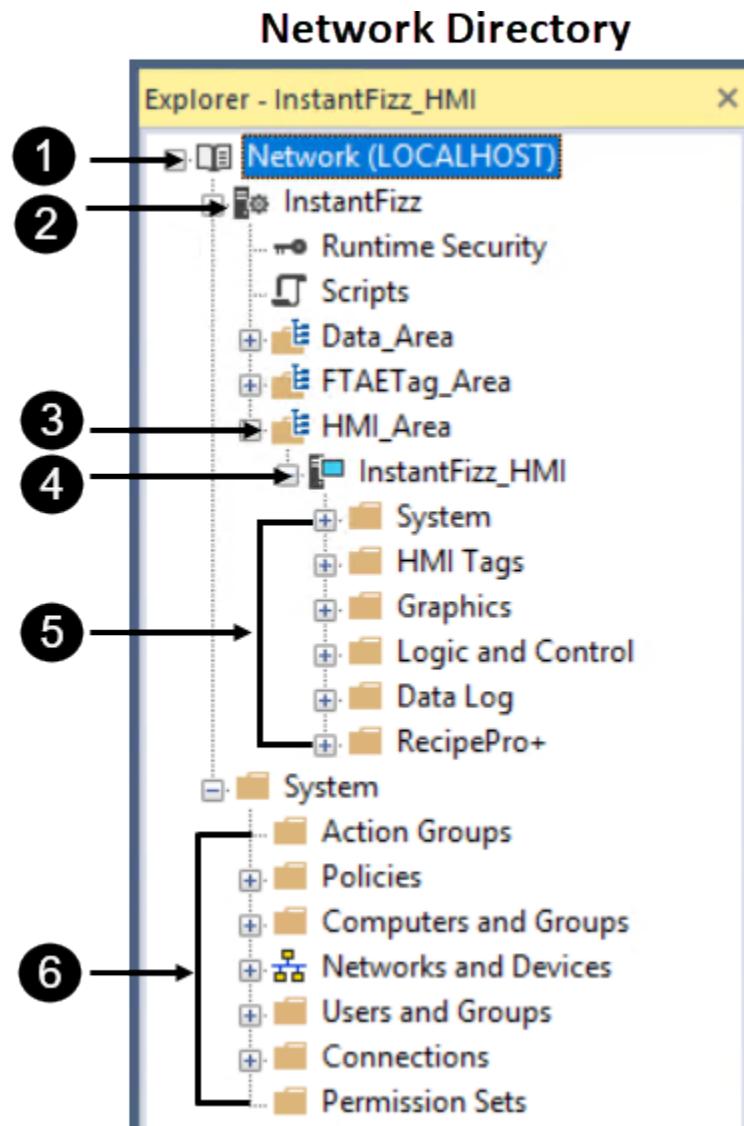
# Working with network distributed applications

This chapter describes:

- What a FactoryTalk View Site Edition network distributed application is.
- Key network distributed application concepts.
- How to create a network distributed application.
- How to add areas and servers to a network distributed application.
- How to set up HMI server properties.
- How to monitor the status of an HMI server.
- How to delete HMI servers.
- How to rename and delete network distributed applications.

## About network distributed applications

In FactoryTalk View Studio, you can create FactoryTalk View Site Edition local station, network station, and network distributed applications. Here is what a network distributed application looks like in the Explorer window:



Item	Description
1	Host computer
2	Application
3	Area
4	HMI server
5	HMI project components and editors
6	FactoryTalk system settings

Network station applications are described in [Working with network station applications on page 131](#). Local station applications are described in [Working with local station applications on page 142](#).

## Parts of a network distributed application

A typical FactoryTalk View SE network distributed application (also called a distributed application) consists of:

- Areas, including nested areas that divide the application into manageable parts or organize it in a way that makes sense for the process it is controlling.  
You can also use areas to separate servers that use the same names, for example, two HMI servers running projects that contain the same set of graphic displays or tags.
- One or more HMI servers that provide FactoryTalk View components and services to application clients.  
FactoryTalk View applications must contain at least one HMI server. A network distributed application can contain up to 10 HMI servers, running on different computers on a network, or up to 10 redundant HMI server pairs.
- HMI project components such as graphic displays, HMI tags, and data log models.
- One or more data servers, with or without the use of HMI tags  
Clients communicate through data servers, which provide access to information in devices and other data servers that comply with the OPC DA 2.05a, UA 1.02 or 1.03 specifications.  
In a network distributed application, you can use multiple FactoryTalk Linx and OPC data servers (including RSLinx Classic), running on different computers. You can also set up a redundant pair of host computers for the data servers in the application. For information about the requirements that apply to redundant data servers, see the product documentation.  
FactoryTalk Linx servers can be set up to subscribe to alarms detected in devices such as Studio 5000 controllers. For more information, see [Setting up FactoryTalk alarms on page 177](#).



**Tip:** Every vendor's OPC data server is different. Some contain their own tag databases, like the tag database in an HMI server, while others reference the tag databases or addresses that exist in controllers, as is the case with FactoryTalk Linx and Studio 5000 controllers.

- A list of users, plus the security codes that allow or deny these users permission to access secured HMI project components at runtime.
- Optionally, one or more FactoryTalk Tag Alarm and Event Servers, to provide alarm monitoring and control for tags in devices that do not have built-in alarm detection. For more information, see [Setting up FactoryTalk alarms on page 177](#).

The software programs supporting a network distributed application—the FactoryTalk Network Directory, the HMI servers, the HMI clients, and the data servers—can be on different computers on the network.

However, all the computers participating in a network distributed application must point at the same Network Directory. For details, see [Setting up the FactoryTalk Directory on page 70](#).

## About FactoryTalk systems

FactoryTalk View SE and other Rockwell Automation software products use a set of common FactoryTalk services to support certain functions, such as diagnostic messages and access to real-time data.

These services, including FactoryTalk Directory, are installed with the FactoryTalk Services Platform when you install FactoryTalk View SE. For an overview of FactoryTalk services, see [FactoryTalk Services Platform on page 31](#).

An automation and control system that uses FactoryTalk services, and integrates FactoryTalk products and components, is known as a FactoryTalk system. A complete FactoryTalk system consists of all the networks, devices, and software applications you have deployed to monitor and control your plant or process. Helping to ensure that the system can provide data in a secure and predictable fashion depends on a number of variables.

## Finding more information about FactoryTalk services

This manual contains information about developing FactoryTalk View SE applications, including information about how FactoryTalk View uses FactoryTalk services.

For additional, detailed information about FactoryTalk systems, services, concepts, and components, see the FactoryTalk Help.

## Key network distributed application concepts

This section presents some of the common terms and concepts that are used to describe FactoryTalk View SE network distributed applications.

## FactoryTalk Network Directory

FactoryTalk Directory centralizes access to application resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed applications, and network station applications. All of the client and server computers participating in a given network distributed application, or network station application must point at the same Network Directory.

For more information about FactoryTalk Network Directory, see [Setting up the FactoryTalk Directory on page 70](#).

---

### NOTE:

Do not install FactoryTalk Directory, FactoryTalk View SE Server, or any other application software on the same computer as the Windows domain controller. This configuration is not supported.

---

## FactoryTalk Security

FactoryTalk View SE applications can use FactoryTalk Security services to authenticate and authorize application users.

During FactoryTalk View SE installation, Windows users with administrative privileges on the computer are set up with full, initial access to FactoryTalk View SE applications managed by a FactoryTalk Local or Network Directory on the same computer.

In FactoryTalk View Studio, you can create FactoryTalk user and group accounts, and then determine which accounts have access to resources such as the Local Directory, or the application.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#). For details, see FactoryTalk Services Platform Help.

## About FactoryTalk Security permissions

If FactoryTalk Security services are used to secure parts of an application, to perform certain tasks, users must have the necessary security permissions.

For example, to create or modify the properties of an application, you must at least be allowed the Common actions Read, List Children, Write, and Create Children, at the FactoryTalk Directory that manages the application.

If you receive a FactoryTalk Security message while trying to perform such a task, contact your system administrator about the permissions you might require.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#).

## HMI servers (network distributed)

HMI servers are software programs that supply information to clients as they request it.

An HMI server stores HMI project components such as graphic displays, and serves these components to clients. An HMI server also manages a database of tags, detects and annunciates, and logs historical data.

In FactoryTalk View Studio, first you create a network distributed application, and then you add one or more HMI servers to the application. Each area or subarea in a network distributed application can contain only one HMI server. For information about:

- Adding an HMI server to an, see [Adding an HMI sever on page 121](#).
- Setting up redundant HMI servers, see [Setting up HMI server redundancy on page 126](#).
- Specifying which components will run when an HMI server starts, see [Selecting startup and shutdown components on page 127](#).

## HMI projects (network distributed)

HMI projects contain graphic displays, data log models, HMI tags, and other services. An HMI project is created when you add an HMI server to a network distributed application.

The HMI project is loaded by the HMI server, either when the first client connects to the server, or when the operating system initializes. For more information, see [Choosing how the server starts on page 125](#).

## HMI clients

HMI clients are software programs that obtain information from, or write information to HMI servers or data servers. FactoryTalk View Studio, the FactoryTalk View SE Administration Console, and the FactoryTalk View SE Client are all HMI clients.

## Areas (network distributed)

All FactoryTalk View applications have one system-defined area called the application root area, which has the same name as the application. The application root area can contain one HMI server, and one or more data servers.

In a network distributed application, you can create additional areas to divide the application into manageable, logical parts, or to organize it in a way that makes sense for the process it is controlling.

For example, an area might represent a portion of a process, or a region within the process facility. An automotive plant could be divided into areas called Press and Fabrication, Body Shop, Paint Shop, Engine, and Transmission; a bakery could be divided into areas called Ingredients, Mixing, Baking, and Packaging.

Alternatively, a plant with identical production lines could be divided into areas called Line 1, Line 2, Line 3, and so on. To add a production line to the application, you could create an area, and then copy the identical HMI server project into the area.

Each area you add to a network distributed application can contain one or more subareas, and one or more data servers. Each area or subarea can contain only one HMI server.

An area or subarea can contain multiple data and alarm servers, but it is best practice for each area to contain only a single data or alarm server.

### About the home area

In an application, the area that contains a given application component, such as a graphic display, is called the home area.

When you refer to an application component without specifying the area, FactoryTalk View SE uses the home area to locate the component.

For example, if an object in a graphic display refers to a tag without specifying an area, FactoryTalk View assumes that the tag and the display are in the same home area.

If the tag cannot be found in an HMI server or a data server in the display's home area, an error is logged when the display is run.

## Absolute and relative references

You can use absolute or relative references to refer to application components, such as graphic displays and tags, when building FactoryTalk View commands or connecting graphic objects to process data.

---

**NOTE:** Local station applications use only relative references.

---

Absolute references work like file paths. Relative references work like file names.

**Absolute references** point directly at a specific component, by referring to the component's name and the area (or areas, in the case of nested areas) in which it is located. For example, an absolute reference to a graphic display called Detail in an area called AssemblyLineNorth is `/AssemblyLineNorth::Detail`.

An absolute reference to an application's root area does not include the application's name, even though the name is shown in the root area in FactoryTalk View Studio.

**Relative references** point at a component relative to the current server or area. For example, a relative reference to a graphic display named Detail is simply the display's name `Detail`. When a relative reference is used, FactoryTalk View assumes that the component is in the current area.

Use relative references, for example, to reuse component names in a network distributed application for a plant that has identical production lines. Such an application might contain different areas to represent each production line; however, each area would contain the same component names.

## Absolute reference syntax

When using absolute references:

- Precede area names with a forward slash ( / ). Also use forward slashes to separate area names from other area names.
- Separate area names from component names with two colons ( :: ).

### Example: Absolute and relative references to tags in the same graphic display

The graphic display named Detail, used in the previous examples, contains an absolute reference to a tag named Speed in the AssemblyLineNorth area. The absolute reference is:

```
/AssemblyLineNorth::Speed
```

The display also contains a relative reference to a tag named Temp. The relative reference is the tag's name:

```
Temp
```

If you were to copy the Detail display to an area named AssemblyLineSouth, the display would still look for the tag named Speed in the AssemblyLineNorth area, because it contains an absolute reference to that tag.

However, the display would look for the tag named Temp in the AssemblyLineSouth area because it contains a relative reference to that tag.

If the tag named Temp cannot be found in the AssemblyLineSouth area when the Detail display is run, a FactoryTalk Diagnostics error will be logged.

### Example: Using absolute and relative references with the Display command

To create a button that an operator can press to open a display named Overview, use the **Display** command in the button's press action, in one of the following ways:

- To create an absolute reference to the display called Overview in the area called Baking, type:  
`Display /Baking::Overview`
- To create an absolute reference to the display called Overview in the area called Packaging, type:  
`Display /Packaging::Overview`

- To create an absolute reference to the display called Overview in the root area of the application, type  
`Display /::Overview`
- To create a relative reference to the display called Overview, in the area where the **Display** command is run, type:  
`Display Overview`



**Tip:** Use the Command Wizard to build commands that take application components as parameters. The wizard supplies the correct syntax, based on the location of the component you select.

## System availability

To minimize data loss and down time, and to help ensure that critical parts of your system are always available to connected clients, FactoryTalk View SE provides the health monitoring and redundancy features.

For information about these and other availability features, see [Setting up FactoryTalk system availability on page 232](#).



**Tip:** You can set up HMI redundancy in network distributed applications only. You can set up Data Server and FTAE Redundancy in network distributed applications and network station applications, but not in local station applications.

## About HMI server redundancy

In process monitoring and control automation, redundancy refers to the ability of the system running the process to continue working correctly, when parts of the system become unavailable or fail.

In a FactoryTalk View SE network distributed application, you can set up redundancy on paired computers, for critical components such as HMI servers. You can set up HMI redundancy in Network Distributed applications only. One computer in the pair hosts the primary server, and the other hosts the secondary (or backup) server.

In the event that the primary HMI server cannot provide service to application clients, the system switches to the secondary server, and FactoryTalk View SE Clients continue to function normally. There is no need to restart client or active server computers.

You can set up Data Server and FTAE Redundancy in Network Distributed applications and Station applications. Redundancy is not available for Local Station applications. For more information, see [Setting up a redundant HMI server pair on page 241](#).



**Tip:** For details about setting up redundancy for other application servers, such as third-party OPC data servers, see the product documentation.

## Language switching

To make user-defined text strings in an application available at runtime in up to 40 different languages, set up language switching for the application.

For more information, see [Setting up language switching on page 210](#).

## Creating network distributed applications

To create a network distributed application, in FactoryTalk View Studio, first you create the application, and then you add elements such as areas, HMI servers, data servers, and FactoryTalk Tag Alarm and Event Servers.

### To create a network distributed application:

1. Select **Start > FactoryTalk View Studio**.
2. In the **Application Type Selection** dialog box, select the application type and click **Continue**.
3. In the **New/Open Site Edition Application** dialog box, click the **New** tab.
4. Type the name and description, select a default language, and click **Create**.
5. (optional) To change the default folder of the HMI projects, use the HMI Projects Folder Path Config Utility to specify a new folder.

## Adding areas and servers to a network distributed application

A network distributed application can consist of one or more areas, up to 10 HMI servers (one per area or subarea), and, if necessary, multiple data servers and FactoryTalk Tag Alarm and Event Servers. You add these elements to the application, after you create it.

## Adding and deleting areas

You can add and delete areas in an application. You cannot copy areas.

### To add an area:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root or right-click an area name, and then click **New Area**.
2. In the New Area dialog box, type a name for the area, and an optional description, and then click **OK**.

### To delete an area:

Right-click the area that you want to delete, and then click **Delete**.

When you delete an area, HMI servers and data servers in the area are not deleted from disk.

## Adding an HMI server (network distributed)

After you create a network distributed application, you must add at least one HMI server, either to the application's root area, or to an area you have added.

A network distributed application can contain multiple HMI servers, but each area in an application can contain only one HMI server.

**To add an HMI server:**

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root or right-click an area name, select **Add New Server > HMI Server**.
2. In the **Add HMI Server Wizard**, in the **Select Operations** window, click one of the following options:
  - **Create a new HMI server.** When you create an HMI server, the server's HMI project is created automatically.
    - **Add Application Library** to add one or multiple Application Libraries.
  - **Copy an HMI server** that already exists. After you have copied an HMI server, changes made to the original do not affect the copy, or vice versa.
  - **Import a project** from RSView32, FactoryTalk View SE, or FactoryTalk View Machine Edition, as the basis for the new HMI server. After you have copied a project, changes made to the original do not affect the copy, or vice versa.
  - **Attach to an existing HMI server** without making a copy of the HMI server.

---

**NOTE:** You cannot attach to an existing HMI server that is being used in another application.

---

3. Click **Next**, and then follow the instructions in the wizard to finish adding the server.

After adding the HMI server, you can set up its properties. For details, see [Setting up HMI server properties on page 139](#).

Then, you can use editors in the Explorer window to create HMI project components, such as graphic displays, HMI tags, and data log models.

## Adding a data server

The application root area can contain one HMI server and one or more OPC Data servers. It is a system best practice to only have one FactoryTalk Linx data server in the root area or individual area.

**To add a FactoryTalk Linx data server:**

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root or right-click the area name, select **Add New Server > Rockwell Automation Device Server (FactoryTalk Linx)**.
2. Set up server properties, such as the name and location of the server, whether to provide redundancy using a secondary server, and whether the server will support alarms and events.  
For details, see the FactoryTalk Linx product documentation.

**To add an OPC data server:**

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root, or right-click the area name, select **Add New Server > OPC DA Server** or **OPC UA Server**.
2. Set up properties as needed. For details, see the product documentation.

For additional information about setting up FactoryTalk Linx and OPC data servers, see [Setting up communications on page 149](#).

## Adding a Tag Alarm and Event Server

Optionally, you can add FactoryTalk Tag Alarm and Event Servers to an application.

Tag Alarm and Event Servers use FactoryTalk Alarms and Events services to monitor and control alarms for tags in programmable controllers (PLC-5 or SLC 500) and other devices that do not have built-in alarm detection.

For more information about Tag Alarm and Event Servers, see [Setting up FactoryTalk alarms on page 177](#).

## Deciding when to use multiple servers

A network distributed application does not require more than one HMI server or data server, but there are circumstances in which it is helpful to use additional servers. For example, you can add servers:

- To set up redundancy for HMI servers or data servers.
- To let different integrators work on different HMI servers without worrying about name collisions or having to integrate all the project files later.

## Adding servers for redundant operation

To provide secondary application servers in the event that the primary servers go out of service, set up redundant HMI servers and data servers.

You must set up redundancy separately for each server in an application. For more information, see [Setting up FactoryTalk system availability on page 232](#).

## Adding servers for load balancing

Over time, an HMI project might increase in size, or the number of clients connecting to an HMI server or data server might increase to the point where the computer running the server becomes overloaded.

To balance the load, install another HMI server or data server on a separate computer, and then move some of the HMI project components, or some of the OPC server tags, to the new server. This distributes processing more effectively across computers.

## Running multiple HMI servers

Each area in a network distributed application can contain only one HMI server.

When deploying a network distributed application for production, ensure that there are no more than two HMI servers running on each host computer. If the HMI servers are primary and secondary partners in a redundant pair, only one server per computer is permitted.

For more information about system limits and requirements, see [About FactoryTalk View SE system limits on page 239](#).

## Providing HMI server names

The FactoryTalk Directory stores HMI server names and does not delete them, even if you remove an HMI server from all the applications in which it is used, and then delete the HMI server's project files.

This means that every HMI server must have a unique name:

- You cannot reuse an HMI server's name, even after you delete the HMI server from an application.
- If multiple users share the same FactoryTalk Directory, the names of all HMI servers created by all users must be unique.

Similarly, if multiple applications share the same FactoryTalk Directory, the name of each HMI server in each application must be unique.

## Monitoring disk space on HMI servers

After deploying a FactoryTalk View SE network distributed application, monitor disk space on computers running HMI servers to make sure that it does not fall below 500 MB.

### To monitor disk space:

1. In the HMI server's Events editor, create an event that uses the free\_bytes function to return the number of free bytes available on the HMI server's hard drive.
2. Create an alarm of FactoryTalk Alarms and Events and set up the alarm conditions for the disk space.

For information about setting up events, see [Adding logic and control on page 423](#).

## Setting up HMI server properties (network distributed)

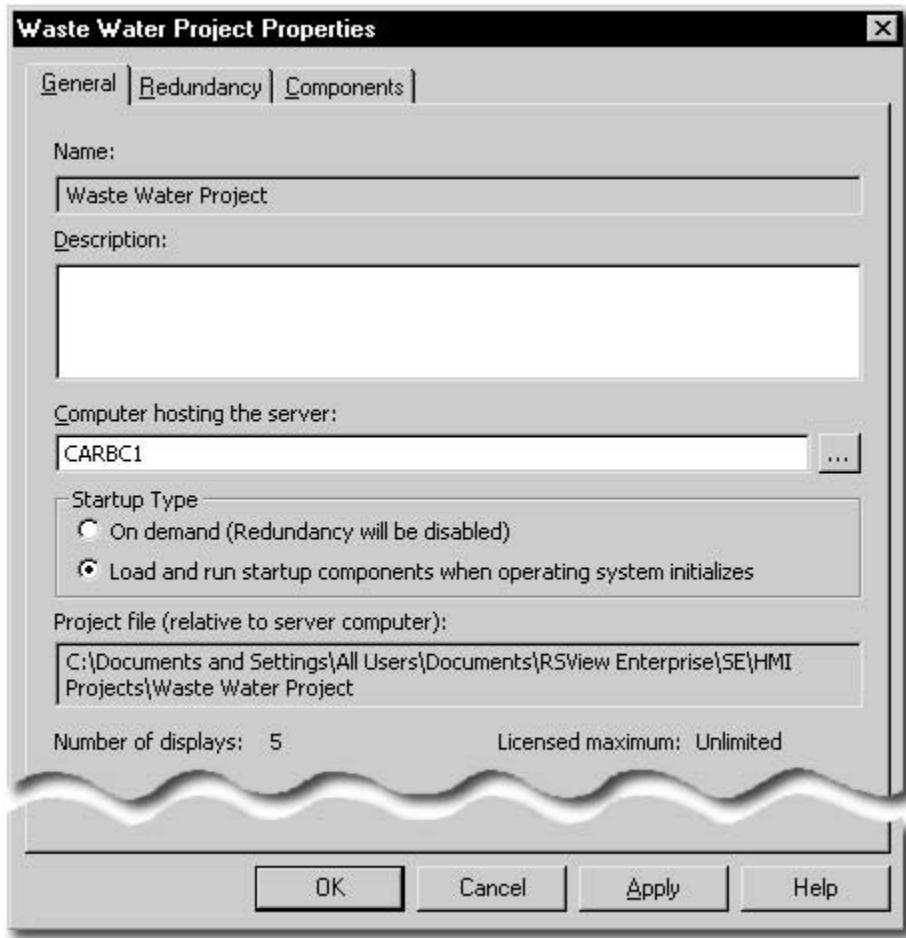
After adding an HMI server to a network distributed application, use the Properties dialog box to:

- Select a startup type for the HMI server.
- Set up redundancy for the HMI server.
- Select the components that will run when the HMI server starts up.
- Specify a macro to run when the HMI server shuts down. If the HMI server is redundant, specify a macro for when the server becomes active, or goes into standby mode.
- Change the name of the computer on which the HMI server is running.

You must set up the properties of each HMI server in a network distributed application separately.

### To open the HMI Server Properties dialog box:

In FactoryTalk View Studio or the FactoryTalk View SE Administration Console, in the Explorer window, right-click the HMI server icon, and then select **Properties**.



For details about options in the HMI Server Properties dialog box, click **Help**.

## Changing the name of the host computer

In the General tab of the HMI Server Properties dialog box, the name of the computer running the HMI server is shown beside the text box **Computer hosting the server**.

To change to the name of the host computer, a copy of the HMI server must exist on the computer that will be the new host.

## Choosing how the server starts

You can set up an HMI server to start on demand, when the first client connects to it, or to start when the operating system initializes.

## Starting the HMI server when the first client connects to it

An HMI server set up to start on demand is loaded when the first client gains access to the HMI server project. The server is unloaded when the last client connected to it is closed.

Start HMI servers on demand only when the control system is under development. In a runtime environment, the HMI server should start when the operating system starts to ensure that no logged data is missed.



**Tip:** An unlicensed HMI server can only be set up to start on demand.

## Starting the HMI server when the operating system initializes

An HMI server set up to start when the operating system initializes is loaded when Windows starts up. You must use this option to set up redundancy for the HMI server.

This startup option is also recommended for production, as it ensures that the HMI server will resume operating automatically if the host computer is restarted.

Because the HMI server runs as a service, users do not have to sign in to Windows to make the HMI server run.

If you set up redundancy for an HMI server, you can run components when the server starts up, or you can specify macros to start components when the server becomes active. For more information, see [Selecting startup and shutdown components on page 148](#).

To unload an HMI server that starts when Windows starts, you can change its startup type to on demand, and then close all connected clients.

---

**NOTE:** If you change an HMI server's startup type, you must restart the server's host computer for the change to take effect.

---

## Setting up HMI server redundancy

To minimize disruptions to clients if the HMI server fails, in the **Redundancy** tab of the HMI Server Properties dialog box, select **Provide redundancy using a secondary server**.



**Tip:** This option is available only if the HMI server's startup type is set to **Load and run startup components when operating system initializes**. For more information about HMI server startup types, see [Starting and stopping HMI server components manually on page 139](#).

## Selecting the secondary server

In FactoryTalk View SE, the redundant (or backup) server is called the secondary server.

To specify the secondary server in a redundant pair, in the **Redundancy** tab of the HMI Server Properties dialog box, provide the name of the computer hosting the secondary HMI server.

The settings you specify in the **Redundancy** tab for the primary HMI server are saved with the application, and are the same for both the primary and the secondary server. No further setup is required on the computer hosting the secondary server.

For more information on setting up redundant HMI servers, see [Setting up FactoryTalk system availability on page 232](#).

## Replicating changes to the standby HMI server

At runtime, you can change the HMI project components in an application while it is running. For example, you can add, modify or delete graphic displays and other components, and have those changes to take effect at connected clients.

To automatically save the project online edits to both active and standby servers, in FactoryTalk View Studio, from the menu, select **Tools > Options**. On the **Settings** tab, make sure the **Save edits to both active and standby HMI servers** option is selected. To keep the HMI memory tag values the same on both active and standby servers, on the same tab, select the **Auto synchronize HMI memory tag values to both active and standby HMI servers** option.

If you do not set up the option, manually replicate the changes from the active server to the standby server. The replicate operation copies the current active server's configuration files, including settings in the Components tab, to the standby server.

For example, if an On Active and a Shutdown macro is selected for the Active server, the settings will be replicated to the Standby server. It means that the same macros specified for the Active server will run when the Standby server becomes active or shuts down. Information that is not included in the replicate operation includes data log files generated at runtime, the current value of HMI memory tags, and retentive tags.

---

**NOTE:** Make sure to verify the current HMI server name and status before the replication. For information about checking server states, see [Monitoring the status of application servers on page 233](#).

---

### To manually replicate the changes:

1. In the FactoryTalk View Studio Explorer window, right-click the HMI server and select **Server Status**.
2. In the **Server Status** dialog box, verify that the current HMI server status is **Active** and the other is **Standby**.



**Tip:** To show the status of the HMI servers, FactoryTalk Alarms and Events servers, and data servers at runtime, add one of the following to the application:

- A **Server Status** display under the **Library** folder.
- A redundancy function, such as **PrimaryServerStatus** and **SecondaryServerStatus**.

3. In the HMI Server Properties dialog box, in the **Redundancy** tab, click **Replicate Active to Standby**.

## Selecting startup and shutdown components (network distributed)

In the Components tab of the HMI Server Properties dialog box, select the checkbox for each of the items you want to run when the HMI server starts or stops, including:

- Which components will run when the HMI server starts up.
- Which macros will run when the HMI server shuts down.
- Which macros will run when the system switches to the secondary HMI server, or back to the primary HMI server.

Where required, specify the name of the component. For details about options in the Components tab, click Help.

For information about macros, see [Adding logic and control on page 423](#).

**NOTE:** The order of items in the Components tab is not the order in which the items will run when the HMI server starts up. If components must run in a particular order, create a macro to start the components, and then select the macro to run when the HMI server starts.

## Running a macro when the HMI server goes into standby mode

The macro that runs when an HMI server becomes active is called the On active macro. An HMI server can become active if another HMI server fails, or if it is the active server at startup time.

Any FactoryTalk View macro can be used as the On active macro. If you have not set up a secondary server in the Redundancy tab, the option to run an On active macro is not available.

## Running a macro when the HMI server goes into standby mode

The macro that runs when an HMI server goes into standby mode is called the On standby macro.

For example, when a primary HMI server recovers, the system switches from the active secondary back to the primary server. Once the primary server becomes active, the secondary switches to standby, and its On standby macro runs.

Any FactoryTalk View macro can be used as the On standby macro. If you have not set up a secondary server in the Redundancy tab, the option to run an On standby macro is not available.



**Tip:** The On standby macro will not run if the active HMI server fails in such a way that it cannot run the macro, for example, if power to the server computer fails.

## Starting and stopping HMI server components manually (network distributed)

For each HMI server in a network distributed application, including the secondary HMI server in a redundant pair, you can start and stop the server's components manually.

### To start HMI server components manually:

1. Select **Start > FactoryTalk View Studio**.
2. In the **Application Type Selection** dialog box, select the application type and click **Continue**.
3. On the **Existing** tab, select the application and click **Open**.
4. In the Explorer window, right-click the HMI server and select **Properties**.
5. In the Components tab, click **Run Startup Components**.

The startup components selected for the HMI server should start running.



**Tip:** If redundancy is set up for the HMI Server, the **Run Startup Components** button is available for the primary and the secondary HMI server. Click either button, depending on which server you want to run startup components for.

## To stop HMI server components manually:

1. Select **Start > FactoryTalk View Studio**.
2. In the **Application Type Selection** dialog box, select the application type and click **Continue**.
3. On the **Existing** tab, select the application and click **Open**.
4. In the Explorer window, right-click the HMI server and select **Properties**.
5. In the Components tab, click **Stop All Running Components**.

All the components running on the computer will stop running, including alarms, data log models, derived tag components, and event components. If a shutdown macro is specified in the Components tab, the specified macro will also run.



**Tip:** If redundancy is set up for the HMI Server, the **Stop All Running Components** button is available for the primary and the secondary HMI server. Click either button, depending on which server you want to stop components for.

## Monitoring the status of an HMI server

In FactoryTalk View Studio, in the Server Status dialog box, you can view the operational status of an HMI server to determine whether the server is ready to provide service, and to troubleshoot problems.

For example, an HMI server in the Active state is fully loaded and ready to provide service to connected clients. A client can be any of FactoryTalk View Studio, a FactoryTalk View SE Client, or the FactoryTalk View SE Administration Console.

For more information about server states, see [Monitoring the status of application servers on page 233](#).

## To open the Server Status dialog box:

In FactoryTalk View Studio, in the Explorer window, right-click the HMI server's icon, and then select **Server Status**.

For details about options in the Server Status dialog box, click **Help**.

## Deleting HMI servers

You can delete an HMI server from an application without deleting the HMI project files, and then add the HMI server to the application later.

You can also add the HMI server temporarily, while you copy components to other HMI servers in the application.

## To delete an HMI server:

1. Disconnect all clients from the HMI server. (A client is any of FactoryTalk View Studio, FactoryTalk View SE Administration Console, or the FactoryTalk View SE Client.)
2. Start FactoryTalk View Studio, and then open the application that contains the HMI server you want to delete.
3. Right-click the HMI server, and then select **Delete**.  
This does not delete the HMI server's project files.

## Deleting HMI server project files

After you delete an HMI server from an application, if you want to delete the project files, you have to delete them separately.

### To delete an HMI server's project files:

1. Disconnect all clients from the HMI server. (A client is any of FactoryTalk View Studio, the FactoryTalk View SE Administration Console, or the FactoryTalk View SE Client.)
2. In Windows Explorer, browse to the HMI projects folder on the computer hosting the HMI server. The default path is C:\Users\Public\Documents\RSView Enterprise\SE\HMI projects.
3. In the HMI projects folder, delete the subfolder that has the same name as the HMI server you deleted from the application.

For example, if you deleted an HMI server named Water, then you will delete an HMI project folder also named Water.

## Renaming and deleting network distributed applications

Use the FactoryTalk View SE Application Manager tool to rename or delete a network distributed application.



**Tip:** You cannot rename or delete a network distributed application that is in use. Ensure that all users disconnect from the application first.

### To rename or delete the application

1. Select **Start > Rockwell Software > FactoryTalk View SE Application Manager**.
2. In the **Common Operations** tab, select your application and follow the wizard.

When you delete a network distributed application, the HMI servers and data servers set up in it are not deleted automatically. For details about:

- Deleting the HMI server files, see [Deleting HMI server project files on page 130](#).
- Removing a data server and deleting its cache files, see [Deleting an OPC data server on page 157](#).

## Backing up and restoring network distributed applications

There are several steps involved in backing up and restoring a FactoryTalk View SE network distributed application.

For information about performing these operations, see the *FactoryTalk View Site Edition Installation Guide*.

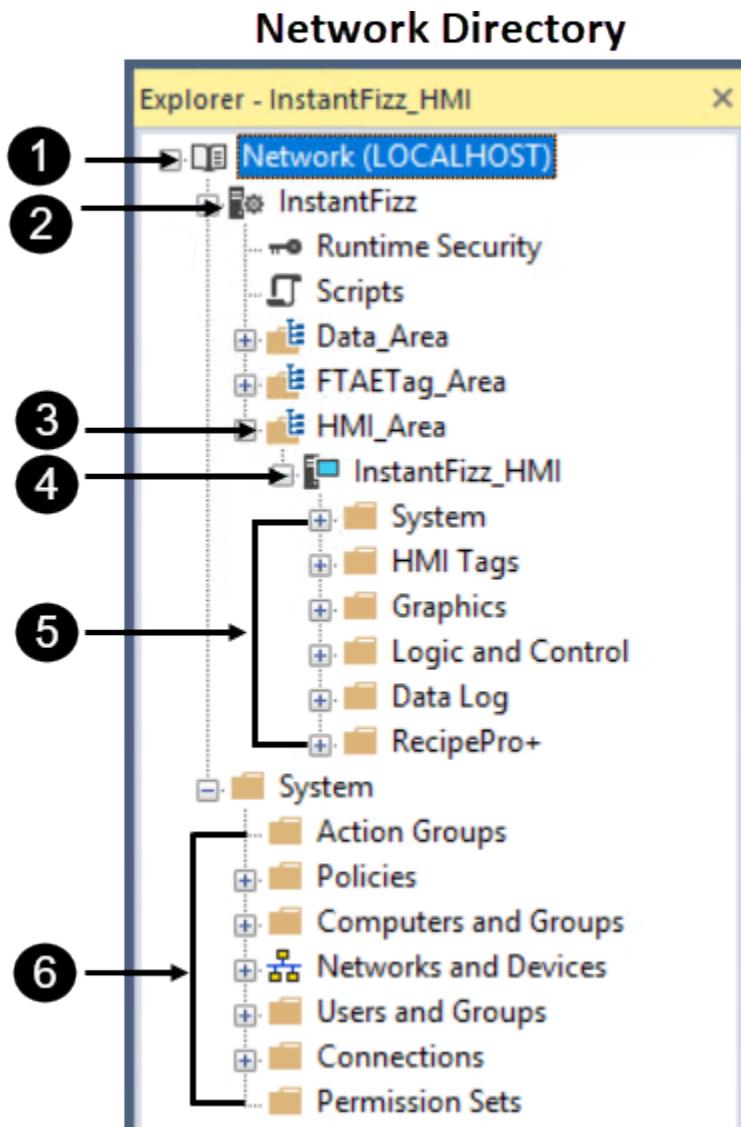
# Working with network station applications

This chapter describes:

- What a FactoryTalk View Site Edition network station application is.
- Key network station application concepts.
- How to create a network station application.
- How to add areas and servers to a network station application.
- How to set up HMI server properties.
- How to monitor the status of an HMI server.
- How to delete HMI servers.
- How to rename and delete network station applications.

## About network station applications

In FactoryTalk View Studio, you can create FactoryTalk View Site Edition local station, network station, and network distributed applications. The following illustration shows what a FactoryTalk View SE network station application looks like in the FactoryTalk View Studio Explorer window:



Item	Description
1	Host computer
2	Application
3	Area
4	HMI server
5	HMI project components and editors
6	FactoryTalk system settings

A network station application expands the capability of local station applications for security and data access. Use a network station application, when you only require a standalone application, but still have the need to:

- Access network-scoped products like FactoryTalk Historian ME, FactoryTalk Historian SE, and FactoryTalk Batch.
- Participate in network-scoped FactoryTalk Security.
- Access a FactoryTalk Linx data server on a remote computer that can be configured for redundancy.

Network distributed applications are described in [Working with network distributed applications on page 114](#). Local station applications are described in [Working with local station applications on page 142](#).

## Parts of a network station application

A typical FactoryTalk View SE network station application consists of:

- One or more areas that divide the application into manageable parts or organize it in a way that makes sense for the process it is controlling. Keep in mind that only one HMI Server can be added to a FactoryTalk View SE network station application.
- An HMI server that provides FactoryTalk View components and services to application clients. A network station application can contain only one HMI server.
- HMI project components such as graphic displays, HMI tags, and data log models.
- One or more data servers, with or without the use of HMI tags

Clients communicate through data servers, which provide access to information in devices and other data servers that comply with the OPC DA 2.05a, UA 1.02 or 1.03 specifications.

In a network station application, you can use multiple FactoryTalk Linx and OPC data servers (including RSLinx Classic), running on different computers. You can also set up a redundant pair of host computers for the data servers in the application. For information about the requirements that apply to redundant data servers, see the product documentation.

FactoryTalk Linx servers can be set up to subscribe to device-based alarms detected in Studio 5000 controllers. For more information, see [Setting up FactoryTalk alarms on page 177](#).



**Tip:** Every vendor's OPC data server is different. Some contain their own tag databases, like the tag database in an HMI server, while others reference the tag databases or addresses that exist in controllers, as is the case with FactoryTalk Linx and Studio 5000 controllers.

- A list of users, plus the security codes that allow or deny these users permission to access secured HMI project components at runtime.

- Optionally, one or more FactoryTalk Tag Alarm and Event Servers, to provide alarm monitoring and control for tags in devices that do not have built-in alarm detection. For more information, see [Setting up FactoryTalk alarms on page 177](#).



**Tip:** Up to 20 network station applications can be configured in one FactoryTalk Directory.

The HMI server and the HMI client are the software programs behind a network station application that must be on the same computer. The FactoryTalk Directory Server, the FactoryTalk Linx data server, and the OPC data servers can be on remote computers.

## About FactoryTalk systems

FactoryTalk View SE and other Rockwell Automation software products use a set of common FactoryTalk services to support certain functions, such as diagnostic messages and access to real-time data.

These services, including FactoryTalk Directory, are installed with the FactoryTalk Services Platform when you install FactoryTalk View SE. For an overview of FactoryTalk services, see [FactoryTalk Services Platform on page 31](#).

An automation and control system that uses FactoryTalk services, and integrates FactoryTalk products and components, is known as a FactoryTalk system. A complete FactoryTalk system consists of all the networks, devices, and software applications you have deployed to monitor and control your plant or process. Helping to ensure that the system can provide data in a secure and predictable fashion depends on a number of variables.

## Finding more information about FactoryTalk services

This manual contains information about developing FactoryTalk View SE applications, including information about how FactoryTalk View uses FactoryTalk services.

For additional, detailed information about FactoryTalk systems, services, concepts, and components, see the FactoryTalk Help.

## Key network station application concepts

This section presents some of the common terms and concepts used to describe FactoryTalk View SE network station applications.

## FactoryTalk Network Directory

FactoryTalk Directory centralizes access to application resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed applications, and network station applications. All of the client and server computers participating in a given network distributed application, or network station application must point at the same Network Directory.

For more information about FactoryTalk Network Directory, see [Setting up the FactoryTalk Directory on page 70](#).

**NOTE:**

Do not install FactoryTalk Directory, FactoryTalk View SE Server, or any other application software on the same computer as the Windows domain controller. This configuration is not supported.

## FactoryTalk Security

FactoryTalk View SE applications can use FactoryTalk Security services to authenticate and authorize application users.

During FactoryTalk View SE installation, Windows users with administrative privileges on the computer are set up with full, initial access to FactoryTalk View SE applications managed by a FactoryTalk Local or Network Directory on the same computer.

In FactoryTalk View Studio, you can create FactoryTalk user and group accounts, and then determine which accounts have access to resources such as the Local Directory, or the application.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#). For details, see FactoryTalk Services Platform Help.

## About FactoryTalk Security permissions

If FactoryTalk Security services are used to secure parts of an application, to perform certain tasks, users must have the necessary security permissions.

For example, to create or modify the properties of an application, you must at least be allowed the Common actions Read, List Children, Write, and Create Children, at the FactoryTalk Directory that manages the application.

If you receive a FactoryTalk Security message while trying to perform such a task, contact your system administrator about the permissions you might require.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#).

## HMI servers (network station)

HMI servers are software programs that supply information to clients as they request it.

An HMI server stores HMI project components such as graphic displays, and serves these components to clients. An HMI server also manages a database of tags and logs historical data.

In FactoryTalk View Studio, first you create a network station application, and then you add one (and only one) HMI server to the application.

**NOTE:** A FactoryTalk View SE network station application must have one and only one HMI server.

You can add the HMI server in the root area of the application (the icon and application name in the Explorer window) or within the areas you have created.

- For information about adding an HMI server to an application, see [Adding an HMI server on page 137](#).

## HMI projects (network station)

HMI projects contain graphic displays, data log models, HMI tags, and other services. Network station applications can contain only one HMI project.

## HMI clients

HMI clients are software programs that obtain information from, or write information to HMI servers or data servers. FactoryTalk View Studio, the FactoryTalk View SE Administration Console, and the FactoryTalk View SE Client are all HMI clients.

## Areas (network station)

All FactoryTalk View applications have one system-defined area called the application root area, which has the same name as the application. The application root area can contain one HMI server and one or more OPC data servers. It is a system best practice to only have one FactoryTalk Linx data server in the root area or individual area.

---

**NOTE:** A FactoryTalk View SE network station application must have one and only one HMI server.

You can add the HMI server in the root area of the application (the icon and application name in the Explorer window) or within the areas you have created.

---

### About the home area

In an application, the area that contains a given application component, such as a graphic display, is called the home area.

When you refer to an application component without specifying the area, FactoryTalk View SE uses the home area to locate the component.

For example, if an object in a graphic display refers to a tag without specifying an area, FactoryTalk View assumes that the tag and the display are in the same home area.

If the tag cannot be found in an HMI server or a data server in the display's home area, an error is logged when the display is run.

## Relative references (network station)

You can have both referencing types in a network station application when using Areas. In a network station application, you use relative references to refer to application components, such as graphic displays and tags, when building FactoryTalk View commands or connecting graphic objects to process data.

Relative references point at a component relative to the current application.

For example, a relative reference to a display named Detail is simply the display's name: Detail. To set up a button in a graphic display to open the Detail display at runtime, use

Display Detail

as the button's press action.

## System availability

To minimize data loss and down time, and to help ensure that critical parts of your system are always available to connected clients, FactoryTalk View SE provides the health monitoring and redundancy features.

For information about these and other availability features, see [Setting up FactoryTalk system availability on page 232](#).



**Tip:** You can set up HMI redundancy in network distributed applications only. You can set up Data Server and FTAE Redundancy in network distributed applications and network station applications, but not in local station applications.

## Language switching

To make user-defined text strings in an application available at runtime in up to 40 different languages, set up language switching for the application.

For more information, see [Setting up language switching on page 210](#).

## Creating network station applications

To create a network station application, in FactoryTalk View Studio, first you create the application, and then you add elements such as areas, HMI server, data servers, and FactoryTalk Tag Alarm and Event Servers.

### To create a network station application:

1. Select Start > FactoryTalk View Studio.
2. In the **Application Type Selection** dialog box, select the application type and click **Continue**.
3. In the **New/Open Site Edition Application** dialog box, click the **New** tab.
4. Type the name and description, select a default language, and click **Create**.
5. (optional) To change the default folder of the HMI projects, use the HMI Projects Folder Path Config Utility to specify a new folder.

## Adding areas and servers to a network station application

A network station application can consist of one or more areas, one HMI server, and, if necessary, multiple data servers and FactoryTalk Tag Alarm and Event Servers. You add these elements to the application, after you create it.

## Adding and deleting areas

You can add and delete areas in an application. You cannot copy areas.

### To add an area:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root or right-click an area name, and then click **New Area**.
2. In the New Area dialog box, type a name for the area, and an optional description, and then click **OK**.

### To delete an area:

Right-click the area that you want to delete, and then click **Delete**.

When you delete an area, HMI servers and data servers in the area are not deleted from disk.

## Adding an HMI server (network station)

After you create a network station application, you must add one HMI server, either to the application's root area, or to the area you have added.

After adding the HMI server, you can set up its properties, and then use editors in the Explorer window to create HMI project components, such as graphic displays, HMI tags, and data log models.

A network station application can contain one and only one HMI server.

### To add an HMI server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application or an area name, select **Add New Server**, and then select **HMI Server**.
2. In the **Add HMI Server Wizard**, in the **Select Operation** window, click **Create a new HMI server**.
3. (optional) To add the Application Libraries, click the **Add Application Library** dropdown list, and then select one or multiple libraries.
4. Click **Next**.
5. Type a name and description for the HMI server, specify the host computer if applicable, and click **Finish**.
6. (optional) To change the default folder of the HMI projects, use the HMI Projects Folder Path Config Utility to specify a new folder.

The **Select Operations** window contains the following options:

- **Create a new HMI server.** When you create an HMI server, the server's HMI project is created automatically.
  - **Add Application Library** to add one or multiple Application Libraries.
- **Copy an HMI server** that already exists. After you have copied an HMI server, changes made to the original do not affect the copy, or vice versa.
- **Import a project** from RSView32, FactoryTalk View SE, or FactoryTalk View Machine Edition, as the basis for the new HMI server. After you have copied a project, changes made to the original do not affect the copy, or vice versa.
- **Attach to an existing HMI server** without making a copy of the HMI server.

---

**NOTE:** You cannot attach to an existing HMI server that is being used in another application.

---

## Adding a data server

The application root area can contain one HMI server and one or more OPC Data servers. It is a system best practice to only have one FactoryTalk Linx data server in the root area or individual area.

**To add a FactoryTalk Linx data server:**

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root or right-click the area name, select **Add New Server > Rockwell Automation Device Server (FactoryTalk Linx)**.
2. Set up server properties, such as the name and location of the server, whether to provide redundancy using a secondary server, and whether the server will support alarms and events.  
For details, see the FactoryTalk Linx product documentation.

**To add an OPC data server:**

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root, or right-click the area name, select **Add New Server > OPC DA Server** or **OPC UA Server**.
2. Set up properties as needed. For details, see the product documentation.

For additional information about setting up FactoryTalk Linx and OPC data servers, see [Setting up communications on page 149](#).

## Adding a Tag Alarm and Event Server

Optionally, you can add FactoryTalk Tag Alarm and Event Servers to an application.

Tag Alarm and Event Servers use FactoryTalk Alarms and Events services to monitor and control alarms for tags in programmable controllers (PLC-5 or SLC 500) and other devices that do not have built-in alarm detection.

For more information about Tag Alarm and Event Servers, see [Setting up FactoryTalk alarms on page 177](#).

## Providing HMI server names

The FactoryTalk Directory stores HMI server names and does not delete them, even if you remove an HMI server from all the applications in which it is used, and then delete the HMI server's project files.

This means that every HMI server must have a unique name:

- You cannot reuse an HMI server's name, even after you delete the HMI server from an application.
- If multiple users share the same FactoryTalk Directory, the names of all HMI servers created by all users must be unique.

Similarly, if multiple applications share the same FactoryTalk Directory, the name of each HMI server in each application must be unique.

## Monitoring disk space on HMI servers

After deploying a FactoryTalk View SE network distributed application, monitor disk space on computers running HMI servers to make sure that it does not fall below 500 MB.

**To monitor disk space:**

1. In the HMI server's Events editor, create an event that uses the free\_bytes function to return the number of free bytes available on the HMI server's hard drive.
2. Create an alarm of FactoryTalk Alarms and Events and set up the alarm conditions for the disk space.

For information about setting up events, see [Adding logic and control on page 423](#).

## Setting up HMI server properties (network station)

After adding an HMI server to a network station application, use the Properties dialog box to:

- Select the components that will run when the HMI server starts up.
- Specify a macro to run when the HMI server shuts down.

### To open the HMI Server Properties dialog box:

In FactoryTalk View Studio or the SE Administration Console, in the Explorer window, right-click the HMI server icon, and then select **Properties**.

For details about options in the HMI Server Properties dialog box, click **Help**.

## Starting and stopping HMI server components manually (network station)

For the HMI server in a network station application, you can start and stop the server's components manually.

### To start HMI server components manually:

1. In FactoryTalk View Studio or the SE Administration Console, open the network station application.
2. In the Explorer window, right-click the HMI server, and then select **Properties**.
3. In the Components tab, click **Run Startup Components**.

The startup components selected for the HMI server should start running.

### To stop HMI server components manually:

1. In FactoryTalk View Studio, open the network station application.
2. In the Explorer window, right-click the HMI server, and then select **Properties**.
3. In the Components tab, click **Stop All Running Components**.

All the components running on the computer will stop running, including alarms, data log models, derived tag components, and event components. If a shutdown macro is specified in the Components tab, the specified macro will also run.

## Monitoring the status of an HMI server

In FactoryTalk View Studio, in the Server Status dialog box, you can view the operational status of an HMI server to determine whether the server is ready to provide service, and to troubleshoot problems.

For example, an HMI server in the Active state is fully loaded and ready to provide service to connected clients. A client can be any of FactoryTalk View Studio, a FactoryTalk View SE Client, or the FactoryTalk View SE Administration Console.

For more information about server states, see [Monitoring the status of application servers on page 233](#).

### To open the Server Status dialog box:

In FactoryTalk View Studio, in the Explorer window, right-click the HMI server's icon, and then select **Server Status**.

For details about options in the Server Status dialog box, click **Help**.

## Deleting HMI servers

You can delete an HMI server from an application without deleting the HMI project files, and then add the HMI server to the application later.

You can also add the HMI server temporarily, while you copy components to other HMI servers in the application.

### To delete an HMI server:

1. Disconnect all clients from the HMI server. (A client is any of FactoryTalk View Studio, FactoryTalk View SE Administration Console, or the FactoryTalk View SE Client.)
2. Start FactoryTalk View Studio, and then open the application that contains the HMI server you want to delete.
3. Right-click the HMI server, and then select **Delete**.

This does not delete the HMI server's project files.

## Deleting HMI server project files

After you delete an HMI server from an application, if you want to delete the project files, you have to delete them separately.

### To delete an HMI server's project files:

1. Disconnect all clients from the HMI server. (A client is any of FactoryTalk View Studio, the FactoryTalk View SE Administration Console, or the FactoryTalk View SE Client.)
2. In Windows Explorer, browse to the HMI projects folder on the computer hosting the HMI server. The default path is C:\Users\Public\Documents\RSView Enterprise\SE\HMI projects.
3. In the HMI projects folder, delete the subfolder that has the same name as the HMI server you deleted from the application.

For example, if you deleted an HMI server named Water, then you will delete an HMI project folder also named Water.

## Renaming and deleting network station applications

Use the FactoryTalk View SE Application Manager tool to rename or delete a network station application.



**Tip:** You cannot rename or delete a network station application that is in use. Ensure that all users disconnect from the application first.

### To rename or delete the application

1. Select **Start > Rockwell Software > FactoryTalk View SE Application Manager**.
2. In the **Common Operations** tab, select your application and follow the wizard.

When you delete a network station application, the HMI servers and data servers set up in it are not deleted automatically. For details about:

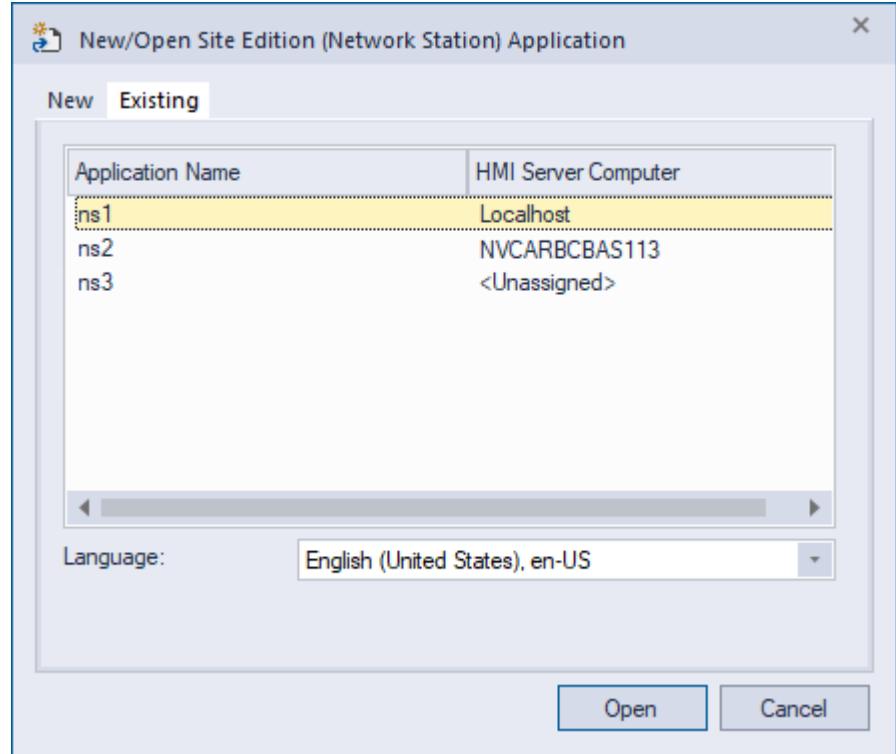
- Deleting the HMI server files, see [Deleting HMI server project files on page 130](#).
- Removing a data server and deleting its cache files, see [Deleting an OPC data server on page 157](#).

## Determining where the application is located

You can browse the properties of any Network Station application from any computer in FactoryTalk Admin Console. And you cannot distinguish between Network Distributed and Network Station applications in FactoryTalk Admin Console.

### To determine the host computer for the application:

1. Select Start > FactoryTalk View Studio.
2. In the Application Type Selection dialog box, select **View Site Edition (Network Station)**, and then click **Continue**.
3. In the New/Open Site Edition (Network Station) Application dialog box, click the **Existing** tab.



The value in the HMI Server Computer column reflects the status of the HMI server added to that application. Status values include:

- Localhost - the HMI server is on the current computer.
- computer name - the HMI server is on the computer named (for example, NVCARBCBAS113).
- <Unassigned> - no HMI server has been added to the application.

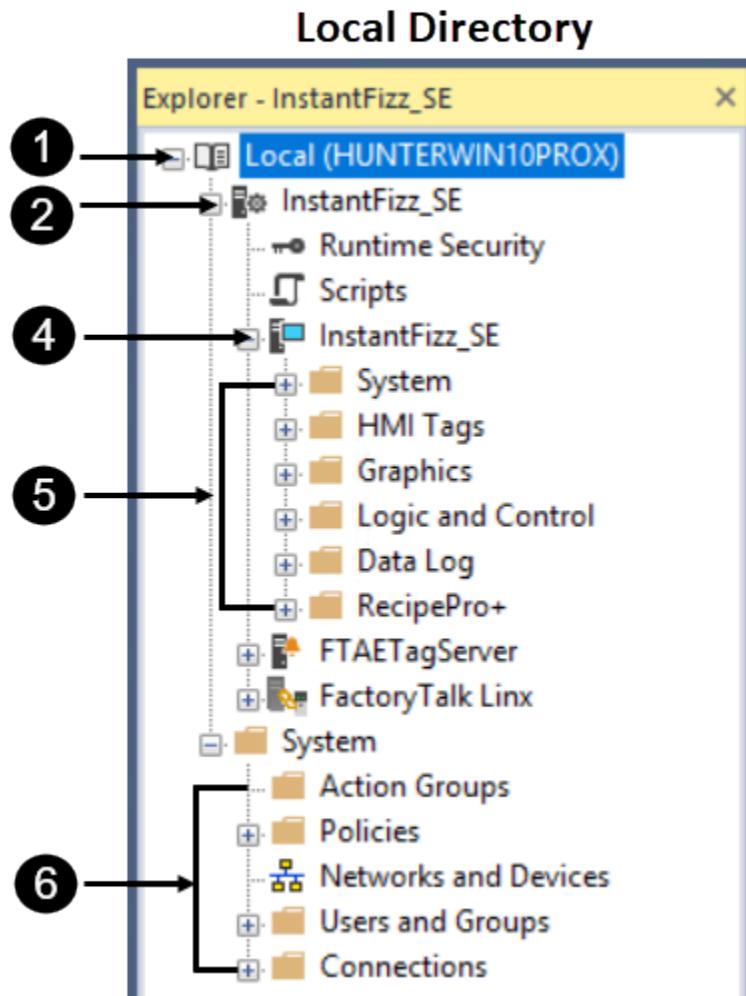
# Working with local station applications

This chapter describes:

- What a FactoryTalk View Site Edition local station application is.
- Key local station application concepts.
- How to create a local station application.
- How to add servers to a local station application.
- Setting up HMI server properties.
- Monitoring the status of an HMI server.
- Renaming, deleting, copying, and backing up local station applications.

## About local station applications

In FactoryTalk View Studio, you can create FactoryTalk View Site Edition network and local station applications. The following illustration shows what a FactoryTalk View SE local station application looks like in the FactoryTalk View Studio Explorer window:



Item	Description
1	Host computer
2	Application

Item	Description
4	HMI server
5	HMI project components and editors
6	FactoryTalk system settings

Network distributed applications are described in [Working with network distributed applications on page 114](#). Network station applications are described in Chapter 7, *Working with network station applications*.

## Parts of a local station application

A typical FactoryTalk View SE local station application consists of:

- The application root area. You cannot add areas to a local station application.
- An HMI server that provides FactoryTalk View components and services to application clients.  
A local station application can contain only one HMI server. The HMI server is created automatically when the application is created, and has the same name as the application.
- HMI project components such as graphic displays, HMI tags, and data log models.
- One or more data servers, with or without the use of HMI tags  
Clients communicate through data servers, which provide access to information in devices and other data servers that comply with the OPC DA 2.05a, UA 1.02 or 1.03 specifications.  
Local station applications support a single FactoryTalk Linx data server, which must run on the same computer as the application. Only OPC data servers (including RSLinx Classic) can be on other computers.  
FactoryTalk Linx servers can be set up to subscribe to device-based alarms detected in Studio 5000 controllers. For more information, see [Setting up FactoryTalk alarms on page 177](#).



**Tip:** Every vendor's OPC data server is different. Some contain their own tag databases, like the tag database in an HMI server, while others reference the tag databases or addresses that exist in controllers, as is the case with FactoryTalk Linx and Studio 5000 controllers.

- A list of users, plus the security codes that allow or deny these users permission to access secured HMI project components at runtime.
- Optionally, a FactoryTalk Tag Alarm and Event Server, to provide alarm monitoring and control for tags in devices that do not have built-in alarm detection. For more information, see [Setting up FactoryTalk alarms on page 177](#).

The software programs behind a local station application—the FactoryTalk Local Directory, the HMI server, the HMI client, and the FactoryTalk Linx data server—must be on the same computer. Only OPC data servers can reside on remote computers.

The location of the FactoryTalk Local Directory that manages a local station application is set up automatically when you install FactoryTalk View Site Edition.

## About FactoryTalk systems

FactoryTalk View SE and other Rockwell Automation software products use a set of common FactoryTalk services to support certain functions, such as diagnostic messages and access to real-time data.

These services, including FactoryTalk Directory, are installed with the FactoryTalk Services Platform when you install FactoryTalk View SE. For an overview of FactoryTalk services, see [FactoryTalk Services Platform on page 31](#).

An automation and control system that uses FactoryTalk services, and integrates FactoryTalk products and components, is known as a FactoryTalk system. A complete FactoryTalk system consists of all the networks, devices, and software applications you have deployed to monitor and control your plant or process. Helping to ensure that the system can provide data in a secure and predictable fashion depends on a number of variables.

## Finding more information about FactoryTalk services

This manual contains information about developing FactoryTalk View SE applications, including information about how FactoryTalk View uses FactoryTalk services.

For additional, detailed information about FactoryTalk systems, services, concepts, and components, see the FactoryTalk Help.

## Key local station application concepts

This section presents some of the common terms and concepts used to describe FactoryTalk View SE local station applications.

### FactoryTalk Local Directory

FactoryTalk Directory centralizes access to application resources and components, such as graphic displays and tags, for all FactoryTalk products participating in a control system.

FactoryTalk Local Directory (also called the Local Directory) manages applications that are confined to a single computer, for example, FactoryTalk View SE local station applications. The Local Directory must reside on the same computer as the local station application.

For more information, see [Setting up the FactoryTalk Directory on page 70](#).

---

**NOTE:**

Do not install FactoryTalk Directory, FactoryTalk View SE Server, or any other application software on the same computer as the Windows domain controller. This configuration is not supported.

---

## FactoryTalk Security

FactoryTalk View SE applications can use FactoryTalk Security services to authenticate and authorize application users.

During FactoryTalk View SE installation, Windows users with administrative privileges on the computer are set up with full, initial access to FactoryTalk View SE applications managed by a FactoryTalk Local or Network Directory on the same computer.

In FactoryTalk View Studio, you can create FactoryTalk user and group accounts, and then determine which accounts have access to resources such as the Local Directory, or the application.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#). For details, see FactoryTalk Services Platform Help.

## About FactoryTalk Security permissions

If FactoryTalk Security services are used to secure parts of an application, to perform certain tasks, users must have the necessary security permissions.

For example, to create or modify the properties of an application, you must at least be allowed the Common actions Read, List Children, Write, and Create Children, at the FactoryTalk Directory that manages the application.

If you receive a FactoryTalk Security message while trying to perform such a task, contact your system administrator about the permissions you might require.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#).

## HMI servers (local station)

HMI servers are software programs that supply information to clients as they request it.

An HMI server stores HMI project components such as graphic displays, and serves these components to clients. An HMI server also manages a database of tags and logs historical data.

In FactoryTalk View Studio, when you create a local station application, the HMI server is created automatically, and given the same name as the application.

A local station application can only contain a single HMI server.

## HMI projects (local station)

HMI projects contain graphic displays, data log models, HMI tags, and other services.

The HMI project is created with the HMI server, when you create a local station application. The HMI project is loaded by the HMI server.

## HMI clients

HMI clients are software programs that obtain information from, or write information to HMI servers or data servers. FactoryTalk View Studio, the FactoryTalk View SE Administration Console, and the FactoryTalk View SE Client are all HMI clients.

## Relative references (local station)

In a local station application, you use relative references to refer to application components, such as graphic displays and tags, when building FactoryTalk View commands or connecting graphic objects to process data.

Relative references point at a component relative to the current application.

For example, a relative reference to a display named Detail is simply the display's name: Detail. To set up a button in a graphic display to open the Detail display at runtime, use

Display Detail

as the button's press action.

## Language switching

To make user-defined text strings in an application available at runtime in up to 40 different languages, set up language switching for the application.

For more information, see [Setting up language switching on page 210](#).

## Creating local station applications

To create a local station application, in FactoryTalk View Studio, first you create the application, and then you add elements such as a FactoryTalk Linx server, a FactoryTalk Tag Alarm and Event Server, and OPC data servers.

### To create a local station application:

1. Select **Start > FactoryTalk View Studio**.
2. In the **Application Type Selection** dialog box, select the application type and click **Continue**.
3. In the **New/Open Site Edition Application** dialog box, click the **New** tab.
4. Type the name and description, select a default language, and click **Create**.
5. (optional) To change the default folder of the HMI projects, use the HMI Projects Folder Path Config Utility to specify a new folder.

The application icon and name are shown in the Explorer window, beneath the Local Directory icon. When you create a local station application, the HMI server is given the same name as the application.

## Importing a project into a new application

You can create a new local station application by importing a project from RSView32, FactoryTalk View SE, or FactoryTalk View Machine Edition.

After you import an HMI project, the changes you make to the original project do not affect the copy, or vice versa.

You cannot import a project into an existing local station application.

For details about importing projects, see the FactoryTalk View Site Edition Help.

## Adding servers to a local station application

local station applications can contain only one HMI server, and one FactoryTalk Linx data server. However, you can add more than one OPC data server, or FactoryTalk Tag Alarm and Event Servers. You add these elements to the application, after you create it.

### To add a FactoryTalk Linx server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root or an area name, select **Add New Server > Rockwell Automation Device Server (FactoryTalk Linx)**.
2. In the FactoryTalk Linx Properties dialog box, select options and provide information in each tab, as described in the sections that follow.
3. When you are finished, click **OK**.



**Tip:** The location is set to *localhost* automatically, and cannot be changed. Local station applications support a single FactoryTalk Linx data server, which must run on the same computer as the application.

For details, see the FactoryTalk Linx product documentation.

### To add an OPC data server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root, or right-click the area name, select **Add New Server > OPC DA Server** or **OPC UA Server**.
2. Set up properties as needed. For details, see the product documentation.

For additional information about setting up FactoryTalk Linx and OPC data servers, see [Setting up communications on page 149](#).

## Adding a Tag Alarm and Event Server

Optionally, you can add FactoryTalk Tag Alarm and Event Servers to an application.

Tag Alarm and Event Servers use FactoryTalk Alarms and Events services to monitor and control alarms for tags in programmable controllers (PLC-5 or SLC 500) and other devices that do not have built-in alarm detection.

For more information about Tag Alarm and Event Servers, see [Setting up FactoryTalk alarms on page 177](#).

## Setting up HMI server properties (local station)

After creating a local station application, use the HMI Server Properties dialog box to:

- Add a description of the HMI server.
- Select the components that will run when the HMI server starts up.
- Specify a macro that will run when the HMI server shuts down.
- Start or stop components manually.



**Tip:** You do not have to specify a startup type for the HMI server in a local station application.

The HMI server always starts automatically, when a FactoryTalk View SE Client connects to the application.

### To open the HMI Server Properties dialog box:

In FactoryTalk View Studio, in the Explorer window, right-click the HMI server icon, and then select **Properties**.

For details about the options in the HMI Server Properties dialog box, click **Help**.

## Selecting startup and shutdown components (local station)

In the Components tab of the HMI Server Properties dialog box, select the checkbox for each of the items you want to run when the HMI server runs, including:

- Which components will run when the HMI server starts.
- Which macro will run when the HMI server shuts down.

---

**NOTE:** The order of items in the Components tab is not the order in which the items will run when the HMI server starts up. If components must run in a particular order, create a macro to start the components, and then select the macro to run when the HMI server starts.

---

## Monitoring the status of an HMI server

In FactoryTalk View Studio, in the Server Status dialog box, you can view the operational status of an HMI server to determine whether the server is ready to provide service, and to troubleshoot problems.

For example, an HMI server in the Active state is fully loaded and ready to provide service to connected clients. A client can be any of FactoryTalk View Studio, a FactoryTalk View SE Client, or the FactoryTalk View SE Administration Console.

For more information about server states, see [Monitoring the status of application servers on page 233](#).

### To open the Server Status dialog box:

In FactoryTalk View Studio, in the Explorer window, right-click the HMI server's icon, and then select **Server Status**.

For details about options in the Server Status dialog box, click **Help**.

## Renaming, deleting, and copying local station applications

Use the FactoryTalk View SE Application Manager tool to rename, delete, or copy a local station application.

### To rename, delete, or copy the application

1. Select **Start > Rockwell Software > FactoryTalk View SE Application Manager**.
2. In the **Common Operations** tab, select your application and follow the wizard.

# Setting up communications

This chapter describes:

- Data servers and FactoryTalk Live Data.
- Basic steps for setting up communications.
- Adding FactoryTalk Linx data servers to an application.
- Setting up communications in FactoryTalk Linx.
- Adding OPC (Open Platform Communications) data servers to an application.

## About data servers

Data servers provide access to devices, making it possible to browse, read, and write values from FactoryTalk View applications.

FactoryTalk View SE supports the following types of data servers:

- Rockwell Automation Device Servers FactoryTalk Linx is the recommended data server for all Rockwell Automation devices and networks.  
FactoryTalk Linx does not require activation or licensing and can be installed as often as needed in any application.  
FactoryTalk Linx also provides service for device-based alarms and events. For more information, see [Setting up FactoryTalk alarms on page 177](#).
- OPC data servers are any data server that supports the OPC DA 2.05a, UA 1.02 and 1.03 specifications.

The OPC Foundation is a non-profit organization that provides standards and technology that enables software from different vendors to work together.

The following OPC Data DA v2.05a servers are available with FactoryTalk View SE.

- **KEPServer Enterprise** is used to communicate with non-Rockwell Automation devices like Siemens or Modicon controllers.  
KEPServer Enterprise requires activation or licensing.
- **RSLinx Classic** should be used if CIP to non-CIP routing is required to reach the device.



**Tip:** If RSLinx Classic runs on the same computer as FactoryTalk View SE Server, RSLinx Classic does not need an activation key.

RSLinx Classic Gateway can run on any computer, but requires an activation.

## About FactoryTalk Live Data

FactoryTalk Live Data manages connections between data servers in an application and FactoryTalk Live Data clients such as FactoryTalk View SE clients.

FactoryTalk Live Data provides data using the web and security-friendly TCP/IP protocol and enables faster disconnect and reconnect.

FactoryTalk View SE has an HMI Tag database, which also uses FactoryTalk Live Data to communicate with FactoryTalk View SE clients.

FactoryTalk Live Data is one of the common services installed with the FactoryTalk Services Platform. For information about other FactoryTalk services, see [FactoryTalk Services Platform on page 31](#).

## Using multiple data servers in an application

A FactoryTalk View SE application can use multiple data servers to:

- Provide redundancy.  
Redundancy is only supported in network applications. To provide redundancy, set up a secondary data server on another computer. On the primary data server's computer, change the settings in the Redundancy tab of the server's Properties dialog box, to switch clients to the secondary server if the primary one fails.
- Communicate with non-Rockwell Automation controllers.  
For example, (1) Data server - FactoryTalk Linx, communicating with Logix Controllers over EtherNet/IP. (2) OPC Data Server - KEPServer Enterprise communicating with a Siemens S7-300 over PROFINET.
- Balance loading.  
If CPU or memory usage is high on a computer running one data server, create another data server on another computer to balance the processing load.

Network applications can use multiple FactoryTalk Linx and OPC data servers, on different computers on the network.

Local applications can only use one FactoryTalk Linx server, which must be on the same computer as the application.



**Tip:** OPC data servers can be on different computers; however, it is not recommended.

## Communicating with multiple controllers

FactoryTalk Linx is designed to serve multiple controllers across multiple networks simultaneously. FactoryTalk Linx recovers from program downloads, disconnects, power-offs, and online changes without affecting service to other controllers.

KEPServer Enterprise can also serve multiple controllers across multiple networks simultaneously.

## Setting up communications in FactoryTalk View SE

These are the basic tasks involved in setting up communications for a FactoryTalk View SE application:

1. Gather information about the network, and the devices that are connected to it. Decide which of the following data servers will be used:
  - Rockwell Automation Device Server (FactoryTalk Linx)
  - Other OPC data servers that comply with the OPC DA 2.05a, UA 1.02 or 1.03 specifications.



**Tip:** FactoryTalk Linx is the recommended server for FactoryTalk View applications. For more information about choosing a data server type, see [About data servers on page 149](#).

2. Install and set up the data servers.
    - FactoryTalk Linx is installed automatically with FactoryTalk View SE. Add a Rockwell Automation Device Server (FactoryTalk Linx) to the FactoryTalk View SE application and set up the server's properties. For more information, see [Adding FactoryTalk Linx data servers to an application on page 151](#).
    - RSLinx Classic and KEPServer Enterprise must be installed and setup separately. Add an OPC data server to the FactoryTalk View SE application, and then set up the server's properties. For more information, see [Adding OPC data servers to an application on page 154](#).

For information about installing FactoryTalk Linx or RSLinx Classic, see the FactoryTalk View Site Edition Installation Guide.
  3. Once the application can communicate with the devices, determine how the application will gain access to values in the devices. You can reference tags in a data server directly, create HMI tags that use the devices as data sources, or use both methods.
- For information about using tags in an application, see [Working with tags on page 159](#).
4. Wherever you need to obtain tag values in the application, create connections in FactoryTalk View SE graphic displays.
- For example, to let an operator read and write values for a particular tag, create an input object in a display, and then connect the object to the tag's name or address.
- For information about supplying tag names for graphic objects, see [Working with tags on page 159](#) and [Creating graphic objects on page 288](#).

## About FactoryTalk Security permissions

If FactoryTalk Security services are used to secure parts of an application, to perform certain tasks, users must have the necessary security permissions.

For example, to create or modify the properties of an application, you must at least be allowed the Common actions Read, List Children, Write, and Create Children, at the FactoryTalk Directory that manages the application.

If you receive a FactoryTalk Security message while trying to perform such a task, contact your system administrator about the permissions you might require.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#).

## Adding FactoryTalk Linx data servers to an application

Network applications can use multiple FactoryTalk Linx and OPC data servers, on different computers on the network.

Local applications can only use one FactoryTalk Linx server, which must be on the same computer as the application.

### To add a FactoryTalk Linx data server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root or an area name, select **Add New Server > Rockwell Automation Device Server (FactoryTalk Linx)**.
2. In the FactoryTalk Linx Properties dialog box, select options and provide information in each tab, as described in the sections that follow.
3. When you are finished, click **OK**.

## Setting up FactoryTalk Linx data server general properties

In the **General** tab of the **FactoryTalk Linx Server Properties** dialog box, type a name and description for the data server, and specify the name of the computer that will host the data server.

### Name

Type a name for the data server. This name will be shown in the Explorer window. The name cannot include dashes or hyphens ( - ).

### Description

Type a description for the data server. For example, the description can consist of the server's location, the name of a contact person or number to contact in case of failure, or version information.

### Computer hosting the FactoryTalk Linx server

Type the name of the computer where the FactoryTalk Linx data server is running. To find and select a computer, click the **Browse** button.

You can change the name of the computer hosting the data server, only in a network application. In a local application, you can only add one FactoryTalk Linx data server, which must be on the same computer as the application.

### Server Type

Select one of the two FactoryTalk Linx data servers: instance01 and instance02.

Instance01 is the base server, which supports all the functions of FactoryTalk Linx. Instance02 is designed to allow applications to have more tags, data clients (for example, HMI clients), and/or controllers without affecting instance01.

## Setting up FactoryTalk Linx data server redundancy

To minimize disruptions to clients in the event that the FactoryTalk Linx server becomes unavailable or fails, set up redundancy for the data server. This option is only available for network applications.

In the **Redundancy** tab of the **Server Properties** dialog box, specify the name of the secondary (or backup) data server, and whether FactoryTalk View should switch back to the primary data server when it becomes available again.



#### Tip:

For a local application, there is no Redundancy tab in the OPC Data Server Properties dialog box. Local applications do not support data server redundancy.

### Provide redundancy using a secondary server

Select or clear this checkbox, depending on whether you want to set up redundancy for the data server.

### Computer running the secondary server

Type the name of the computer where the secondary data server is running. To find and select a computer, click the **Browse** button.

### Switch back options

To prevent FactoryTalk View SE from switching back to the primary data server when it becomes available, select the option, **Continue using the secondary server even when the primary becomes available again**.

To make FactoryTalk View SE switch back to the primary data server when it becomes available, select the option, **Switch over to the primary server when it becomes available again**.

## Enabling support for FactoryTalk Alarms and Events

To enable a FactoryTalk Linx server to use FactoryTalk Alarms and Events services to receive and send alarms detected in Studio 5000 controllers, in the Alarms and Events tab in the FactoryTalk Linx Server Properties dialog box, select the checkbox, **Enable alarm and event support**.

For more information, see [Setting up FactoryTalk alarms on page 177](#).

## Deleting a FactoryTalk Linx data server

If a FactoryTalk Linx data server is no longer required, you can delete it from the application.

### To delete a FactoryTalk Linx data server:

In FactoryTalk View Studio, in the Explorer window, right-click the FactoryTalk Linx data server you want to delete, and then click **Delete**.

## Setting up communications for FactoryTalk Linx

Use the FactoryTalk Linx Communication Setup editor to add or remove drivers and devices, set up driver and device properties, create device shortcuts, enable shortcuts to support FactoryTalk Alarms and Events services, or browse for tags in an offline tag file.

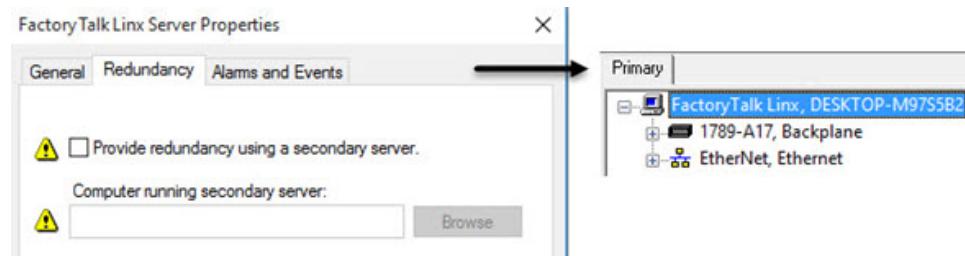
### To open the Communication Setup editor:

1. In FactoryTalk View Studio, in the Explorer window, click the + symbol beside the FactoryTalk Linx data server icon.
2. Double-click the **Communication Setup** icon.

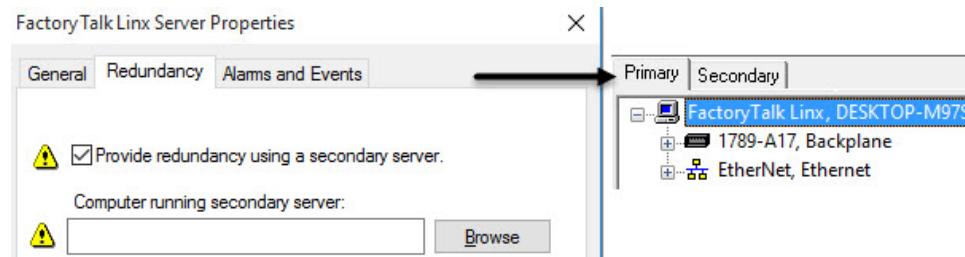
For details about options in the Communication Setup editor, click **Help**.

## About the Primary and Secondary tabs

If you have not set up redundancy for a FactoryTalk Linx data server, the Communication Setup editor will contain only a Primary tab.



In a network application, if you set up both a primary and a secondary FactoryTalk Linx data server, the Communication Setup editor has a Primary and a Secondary tab.



You can use these tabs to point FactoryTalk Linx at different networks or different hardware in case of failure.

---

**NOTE:** When specifying device shortcuts for a redundant server pair, be sure to use exactly the same shortcut names for the primary and secondary servers. If the names are not the same, tag references that use these shortcuts will not be able to obtain data reliably.

---

## Adding OPC data servers to an application

Add an OPC data server for use with third-party devices such as Modicon or Siemens.

You can add more than one OPC data server to a local or a network application. In a network application, an area can contain multiple OPC data servers.

For details about adding FactoryTalk Linx data servers, see [Using multiple data servers in an application on page 150](#).

### To add an OPC data server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root, or right-click the area name, select **Add New Server > OPC DA Server** or **OPC UA Server**.
2. Set up properties as needed. For details, see the product documentation.

## Setting up OPC data server general properties

### OPC DA Server

On the **General** tab of the **OPC Data Server Properties** dialog box, type a name and description for the data server, and specify a computer and an OPC server name.

#### Name

Type a name for the data server. This name will be shown in the Explorer window. The name cannot include dashes or hyphens ( - ).

#### Description

Type a description for the data server. For example, it can describe the server's location, the name of a contact person or number to contact in case of failure, or version information.

#### Computer that will run the OPC server

Type the name of the computer where the OPC server is running. To find and select a computer, click the **Browse** button.



**Tip:** The Browse button is available only if you select the option, **Server will be hosted on remote computer**.

#### OPC server name (Prog ID)

Type the programmatic ID of the OPC server. To browse for the programmatic ID, click the **Browse** button.

To use RSLinx Classic as the OPC server, click the **Browse** button, and then click **RSLinx Remote OPC Server**.



**Tip:** In a network application, always select **RSLinx Remote OPC Server**, instead of RSLinx OPC Server.

### OPC UA Server

On the **Connector Settings** tab of FactoryTalk Linx OPC UA Connector, type a name for the FactoryTalk Linx OPC UA Connector, and specify a computer hosting the FactoryTalk Linx OPC UA Connector.

#### Name

Type a name for the FactoryTalk Linx OPC UA Connector. The name must be unique and cannot be changed once it's created.

#### Computer hosting the FactoryTalk Linx OPC UA Connector

Enter the name of the computer hosting the FactoryTalk Linx OPC UA Connector. Only one Connector can run per host computer.

On the **OPC UA Servers** tab, configure settings and view the diagnostic information for each OPC UA Server that communicates with a FactoryTalk Linx OPC UA Connector.

#### OPC UA Servers list

Add a new OPC UA Server to the FactoryTalk Linx OPC UA Connector.

#### UA Server Properties

Select a server from the OPC UA Servers list to view or edit its properties. Some properties have predefined diagnostic item counters that can be used for monitoring. If needed, use Reset Counters command to reset diagnostic counter values to 0 for the server. Not all counters are resettable.

## Setting up OPC data server redundancy

To minimize disruptions to clients in the event that the data server becomes unavailable or fails, set up redundancy for the data server. This option is only available for network applications.

### OPC DA Server

On the **Redundancy** tab of the **Server Properties** dialog box, specify the name of the secondary (or backup) data server, and whether FactoryTalk View should switch back to the primary data server when it becomes available again.



**Tip:** For a local application, there is no Redundancy tab in the OPC Data Server Properties dialog box.

Local applications do not support data server redundancy.

#### Provide redundancy using a secondary server

Select or clear this checkbox, depending on whether you want to set up redundancy for the data server.

#### Computer running secondary server

Type the name of the computer where the secondary data server is running. To find and select a computer, click the **Browse** button.

#### Switch back options

To prevent FactoryTalk View SE from switching back to the primary data server when it becomes available, select the option, **Continue using the secondary server even when the primary becomes available again**.

To make FactoryTalk View SE switch back to the primary data server when it becomes available, select the option, **Switchback to the primary server when it becomes available**.

### OPC UA Server

On the **Redundancy** tab of FactoryTalk Linx OPC UA Connector, set up options for a secondary OPC UA server to use if the primary OPC UA server fails.

#### Secondary Server Setting

Identifies the server providing redundancy.

Under **Computer hosting the FactoryTalk Link OPC UA Connector**, select Browse to locate the server on the network or type the server name if the server has not been added to the network directory yet.

#### Switchover Options

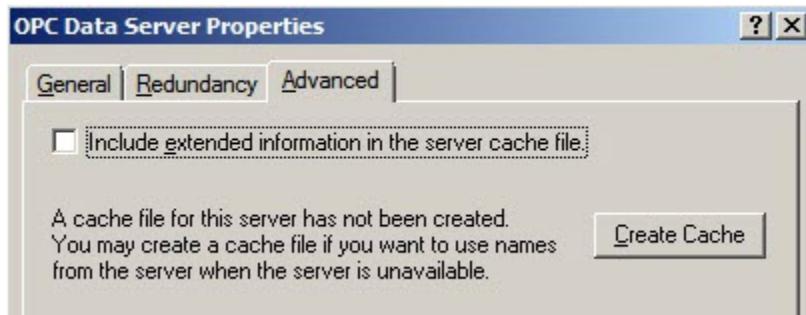
The FactoryTalk Linx OPC UA Connector has the option to **continue using secondary server even when the primary server becomes available again** to indicate that the system won't automatically switch to the primary server when it is restored.

### Server switchover

After selecting a server in the **Server Status** box, use the **Switch to Primary** and **Switch to Secondary** buttons to manually switch to the primary or secondary server.

## Setting up advanced properties

In the Advanced tab, set up a cache for tags on the data server. A cache lets you view tag names when you are not connected to the data server.



### Include extended information in the server cache file

Select this checkbox to make available additional information about tags, for example, their data types, when you are not connected to the data server.

#### To create a data server cache:

1. To provide the list of tags, ensure that the OPC data server is running, and that the devices are connected.
2. In the Advanced tab of the OPC Data Server Properties dialog box, select the checkbox, **Include extended information in the server cache file**.
3. Click **Create Cache**, and then click **OK**.

## Synchronizing a data server's cache

If tags are added, modified, or deleted on the data server, you must synchronize the cache manually. You can synchronize a data server's cache only after you have created one.

#### To synchronize a data server's cache:

1. To provide the list of tags, ensure that the OPC data server is running, and that the devices are connected.
2. In FactoryTalk View Studio, in the Explorer window, right-click the data server, and then click **Properties**.
3. In the Advanced tab of the OPC Data Server Properties dialog box, click **Synchronize**, and then click **OK**.

## Deleting an OPC data server

If an OPC data server is no longer required, you can delete it from the application. When you delete a data server, its cache files are also deleted.

#### To delete an OPC data server:

In FactoryTalk View Studio, in the Explorer window, right-click the OPC data server you want to delete, and then click **Delete**.

# Working with tags

This chapter describes:

- Data server tags, HMI tags, and client tags.
- Using tag data in a FactoryTalk View Site Edition application.
- How tag references work.
- Parts of the Tag Browser.
- Searching for and selecting tags.
- Browsing for offline tags.
- What HMI tags are.
- How to organize HMI tags.
- Parts of the HMI Tags editor.
- Creating, modifying, and deleting HMI tags.
- Selecting a data source for HMI tags.
- Creating HMI tags without using the Tags editor.
- What client tags are.

## About data server tags, HMI tags, and client tags

FactoryTalk View SE supports the OPC DA 2.05a, UA 1.02 and 1.03 specifications for exchanging data among automation or control applications, field systems or devices, and business or office applications.

A tag is a logical name that represents a variable in a network device or in a computer's memory (RAM). In FactoryTalk View Site Edition, you can use the following types of tags:

- **Data server tags** (also called direct reference tags, or data items in the OPC DA and UA specification) provide direct access to the basic attributes of all OPC DA and UA-compliant tags, through the data servers you add to an application.
- **HMI tags** provide additional properties for runtime security and data manipulation. HMI tags are created in the Tags editor and stored in an HMI server's tag database. For information about HMI tags, see [Working with HMI tags on page 168](#).
- **Client tags** are memory tags local to each FactoryTalk View SE client computer in a FactoryTalk network distributed or network station application. Client tag values are independent for each running client and independent between sessions logged on with different user accounts. For more information, see [Client tags on page 176](#).

## Using direct referencing to eliminate duplication

For some uses in an application, using direct references to tags in devices, or to tags located in an OPC server's database, offers advantages over using HMI tags.

For example, use data server tags to add, modify, or delete tags in a device without having to duplicate the changes in the HMI server's tag database.

## Providing access to complex data types

Some devices (Studio 5000 controllers, for example) support data types such as arrays and structures, which can contain hundreds of member elements.

Use data server tags to reference the tag values directly, and eliminate the need to create an HMI tag for each member element.

## Using the controller tag extended properties

Starting from version 21, Logix controllers support the extended tag properties that are configurable in Studio 5000 Logix Designer®.

Starting from FactoryTalk View version 9.00.00, you can use the following extended tag properties when a controller tag type can be referenced:

- **.@Description:** Description of the tag
- **.@EngineeringUnit:** A system of measurements of physical quantities
- **.@Min:** The minimum numeric value for the non-Boolean tag being measured
- **.@Max:** The maximum numeric value for the non-Boolean tag being measured
- **.@State0:** The false Boolean state of the tag
- **.@State1:** The true Boolean state of the tag

Starting from Logix Designer version 33 and FactoryTalk Linx version 6.20.00, you can also use the following extended tag properties:

- **.@Library:** Associates instructions with a display library for faceplate call-up.
- **.@Instruction:** Associates the instruction name for faceplate call-up.
- **.@Label:** A label that appears on the graphic HMI symbol.
- **.@Area:** The area name for area-based security in the HMI.
- **.@URL:** The Uniform Resource Locator (URL) for accessing the help content from HMI.
- **.@Navigation:** A path to another source of information. For example, a document link or website.
- **.@Name:** A text string ranging from 0 through 512 characters that defines the tag name in the client application.
- **.@Quality:** A string following the OPC DA standard to indicate the tag status. The following table lists the quality codes and their meanings:

Value	Description
0	Bad
4	Bad - Configuration Error in Server
8	Bad - Not Connected
12	Bad - Device Failure
16	Bad - Sensor Failure
20	Bad - Last Known Value Passed
24	Bad - Common Failure
28	Bad - Out of Service
64	Uncertain
68	Uncertain - Last Usable Value
80	Uncertain - Sensor not Accurate
84	Uncertain - Engineering Units exceeded
88	Uncertain - Value from Multiple Sources

Value	Description
192	Good
216	Good - Local Override, Value Forced

To obtain the quality value, for example, `[New_shortcut]Tag1.@Quality`, the base tag `[New_shortcut]Tag1` must be added to the client application as well.

- **.@Timestamp:** A string that follows the format of Year/Month/Day/Hour/Minute/Second to define the time when FactoryTalk Linx data server polls the data to the client.
- **.@Scope:** A string that defines a tag's controller and program scope.

To read the values, use the format **TagName.@ExtendedProperty** where *TagName* is the controller tag and *ExtendedProperty* is the extended tag property, for example, `[Shortcut]SodaCIPTanks.Water.Value.@Min`.

## Using the extended capabilities of HMI tags

To take advantage of extended capabilities, such as runtime security, scaling or offsetting of tag values, or more flexible addressing, you can create HMI tags in FactoryTalk View Studio, in the Tags editor.

## Securing tag or device values

HMI tags let you secure tag or device values at runtime.

For example, to prevent users from changing a value in a device, create an HMI tag for the device's address, and then assign a security code to the HMI tag. For details, see [Assigning security codes to HMI tags on page 93](#).

## Scaling and offsetting values

HMI tags provide scale and offset capabilities, which are not supported by some data servers.

Use scale and offset to modify the raw data that comes from the network device before it is saved in the computer's memory, or to modify a value specified in FactoryTalk View before it is written to a device or data server.

Scale is a multiplication factor – the value from the device is multiplied by the scale. Offset is a fixed value – after the value from the device is multiplied by the scale, the offset amount is added.

## Specifying minimum and maximum values

With HMI tags, you can specify the minimum and maximum values that can be written to the network device or data server.

The minimum and maximum values do not affect what is read from the device or server.

For example, if you specify a minimum of 0 and a maximum of 100, FactoryTalk View would be able to read a value of 200 from a device and store it in memory, but would not be able to write this value to the device.



**Tip:** To further define a range of values that can be written to a tag, you can specify a minimum and maximum value for FactoryTalk View SE numeric input objects. For details, see [Validating operator input on page 296](#).

## Storing values in memory

Create HMI memory tags to store values without the need for an attached or accessible device. For example, you can use memory tags to:

- Store the result of a calculation.
- Act as a temporary counter or index.
- Maintain information about the system's current state, for example, which graphic display was last shown.

For more information about HMI memory tags, see [Data sources for HMI tags on page 169](#).

## Reusing HMI tag names

HMI tags do not require hard-coded physical addresses or device-specific variable names. This means you can reuse an application with other devices, simply by changing the physical addresses the tag names are mapped to.

HMI tags can also have descriptive names, which you cannot create in some controllers or OPC servers.

## Using tag data in a FactoryTalk View SE application

A FactoryTalk View SE application can use a mix of data server tags and HMI tags, depending on needs.

Following are the basic tasks involved in setting up the two types of tags.

### Setting up data server tags

To use data server tags, such as those found in Studio 5000 and other OPC-compliant devices, you can refer directly to the tag's location wherever the tag data is needed.

These are the basic tasks involved in setting up data server tags for an application:

1. Create the tag in the OPC server or processor, or use an existing tag in the processor. For information about creating tags:
  - In OPC data servers that have their own tag database, see the product documentation.
  - In a Studio 5000 controller, see the programming software documentation for the controller.
2. In FactoryTalk View Studio, in the Explorer window, if it does not have a data server, add one to the application.
3. Connect application components to the data server tags, wherever the application needs to use the tag values.

For example, in a FactoryTalk View SE graphic display, connect the animation of a graphic object to a data server tag. At runtime, tag values are passed to the object, changing its appearance as the value changes.

### Setting up HMI tags

To use tags stored in an HMI server's tag database, connect the tags to addresses in network devices, and then refer to the HMI tags wherever the tag data is needed.

These are the basic tasks involved in setting up HMI tags for an application:

1. In FactoryTalk View Studio, in the Explorer window, if it doesn't have an HMI server, add one to the application. Each HMI server contains one tag database.
2. In the Tags editor, create HMI device tags by mapping tag names to data server or DDE addresses, or create HMI memory tags. For details, see [Creating, modifying, and deleting HMI tags on page 172](#).
3. Connect application components to the HMI tags, wherever the application needs to use the tag values. For example, in a FactoryTalk View SE graphic display, connect a numeric input object to an HMI tag. At runtime, tag values are passed to the object and the operator can read from or write to the tag.

## Specifying tag names where tag data is needed

Specify data server or HMI tag names in an application, wherever live tag data is needed.

For example, you could create a graphic object that represents a vat on a production line, and then set up the object to show the level of the vat at runtime.

To do this, attach the Fill animation to the object using a tag that is updated by a network device that monitors the vat level. At runtime, the value of the tag will determine the fill level of the graphic object.

In the Animation dialog box, to connect the tag to the fill level, you can:

- Type the tag name.



**Tip:** You can type the name of a tag that doesn't exist. If you do this, to avoid errors at runtime, when you create the tag, make sure you spell the tag's name consistently.

Use letters in the tag name in addition to numbers. Tag names with over 30 numbers may not show properly in the tag browser.

- Find and select the tag in the Tag Browser.

You can browse while online and connected to a device, or you can browse for tags from an offline file, such as a PLC program file. For more information, see [Parts of the Tag Browser on page 165](#).

- Create the tag, if it does not exist.

Create data server tags in the device or OPC server. For example, in a Studio 5000 device, create the tag using Studio 5000 Logix Designer programming software.

Create HMI tags in the Tags editor in FactoryTalk View Studio. For details, see [Creating, modifying, and deleting HMI tags on page 172](#).

## Logging tag values

To log tag values, in FactoryTalk View Studio, create data log models that specify which values to log, and when. For tags in a data log model to be polled and their values logged, the data log model needs to be started at runtime.

For details about setting up data logging, see [Setting up data logging on page 372](#).

## Observing tag-related limits

A graphic display can contain up to a total of 3000 connections, whether they originate from the expressions or the tags.

- Each expression associated with an object is counted as one connection regardless of the number of tags in the expression.
  - Each animation of an object (except the Touch animation) is counted as one connection.
  - Each connection in the Connections property of an object that is linked to a tag is counted as one connection.
  - Each pen configured in a Trend object is counted as one connection.
  - Duplicate references of the same expression or tag connection are counted as the additional connections.
- For example, one display can contain up to 3000 numeric inputs, even if all numeric input objects refer to the same tag.



**Tip:** Tags associated with embedded variables do not count towards the limit.

Each HMI server in an application can have up to 40,000 HMI tags. Of these, 10,000 can be HMI analog tags.



**Tip:** FactoryTalk Linx is the recommended data communications software for FactoryTalk View applications.

## About tag references

When you type a tag name, you are creating a reference to the tag.

For example, to animate a graphical representation of a fan in a graphic display, you might refer to a tag called FanRunning, to use its value for the animation.

FactoryTalk View SE network distributed and station applications can use absolute or relative references to tags. Local station applications can only use relative references. For more information, see [Absolute and relative references on page 118](#).

## Absolute references

Absolute references point directly at a specific tag, by referring to the tag's name and the area (or areas, in the case of nested areas), in which it is located.

For example, an absolute reference to a tag called Extractor in the Fans subarea of the Cooling area is:

```
/Cooling/Fans::Extractor
```

Use absolute references to ensure that a specific tag in a specific location is used, regardless of where it is referenced from.

## Relative references

Relative references point at a tag relative to the current server or area. A relative reference to a tag called Extractor is simply the tag's name:

**Extractor**

When a relative reference is used, FactoryTalk View assumes that the tag is in the current area.

Use relative references, for example, to reuse component names in a network distributed application for a plant that has identical production lines.

The application might contain different areas to represent each production line; however, within each area, the same names would be used for application components such as graphic displays.

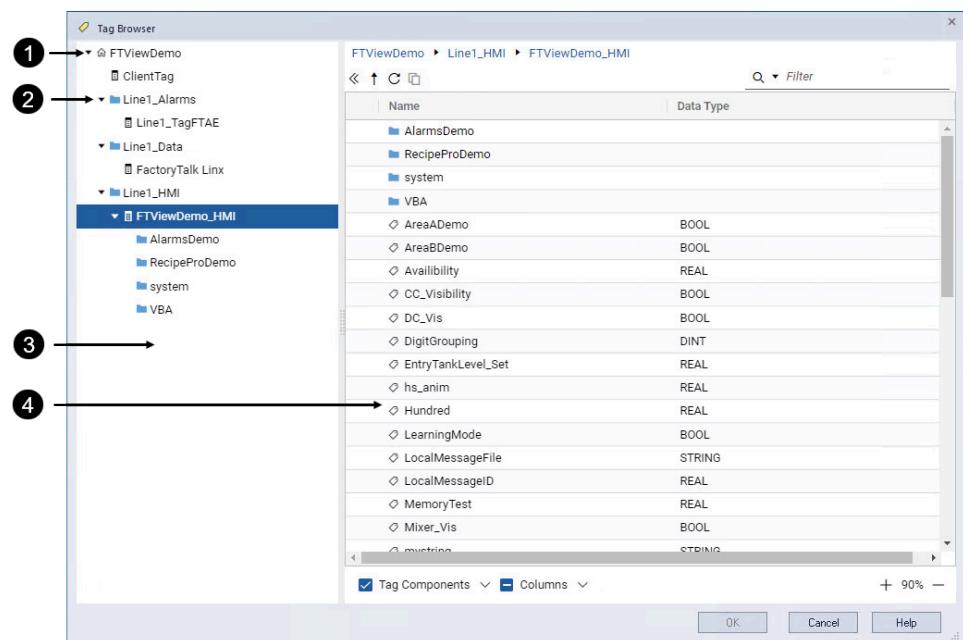
## Parts of the Tag Browser

To use a tag in a FactoryTalk View SE application, you can type the tag's name and path, or you can use the Tag Browser to search for and select the tag.

### To open the Tag Browser:

How you open the Tag Browser depends on where you are in FactoryTalk View SE:

- In the **Command Wizard**, for commands that take tags as parameters, click the **Browse** button beside the Tag field.
- In the Tags editor, when creating an HMI device tag, click the **Browse** button beside the Address field.
- In the Graphics editor, in the Animation dialog box, click **Tag**.
- In the Properties dialog box for a graphic object, in the Connections tab, click the **Browse** button in the Tag column.
- In the Expression editor, position the cursor where you want to insert the tag name, and then click **Tags**.
- In the Tags in Model tab in the Data Log Models editor, click the **Browse** button beside the Tag[s] to Add field.



Item	Description
1	Application root
2	Area folder
3	Folder pane

Item	Description
4	Tags pane

## Viewing tags in folders

In the Tag Browser, the left pane shows the application root, plus the application's HMI servers, data servers, and client tags. In a network distributed application, there is also a folder representing each area in the application.

## Finding tags in the home area

To locate the tags in the home area, right-click a blank area of the folder pane, and then select **Go To Home Area**. The home area is selected automatically.

## Searching for and selecting tags

Use Find to search for every occurrence of a specific tag or text string throughout many different components within FactoryTalk View SE applications. Find also locates all the text strings within each component that refer to the tag. Cross Reference shows a list of all the tags, all the components that refer to those tags, and the text strings.

You can also use the tag browser.

## Finding a tag or text string

To find a tag or text string:

1. From the toolbar, click  or from the Tools Menu, select Find.
2. Type the text string in **Find what**, or identify the tag to search for:
  - a. Click the browse button () next to **Find what**.
  - b. From the Tag Browser, navigate to the correct folder in the left pane, select the tag in the right pane, and then click **OK** to save the selection and close the browser.
3. If needed, identify the type of product components to search in **Find within** (all components are selected by default):
  - a. Click the browse button () next to **Find within**.
  - b. From the **Find within** dialog box, select the product component types to be searched, and then click **OK** to save the selection and close the dialog box.
4. If you are using a network distributed application, identify the HMI servers of the search in **Find where** (all servers are selected by default):
  - a. Click the browse button () next to **Find where**.
  - b. From the **Find where** dialog box, select the specific HMI servers to be searched then click **OK** to close the **Find where** dialog box.
5. Identify the word search limitations in **Search Options**.
6. Select the **Direction** in which to search:
  - a. Select **Up** to search for the tag or text string in all components before the shown one.
  - b. Select **Down** to search for the tag or text string in all components after the shown one.

7. Click **Find Next**. The location of the found tag or text string is shown in **Found item**. Click **Find Next** again to find each item in the search order. You can click the found item to open it in the appropriate editor.
8. If needed, click **Find All** to view a spreadsheet of all items that match your search criteria. Double-click the found item in the spreadsheet to open it.

## Finding a list of all tags or text strings

To find a list of all tags or text strings:

1. From the toolbar, click  or from the Tools Menu, select Cross Reference.
2. Type the text string in **Find what**, or identify the tag to search for:
  - a. Click the browse button () next to **Find what**.
  - b. From the Tag Browser, navigate to the correct folder in the left pane, select the tag in the right pane, and then click **OK** to save the selection and close the browser.
3. If needed, identify the type of product components to search in **Find within** (all components are selected by default):
  - a. Click the browse button next to **Find within**.
  - b. From the **Find within** dialog box, select the product component types to be searched, and then click **OK** to save the selection and close the dialog box.
4. If you are using a network distributed application, identify the HMI servers of the search in **Find Where** (all servers are selected by default):
  - a. Click the browse button () next to **Find where**.
  - b. From the Find where dialog box, select the specific HMI servers to be searched, and then click **OK** to close the Find where dialog box.
5. Click **Search** to show all the found items in a spreadsheet form.
6. You can double-click any of the found items to open it with the appropriate editor.

## Finding tags with Tag Browser

In the Tag Browser, select the folder where you want to search for tags, and then select the tags you need. You can select a single tag or multiple tags, depending on where you opened the Tag Browser.

For example, you can select multiple tags when you open the browser from the Data Log Models and DataLogPro editor.

The tag or tags you select can be from a data server or an HMI server. If multiple servers use the same tag name, the server name is shown with the tag name.

## Filtering tags

To show only tags whose names match a pattern, type the pattern in the **Filter** box.

### To filter tags:

- In the **Filter** box, type a string to specify which tags should be shown.
- To remove the filter, clear the text string from the **Filter** box.

## Browsing for offline tags

In the Tag Browser, tags that are available when the application is connected to a device are in the Online folder. Tags that are available from an offline file, such as a PLC program file, are in the Offline folder.

## Browsing from FactoryTalk Linx

For each FactoryTalk Linx shortcut in an application, an Offline and Online folder is shown in the Tag Browser.

Use the Offline folder to browse for tags in a PLC or Studio 5000 program stored on disk. If the shortcut in FactoryTalk Linx has access to symbols, you can also browse for them in the Offline folder.

For details about browsing offline for tags or adding symbols to a shortcut in FactoryTalk Linx, see the FactoryTalk Linx Help.

### To browse an offline file in the Communication Setup dialog box:

1. In FactoryTalk View Studio, in the Explorer window, double-click the FactoryTalk Linx icon.
2. Double-click Communication Setup to open the FactoryTalk Linx Communication Setup dialog box.
3. Click the **Browse** button beside Offline Tag File.

## Browsing from other OPC servers

To browse for offline tags from OPC servers other than RSLinx Classic, set up a data server cache.

When you are not connected to the device, you will find the offline tags in the same Tag Browser folder that contains the online tags.

For details about setting up and managing an OPC data server's cache files, see [Setting up advanced properties on page 157](#).

## Working with HMI tags

In FactoryTalk View SE, HMI servers have tag databases that can store HMI tags.

You can use HMI tags in an application, to take advantage of extended properties for securing tag or device values, manipulating data, or triggering alarms.

You can create HMI tags, or modify the properties of existing tags, in FactoryTalk View Studio, in the Tags editor. For more information, see [Creating, modifying, and deleting HMI tags on page 172](#). In a network distributed application, you can create or modify HMI tags locally (on the same computer) or remotely (in FactoryTalk View Studio on a different computer).

## HMI tag types

An HMI server's tag database can contain the following types of tags.

Tag	Type of data stored
Analog	Range of values. HMI analog tags can represent variable states such as temperature or the position of rotary controls.

Tag	Type of data stored
Digital	0 or 1. HMI digital tags can represent devices that can only be on or off, such as switches, contacts, and relays.
String	ASCII string, series of characters, or whole words (maximum of 82 characters). HMI string tags can represent devices that use text, such as a barcode scanner that uses an alphanumeric product code.
System	Information generated while the system is running. FactoryTalk View creates system tags when it creates an HMI project. The system tags are stored in the tag database, in the folder called System. You can use system tags anywhere you would use any other type of tag. You cannot modify system tag properties.

## Data sources for HMI tags

When creating an HMI analog, digital, or string tag, you specify the source of the tag's values. This is called the data source.

You can specify two types of data source for an HMI tag:

- An HMI tag with **device** as its data source receives data externally, from a source other than FactoryTalk View SE. The data can come from a programmable controller or other device via RSLink, or from another OPC or DDE data server.
- An HMI tag with **memory** as its data source receives data internally, from FactoryTalk View SE. A memory tag can be used to store values internally.

### About retentive memory tags

By default, the value of an HMI memory tag reverts to the initial value specified for the tag, whenever the HMI server containing the tag is shut down and restarted.

To set up a memory tag to retain the last value it was set to, even after the HMI server is shut down, in the Tags editor, select the Retentive checkbox for the tag.

A retentive memory tag will always have the last value it had prior to the HMI server shutting down. In other words, the tag's value will not revert to the initial value specified for the tag.

## Managing HMI tags

To make HMI tags easier to work with, especially if the tag database is large, organize related tags by establishing naming conventions for them, or by grouping them in folders.

## Naming HMI tags

Establishing naming conventions for HMI tags makes using wildcard characters more effective, when searching for and displaying tags during development and at runtime.

Tag names can use up to 255 characters of the following types:

- A to Z (upper and lower case)
- 0 to 9
- Underscore ( \_ ) and dash ( - )

For HMI tags stored in folders, the folder name becomes part of the tag name, and the backslash that separates folder names counts as one character.

Tag names can be mixed case. Tag names preserve upper and lower case for readability but they are not case-sensitive. For example, the tag name *MixerValve1* is the same as *mixervalve1*.

### Enclosing tag names in brackets

In an expression, tag names that start with a number or contain a dash must be enclosed in brackets, for example, {N33-0}.

Also use brackets with wildcard characters to represent multiple tags in an expression, for example, {alarm\*}.

For more information about using tags in expressions, see [Adding logic and control on page 423](#).

## Grouping HMI tags in folders

To organize tags, create folders in the Tags editor, for tags that are related to one another.

When naming tags stored in a folder, to separate the folder name from the rest of the tag name, use a backslash ( \ ). For example, the names of tags in a folder called *Pump* would start with *Pump\*.

To further organize tags, you can also nest folders. For example, if a plant is divided into zones, you can organize the tag database first by zone, then by machines in the zone, and finally by devices in each machine. In this case, the names of tags in the nested folder might start with *Area1\Machine1\Pump*.



**Tip:** Do not place all HMI tags in the root folder of the database. HMI tags contained in nested folders do not contribute to the total number of tags in the root folder. It is recommended that you limit the number of tags in any folder to less than 1000.

For details about creating, duplicating and deleting folders in the Tags editor, see the FactoryTalk View Site Edition Help.

## Viewing tag statistics

The Tag Statistics dialog box shows the number of tags that an HMI server contains, and provides information about the tags, such as the date the tag database was last modified.

## To view tag statistics:

1. In FactoryTalk View Studio, in the Explorer window, open the Tags editor.
2. In the Tags editor menu, select the **Tag Statistics** icon.

## Parts of the Tags editor

In FactoryTalk View Studio, use the Tags editor to:

- Create HMI tags and modify their properties.
- Duplicate and delete HMI tags.
- Create folders to group related tags.
- Duplicate, nest, and delete folders.

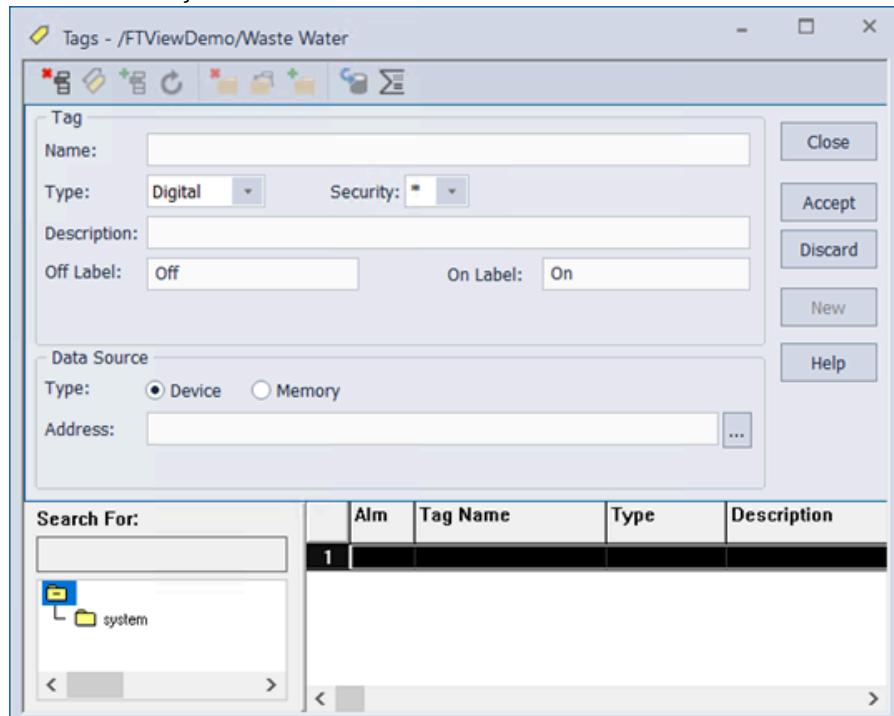
While the Tags editor is open, use items on the View menu and the toolbar, to control the editor's appearance.

## To open the Tags editor:

1. In FactoryTalk View Studio, in the Explorer window, open the HMI tags folder.
2. Right-click the Tags icon, and then select **Open**.

You can also double-click the Tags icon, or drag the icon into the FactoryTalk View Studio workspace.

The main parts of the Tags editor are the form, the query box, the folder hierarchy, and the spreadsheet, as shown in the following illustration.



## Accept and Discard buttons (Tags editor)

When you enter information in the Tags editor, the **Prev** and **Next** buttons change to the **Accept** and **Discard** buttons.

Click **Accept** to save tag information. Click **Discard** to cancel changes to a tag's properties.

## Form and spreadsheet (Tags editor)

In the upper part of the form, define the basic characteristics of the tag, such as tag name, type, security, and specifics related to the tag type.

In the lower part of the form, specify the data source (where the tag's values will come from).

## Query box

Use the query box to select the tags you want to show in the spreadsheet.

This means that you can modify tags in different folders without browsing the folder hierarchy. You can select a single tag by typing the tag name, or select multiple tags by typing wildcard characters.

This wildcard character	Does this
?	Matches any single character.
*	Matches any number of characters, including the backslash (\ ) character. Use this wildcard character by itself, to display all the tags in the tag database.

## Folder hierarchy

The hierarchy and spreadsheet work together. The hierarchy shows the tag folders, and the spreadsheet shows the tags within the folders.

A folder icon called root is always present in the hierarchy window, at the top of the folder hierarchy. This folder contains all the tag folders you and FactoryTalk View create.

If a folder icon has a plus (+) sign on it, the folder contains one or more nested folders. If the folder's icon is blank, it does not contain any other folders.

When you nest folders, remember that the backslash in a folder name counts as one character in the tag name.

## Creating, modifying, and deleting HMI tags

Use the Tags editor to create, modify, duplicate, and delete HMI tags.

For details about these operations, see the FactoryTalk View Site Edition Help.

### To create an HMI tag:

1. In FactoryTalk View Studio, in the Explorer window, open the HMI Tags folder, and then double-click the Tags icon.
2. In the Tags editor, if you want to add the new tag to a folder, double-click the folder in the folder hierarchy window (under the Search For field).
3. To clear the form and position the cursor in the Name box, click **New** in the form section of the editor. If you have opened a folder, the folder's name is inserted in the Name box automatically, and the cursor is positioned to the right of the folder name.
4. Type a name for the tag, and then select its type.

5. In the Tag section, specify any additional information required for the type of tag you are creating.
6. Select a data source and specify the required information.
7. Click **Accept** to save the tag to the database.

### To modify a tag:

1. In the Tags editor, find and select the tag you want to modify.  
Tag attributes are shown in the form section of the editor.
2. Modify selected attributes.  
You can modify all parts of an existing tag, except the tag name and tag type.
3. Click **Accept** to save changes.

### To duplicate a tag:

1. In the Tags editor, find and select the tag you want to duplicate.
2. From the Edit menu, select **Duplicate**  
You can also click the **Duplicate** button on the editor's toolbar.  
A new row is inserted under the highlighted row. The new row contains all the same information except the tag name.
3. In the Tag Name field, type a name for the new tag.
4. Click **Accept**.

### To delete a tag:

1. In the Tags editor, find and select the tag you want to delete.
2. From the Edit menu, select Delete.  
You can also click the Delete button on the editor's toolbar.

---

**NOTE:** Delete tags carefully. Once you click Delete, the tag is deleted. There is no confirmation message, and you cannot undo the deletion.

---

## Selecting data source for HMI tags

The data source for an HMI tag can be an external device or FactoryTalk View.

### Getting HMI tag data from a device

An HMI tag with the device as its data source receives its data from a source external to FactoryTalk View. The data can come from:

- Allen-Bradley or Studio 5000 programmable controllers through FactoryTalk Linx.
- Network devices through an OPC or DDE data server.
- Another Windows program through an OPC or DDE data server.

For information about setting up OPC communications, see [Setting up communications on page 149](#).

For information about DDE communications, see [Setting up DDE communications for HMI tags on page 472](#).

**To create a device tag:**

1. In the form section of the Tags editor, click **Device**.
2. In the Address box, type the name of the data server tag or DDE item.
3. To find and select a tag's address, click the **Browse** button.

**Addressing syntax for OPC tags:**

The syntax for OPC tag addresses is:

[AccessPath]Address

or

:: [AccessPath]Address

where :: is required for addresses that contain an initial colon (:).

The access path is one of the following:

- For communications with an RSLinx Classic OPC server, the access path is the name of a device shortcut or DDE/OPC topic in RSLinx Classic.
- For communications with other OPC servers, the access path may be optional. For information about the syntax for the access path, see the OPC server documentation.

The square brackets around the access path are part of the syntax. They do not indicate optional parameters.

**Getting HMI tag data from memory**

An HMI tag with Memory as its data source receives its data from internally, from FactoryTalk View.

**To create a memory tag:**

1. In the form section of the Tags editor, click **Memory**.
2. In the Initial Value field, type the tag's starting value. This is the value that the memory tag will have when you first load the HMI project.

To ensure that a memory tag uses a particular value when the project starts, use the Set or =(Equal) commands in a startup macro to specify the tag's value.

For details about using macros, see [Adding logic and control on page 423](#).

The value of a memory tag can also be set using derived tags, events, or graphic objects.

3. To retain the tag's value retained when an HMI server shuts down, select the **Retentive** checkbox.

To let the memory tag revert to its initial value the next time the HMI server starts, clear the **Retentive** checkbox.

**Creating HMI tags without using the Tags editor**

In addition to creating tags in the Tags editor, you can add tags to an HMI server's tag database using one of the following methods:

- Create the tags in a third-party application, and import them into FactoryTalk View.
- Import tags from a PLC or SLC database.

## Creating tags in a third-party application

You can use a third-party spreadsheet editor such as Microsoft Excel to create tags, and then import them into FactoryTalk View using the Tag Import and Export Wizard. For details, see [Importing and exporting tags on page 175](#).

When you import tags, they can be merged with tags already in the HMI server's tag database. This means that any tags in the database with the same name are updated with the new information.

## Importing and exporting tags

You can also import PLC or SLC databases into FactoryTalk View using the Tag Import and Export Wizard. The Tag Import and Export Wizard must be run on the same computer as the HMI server.

To open the wizard, use one of these methods:

- In FactoryTalk View Studio, from the Tools menu, select **Tag Import and Export Wizard**.
- Select **Start > Rockwell Software > Tag Import and Export Wizard**.

For details about options in the Tag Import and Export Wizard, click **Help**.

## Importing tags from a PLC database

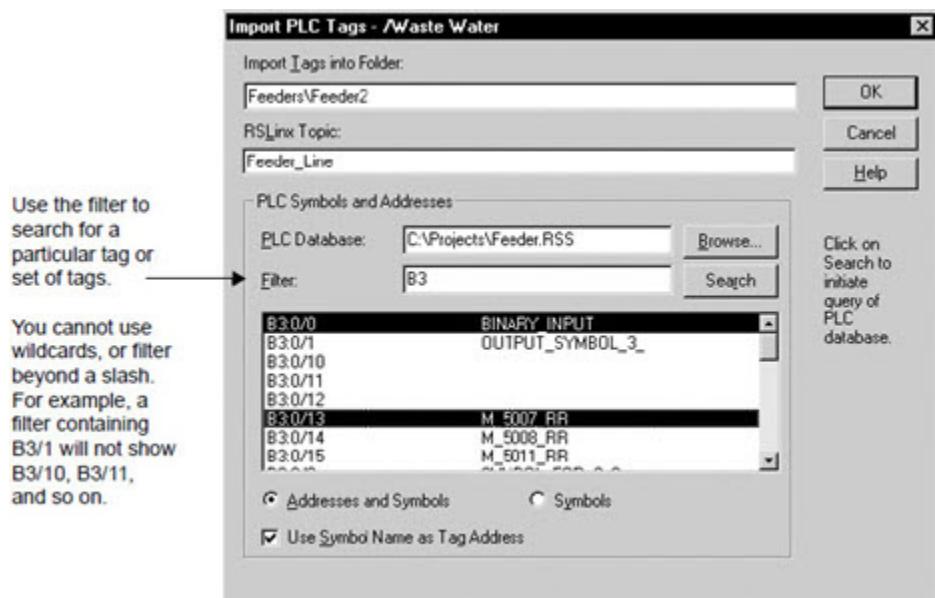
To import tags selectively from a PLC database into the FactoryTalk View HMI tag database, use the Import PLC Tags dialog box.

Tags imported in this way are copied into the database—they are not shared with the source database. This means changes to tags in FactoryTalk View do not affect the database from which they have been imported and vice versa.

## Opening the Import PLC Tags dialog box

To open the Import PLC Tags dialog box:

- In the Tags editor, click the **DB Browser** button  on the toolbar, or from the **Edit** menu, select **Other Databases**.



You can import tags from any of these databases:

- RSLogix 5/500, saved as an external database, with file extension .ctd
- RSLogix 5 internal database, with file extension .rsp
- RSLogix 500 internal database, with file extension .rss

For PLC and SLC addresses, the Import PLC Tags dialog box shows only addresses that are used in the symbol or address list of the PLC programming software.

For details about options in the Import PLC Tags dialog box, click **Help**.

## Client tags

Client tags are memory tags local to each FactoryTalk View SE client computer in a FactoryTalk network distributed or network station application. Unlike server-based HMI tags whose value is the same on all the client computers, client tag values are independent for each running client. Use client tags for functions that must be specific to each client computer.

Additionally, on a single client computer, client tags are independent between sessions that are logged on with different user accounts. When accessing the same display client from different user sessions or thin clients, the client tag values are local per session or thin client.



**Tip:** Because client tags are only available for client computers, do not use them for server-based functions, such as Derived Tags, Events, and Scripts.

Client tags are in the **@ClientTags** folder in the Tag Browser. Client tags' supported data types are Boolean, floating point, long integer, and string. Each type includes 10 tags by default.

## Working with client tags

Unlike HMI tags in a network distributed application that are global, HMI tags can be used for different tag values on different client computers. For example, if an application requires that object animation (button visibility, for example) be specific or unique to each running client, client tags on each client computer can provide such capability.

Client tags can be used just like any other tags. You can browse client tags in the **@ClientTags** folder in the Tag Browser, set client tag values, and add client tags in the expression. To customize the default client tags, use Tag Import and Export Wizard to export or import client tags. To open the wizard, click the Windows **Start** menu, and then type Tag Import and Export Wizard.

# Setting up FactoryTalk alarms

This chapter describes:

- What FactoryTalk Alarms and Events is.
- Key FactoryTalk alarm concepts.
- The basic steps involved in setting up FactoryTalk alarms.
- Setting up system-wide alarm and event policies.
- Working with Rockwell Automation Device Servers.
- Working with Tag Alarm and Event Servers.
- Setting up FactoryTalk tag-based alarms.
- Setting up OPC UA alarm monitoring.
- Setting up alarm and event history logging.
- Setting up alarm and event displays.
- Working with alarm sources in the Alarm Status Explorer.
- Viewing alarm and event history logs.
- Using tags to interact with alarms or to obtain alarm status.
- Using FactoryTalk alarm functions in expressions.
- Importing and exporting alarms.

## About FactoryTalk Alarms and Events

Alarms are an important part of most plant control applications because they alert operators when something goes wrong.

An alarm can signal that a device or process has ceased operating within acceptable, predefined limits, or it can indicate breakdown, wear, or a process malfunction. Often, it is also important to have a record of alarms and whether they were acknowledged.

In FactoryTalk View SE, use FactoryTalk Alarms and Events services to centralize the distribution of device-based and tag-based alarm data to runtime clients, through FactoryTalk alarm servers that you add to a FactoryTalk View SE application. FactoryTalk Alarms and Events also supports OPC UA alarms, in which case alarm conditions are provided by third-party OPC UA servers, and then the alarm data is routed through FactoryTalk Linx OPC UA Connector.

The choices you make will depend on factors such as the design of your application, the processes you need to monitor for alarms, the types of devices used in the application, and whether you want to build alarm detection into those devices.

## Advantages in using FactoryTalk Alarms and Events services

FactoryTalk Alarms and Events offers a single, integrated set of alarm information, distributed through supporting alarm servers. This means that all clients in a FactoryTalk system can receive a consistent view of device-based, tag-based, and OPC UA alarm activity.

FactoryTalk Alarms and Events services are also integrated with FactoryTalk Security services. This means you can set up FactoryTalk system users and computers with different levels of access to alarm sources and data.

The decisions you make about setting up alarm monitoring and control will depend on the design of your FactoryTalk View SE application, the types of devices the application requires, and the processes you need to monitor for alarms.

With FactoryTalk Alarms and Events, a FactoryTalk View application can include the following alarm monitoring:

- Device-based alarms, including:
  - Logix instruction-based alarms  
Alarm instructions are programmed and then downloaded into the Studio 5000 controllers. The controller detects alarm conditions and publishes the alarm information, which is routed to FactoryTalk Alarms and Events.
  - Logix tag-based alarms  
Logix tag-based alarms associate alarm conditions with tags for Studio 5000 controllers. Logix tag-based alarms monitor tag values to determine the alarm condition, but they are not part of the logic program and do not increase the scan time for a project. Logix tag-based alarms are supported only on Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers.
- Server tag-based alarms  
Alarm conditions are defined and managed with software that monitors programmable controllers through tags. Tag-based alarm monitoring is supported by Rockwell Automation controllers communicating through Rockwell Automation device servers (FactoryTalk Linx), third-party controllers communicating through OPC data servers, and HMI tags. Use server tag-based alarm monitoring if you do not want to use the pre-built instructions in Studio 5000 Logix Designer, or do not have Studio 5000 controllers.
- OPC UA alarms  
Alarm conditions are provided by third-party OPC UA servers. The alarm data is then routed to FactoryTalk Alarms and Events through FactoryTalk Linx OPC UA Connector.

## Finding more information about FactoryTalk Alarms and Events

This section provides an overview of the key FactoryTalk Alarms and Events concepts, and describes some of the basic tasks involved in setting up FactoryTalk alarms in a FactoryTalk View SE application.

For example, this manual contains basic information about:

- Setting up Rockwell Automation Device Servers (FactoryTalk Linx), so that FactoryTalk View SE Clients can receive FactoryTalk device-based alarm information.
- Setting up Tag Alarm and Event Servers, so that FactoryTalk View SE Clients can receive FactoryTalk tag-based alarm information.
- Setting up OPC UA alarm monitoring, so that FactoryTalk View SE Clients can receive alarm information from third-party OPC UA servers.
- Setting up runtime alarm monitoring and control, using the FactoryTalk objects available in FactoryTalk View Studio, in the Graphics editor.
- Setting up FactoryTalk alarm and event history logs, and viewing alarm history information.

For information about creating FactoryTalk View SE applications and components, see other chapters in this manual, or see *FactoryTalk View Site Edition Help*.



**Tip:** This manual does not contain information about how to use Studio 5000 Logix Designer to program alarm instructions into Studio 5000 controllers. For details, see the Studio 5000 Logix Designer documentation, or see *FactoryTalk Alarms and Events System Configuration Guide*.

## FactoryTalk Alarms and Events Help

For comprehensive information about FactoryTalk Alarms and Events, including detailed setup instructions, see *FactoryTalk Alarms and Events Help*.

To open the Help, use one of these methods:

- Select **Start > Rockwell Software > FactoryTalk Help**.  
In the FactoryTalk Help file, open the book *FactoryTalk Alarms and Events Help*.
- Click **Help** in editors and dialog boxes for help with a specific object.

## FactoryTalk Alarms and Events System Configuration Guide

For step-by-step instructions about setting up FactoryTalk Alarms and Events quickly, from a FactoryTalk system perspective, see the *FactoryTalk Alarms and Events System Configuration Guide*.

To open the System Configuration Guide, in FactoryTalk View Studio, from the Help menu, select **Online Books > FactoryTalk Alarms and Events System Configuration Guide**.

## Key FactoryTalk Alarms and Events concepts

This section presents some of the common terms and concepts used to describe FactoryTalk Alarms and Events services, components, and data.

### Alarms and events

A FactoryTalk Alarms and Events system produces two types of events:

- **Condition-related events** are associated with changes in an alarm's state. For example, a condition-related event is generated when an alarm changes to In Alarm, Acknowledged, Return to Normal or Disabled. Also known as alarms, alarm conditions, or alarm activity, condition-related events are recorded in FactoryTalk alarm and event displays and logs.
- **Tracking-related events** are associated with monitoring or auditing changes to the FactoryTalk system. For example, a tracking-related event is generated when an operator acknowledges an alarm. Tracking-related events associated with alarm activity are recorded in FactoryTalk alarm and event logs only.

For details about these types of events, see the FactoryTalk Alarms and Events Help.

### Alarm servers

FactoryTalk Alarms and Events services make alarm information available to subscribing FactoryTalk View SE Clients through the following alarm servers:

- **Rockwell Automation Device Servers (FactoryTalk Linx)** provide access to alarms detected in Studio 5000 controllers that support built-in alarm detection.
- **Tag Alarm and Event Servers** provide access to alarms set up for tags in:
  - Older devices (PLC-5 or SLC 500) that don't support built-in alarm detection.
  - Third-party devices communicating through OPC data servers.

- An HMI server's tag database (HMI tags).



**Tip:** You can also set up tag-based alarms for Studio 5000 controllers that do support device-based alarms, if you prefer not to set up built-in alarm detection.

- **Third-party OPC UA servers** provide the alarm data to FactoryTalk Alarms and Events via FactoryTalk Linx OPC UA Connector that acts as a proxy between the OPC UA server and a FactoryTalk application.

For information about setting up alarm servers in a FactoryTalk View SE application, see:

- [Working with Rockwell Automation Device Servers on page 189](#)
- [Working with Tag Alarm and Event Servers on page 191](#)
- [Setting up OPC UA alarm monitoring on page 195](#)

## Level alarm

A level alarm defines multiple conditions that evaluate a single level tag (sometimes called an analog tag). Each level can uniquely identify an alarm threshold, a severity, and an alarm message.

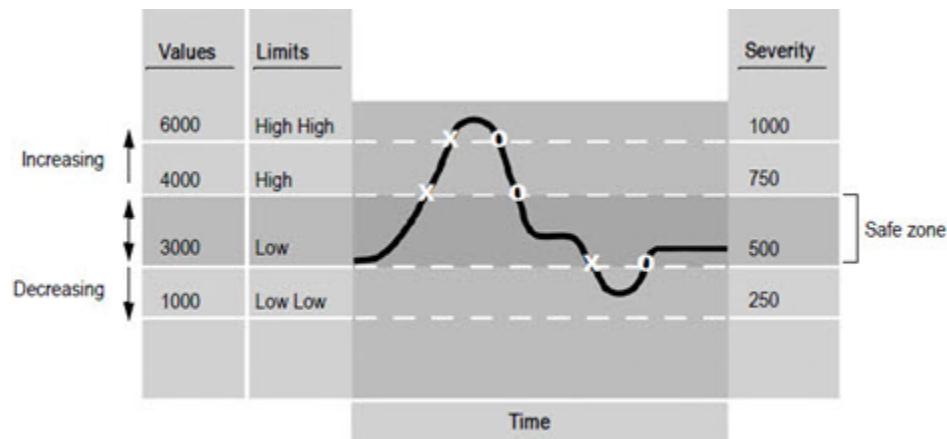
## Level alarm limits

When defining a level alarm, you can assign any of the following limits, with different levels of severity:

- High High
- High
- Low
- Low Low

The High and High High alarm conditions monitor for a rising value, and the Low and Low Low conditions monitor for a decreasing value.

### Example: Alarm limits



In this example, a tag's value changes as it monitors a motor's revolutions per minute (rpm). An X in the illustration shows when the alarm condition goes into alarm, and an O shows when the alarm condition returns to normal.

With the given limit settings, the motor must run between 3000 rpm and 4000 rpm (the safe zone), otherwise, an alarm will be triggered.

If the motor speed	It triggers an alarm of this severity
Exceeds 4000 rpm	750 (High limit)
Exceeds 6000 rpm	1000 (High High limit)
Falls below 3000 rpm	500 (Low limit)
Falls below 1000 rpm	250 (Low Low limit)

## Variable limits and alarm faults

Limits can be constant or variable. Variable limits are derived from the value of another tag. As the value of the specified tag changes, the limit changes.

A variable limit must not become higher than the limit above it, or lower than the limit below it. If this happens, an alarm fault is generated for the tag being monitored.

To correct an alarm fault, you must change the variable limit so it does not overlap either of its neighbors. This can become complex when the neighboring limits are also variable, because these boundaries are determined dynamically at runtime.

When the faulty limits return to their normal operating range, the alarm fault is cleared, the out-of-alarm-fault status is generated and logged, and normal alarm monitoring for the alarm condition resumes.

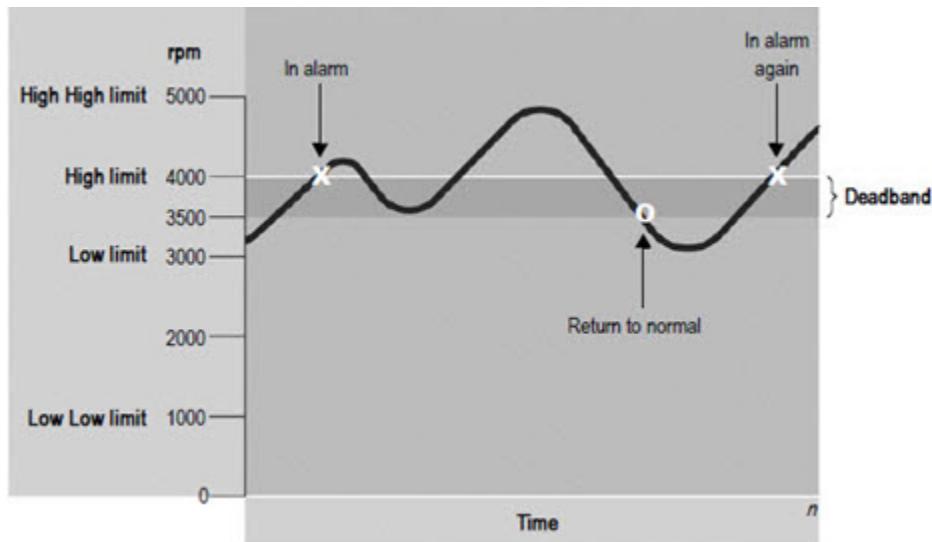
## Deadband

With some types of measured values, such as line pressure, tag values can fluctuate rapidly above and below a critical limit.

Where such conditions exist, you can create a deadband as a buffer zone to prevent the fluctuations from retriggering unnecessary alarms.

For the High and High High alarm conditions, the tag value must drop below the alarm limit minus the deadband, before the alarm condition goes Normal (Out of Alarm). For the Low and Low Low alarm conditions, the tag value must go above the alarm limit plus the deadband, before the alarm condition goes Normal.

In the following illustration, an increasing limit of 4000 rpm, with a deadband value of 500 rpm, means that the rpm has to fall to 3500 and then rise above 4000, before the rpm value will trigger an alarm again.



A deadband range can only be an absolute (constant) value. If a buffer is not required, the deadband must be set up as zero.

With a deadband of zero, alarms will be triggered as soon as the tag value crosses any of its limits.

For information about how deadbands work with deviation alarms, see *FactoryTalk Alarms and Events Help*.

---

**NOTE:** Use the deadband carefully in safety-critical applications. In the preceding illustration, for example, the variable has to fluctuate by more than 500 rpm before an alarm is triggered again.

---

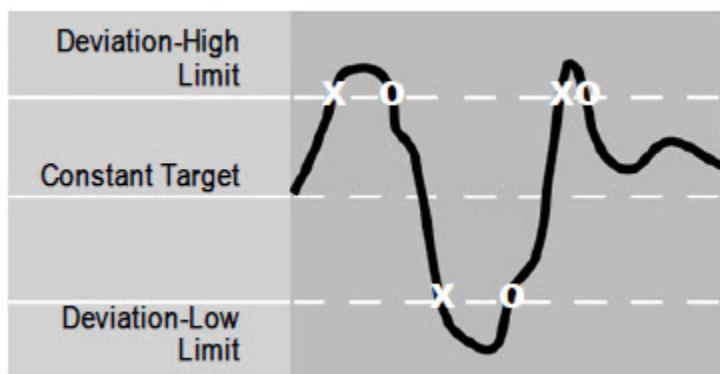
## Deviation alarms

A deviation alarm compares the value of an input tag to the value of a target for a deviation value.

The target can be either a constant or a tag, but the deviation value can only be an absolute value (a constant, not a tag value).

If the target differs from the input tag by greater or less than the deviation value, an alarm occurs.

In the following illustration, an X shows when the tag goes into alarm, and an O shows when the tag returns to normal.



## Digital alarms

A digital alarm monitors the value of a tag or input for either of these alarm conditions: the value is equal to zero, or the value is not equal to zero.

FactoryTalk Alarms and Events does not support tag-based, digital change-of-state alarms. Change-of-state alarms are triggered by any change in the digital tag being monitored, or when the tag's value changes to either 0 or 1.

Digital alarms can be latched. This means that the digital alarm will remain In Alarm, even if its alarm condition returns to normal, until an operator resets the alarm.

## FactoryTalk alarm and event displays

To monitor and respond to FactoryTalk device-based, tag-based, and OPC UA alarms at runtime, use the following FactoryTalk Alarm and Event objects, embedded in FactoryTalk View SE graphic displays.

### Alarm and event summary

The Alarm and Event Summary shows alarm information from Rockwell Automation Device Servers (FactoryTalk Linx), Tag Alarm and Event Servers, and third-party OPC UA servers that support FactoryTalk Alarms and Events.

Use the alarm and event summary to view, acknowledge, suppress, shelve, unshelve, and disable the alarms on display. The alarm and event summary also provides a detailed view of selected alarms.

An alarm and event summary can display alarms from more than one area and alarm server in a FactoryTalk View SE application.

For information about setting up an alarm and event summary, see [Setting up an alarm and event summary on page 198](#).

### Alarm and event banner

The alarm and event banner can show up to five of the highest priority, most severe, and most recent alarms in the FactoryTalk system.

Use the alarm and event banner to monitor and acknowledge the critical system alarms. You can also open an alarm and event summary, from an alarm and event banner.

For information about setting up an alarm and event banner, see [Setting up an alarm and event banner on page 198](#).

### Alarm status explorer

The alarm status explorer shows all of the alarm sources in an application.

Use the alarm status explorer to view alarm sources, suppress or unsuppress, shelve or unshelve, enable or disable alarms.

For information about setting up an alarm status explorer, see [Working with alarm sources in the Alarm Status Explorer on page 200](#).

## Alarm and event log viewer

The alarm and event log viewer shows the information stored in the alarm and event logs, generated by the FactoryTalk Alarms and Events Historian.

Use the alarm and event log viewer to view, sort, filter, and print historical alarm and event information.

For information about setting up an alarm and event log viewer, see [Viewing alarm and event history logs on page 201](#).

## Automatic diagnostic event summary

The automatic diagnostic event summary shows diagnostic events collected by Rockwell Automation controllers via Ethernet. It is enabled by the Automatic Diagnostics feature. Automatic Diagnostics is supported only on Compact GuardLogix 5380, CompactLogix 5380, CompactLogix 5480, ControlLogix 5580, and GuardLogix 5580 controllers, and the firmware revision must be 33 or later.

Use the automatic diagnostic event summary to suppress, unsuppress, filter, and sort diagnostic events at runtime.

For information about setting up an automatic diagnostic event summary, see [Setting up an automatic diagnostic event summary on page 200](#).

## Alarm priority and severity

An individual alarm in a FactoryTalk system can have a severity value of 1 to 1000, as well as a priority value that covers a range of severities.

Priority values group severities into levels of urgency that an operator can easily identify.

Alarm priorities can be defined in an alarm server (Rockwell Automation Device Server or Tag Alarm and Event Server), or in the FactoryTalk Directory.

If priority values are not defined in an alarm server, then the following system-wide settings are in effect for all alarm servers in the FactoryTalk Directory.

	This priority value	Includes this range of severities
	Urgent	751 to 1000
	High	501 to 750
	Medium	251 to 500
	Low	1 to 250

You can change the range of severities associated with a given priority value.

If you change the settings held in the FactoryTalk Directory, the changes will apply to all FactoryTalk alarm servers the directory manages. For more information, see [Setting up system-wide alarm and event policies on page 188](#).

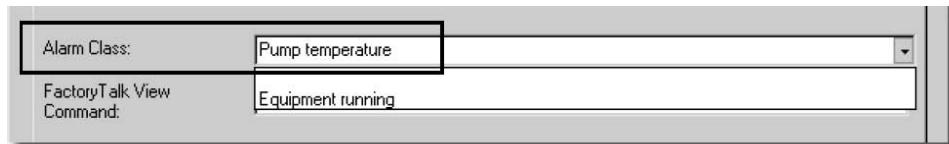
If priority values are defined at an alarm server, the server-defined priorities take precedence over the system-wide priorities.

## Alarm class

Use alarm classes to further group related alarms, based on characteristics other than priority and severity.

For example, you might group alarms based on their function, to make it easier for operators to identify alarms related to temperature, pressure, tank levels, equipment running, or valves that fail to open or close.

To create an alarm class, in the Alarm and Event Setup editor, in any of the Digital, Level, or Deviation Properties dialog boxes, type the class name in the Alarm class box, or select one from the list of classes.



Each time you create an alarm class, it is added to the list. In the previous illustration, for example, Pump temperature will be listed with Equipment running, the next time you create an alarm.

For more information about alarm classes, see the FactoryTalk Alarms and Events Help.

## Alarm states

Alarm states indicate the current status of an alarm.

When shown in an alarm and event summary or banner, an alarm condition can have one of the following states.

This icon	Indicates that the alarm condition has this state
	In Alarm, Unacknowledged
	In Alarm, Acknowledged
	<ul style="list-style-type: none"> <li>• In Alarm, Unacknowledged, Suppressed</li> <li>• In Alarm, Unacknowledged, Suppressed, Shelved</li> </ul>
	<ul style="list-style-type: none"> <li>• In Alarm, Acknowledged, Suppressed</li> <li>• In Alarm, Acknowledged, Suppressed, Shelved</li> </ul>
	In Alarm, Unacknowledged, Shelved
	In Alarm, Acknowledged, Shelved
	<ul style="list-style-type: none"> <li>• Normal, Unacknowledged, Disabled</li> <li>• Normal, Unacknowledged, Disabled, Suppressed</li> <li>• Normal, Unacknowledged, Disabled, Shelved</li> <li>• Normal, Unacknowledged, Disabled, Suppressed, Shelved</li> <li>• Normal, Acknowledged, Disabled</li> <li>• Normal, Acknowledged, Disabled, Suppressed</li> <li>• Normal, Acknowledged, Disabled, Shelved</li> <li>• Normal, Acknowledged, Disabled, Suppressed, Shelved</li> </ul>
	Normal, Unacknowledged

This icon	Indicates that the alarm condition has this state
	Normal, Acknowledged
	<ul style="list-style-type: none"> <li>Normal, Unacknowledged, Suppressed</li> <li>Normal, Unacknowledged, Suppressed, Shelved</li> </ul>
	<ul style="list-style-type: none"> <li>Normal, Acknowledged, Suppressed</li> <li>Normal, Acknowledged, Suppressed, Shelved</li> </ul>
	Normal, Unacknowledged, Shelved
	Normal, Acknowledged, Shelved
	Active, Faulted
	Normal, Faulted
	Normal, Suppressed
	Active
	Normal
	Active, Suppressed
	Normal, Disabled

## Alarm tags

Tags that you can use to obtain and change the state of an alarm are called alarm tags.

Use alarm tags to monitor and interact with alarms at runtime, without the need for an operator to use alarm and event objects in a graphic display.

For example, an operator can use the OperAck (or OperAckAll) alarm tag to acknowledge an alarm when it occurs.

To use alarm tags for an alarm, in the Alarm Setup editor, in the alarm's Properties dialog box, you must select the **Show Alarm as a Tag** option.

After you set up an alarm to show as a tag, you can select the alarm in the Tag Browser, and then select the alarm tags you want to use. For more information, see [Using tags to interact with alarms or obtain their status on page 201](#).

## Alarm status tags

Alarm status tags let you use handshaking to keep alarm state changes synchronized with a controller (PLC-5 or SLC 500).

For example, if an Acknowledged status tag is specified for an alarm, when an operator acknowledges the alarm, the status tag is set to 1.

You can use status tags to handshake changes in the In Alarm, Disabled, Suppressed, and Acknowledged alarm states.

To assign status tags to tag-based alarms, in the Alarm and Event Setup editor, select the Status Tags tab in any of the Digital, Level, or Deviation Properties dialog boxes. For more information, see [Setting up status tags for tag-based alarms on page 193](#).

## Alarm messages

Alarm messages report information about alarms.

For each alarm condition, you can create a message (or specify an existing message) to display when the alarm condition occurs. You can also use placeholders and insert variables in alarm messages.

To create and modify tag-based alarm messages, in the Alarm and Event Setup editor, select the Messages tab or open any of the Digital, Level, or Deviation Properties dialog boxes. For more information, see [Setting up tag-based alarm messages on page 193](#).

## Alarm audit, diagnostic, and history logs

FactoryTalk Alarms and Events generates three types of messages for logging:

- **Audit messages** are generated in response to actions performed in alarm and event objects, such as modifying or acknowledging an alarm.  
FactoryTalk Diagnostics routes audit messages to the Local Log (and to the FactoryTalk Audit Log, if you have the FactoryTalk AssetCentre software installed). You can look at the audit messages logged to the Local Log in the Diagnostics Viewer.
- **Diagnostic messages** inform operators of system activity, in the form of information, error, and warning messages.  
FactoryTalk Diagnostics routes these messages to a Diagnostics Local Log, the Diagnostics List, and to a central ODBC log, if one is available. You can look at the diagnostic messages in the Diagnostics Viewer.
- **Historical alarm and event messages** record all the alarm activity that occurs at an alarm server at runtime.  
The Alarm and Event Historian manages connections between alarm servers and databases, and logs data from each alarm server to an alarm history database.  
You can look at the historical alarm information in the Alarm and Event Log Viewer.

For information about FactoryTalk Diagnostics, see the FactoryTalk Help. For more information about historical alarm and event logging, see [Setting up alarm and event history logging on page 196](#).



**Tip:** The Alarm and Event Historian is installed with the FactoryTalk Alarms and Events software. To log alarm and event data, you must also install Microsoft SQL Server separately on the computers where the data will be stored. For details, see the FactoryTalk Alarms and Events Help.

## Summary of basic steps for setting up FactoryTalk alarms

To set up FactoryTalk alarm monitoring, plan the design of the FactoryTalk View SE application first, and decide which processes you want to monitor for alarms.

Then, in FactoryTalk View Studio, create the application and set up the FactoryTalk Alarms and Events components you need. Following are the basic tasks involved:

1. Set up system-wide alarm and event policies for the application.  
For more information, see [Setting up system-wide alarm and event policies on page 188](#).
2. If the application is to use device-based alarms, in Studio 5000 Logix Designer, program alarms or alarm instructions into Studio 5000 controllers that support built-in alarm detection.

- For more information, see the Studio 5000 Logix Designer documentation.
3. To let runtime clients receive device-based alarm data, add one or more Rockwell Automation device servers (FactoryTalk Linx) to the application, and set up the servers to support FactoryTalk Alarms and Events.  
For more information, see [Working with Rockwell Automation Device Servers on page 189](#).
  4. If the application is to use tag-based alarms, add one or more Tag Alarm and Event Servers.  
For more information, see [Working with Tag Alarm and Event Servers on page 191](#).
  5. In the Alarm and Event Setup editor, create digital, level, and deviation alarm conditions for the tags you want to monitor for alarms.  
For more information, see [Setting up FactoryTalk tag-based alarms on page 192](#).
  6. If the application is to capture alarm information originated from third-party OPC UA servers, add FactoryTalk Linx OPC UA Connector as the proxy to route the alarm information to FactoryTalk Alarms and Events.  
For more information, see [Setting up OPC UA alarm monitoring on page 195](#).
  7. Set up historical alarm and event logging.  
For more information, see [Setting up alarm and event history logging on page 196](#).
  8. In FactoryTalk View Studio, in the Graphics editor, set up FactoryTalk alarm and event objects in graphic displays, to monitor and interact with device-based and tag-based alarms at runtime.  
For more information, see [Setting up alarm and event displays on page 197](#).
  9. Set up FactoryTalk View SE Clients to run the graphic displays.  
For more information, see the FactoryTalk View Site Edition Help.

## Setting up system-wide alarm and event policies

FactoryTalk Alarms and Events policies are stored at the FactoryTalk Directory and apply system-wide to all alarm servers the directory manages.

In FactoryTalk View Studio, you can modify system-wide alarm settings, such as the range of severities set up for alarm priority values, and whether to send audit messages to the alarm and event log.

You can also modify the severity settings associated with system events, such as loss of connection to a controller, or with events that track operator actions, such as acknowledging an alarm.

## To modify system-wide alarm settings

1. In FactoryTalk View Studio, in the Explorer window, expand the System, Policies, System Policies, and FactoryTalk Alarms and Events folders.
2. Double-click **System Settings**.
3. In the System Settings Properties dialog box, modify special events, event buffering, and alarm priority settings, as needed. For details about each option, click **Help**.



**Tip:** To modify the severity range for the alarm priorities, change the values in the Low boxes only. Priorities defined at an alarm server override the system-wide settings held at the FactoryTalk Directory.

## To modify severity settings for system events

1. In FactoryTalk View Studio, in the Explorer window, expand the System, Policies, System Policies, and FactoryTalk Alarms and Events folders.
2. Double-click **Severity Settings**.
3. In the Severity Settings Properties dialog box, modify the severity values associated with selected system events.

For details about options in the Severity Settings Properties dialog box, click **Help**.

## Securing access to FactoryTalk alarm information

For FactoryTalk products like FactoryTalk View SE, the FactoryTalk Directory stores information about which users are allowed access to the parts of a control system.

You can secure access to alarms, and to alarm information in an application, by determining which users are allowed to perform alarm-related actions, or gain access to areas that contain alarm servers.

## About FactoryTalk Security permissions

If FactoryTalk Security services are used to secure parts of an application, to perform certain tasks, users must have the necessary security permissions.

For example, to create or modify the properties of an application, you must at least be allowed the Common actions Read, List Children, Write, and Create Children, at the FactoryTalk Directory that manages the application.

If you receive a FactoryTalk Security message while trying to perform such a task, contact your system administrator about the permissions you might require.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#).

## Working with Rockwell Automation Device Servers

To enable FactoryTalk View SE Clients to receive and display FactoryTalk alarms detected in Studio 5000 controllers, do the following:

1. Add one or more Rockwell Automation Device Servers (FactoryTalk Linx) to a FactoryTalk View SE network distributed application.  
In a local station application, you can only add one FactoryTalk Linx server, which must reside on the same computer as the application.
2. Set up the device server to support FactoryTalk Alarms and Events.
3. Create shortcuts in FactoryTalk Linx, to the Studio 5000 controllers that are detecting alarms, and enable the device shortcuts to support FactoryTalk Alarms and Events.



**Tip:** This manual does not contain information about how to use Studio 5000 Logix Designer to program alarm instructions into Studio 5000 controllers, nor does it describe how to set up redundancy for these controllers. For details, see the product documentation.

### To add a FactoryTalk Linx server to a network distributed application:

1. In the FactoryTalk View Studio, in the Explorer window, right-click the application name or an area, select **Add New Server > Rockwell Automation Device Server (FactoryTalk Linx)**.
2. Provide a name and an optional description, and then specify the name of the computer hosting FactoryTalk Linx.

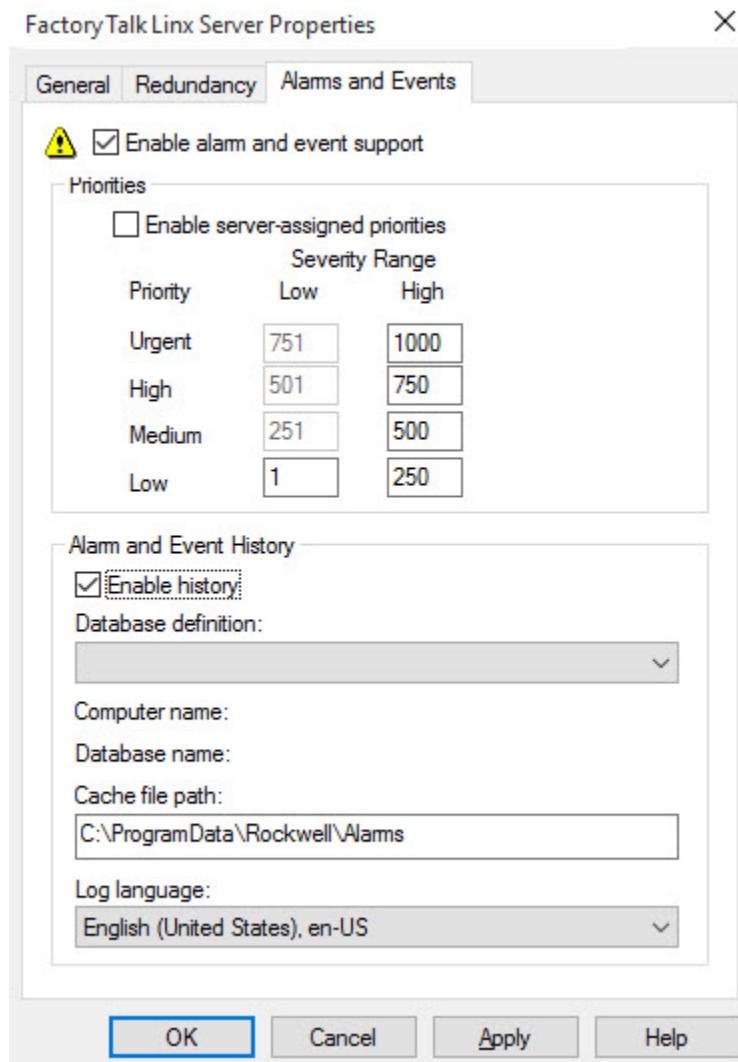
## Setting up support for FactoryTalk Alarms and Events

After you add a FactoryTalk Linx server to an application, you must set up the server to support FactoryTalk Alarms and Events.



**Tip:** If you do not enable this when the FactoryTalk Linx server is created, you can modify the server, but the server will have to be restarted.

To do this, in the server's Properties dialog box (shown in the next illustration), select the checkbox, Enable alarm and event support.



To enable alarm and event history logging on the computer running the device server, select the checkbox, Enable history.

If you enable logging, you must specify a database definition for the server. For more information, see [Setting up alarm and event history logging on page 196](#).

Optionally, you can set up server-assigned priority values for the alarm server. To modify the severity range, change values in the Low boxes only. Priorities defined at an alarm server override the system-wide settings held at the FactoryTalk Directory.

## Specifying a device-based alarm source

In FactoryTalk Linx, in the Communication Setup editor, create shortcuts to device-based alarm sources in an application. Then, enable the shortcuts to support FactoryTalk Alarms and Events.

### To open the Communication Setup editor:

In FactoryTalk View Studio, in the Explorer window, expand the FactoryTalk Linx server icon, and then double-click **Communication Setup**.

For details about options in the Communication Setup editor, click **Help**.

## Working with Tag Alarm and Event Servers

An integrated FactoryTalk Alarms and Events system can include tags in older devices (PLC-5 or SLC 500) that don't have built-in alarm detection, tags in third-party devices communicating through OPC data servers, or tags in an HMI server's tag database.

To set this up, add one or more Tag Alarm and Event Servers to a FactoryTalk View SE network distributed application, and then create digital, level, and deviation alarm definitions, create alarm messages, and specify tag update rates.

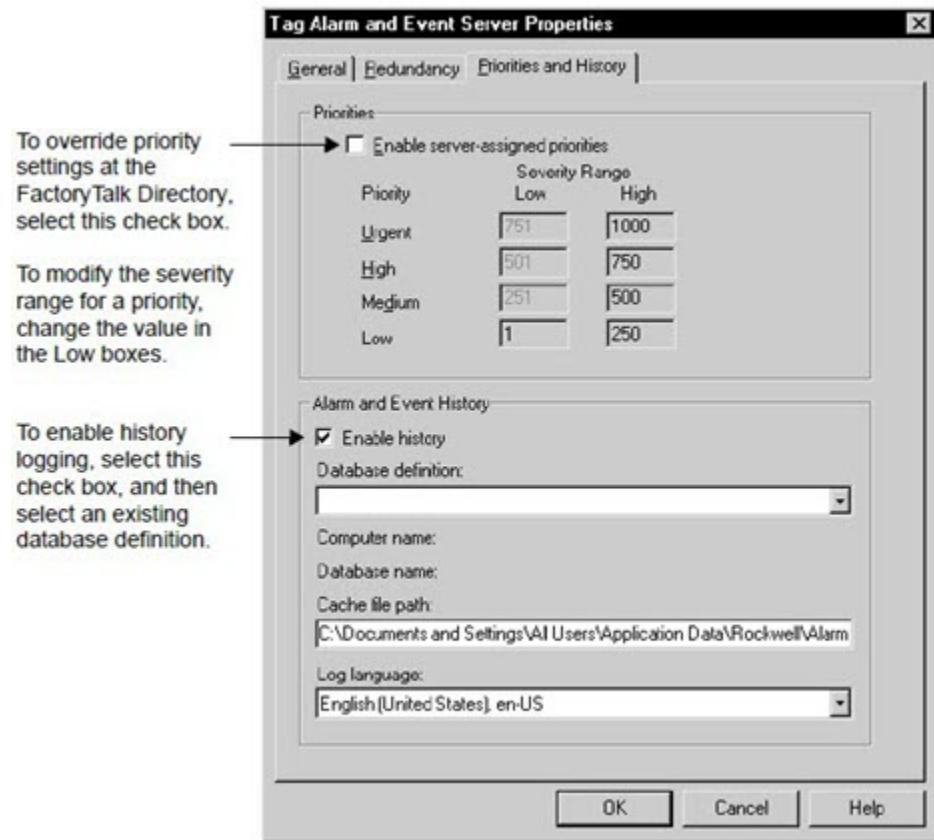
For more information, see [Setting up FactoryTalk tag-based alarms on page 192](#).

### To add a Tag Alarm and Event Server to a network distributed application:

1. In the FactoryTalk View Studio, in the Explorer window, right-click the application name or an area, select **Add New Server > Tag Alarm and Event Server**.
2. Provide a name and an optional description, and then specify the name of the computer hosting the alarm server.

## Setting up alarm priorities and history logging

To enable alarm and event history logging on the computer running the Tag Alarm and Event Server, select the checkbox, Enable history, as shown in the next illustration.



If you enable logging, you must specify a database definition for the server. For more information, see [Setting up alarm and event history logging on page 196](#).

Optionally, you can set up server-assigned priority values for the alarm server. To modify the severity range, change values in the Low boxes only. Priorities defined at an alarm server override the system-wide settings held at the FactoryTalk Directory.

The Redundancy tab provides settings for defining a secondary server.

## Setting up FactoryTalk tag-based alarms

Set up tag-based alarms to include the following in an integrated FactoryTalk Alarms and Events system:

- Older devices (PLC-5 Or SLC 500) that don't support built-in alarm detection.
- Third-party devices communicating through OPC data servers.
- HMI tags in an HMI server's tag database.



**Tip:** If you prefer not to set up built-in alarm detection, you can also set up tag-based alarms for Studio 5000 controllers that do support device-based alarms.

In the Alarm and Event Setup editor, you can create and modify:

- **Digital alarms** are either on or off. A digital alarm is triggered when the tag being monitored has a value of 1, or a value of 0.
- **Level alarms** obtain data from analog tags. A level alarm is triggered when the value of the tag being monitored crosses predefined limits. A single tag can generate several alarms of different severities, at various limits (also called thresholds).
- **Deviation alarms** compare the value of an input tag to the value of a target for a deviation value. A deviation alarm is triggered when the target differs from the input tag by greater or less than the deviation value.
- **Server-based events** are used to create a notification in the FactoryTalk Alarms and Events system when a control system event occurs.

### To open the Alarm and Event Setup editor:

1. In FactoryTalk View Studio, in the Explorer window, expand the Tag Alarm and Event Server you want to set up alarms for.
2. Double-click **Alarm and Event Setup**, or right-click the icon, and select **Open**.

For information about setting up alarms and events, see *FactoryTalk Alarms and Events Help*.

## Setting up status tags for tag-based alarms

Digital tags that you assign to alarm states are called alarm status tags.

Alarm status tags let you use handshaking to keep alarm state changes synchronized with a controller (PLC-5 or SLC 500).

For example, if an Acknowledged status tag is specified for an alarm, when an operator acknowledges the alarm, the status tag is set to 1.

For digital, level, and deviation alarms, you can assign tags to the Disabled, Suppressed, In Alarm, and Acknowledged states.

For level alarms, you can assign different In Alarm and Acknowledged status tags, for each alarm level (High High, High, Low, or Low Low).



**Tip:** Status tags provide alarm state information. Do not write to tags assigned as status tags.

### To assign status tags to a tag-based alarm:

1. In the **All Alarms** tab of the **Alarm and Event Setup** editor, select the alarm you want to assign status tags to, or click the **New** button on the toolbar to create an alarm.
2. Click the **Status Tags** tab, and then type or browse for the digital tags to be used as status tags.

## Setting up tag-based alarm messages

Alarm messages report information about alarms.

In the Alarm and Event Setup editor, the Messages tab shows all of the messages created for alarms in the Tag Alarm and Event Server.

For each message listed in the Messages tab, the ID column shows a message identifier, and the Usage column shows the number of alarms currently using the message.

You can create messages or modify existing ones using the Alarm Message Editor.

## Opening the Alarm Message Editor

To open the Alarm Message Editor, use one of these methods:

- In the Alarm and Event Setup editor, click the Messages tab, and then click the New icon on the editor's toolbar.
- In either of the Digital or Deviation Alarm Properties dialog box, click **Edit** under the Message box.
- In the Level Alarm Properties dialog box, click the Messages tab, and then click **Edit** under any of the message boxes.

For level alarms, you can create different messages for each alarm level (High High, High, Low, Low, Low). The alarm limit must be selected in the Level tab for the message box to be available.

For details about options in the Alarm Message Editor, click **Help**.

## Modifying an existing alarm message

To modify an existing alarm message, use one of these methods:

- In the Alarm and Event Setup editor, click the Messages tab, and then double-click the message you want to modify.
- In the Digital, Level, or Deviation Alarm Properties dialog box, under the Message box, click **Edit**.



**Tip:** When you modify a message, changes are copied wherever the message is used. To check the number of alarms a message is assigned to, in the Alarm and Event Setup editor, click the Messages tab. The number is shown beside the message in the Usage column.

## Adding variables to alarm messages

In the Alarm Message Editor, you can embed one or more variables into alarm messages.

For example, you might embed variables for the input tag value and the alarm's severity. At runtime, the message will show the value of the tag and the alarm severity at the time the message is logged.

For details about embedding variables in an alarm message, and the syntax that variables use, click **Help** in the Alarm Message Editor.

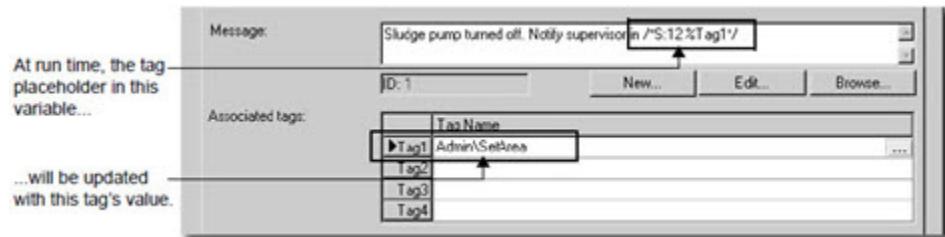
For information about using variables in other parts of a FactoryTalk View SE application, see [Creating embedded variables on page 365](#).

## Using tag placeholders in alarm messages

Tag placeholders are another type of variable that you can add to an alarm message, to include specific values in the message at runtime.

In the Alarm Message Editor, in the Variable list, select up to four tag placeholders (Tag1, Tag2, Tag3, Tag4) to add to an alarm message. Then, specify the tag to associate with the placeholder in the Alarm Properties dialog boxes where the message is used.

In the following illustration, the placeholder **Tag1** is associated with a tag named Admin\SetArea.



For details about using tag placeholders in an alarm message, click **Help** in the Alarm Properties dialog box.



**Tip:** In addition to providing values to alarm message variables, the values of associated tags can be shown with alarm conditions or used to filter information in an alarm and event summary or log viewer.

For information about using tag placeholders in other parts of a FactoryTalk View SE application, see [Creating graphic displays on page 259](#).

## Specifying tag update rates

The tag update rate determines how frequently to update the values of tags being monitored for alarms.

The rate at which the system checks for alarms will not be faster than the specified tag update rate. The default rate is 2 seconds.

In the Tag Update Rates tab, specify a rate for each tag being monitored for alarms. Base the rate on how often you expect tag values to change.

For example, if you are monitoring temperatures that change slowly, you can check for alarms less often than if you are monitoring manufacturing processes that change rapidly.

## Setting up OPC UA alarm monitoring

Set up OPC UA alarm monitoring to capture alarm information originated from third-party OPC UA servers. Unlike tag-based or device-based alarm monitoring, OPC UA alarm monitoring does not require a Tag Alarm and Event Server or FactoryTalk Linx.

For OPC UA alarm monitoring, at least one third-party OPC UA server must be able to communicate with the third-party devices that you want to monitor for alarm conditions. Then, add FactoryTalk Linx OPC UA Connector to an application or area. After defining an endpoint URL to an OPC UA server and enabling alarm and event support, FactoryTalk Linx OPC UA Connector routes the alarm information to FactoryTalk Alarms and Events.

## Enabling alarm and event in FactoryTalk Linx OPC UA Connector

FactoryTalk Linx OPC UA Connector is required for OPC UA alarm monitoring.

## To enable alarm and event in FactoryTalk Linx OPC UA Connector:

1. In FactoryTalk Linx OPC UA Connector, on the **OPC UA Servers** tab, under **General**, specify the **Endpoint URL**.  
**Communicating with server** shows indicating that the connection is established.
2. Under **Alarm Settings**, select **Enable alarm and event support**.  
**Communicating with server** shows indicating that the connection is established.
3. Click **Apply**, and then click **OK**.

## Setting up alarm and event history logging

FactoryTalk Alarms and Events generates messages to record the alarm activity that occurs at tag-based, device-based, and OPC UA alarm servers in an application.

The Alarm and Event Historian software, which is installed with FactoryTalk Alarms and Events, logs alarm information to the databases set up to store logged data. It also manages connections and data buffering between alarm servers and their databases.



**Tip:** FactoryTalk Alarms and Events also generates audit and diagnostic messages. For information about logging and viewing these messages, see the FactoryTalk Alarms and Events Help.

The following are the tasks involved in setting up FactoryTalk alarm logging:

- Install Microsoft SQL Server and supporting components, on computers that will host alarm and event log databases. For details, see the FactoryTalk Alarms and Events Help.
- Create database definitions at the FactoryTalk Directory. For more information, see [Defining an alarm and event log database on page 196](#).
- Set up support for alarm and event history logging for each alarm server in an application. For more information, see [Enabling alarm and event history logging on page 197](#).
- Set up alarm and event log viewers to display logged data at runtime. For more information, see [Viewing alarm and event history logs on page 201](#).

## Defining an alarm and event log database

The FactoryTalk Alarm and Event Historian routes FactoryTalk alarm-related data from alarm servers to Microsoft SQL Server databases. Information is logged when:

- A tag goes into alarm or returns to normal.
- An alarm is acknowledged.
- An alarm is suppressed or unsuppressed.
- An alarm is enabled or disabled.
- An alarm fault is generated or cleared.
- An alarm is reset.
- An alarm is shelved or unshelved.

Database definitions are stored in the FactoryTalk Directory, and can be shared by multiple alarm servers in an application.

You specify the database definition that an alarm server will use in the server's Properties dialog box. For more information, see [Enabling alarm and event history logging on page 197](#).

## To create an alarm and event log database definition:

1. In FactoryTalk View Studio, in the Explorer window, expand the System and Connections folders.
2. Right-click the Databases folder, and then select **New Database**.

For details about options in the Alarm and Event Historian Database Properties dialog box, click **Help**.

## Enabling alarm and event history logging

To set up an alarm server to log alarm activity, for each Rockwell Automation Device Server (FactoryTalk Linx), and Tag Alarm and Event Server in an application, you must enable alarm and event history logging, and then select a database for storing the data.

You do this in the alarm server's Properties dialog box, in the Alarm and Event History section. For an illustration, see [Setting up alarm priorities and history logging on page 192](#).



**Tip:** For a FactoryTalk Linx server, Alarm and Event History settings are in the Alarms and Events tab. For a Tag Alarm and Event server, the settings are in the Priorities and History tab.

## To view the alarm servers using a database definition:

1. In FactoryTalk View Studio, in the Explorer window, expand the System, Connections, and Databases folders.
2. Right-click the database definition that you want to check, and then select **Properties**.
3. In the Alarm and Event Historian Database Properties dialog box, click **Show Usage**.

## Setting up alarm and event displays

To monitor and respond to FactoryTalk alarms at runtime, use FactoryTalk alarm and event objects hosted in FactoryTalk View SE graphic displays.

In FactoryTalk View Studio, in the Graphics editor, you can create the following objects:

- Use the **alarm and event summary** to view, acknowledge, suppress, and disable alarms from multiple FactoryTalk alarm sources. For information about setting up an alarm and event summary, see [Setting up an alarm and event summary on page 198](#).
- Use the **alarm and event banner** to monitor and acknowledge the most serious FactoryTalk alarms in the system. For information about setting up an alarm and event banner, see [Setting up an alarm and event banner on page 198](#).
- Use the **automatic diagnostic event summary** to suppress, unsuppress, filter, and sort diagnostic events at runtime. For information about setting up an automatic diagnostic event summary, see [Setting up an automatic diagnostic event summary on page 200](#).
- Use the **alarm and event log viewer** to view, sort, filter, and print historical alarm information. For information about the alarm and event log viewer, see [Viewing alarm and event history logs on page 201](#).
- Use the **alarm status explorer** to view alarm sources, suppress or unsuppress, and enable or disable alarms. For information about the alarm status explorer, see [Working with alarm sources in the Alarm Status Explorer on page 200](#).

For detailed setup instructions, see the FactoryTalk Alarms and Events Help.

## Setting up an alarm and event summary

When setting up an alarm and event summary, you specify which alarm information will be shown at runtime, and how the information will appear in the summary.

For example, you can specify a different color for each priority—blue for low, yellow for medium, orange for high, and red for urgent—so that operators can identify the priority of alarms at a glance.

A FactoryTalk alarm and event summary can occupy part of a FactoryTalk View SE graphic display, or it can fill the entire display.

### To create an alarm and event summary object:

1. In FactoryTalk View Studio, in the Explorer window, expand the application, area, and HMI server icons, where you want to create the alarm and event summary display.
2. To create a graphic display, expand the Graphics folder, right-click Displays, and then click **New**.
3. In the Graphics editor, select **Objects > Alarm and Event > Summary**. You can also select Alarm and Event Summary in the Toolbox.
4. Drag the alarm and event summary tool on the graphic display, to the desired size and shape, and then release the mouse.

You can move and resize the Alarm and Event Summary Design View object, as needed. To set up properties, double-click the object (or right-click, and then click **Properties**).

For details about options in the Alarm and Event Summary Properties dialog box, click **Help**.

## Setting up an alarm and event banner

When setting up an alarm and event banner, you specify which alarm information will be shown at runtime, and how the information will appear in the banner.

For example, to notify operators when the most severe alarms occur, set up these alarms to blink and sound an alarm bell. To further indicate severity, adjust the alarm bell beep rate.

A FactoryTalk alarm and event banner can be docked to the inside of the FactoryTalk View SE Client window. For more information, see [Docking a banner display to the runtime client window on page 199](#).

### To create an alarm and event banner:

1. In FactoryTalk View Studio, in the Explorer window, expand the application, area, and HMI server icons, where you want to create the alarm and event banner display.
2. To create a graphic display, expand the Graphics folder, right-click Displays, and then select **New**.
3. In the Graphics editor, select **Objects > Alarm and Event > Banner**. You can also select Alarm and Event Banner in the Toolbox.
4. Drag the alarm and event banner tool on the graphic display, to the desired size and shape, and then release the mouse.

You can move and resize the Alarm and Event Banner Design View object, as needed. To set up properties, double-click the object (or right-click, and then click **Properties**).

For details about options in the Alarm and Event Banner Properties dialog box, click **Help**.

## About the alarm and event banner at runtime

The alarm and event banner shows only the most serious alarms and events in the system. If you want operators to focus on alarms that need immediate attention, provide this object.

You can set up the appearance and behavior of the alarm banner, but it can only show up to five of the most recent, most severe alarms. An operator can use the alarm banner to:

- Acknowledge the selected alarm (and enter an optional comment).
- Silence alarms.
- Enable or disable the alarm bell.



**Tip:** The alarm bell is a feature of the alarm and event banner only. In an alarm and event summary, alarms cannot be set up to sound a bell when they occur.

- Run the command associated with an alarm.
- Refresh the alarm list.
- Open an alarm and event summary.

For more information, see the FactoryTalk Alarms and Events Help.

## Docking a banner display to the runtime client window

At runtime, graphic displays can be docked to an edge of the FactoryTalk View SE Client window, so that they are available to operators at all times.

For example, to let an operator continuously monitor the most serious alarms in an application, dock an alarm banner display to the top or bottom of the client window.

### Example: Docking an alarm banner when the FactoryTalk View SE Client starts up

To dock an alarm banner display to the bottom edge of the FactoryTalk View SE Client window when the client starts up, follow these steps:

1. In FactoryTalk View Studio, create a macro that contains the command

```
Display <Alarm Banner> /DB
```

where *Alarm Banner* is the name of the graphic display that contains the alarm banner.



**Tip:** /DB is the **Display** command parameter for docking a display at the bottom of the FactoryTalk View SE Client window. For a full list of docking parameters, see [Display command parameters for docking displays on page 285](#).

2. In the FactoryTalk View SE Client Wizard, select the macro that contains the **Display** command for docking the banner.
3. Save the client configuration file.

When you start the client, the alarm banner will be attached to the bottom edge of the client window.

Opening and closing other graphic displays will not interfere with the position or appearance of the docked banner display.

## Setting up an automatic diagnostic event summary

When setting up an automatic diagnostic event summary, you specify which diagnostic events will be shown at runtime, and how the information will appear in the automatic diagnostic event summary.

For example, you can specify a different text color for each event type—blue for Active and Unsuppressed, black for Active and Suppressed, and gray for Inactive and Suppressed—so that operators can identify the type of diagnostic events at a glance.

A FactoryTalk automatic diagnostic event summary can occupy part of a FactoryTalk View SE graphic display, or it can fill the entire display.

### To create an automatic diagnostic event summary:

1. In FactoryTalk View Studio, in the Explorer window, expand the application, area, and HMI server icons, where you want to create the automatic diagnostic event summary display.
2. To create a graphic display, expand the Graphics folder, right-click Displays, and then click **New**.
3. In the Graphics editor, select **Objects > Alarm and Event > Automatic Diagnostic Event Summary**. You can also select Automatic Diagnostic Event Summary in the Toolbox.
4. Drag the automatic diagnostic event summary tool on the graphic display, to the desired size and shape, and then release the mouse.

You can move and resize the Automatic Diagnostic Event Summary Design View object, as needed. To set up properties, double-click the object (or right-click, and then click **Properties**).

For details about options in the Automatic Diagnostic Event Summary Properties dialog box, click **Help**.

## Working with alarm sources in the Alarm Status Explorer

The alarm status explorer shows all of the FactoryTalk Alarms and Events alarm sources and alarms in an application.

Use the alarm status explorer to view alarm sources, and to suppress, unsuppress, enable, or disable alarms. The alarm status explorer is the only FactoryTalk alarm and event object that you can use to unsuppress and enable alarms.



**Tip:** You can also use alarm tags to suppress, unsuppress, enable, and disable alarms. For more information, see [Using tags to interact with alarms or obtain their status on page 201](#).

### To create an alarm status explorer:

1. In FactoryTalk View Studio, in the Explorer window, expand the application, area, and HMI server icons, where you want to create the alarm status explorer display.
2. To create a graphic display, expand the Graphics folder, right-click Displays, and then select **New**.
3. In the Graphics editor, select **Objects > Alarm and Event > Status Explorer**. You can also select Alarm Status Explorer.
4. Drag the alarm status explorer tool on the graphic display to the desired size and shape, and then release the mouse. You can move and resize the alarm status explorer object, as needed.
5. To set up properties, right-click the object, and then click **Properties**. For details about options in the Alarm Status Explorer Properties dialog box, click **Help**.

To set the width of the application and area display, and of columns in the alarm source list, double-click the alarm status explorer object. This places the object in Edit mode, so that you can drag column headers and the vertical split bar.



**Tip:** The design-view alarm status explorer contains a default tree-view of an application named Line1, with Machine1 and Machine2 areas, and subareas. To view the actual application, test the graphic display in FactoryTalk View Studio, or run it in a FactoryTalk View SE Client.

## Viewing alarm and event history logs

The Alarm and Event Log Viewer shows the contents of the alarm and event history logs.

When you create an alarm and event log viewer, you select which local or remote database to query, from the list of database definitions held in the FactoryTalk Directory.

### To create an alarm and event log viewer:

1. In FactoryTalk View Studio, in the Explorer window, expand the application, area, and HMI server icons, where you want to create the alarm and event log viewer.
2. To create a graphic display, expand the Graphics folder, right-click Displays, and then select **New**.
3. In the Graphics editor, select **Objects > Alarm and Event > Log Viewer**. You can also select Alarm and Event Log Viewer in the Toolbox.
4. Drag the alarm and event log viewer tool on the graphic display to the desired size and shape, and then release the mouse.

The Alarm and Event Log Viewer Design View object is placed in the display. You can move and resize the object as needed.

To set up properties, double-click the object. For details about options in the Properties dialog box, click **Help**.

## Using tags to interact with alarms or obtain their status

Alarm tags let you monitor and interact with alarms without the need for an alarm and event object in a graphic display.

Use alarm tags when an alarm display, such as an alarm and event summary, is not available for responding to alarms. You can use alarm tags to acknowledge, enable, disable, suppress, unsuppress, or reset alarms.

For example, to let an operator acknowledge a tag-based alarm, expose the alarm as a tag, and then provide a way for the operator to set the OperAck (or OperAckAll) alarm tag to 1.

You can also use alarm tags to obtain information about an alarm's state. For a list of alarm tags and their functions, and for information about device-based alarm tags, see the FactoryTalk Alarms and Events Help.

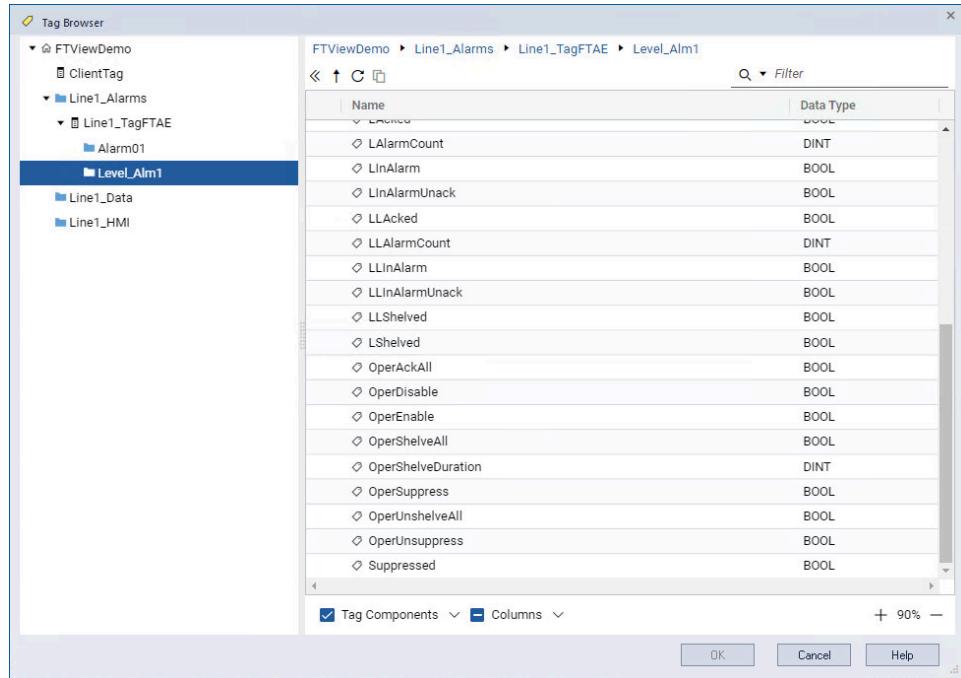


**Tip:** You can also use FactoryTalk alarm functions in FactoryTalk View expressions, to obtain information about alarms. For more information, see [Using FactoryTalk alarm functions in expressions on page 203](#).

### To expose an alarm as a tag:

1. In the Alarm and Event Setup editor, open the Alarm Properties dialog box for the digital, level, or deviation alarm you want to expose as a tag.
2. In the Digital, Level, or Deviation tab, select the checkbox, Show Alarm as a Tag.

The next illustration shows how alarm tags are shown in the Tag Browser. Level\_Alm1 is a tag-based alarm definition exposed as a tag. When you select it in the Tag Browser, its alarm tags become available on the right. You can use the Oper\* tags to let an operator respond to instances of the alarms at runtime.



### Example: Using alarm tags to acknowledge an alarm

An operator is expected to acknowledge alarms associated with a faceplate in a graphic display that does not contain an alarm and event summary or banner.

To let the operator acknowledge High High alarm conditions for the alarm, named Level\_Alm1, you might try the following these steps:

1. In the Alarm and Event Setup editor, open the Properties dialog box for Level\_Alm1, and then select the checkbox, Show Alarm as a Tag.
2. In FactoryTalk View Studio, open the graphic display that the operator will be monitoring, and then create this text object: **High High Alarm**.  
In the text object's Properties dialog box, select red as the Fore color. Make the object big enough to catch the operator's attention when it becomes visible.
3. Right-click the text object, select **Animation > Visibility**.

4. To make the **High High Alarm** object visible when a High High alarm occurs for Level\_Alm1:
  - In the Expression box, type Level\_Alm1.HHInAlarm.
  - For the Expression true state, select Visible.
5. In the Animation dialog box, click the Touch tab. To let the operator acknowledge the alarm by clicking the **High High Alarm** object:
  - In the Press action box, type Optionally, run a command or macro as the release action, to notify the operator's supervisor when the alarm is acknowledged.

At runtime, when the **High High Alarm** object becomes visible, the operator can press the object to acknowledge the alarm.

The object will remain visible until the alarm returns to normal. When that happens, the HHInAlarm status tag is reset to 0, making the **High High Alarm** text object invisible.

For more information about setting up animation for FactoryTalk View graphic objects, see [Creating graphic objects on page 288](#).

## Using FactoryTalk alarm functions in expressions

To retrieve information about FactoryTalk alarms for individual tags, or for groups of tags, use FactoryTalk alarm functions in expressions.

You can use the FactoryTalk alarm functions wherever expressions are supported in a FactoryTalk View SE application. For example, use an alarm function to set up a numeric display object, or define a derived tag.

For information about creating expressions, see [Creating expressions on page 344](#).

## Retrieving information about the severity of alarms

FactoryTalk View SE provides alarm functions that you can use in expressions to retrieve the highest severity value of specified alarms that are acknowledged or unacknowledged.

Use the high severity functions to alert operators when high-severity alarms occur. For example, you could set up a graphic object to turn red and blink, if the specified function returns a value greater than 900 for alarms in the operator's area.

## Highest severity value of acknowledged alarms

The following expression returns the highest severity value associated with the specified alarms, for alarms that have been acknowledged:

```
AE_HighSeverityAcked("Filter")
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms specified has the state In Alarm and Acknowledged, the expression result is an integer, from 1 to 1000. If none of the alarms has the state In Alarm and Acknowledged, the expression result is 0.

## Highest severity value of unacknowledged alarms

The following expression returns the highest severity value associated with the specified alarms, for alarms that are unacknowledged:

```
AE_HighSeverityUnAcked( "Filter" )
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms specified has the state In Alarm and Unacknowledged, the expression result is an integer from 1 to 1000. If none of the alarms is In Alarm and Unacknowledged, the expression result is 0.

## Retrieving information about the number of alarms

FactoryTalk View SE provides alarm functions that you can use in expressions to retrieve the number of specified alarms that are in different states.

Use the alarm count functions, for example, to alert operators when a number of alarms are waiting to be acknowledged.

Each alarm condition for an alarm is counted. For example, if both the High and High High alarm conditions are active for a level alarm the In Alarm count will be 2.

## Number of disabled alarms

The following expression returns the number of specified alarms that have been disabled:

```
AE_DisabledCount( "Filter" )
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms has the state **Disabled**, the expression result is an integer, from 1 to the total number of alarms specified. If none of the alarms is **Disabled**, the expression result is 0.

## Number of shelved alarms

The following expression returns the number of specified alarms that have been shelved:

```
AE_InAlmShelvedCount( "Filter" )
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms has the state **In Alarm** and **Shelved**, the expression result is an integer, from 1 to the total number of alarms specified. If none of the alarms is **In Alarm** and **Shelved**, the expression result is 0.

## Number of shelved alarms returned to normal

The following expression returns the number of specified alarms that are shelved, but have returned to normal:

```
AE_NormalShelvedCount( "Filter" )
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms specified has the state **Normal** and **Shelved** and none of the alarms specified is disabled, the expression result is an integer from 1 to the total number of FactoryTalk alarms in the system. If none of the alarms is **Normal** and **Shelved**, the expression result is 0.

## Number of suppressed alarms

The following expression returns the number of specified alarms that have been suppressed:

```
AE_InAlmSuppressedCount("Filter")
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms has the state **In Alarm** and **Suppressed**, the expression result is an integer, from 1 to the total number of alarms specified. If none of the alarms is **In Alarm** and **Suppressed**, the expression result is 0.

## Number of suppressed alarms returned to normal

The following expression returns the number of specified alarms that are suppressed, but have returned to normal:

```
AE_NormalSuppressedCount("Filter")
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms specified has the state **Normal** and **Suppressed** and none of the alarms specified is disabled, the expression result is an integer from 1 to the total number of FactoryTalk alarms in the system. If none of the alarms is **Normal** and **Suppressed**, the expression result is 0.

## Number of acknowledged alarms

The following expression returns the number of specified alarms that have been acknowledged:

```
AE_InAlmAckedCount("Filter")
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms has the state **In Alarm** and **Acknowledged**, the expression result is an integer, from 1 to the total number of alarms specified. If none of the alarms is **In Alarm** and **Acknowledged**, the expression result is 0.

## Number of unacknowledged alarms

The following expression returns the number of specified alarms that are unacknowledged:

```
AE_InAlmUnackedCount("Filter")
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms has the state In Alarm and Unacknowledged, the expression result is an integer from 1 to the total number alarms specified. If none of the alarms is In Alarm and Unacknowledged, the expression result is 0.

## Number of unacknowledged alarms returned to normal

The following expression returns the number of specified alarms that are unacknowledged, but have returned to normal:

```
AE_NormalUnackedCount("Filter")
```

where *Filter* is an absolute or relative path to one or more FactoryTalk alarms. For information about specifying the alarm source, see [Specifying the FactoryTalk alarm source in an expression on page 206](#).

If any of the alarms specified has the state Normal and Unacknowledged, the expression result is an integer from 1 to the total number of FactoryTalk alarms in the system. If none of the alarms is Normal and Unacknowledged, the expression result is 0.

## Specifying the FactoryTalk alarm source in an expression

When you use a FactoryTalk alarm function in an expression, you must specify the alarm source in double quotes.

In the Expression editor, you can type the path to the alarm source, or click **Alarm** to open to find and select alarms. For details about using the Alarm Source Browser, see the FactoryTalk Alarms and Events Help.

## Using absolute and relative references to alarms

In a FactoryTalk alarm function, you can specify an absolute or a relative path to one or more alarms.

### Absolute path format

The format for the absolute path to an alarm is:

- /<AreaPath>::<Filter>
- /<GroupPath>::<Filter>

where

- *AreaPath* includes the application and area path, but not the application name.
- *GroupPath* includes the application, area path, group path, but not the application name.
- *Filter* specifies a single alarm name, or uses the wildcard characters ? and \* (question mark and asterisk) to specify multiple alarms.

For example, /MixerArea/MixerLine1:[CLX1]Line1MixingTank1, is a valid absolute reference to an alarm named [CLX]Line1MixingTank1.

### Relative path format

The format for the relative path is:

```
[ :: ]<Filter>
```

where

- *Filter* indicates the name of the alarm.

In a relative reference, the colons (:) preceding the alarm name are optional. For example, MixingTank1 is a valid relative reference to an alarm named MixingTank1.

For more information about using absolute and relative references in FactoryTalk View SE applications, see [Absolute and relative references on page 118](#).

## Using wildcards to specify multiple alarms

In a FactoryTalk alarm function, you can use wildcard characters in the alarm name portion of the path to the alarm source, to retrieve information about multiple tags at once.

This wildcard character	Does this
?	Matches any single character.
*	Matches any number of characters, including the backslash (\) character.

For example, the following expression will return the total number of In Alarm and Acknowledged alarms, for all alarms in the Studio 5000 device referenced by the shortcut *MixerController*, in the subarea named *MixerLine1*, in the area named *MixerArea*:

```
AE_InAlmAckedCount (" /MixerArea/MixerLine1::[MixerController]*" )
```



**Tip:** You cannot use wildcard characters in the area portion of an alarm source path. For example, you cannot specify /MixerArea/Mixer\*: for the area path, in the previous example.

## Using tag placeholders to specify the alarm name

In a FactoryTalk alarm function, you can use tag placeholders in the alarm name portion of the path to the alarm source.

A tag placeholder is the crosshatch character (#) followed by a number from 1 to 500. A tag placeholder can also contain wildcard characters and folder names.

The tag definition in the following example contains the placeholder #1:

```
AE_InAlmAckedCount (" /MixerArea/MixerLine1::[CLX]Line1.Mixer_#1" )
```

At runtime, tag placeholders in a graphic display are replaced with the values of tags specified in parameter files or parameter lists, which are loaded with the display.

For more information, see [Using placeholders to specify tag values on page 275](#).

## What happens if the alarm source becomes unavailable

When a FactoryTalk alarm function in an expression is first initialized, only alarms from connected alarm servers are counted in the evaluation.

At runtime, if a FactoryTalk alarm function refers to an alarm source that becomes unavailable, the function is evaluated but returns nothing to the expression evaluation.

For example, a device-based alarm function is not counted, if the corresponding alarm is deleted from the Studio 5000 controller, or if the FactoryTalk Linx alarm server:

- Loses its connection to the Studio 5000 controller that is detecting alarms.
- Begins or is in the process of downloading a program.
- Switches from run to test program mode.
- Experiences a major or nonrecoverable program fault.
- Cannot subscribe to all of the alarms at the Studio 5000 controller.

If the condition causing the disconnection is resolved, the expression can be reevaluated, with the restored alarm function included.

## About controller status system alarms

When an alarm source becomes unavailable, the system generates a special type of alarm, called a controller status alarm, in alarm and event displays and logs.

Controller status alarms provide information about the condition behind an alarm source disconnection.



**Tip:** An expression that contains a FactoryTalk alarm function will return 0, if the alarm function references an alarm from a disconnected controller. Provide a way for operators to monitor for controller status alarms, so that they can confirm whether 0 is a valid expression value.

For details about controller status alarms and the conditions that cause them, see the FactoryTalk Linx Help.

## Importing and exporting alarms

Tag-based alarms and alarm messages can be exported from and imported into a Tag Alarm and Event Server.

For example, you might export and import alarms and messages to:

- Move or copy alarms from one Tag Alarm and Event Server to another.
- Create alarms using programming tools such as Microsoft Visual Basic or C++, and then import the program's output file.
- Add alarms to an exported alarm file, using Microsoft Excel. You can then import the file with the new alarms.
- Modify alarms in an exported file, by searching for one set of tags, and then replacing the tags with another set. You can then import the file with the modified alarms.
- Translate alarm messages into other languages.
- Generate reports about alarms.
- Convert alarms from one alarm and events system to another, using conversion programs.

For details about importing and exporting alarms and messages, see the FactoryTalk Alarms and Events Help.

## To export alarms and messages:

1. In FactoryTalk View Studio, in the Explorer window, right-click the Tag Alarm and Event Server containing the alarms you want to export, and then select **Import and Export**.
2. In the Alarm Import and Export Wizard, click the export operation that uses the desired format, and then click **Next**.
3. To complete the export operation, follow the instructions on the screen.  
For details about options in the Alarm Import and Export Wizard, click **Help**.

## To import alarms and messages:

1. In FactoryTalk View Studio, in the Explorer window, right-click the Tag Alarm and Event Server, into which you want to import tags, and then select **Import and Export**.
2. In the Alarm Import and Export Wizard, click the import operation that uses the desired format, and then click **Next**.
3. To complete the import operation, follow the instructions on the screen.  
For details about options in the Alarm Import and Export Wizard, click **Help**.

## About import and export formats

FactoryTalk Alarms and Events supports Microsoft Excel Workbook (.xls) and XML (.xml) import and export formats.

Use Microsoft Excel to modify existing alarms, for example, to translate text strings into foreign languages, or to search and replace text across multiple alarms.

Use XML to create alarms programmatically, using third-party tools, and then generate an XML file that you can import.

You can also use the XML format to back up alarms, or to move definitions from one Tag Alarm and Event Server to another.



**Tip:** The XML format is recommended for importing or exporting large numbers of alarms.

# Setting up language switching

This chapter describes:

- What language switching is.
- Text strings you can and cannot view in different languages at runtime.
- The basic steps involved in setting up language switching.
- Setting up font support for Windows languages.
- Selecting a language for a new FactoryTalk View application.
- Adding languages to an application.
- Setting up the default application language.
- Exporting application text strings.
- Working with exported Microsoft Excel spreadsheets.
- Working with exported Unicode text files.
- Importing translated or modified text strings.
- Setting up runtime language switching.
- Support for multiple languages in the graphic libraries.

## About language switching

Language switching allows operators to view user-defined text strings in an application in up to 40 different languages. At runtime, a FactoryTalk View SE Client can switch between any of the languages the application supports. In a network distributed application, multiple clients can also run in different languages at the same time.

With language switching, you can:

- Develop an application in a language, export the application's text strings, and then import strings translated into multiple languages back into the application.
- Deploy the same network distributed application in different countries.
- Provide operators in multilingual countries with the ability to choose which language they use.
- Import application components developed in different countries into a single application that supports multiple languages.

In FactoryTalk View Studio, use the Language Configuration dialog box to export text strings for one language, or for all languages in an application.

You can translate or modify the exported strings, and then import them back into the application. For information about:

- Exporting text strings, see [Exporting application text strings on page 217](#).
- Importing translated text, see [Importing translated or modified text strings on page 224](#).

## Text strings you can view in different languages at runtime

In applications that support multiple languages, the following types of text strings can be viewed in different languages at runtime:

- Captions, tool tip text, time and date embedded variables, and numeric embedded variables defined for FactoryTalk View graphic objects
- Titles typed in the Display Settings dialog box, for FactoryTalk View graphic displays

- FactoryTalk View local messages
- FactoryTalk View text objects

**Tip:**

If a Text object has its **Size to Fit** property set to **True**, the object's size might vary for different languages.

- Text strings defined for FactoryTalk Alarm and Event summaries, banners, and log viewers. For details, see the FactoryTalk Alarms and Events Help.
- Hard-coded text strings defined for TrendPro and XY Plot objects, such as menu tooltips and column names
- Hard-coded text strings defined for signing in, password, and electronic signature dialog boxes
- Text strings defined in the extended properties of controller tags

**NOTE:** Make sure that you define the controller default language when multi-language is enabled.

- If the client-requested language is not available in the controller, the application shows the default language defined and enabled in the HMI.
- If the HMI default language is not defined in the controller, the application uses the controller default language or the single language that does not have an ID.

## Specifying time, date, and numeric formats

During development or at runtime, FactoryTalk View uses the Windows regional settings for the current application language when you enter, modify, or display information (such as dates and numbers) in an application.

You can also change the formats in the Windows Region and Language so that the display of FactoryTalk View matches the standards or language used in the country or region where you are located.

For example, for a numeric display object at runtime, the decimal number **123.45** is shown as:

- **123.45**, if the current application language is English (United States).
- **123,45**, if the current application language is German (Germany).
- **123,45**, if the current application language is German (Germany), and in the Windows regional settings, you customize the decimal symbol of the German (Germany) format to a period (.)

## Text strings you cannot view in different languages

In applications that support multiple languages, some text strings will not be able to switch languages at runtime.

## Exported text that does not support language switching

If you export text strings for a single application language to a Unicode text file, the caption property of any third-party ActiveX objects will be exported. However, even if these strings are translated, they cannot be viewed in different languages at runtime.

## Text that cannot be exported

The following user-defined text strings are visible at runtime, but cannot be exported for translation:

- Command strings, including those you type for the **Remark** command
- Tag descriptions for tags in the HMI tag database
- Alarm Fault List messages shown in FactoryTalk Alarm and Event summaries, banners, and log viewers
- String constants in expressions, and in all components that use expressions  
For example, you might use string constants in expressions if an application displays different strings in a string display object, based on the result of an expression.
- The text shown in the title bar of the FactoryTalk View SE Client window

## Text that is part of the FactoryTalk View software

The following text strings are part of the FactoryTalk View Studio or FactoryTalk View SE Client software, and cannot be exported for translation:

- Tooltips on buttons belonging to the FactoryTalk View Studio or FactoryTalk View SE Client software, for example, the buttons on FactoryTalk View Studio toolbars
- Text in the on-screen keyboard or numeric keypad
- Text in the FactoryTalk View SE Client Login dialog box
- Text in the RecipePro+ dialog box

## Text shown in FactoryTalk View SE system tags

You can use system tags in an application to show text in graphic displays. For example, the tag, system \dateandTimeString, shows the current date and time.

The changes to Windows regional settings are not reflected in the following tags. The format used is defined by the operating system and cannot be changed.

- System\Time
- System\Date
- System\DateAndTimeString
- System\MonthString
- System\AlarmMostRecentDate
- System\AlarmMostRecentTime
- System\AlarmStatus

## Text that is not visible at runtime

Some text that you define for an application is never visible to an operator at runtime. For that reason, FactoryTalk View deliberately does not export this text. The properties of graphic objects shown in the following table are such text strings.

In this product	This graphic object	Has these properties defined using text that cannot be exported
FactoryTalk View Site Edition and FactoryTalk View Machine Edition	all objects	Name Description

In this product	This graphic object	Has these properties defined using text that cannot be exported
		Image St_Image LinkedObject
	Display list selector	St_Dispatch St_Parameter
FactoryTalk View Site Edition	Legacy Recipe	Value
	RecipePro+	All properties
	Numeric input and string input	TagName
	Numeric display and string display	Expression
	Tag label	TagName PropertyName Caption
	Trend	All properties, <b>except</b> ChartTitle, Pen Description, and EngineeringUnits.  These three properties <b>are</b> exported.
FactoryTalk View Machine Edition	Goto display button	Display Parameter
	Local message display	MessageFile
	Alarm list, alarm banner, and alarm status list	FilteredTriggers
	Macro button	Macro

## Summary of steps for setting up language switching

These are the tasks involved in setting up language switching for an application:

1. Install the Windows languages that the application will use.
2. Select a language in which to create the application and its components.
3. Add the languages that you want the application to support.



**Tip:** You can add languages to an application before or after you export the application's text strings. To translate multiple languages at once, add languages before exporting the text strings.

4. Export the application text strings for translation.

5. Import translated text strings for each of the supported languages.
6. Provide ways for operators to switch between the application languages at runtime.

To set up FactoryTalk View SE Clients:

1. On client computers, install the Windows languages that the application supports.
2. In client setup files, specify an initial language for the client to run in.

For information about:

- Exporting application text strings, see [Exporting application text strings on page 217](#).
- Importing translated strings, see [Importing translated or modified text strings on page 224](#).
- Setting up language switching at runtime, see [Setting up runtime language switching on page 230](#).

## About FactoryTalk Security permissions

If FactoryTalk Security services are used to secure parts of an application, to perform certain tasks, users must have the necessary security permissions.

For example, to create or modify the properties of an application, you must at least be allowed the Common actions Read, List Children, Write, and Create Children, at the FactoryTalk Directory that manages the application.

If you receive a FactoryTalk Security message while trying to perform such a task, contact your system administrator about the permissions you might require.

For an overview of FactoryTalk Security services, see [Setting up security on page 77](#).

## Setting up font support for Windows languages

The operating systems supported by FactoryTalk View SE support all of the necessary fonts by default. To make language switching work, you do not need to install any Language Interface Packs (LIP).

## Selecting a language for a new FactoryTalk View application

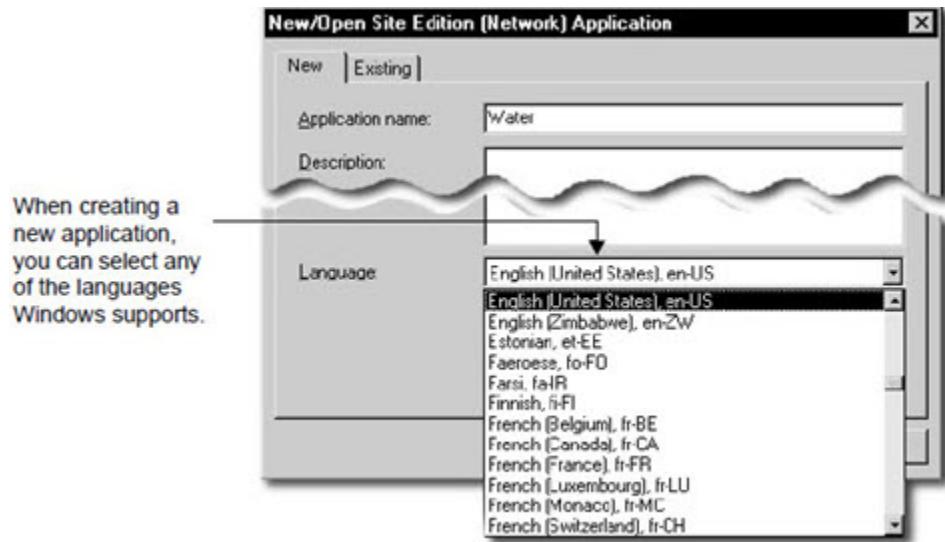
When you create a FactoryTalk View application, you must select a language for the application. The language you select becomes the default application language.

After setting up components for the application, such as data servers, HMI servers, and graphic displays, you can export text strings in the application for translation. Then, you can import translated strings for the languages you add to the application.

### To select a language when creating an application:

1. Start FactoryTalk View Studio, select an application type, and then click **Continue**.
2. In the New/Open Site Edition (...) Application dialog box, click the New tab
3. Type an application name, select an application language, provide an optional description, and then click **Create**.

When creating an application, you select a language from the full list of languages that Windows supports, as shown in the following illustration.



## About the current application language

When you open an existing application and select a language from the list of languages the application supports, that language becomes the current application language.

For example, if an application supports English and Spanish, and you select Spanish when you open the application, then Spanish is the current application language.

This means that when you save application components, such as graphic displays, the text strings they contain will be saved in Spanish. Text strings in the application that are unavailable in Spanish, will be treated as undefined.

Undefined text strings in an application show as question marks (?), unless the application is set up to show undefined strings in the default language. For more information, see [Setting up a default application language on page 216](#).

## Adding languages to an application

Use the Language Configuration dialog box to add up to 40 languages that you want the application to support. You can add languages before or after you export the application's text strings for translation.

To translate multiple languages at once, add languages to the application first, and then export text strings for all the languages, to a Microsoft Excel spreadsheet. For details, see [Exporting application text strings on page 217](#).



**Tip:** You cannot switch dynamically between application languages while developing an application. To change the current language, you must close the application, and then select a different language when you reopen the application.

### To add languages to an application:

1. In **FactoryTalk View Studio**, click **Tools > Languages**.
2. In the **Language Configuration** dialog box, in the **General** tab, click **Add**.

3. In the **Add Language** dialog box, select the language to be added to the application, and then click **OK**.  
The **Add Language** dialog box displays the name and RFC1766 identifier of each of the languages supported by the design-time computer. If the selected language is not installed, a warning message is displayed.

For details about options in the Language Configuration dialog box, click **Help**.

## Setting up a default application language

The language you select when creating an application becomes the default language for the application.

### To change the default language:

1. In FactoryTalk View Studio, click **Tools > Languages**.
2. In the **Language Configuration** dialog box, select the language the default is to be changed to.
3. Select **Set As Default**. A check mark is displayed beside the newly selected default language.

A check mark beside the language you selected indicates that it is the default application language. For information about using the default language to show undefined text strings, see "Showing undefined text strings in the default language," next.

When you open an existing FactoryTalk View SE application, if the application supports several languages and no default is specified, the following dialog box opens, so that you can select a default language.



## Showing undefined text strings in the default language

At runtime, if a text string is undefined for the language an application is currently running in, it is displayed as a single question mark (?), unless you have specified that undefined strings are to be displayed using strings from the application's default language.

If you prefer undefined strings to display in a specific language, in the Language Configuration dialog box, set up that language as the default application language, and then select the checkbox, **Display undefined strings using the default language**.

For example, use this option to always view displays used for maintenance purposes in English, regardless of the current application language.

Using this option will also ensure that local message strings show correctly when you switch application languages at runtime.

## Exporting application text strings

Use the Language Configuration dialog box to export text strings in an application, in order to translate or modify them.

If the application contains multiple languages, you can export text strings for the selected language only, or for all languages in the application.

To export strings only for the currently selected (highlighted) language, and save them to a text file in Unicode format, select the option, Export strings for <aa-BB> to one or more Unicode text files (where aa-BB is the RFC1766 code for the selected language).

For details about modifying Unicode text files in Microsoft Excel and in Notepad, see the FactoryTalk View SE Help.

To export strings for all languages in the application, and save them to a Microsoft Excel spreadsheet, select the option, Export strings for all languages to an Excel spreadsheet. This option lets you translate or modify multiple languages at the same time.

If you export all strings in an application, you can choose to optimize duplicate strings during the export operation. This means that any string that occurs multiple times in the application will be shown only once in the Excel spreadsheet. For more information, see [Working with duplicate text strings on page 221](#).



**Tip:** If the application contains only one language, you can still select the option Export strings for all languages to an Excel spreadsheet, if you want to save strings to a spreadsheet instead of a Unicode text file.

### To export text strings in an application:

1. Make sure that the HMI project that you are going to export text from is not being edited or modified. If components are being edited remotely and have not been saved, the exported file will not contain the unsaved changes.
2. In **FactoryTalk View Studio**, click **Tools > Languages**.
3. If you are exporting to Unicode text files, in the **Language Configuration** dialog box select the language for which the strings are being exported, and then click **Export**.  
If you are exporting to an Excel spreadsheet, simply click **Export**.
4. In the **Export Operation** dialog box, select whether to export the strings to Unicode files or an Excel spreadsheet.  
If you are exporting to an Excel spreadsheet, you can also choose to optimize the strings being exported and to open the file in Excel as soon as the export is complete.
5. If you are exporting strings from a network distributed application or a network station application, the **Select Servers** window opens. Select the checkbox for each HMI server from which you want to export text, and then click **Next**.  
If you are exporting text from a local station application, this step does not apply and the wizard goes directly to the **Select Destination** window.
6. For either local station, network station or network distributed applications, in the **Select Destination** window, click **Finish** to export text to files in the default folder.
7. To export text to files in another folder, click the **browse button (...)** to navigate to a folder, and then click **Finish**.

## Troubleshooting export problems

If errors occur during an export operation, or if you cancel the export while it is in progress, a message is logged to the Diagnostics List and log file.

In addition, export errors are shown in a log file called ExportErrors.txt, which is saved in the folder **C:\Users\Public\Public Documents\RSView Enterprise\Strings** on the local computer.



**Tip:** Existing export error log files are overwritten for each subsequent export operation that generates errors.

## Export file formats

During the export operation, files containing the exported text strings are saved by default in the Strings folder on the local computer. For the location of the Strings folder, see the previous section.

The type of file saved depends on the type of export operation. Text strings exported:

- For all application languages, are saved in a Microsoft Excel spreadsheet.
- For the selected application language, are saved in a Unicode text file.

For more information about exporting application text strings, see [Exporting application text strings on page 217](#).

### Excel spreadsheet file format

Text strings exported for all languages in an application are saved in an Excel spreadsheet.

The format for the file name is *<application>\_<export version>.xls*, where:

- *<application>* is the name of the application.
- *<export version>* is a number indicating the export version.

The first time text strings are exported from an application, the export version is 1. Subsequent export files for the same application are appended with 2, 3, 4 and so on.

The Excel spreadsheet contains additional information about the exported strings, for example, the names of the HMI servers and graphic displays that contain the strings. For more information, see [Working with text strings exported to an Excel spreadsheet on page 219](#).

### Unicode text file format

Text strings exported for the selected application language are saved in a text file in Unicode format.

For network distributed applications, text strings from each HMI server in the application are exported to separate text files. The file name format is *<application>\_<area>\_<HMI server>\_<RFC1766>.txt*, where:

- *<application>* is the name of the application.
- *<area>* is the name of the area.
- *<HMI server>* is the name of the HMI server containing the exported text strings.

- <RFC1766> is the RFC1766 name associated with the language selected for the export operation.

**Tip:**

In FactoryTalk View Studio, in the Language Configuration tool, the Add Language dialog box provides a list of Windows languages and their RFC1766 names. For details, see [Adding languages to an application on page 215](#).

For local applications, which contain only one HMI server, text strings are exported to a single, Unicode text file with the format <application><HMI server><RFC1766>.txt.

For information about the format and schema of Unicode text files, see [Working with strings exported to a Unicode text file on page 221](#).

## Working with text strings exported to an Excel spreadsheet

Text strings exported for all languages in an application are saved to a Microsoft Excel spreadsheet. This lets you translate or modify multiple languages at the same time.

To ensure that the spreadsheet can be imported successfully after it has been modified, give the information in this section to translators.

The following illustration shows the format of the exported spreadsheet.

Server	Component Type	Component Name	Description	REF	en-US
/Application/Area:HMI Server	Global Objects Display	Waste Water	Button1.Up.Caption		Waste Water
/Application/Area:HMI Server	Global Objects Display	Waste Water	Button10.Up.Caption	REF:1	Digestion
/Application/Area:HMI Server	Global Objects Display	Waste Water	Button16.Up.Caption	REF:2	Overview
/Application/Area:HMI Server	Global Objects Display	Waste Water	Button7.Up.Caption	REF:3	Lift Station
/Application/Area:HMI Server	Global Objects Display	Waste Water	Button9.Up.Caption	REF:4	Aeration
/Application/Area:HMI Server	Global Objects Display	Waste Water	Title Bar		
/Application/Area:HMI Server	Graphic Display	Aeration	Text11.Caption		**REF:4**
/Application/Area:HMI Server	Graphic Display	Aeration	Text26.Caption		Compressor
/Application/Area:HMI Server	Graphic Display	Aeration	Text27.Caption		Aeration basin
/Application/Area:HMI Server	Graphic Display	Aeration	Text29.Caption	REF:5	Sludge pump

The spreadsheet provides the following information for each exported text string:

- **Server** identifies the name of the HMI server that contains the exported text string. For network distributed applications, which can contain multiple HMI servers, the server name format is /<application>/<area>:<HMI server>. For local station applications, which can contain only one HMI server, the format is /<application>:<application>.



**Tip:** HMI server names in the Server column are case-sensitive. Ensure that the server names in the spreadsheet match those in the application. If not, some strings might not be imported.

- **Component Type** identifies the type of component that contains the text string. The component type can be either a standard graphic display, or a global object display.
- **Component Name** indicates the name of the standard display, global object display, or local message file that contains the text string.
- **Description** identifies where the text string is used in the standard display, or global object display.
- **REF** displays a unique reference number beside the first instance of each text string with duplicates in the HMI server.

A REF column is shown in the spreadsheet for each application language, only if the checkbox, **Optimize duplicate text strings**, was selected for the export operation.

- The <RFC1766 name> column contains the exported text strings. The number of columns depends on the number of languages in the application, when the strings were exported.

The column heading is the RFC1766 name for the language of the text strings in that column. For example, en-US is the RFC1766 name of English (United States).

In FactoryTalk View Studio, in the Language Configuration tool, the Add Language dialog box provides a list of Windows languages and their RFC1766 names. For details, see [Adding languages to an application on page 215](#).



**Tip:** The first column, containing exported text strings, will be for the default application language, if one is specified. Columns for all other application languages will follow in alphabetical order.

## Maintaining the format of the spreadsheet

To ensure that the spreadsheet can be imported successfully, you must preserve parts of the spreadsheet's format.

- Do not change the default order of the Server, Component Type, Component Name and Description columns.
  - Do not insert any blank rows or columns in the spreadsheet. The import operation will stop at a blank row or column. Anything after that will not be imported.
  - You can change the order of the columns containing the text strings for each language. However, if you move one of these columns, remember to move the associated REF column with it, if one exists.
- The REF column must always be to the left of the strings column.
- Do not delete a REF column, or any of the unique duplicate string identifiers in a REF column.
  - Do not delete a row that contains a unique duplicate string identifier in the REF column.

## Modifying or translating text strings

In the spreadsheet, translate only the \*\*UNDEFINED\*\* text strings for each application language.

For example, in the following illustration, all the text strings for Danish (da-DK) need to be translated.

Component Name	Description	REF	da-DK	REF	en-US
Alarm Summary	Button1.Up.Caption	REF:104	**UNDEFINED**	REF:20	Water Overview
Alarm Summary	Button2.Up.Caption		**UNDEFINED**		Alarm Log
Alarm Summary	Text1.Caption		**UNDEFINED**		Alarm Summary Utilities
Alarm Summary	Text4.Caption	REF:108	**UNDEFINED**	REF:21	Items in Summary:
Alarm Summary	Text5.Caption	REF:110	**UNDEFINED**		**REF:4**
Water Overview	Button1.Up.Caption	REF:120	**UNDEFINED**	REF:24	Alarm Log Viewer
Water Overview	Button2.Up.Caption	REF:122	**UNDEFINED**	REF:25	Alarm Summary
Water Overview	Button3.Up.Caption	REF:124	**UNDEFINED**	REF:26	FT Diagnostics Viewer
Water Overview	cmdClose.Up.Caption		**UNDEFINED**		Close SE Client
Water Overview	Text1.Caption		**REF:104**		**REF:20**
Water Overview	Text10.Caption	REF:126	**UNDEFINED**	REF:27	Industrial

In the da-DK column, \*\*REF:104\*\* indicates a duplicate instance of the undefined string in the first row. In this case, translate only the undefined string. The duplicate will be translated automatically when the spreadsheet is imported into the application.

For more information about duplicate text strings, see "Working with duplicate text strings," next.



**Tip:** Undefined text strings in an application always export as **\*\*UNDEFINED\*\***, regardless of whether you have selected the option, **Display undefined text strings in the default language**. For more information about this option, see [Setting up a default application language on page 216](#).

## Working with duplicate text strings

If the checkbox, Optimize duplicate strings, was selected for the export operation, text strings that occur multiple times in the application will be shown only once in the Excel spreadsheet.

In the following illustration, the text string Primary Treatment is marked with the unique identifier REF:15. Duplicate instances of the Primary Treatment string are marked with the same, unique identifier.

Component Name	Description	REF	en-US	REF	sv-SE
Waste Water	Button1.Up.Caption		Waste Water		**UNDEFINED**
Lift Station	Text13.Caption		Lift station #1		**UNDEFINED**
Lift Station	Text14.Caption		Pump #1		**UNDEFINED**
Overview	Group2.ToolTipText	REF:15	Primary Treatment	REF:30	**UNDEFINED**
Overview	Polygon414.ToolTipText		**REF:15**		**REF:30**
Overview	Polygon420.ToolTipText		**REF:15**		**REF:30**
Overview	Polygon426.ToolTipText		**REF:15**		**REF:30**
Overview	Polygon432.ToolTipText		**REF:15**		**REF:30**
Overview	Polygon438.ToolTipText		**REF:15**		**REF:30**
Overview	Text10.Caption		Town		**UNDEFINED**
Overview	Text11.Caption		Local sewers		**UNDEFINED**

↑  
Unique reference number attached to first instance of Primary Treatment.

Duplicate strings replaced with Primary Treatment reference number.

Only the first instance of the Primary Treatment string needs to be translated or modified. The duplicate references will be updated automatically when the spreadsheet is imported.



**Tip:** The REF column should only contain valid duplicate string identifiers. Do not delete any of these identifiers, or type any other text in the REF column.

If a string has been translated elsewhere in the spreadsheet, the optimization process will also translate the undefined string. In these cases, duplicate undefined text strings do not have to be translated manually.

In the illustration, for example, you would only have to translate the Primary Treatment string once into Swedish (sv-SE). Other, undefined instances of the same string will be translated automatically when the spreadsheet is imported.

## Working with strings exported to a Unicode text file

This section contains important information for translators about the format and schema of exported Unicode text files.

FactoryTalk View requires that parts of the Unicode text file remain as they were when exported. To ensure that the text file can be imported successfully, after it has been modified, give the information in this section to translators.

## File name and format

Text strings exported for the selected application language are saved in a text file in Unicode format.

You can rename the exported file, for example, to distinguish the translated version from the original. However, to import the text successfully, you must ensure that the file is saved as tab-delimited text, in Unicode Text format.

## Opening a Unicode text file in Microsoft Excel

When you open the text file in Microsoft Excel, the Text Import Wizard opens.

1. Open **Excel** and click **File > Open**.
2. In the **Open** dialog box, in the **Files of type** field, make sure that **All Files** is selected, then browse to and select the file you are going to edit, and click **Open**.  
When the text file is opened, the Excel Text Import wizard starts up.
3. In the **Text Import** wizard, select these options, and then click **Next**.  
Choose the file type that best describes your data: Delimited  
Start import row at: 1  
File origin: Windows (ANSI)
4. In the **Text Import** wizard, select these options, and then click **Next**.  
Delimiters the data must contain: Tab (make sure that the other checkboxes are cleared)  
Treat consecutive delimiters as one: clear the checkbox
5. In the **Text Import** wizard, for Column data format, select **General**.
6. Click **Finish**.

## Saving a Unicode text file in Microsoft Excel

1. Click **File > Select Save As**.
2. In the **Save As** dialog box, in the **Save as type** field, make sure that **Unicode Text (\*.txt)** is selected, and then click **Save**.
3. Click **Yes**, when a message appears stating that the file may contain features that are not compatible with Unicode Text, and asking if you want to keep the workbook in this format. This saves the file but does not close it.
4. Click **File > Close**, and then click **Yes** when you are asked whether you want to save the changes. You can then save the file using its original name or specify a new name.
5. Click **Yes**, when you are prompted again with the message described in step 3.

## Differences in format for Unicode files saved in Excel

If you use Notepad to open a Unicode text file that was saved in Excel, you will notice some differences from a file edited and saved in Notepad.



**Tip:** You do not have to change the format of the file before you import it into FactoryTalk View Studio.

The differences are:

- Double quotes surrounding the string definitions are removed for most strings.
- String definitions, containing embedded double quotes, or other characters that Excel treats as special characters, such as commas, are enclosed within double quotes.
- Any embedded double quotes are converted to a pair of double quotes.

## Saving a Unicode text file in Notepad

When saving the file, save it using the Unicode encoding option in the Save As dialog box.

### File schema

#### Comments

The text file uses the # symbol as a comment delimiter if it is the first character on a line.

#### Header

The first seven lines of the text file contain header information that must not be translated or modified.

#### Body

The body of the text file starts on line eight, and includes the following fields:

Field	Component type	Component name	String reference	"String definition"
Example	Graphic Display	Pump station	1	"Stop motor"

The file is sorted alphabetically by component name, and then numerically by string reference number.

Each string reference number refers to a different object in the component. In the preceding example, string reference 1 might refer to a push button in the graphic display called Pump station.

In the translated text file, the only text that needs to be modified is the text inside the quotation marks in the string definition column. For example, translated into German, the file would look like this:

Field	Component type	Component name	String reference	"String definition"
Example	Graphic Display	Pump station	1	"Motor abschalten"

Do not change the entries in the component name column, unless the component was renamed in the application after the text was exported.

---

**NOTE:** Do not modify the component type or string reference number. The string reference number is unique to FactoryTalk View. Modifying the component type or string reference number prevents FactoryTalk View from identifying the object correctly when you import the text.

---

## Working with pairs of double quotes

If a text string contains double quotes, the whole string definition must also be enclosed in double quotes. For example:

Call "Duty Manager"

must be entered in the string file as:

"Call "Duty Manager""

## Importing text containing multiple sets of double quotes

If the string definition contains an odd number of double quotes next to each other, they will be rounded to an even number and each pair will be imported as one double quote.

For example, the string:

```
"Call """Duty Manager"""
```

will be imported as:

```
Call "Duty Manager"
```

## Working with backslashes and new-line characters

To force the text to begin on a new line, precede the text with the backslash character \ and the new line character **n**.

For example:

```
Motor\nabschalten
```

is shown in the application as:

```
Motor
```

```
abschalten
```

To include a backslash in the text, type two backslashes (\\\). For example, to include the characters \n in the text, type \\n.

## Importing text containing multiple backslashes:

If the imported text file contains an odd number of backslashes next to each other, one of the backslashes will be ignored.

For example, the string:

```
Seven\\Eight
```

is imported into the application as:

```
Seven\Eight
```

## Importing translated or modified text strings

Use the Language Configuration dialog box to import text strings into an application, after translating or modifying them.

To avoid import errors, ensure that text files and spreadsheets to be imported are translated according to the prescribed formats and schema. For important information to share with translators, see:

- [Working with text strings exported to an Excel spreadsheet on page 219](#).
- [Working with strings exported to a Unicode text file on page 221](#).

If you are importing text into a network distributed application, you can import text for multiple HMI servers at the same time.

Text strings that exist in an application, but do not exist in the import file, are not deleted or modified during the import operation. This lets you import only the text strings that have been modified.

### To import text strings into an application:

1. Make sure that the HMI project that you are going to import text to is not being edited or modified.
  - If components are being edited remotely and are saved after you have imported text for them, the imported text will be overwritten.
  - If a component is being edited locally and has not been saved, you will be prompted to save it. If you click Yes, the edits will be saved, the component closed, and then the import will continue. If you click No, the edits will not be saved, but the component will be closed and the import will continue.
  - If a component is being edited remotely, you will not be prompted to save it and the import will be made to the most recently saved version of the component.
2. Create a backup of the text currently in the application by exporting it.  
This allows you to restore the original text to the application if an error occurs while importing, or if you cancel the import before it is complete. If you cancel the import while it is in progress, any text strings changed prior to canceling are not restored to their original values. To restore text strings as they were originally in the application, import the text from the backup file.
3. Click **Tools > Languages**.
4. If you are importing Unicode text files, in the **Language Configuration** dialog box select the language of the strings that are being imported, and then click **Import**.  
If you are importing an Excel spreadsheet, simply click **Import**.
  - An import from Excel will import every row in the spreadsheet until it reaches a blank row, and every column until it reaches a blank column. Anything after that will not be imported.
5. In the **Import Operation** dialog box, select whether to import the strings from a Unicode file or an Excel spreadsheet.
6. If you are importing Unicode strings for multiple HMI servers in a network distributed application, in the **Select Files** window click **Browse (...)**, select the files to be imported into the application, then click **Finish**.  
If you are importing an Excel file, in the **Select File** window, click the **browse button (...)**, select the file to be imported into the application, and then click **Finish**.

## Troubleshooting import problems

If errors occur during an import operation, or if you cancel the import while it is in progress, a message is logged to the Diagnostics List and log file.

In addition, import errors are shown in a log file called ImportErrors.txt, which is saved in the folder **C:\Users\Public\Public Documents\RSView Enterprise\Strings** on the local computer.



**Tip:** Existing import error log files are overwritten for each subsequent import operation that generates errors.

If it seems that some but not all of the text in an HMI server has been modified, the import might have been canceled. If you cancel the import before it is complete, any text strings that were changed are not restored to their original values.

To restore the original text in the application, import the text from the backup text file you created in step 2 on [Importing translated or modified text strings on page 224](#).

If it seems that text has been imported for some HMI servers but not for others, check the ImportErrors.txt file for an error message. The causes of common errors and their remedies are described in the following tables.

## Common errors when importing Unicode text files

The following table describes error messages that might be logged in the ImportErrors.txt file, while a Unicode text file is being imported into an application.

'Line x' in the error messages refers to the line number in the text file.

Error message	Cause and solution
ServerName defined in <i>FileName</i> .txt does not exist. None of the strings in this file were imported.	The name of the HMI server is invalid. If the HMI server was renamed, open the file, and then correct the name of the HMI server.
File " <i>FileName</i> ", Line x. <i>ComponentType</i> is not a valid component type for the application. The string was not imported	The component type has been modified in the text file. Open the text file, and then correct the text for the component type.
File " <i>FileName</i> ", Line x. <i>ComponentName</i> is not a valid component for the application. The string was not imported.	The component name has been modified in the text file, or in the application. Open the text file, and then correct the text for the component name.
File " <i>FileName</i> ", Line x. The string reference must be an unsigned long integer value between 1 and 4294967295. The string was not imported.	The string reference number has been modified in the text file, and the new string reference number is invalid. Open the backup text file, and then copy the correct string reference number into the translated text file.
File " <i>FileName</i> ", Line x. The string was not used in the application and was not imported.	This error occurs if: <ul style="list-style-type: none"> <li>the string reference number has been modified in the text file, and the new string reference number is not used in the application. Open the backup text file, and then copy the correct string reference number into the translated file.</li> <li>the object was deleted from the application after the text was exported. If this is correct, ignore the error.</li> </ul>
File " <i>FileName</i> ", Line x. The string definition must be contained within double quotes. The string was not imported.	The translated string definition includes embedded double quotes, but the string definition itself was not enclosed in double quotes.  Open the text file, and then enclose all string definitions containing embedded double quotes in double quotes. For example, the string definition Start "Backup motor" must be enclosed in double quotes, like this:  "Start "Backup motor""

Error message	Cause and solution
File " <i>FileName</i> ", Line x. Invalid line format!	A line in the import file does not contain all the component name or string reference number fields. The import continues with the next line in the file. Open the backup text file, and then copy the missing fields into the translated text file.
Unable to open { <i>FileName.txt</i> }. None of the strings from this file were imported.	The text file could not be opened. Make sure that the text file is in the folder from which you are importing files, and that you can open the text file in Notepad or Microsoft Excel.

## Errors when importing Excel spreadsheet files

The following table describes error messages that might be logged in the ImportErrors.txt file, while a Microsoft Excel spreadsheet file is being imported into an application.

Error message	Cause and solution
String import was interrupted by the user.	The user canceled or interrupted the string import operation.
" <i>RFC1766 Name</i> " does not exist in this application. No strings for this language were imported.	The language associated with a column in the spreadsheet does not exist in the application. No strings associated with this language will be imported.  The import will continue, but there will be no further attempt to import strings for this language.
" <i>Component Type</i> " is not valid. No strings for this component type were imported.	The component type for a string in the spreadsheet is not valid. No strings associated with this component type will be imported.  The import will continue, but there will be no further attempts to import strings and no further error messages logged for this component type. Valid component types are "Graphic Display" and "Global Object Display."
" <i>Server Name</i> " does not exist in this application. No strings for this server were imported.	The server name associated with a string in the spreadsheet does not exist in the application. No strings associated with this server will be imported.  The import will continue, but there will be no further attempts to import strings and no further error messages logged for this server.
" <i>Server Name</i> " is unavailable. No strings for this server were imported.	The server name associated with a string in the spreadsheet exists, but the HMI server is not available. No strings associated with this server will be imported.  The import will continue, but there will be no further attempts to import strings and no further error messages logged for this server.
" <i>Server Name_Component Type_Component Name</i> " does not exist in this application. No strings for this component were imported.	The component name for a string in the spreadsheet does not exist in the application. No strings associated with this component name will be imported.

Error message	Cause and solution
	The import will continue, but there will be no further attempts to import strings and no further error messages logged for this component name.
<p>"Server Name_Component Type_Component Name_Description" does not exist in this application. No strings for this component were imported.</p>	<p>The description for a string in the spreadsheet does not exist in the application. No strings with this description will be imported.</p> <p>The import will continue, but there will be no further attempts to import strings and no further error messages logged for this description.</p>
<p>"Optimized Duplicate String Reference" is pointing to a string that does not exist in the spreadsheet. No strings with this optimized duplicate string reference were imported.</p>	<p>Optimized duplicate string reference in the spreadsheet refers to a string that does not exist.</p> <p>The import will continue, but there will be no further attempts to import strings and no further error messages logged for this duplicate string reference.</p>
<p>Unable to save "Display Name." No strings for this component name were imported.</p>	<p>A graphic display or global object display cannot be saved. The import will continue.</p>
<p>Unable to open "Display Name." No strings for this component name were imported.</p>	<p>A graphic display or global object display cannot be opened. The import will continue.</p>
<p>Unable to import strings into "HMI Server Name" as the currently logged in user doesn't have write access to the area.</p>	<p>The currently logged on user does not have FactoryTalk Security Common Write access to the area containing the HMI server whose strings are being imported.</p> <p>Applies to FactoryTalk View SE network distributed applications.</p> <p>No strings associated with this server will be imported.</p> <p>However, the import will continue for any other HMI servers in the application.</p>
<p>Unable to import strings as the currently logged in user doesn't have write access to the application.</p>	<p>The currently logged on user does not have Common Write access to the application whose strings are being imported.</p> <p>Applies to FactoryTalk View SE local station, network station, and FactoryTalk View ME applications. No application strings will be imported.</p>
<p>Unable to import one or more strings, as one or more cells were not formatted as text or numeric values.</p>	<p>A spreadsheet cell containing a string for a language is not formatted for text or numeric values. The string in that cell will not be imported. The import will continue. This error will only be reported once for a single import operation.</p>
<p>Unable to import one or more strings, as one or more cells were not formatted as text or numeric values.</p>	<p>A spreadsheet cell containing a server name, component type, component name, description, or string reference is not formatted for text or numeric values.</p> <p>No strings in that row will be imported. The import will continue.</p> <p>This error will only be reported once for a single import operation.</p>

Error message	Cause and solution
Unable to import the spreadsheet, as one or more of the header cells were not formatted as text or numeric values.	A header cell is not formatted for text or numeric values. No strings will be imported into the application. The import will stop.

## Common errors when importing local messages

The following table describes error messages that might be logged in the ImportErrors.txt file, while local message text is being imported into an application.

Error message	Cause and solution
Unable to save " <i>Local Message File Name</i> ." No strings for this component name were imported.	Imported local message file cannot be saved. The import will continue.
Unable to open " <i>Local Message File Name</i> ." No strings for this component name were imported.	Imported local message file cannot be opened. The import will continue.

## Switch application languages at design time

If an application is designed with multiple languages, you may need to switch application languages at design time to check the layout of translated text.



**Tip:** If any of the following editors is opened, these editors will be forced to be closed and will not be opened automatically after you switch application languages:

- Communication Setup
- Alarm and Event Setup

If any of the following dialog boxes is opened and has unsaved changes, a message prompts you to save the changes:

- Display
- Global objects
- Local Messages

### To switch application languages at design time:

1. In FactoryTalk View Studio, on the toolbar, select **Change Application Language** .
2. In the **Change Application Language** dialog box, select the language you want to change to.



**Tip:** The language list shows the languages added to the application, first with the language name and then with the language RFC1766 identification code.

3. Click **OK**.

## Setting up runtime language switching

When you create a FactoryTalk View SE Client configuration file, you select the initial runtime language from the list of application languages. Once the client is running, the operator can switch from the initial language to any other application language.

For information about setting up client files, see the FactoryTalk View Site Edition Help.

## Using the Language command to switch languages

In FactoryTalk View SE, use the **Language** command to switch languages at runtime.

The Language command uses the following syntax:

```
Language <RFC1766 name>
```

where *<RFC1766 name>* is a code name for the language that you want to switch to.

To enable language switching in an application, provide operators with a way to run the **Language** command, for each language the application supports. For example, create buttons in a graphic display that use the **Language** command as the press action.

To find the RFC1766 name associated with a Windows language, open the Add Language dialog box in FactoryTalk View Studio. For details, see [Adding languages to an application on page 215](#).



**Tip:** If you create buttons for switching languages at runtime, be aware that button captions will be translated when a language switch occurs. To avoid confusion, use graphics instead of text to represent the different languages. For example, use an image of a country's flag on a button.

## Support for multiple languages in the graphic libraries

By default, text strings in the graphic libraries installed with FactoryTalk View SE never display as undefined. Instead, they always display in the language they were shipped in, regardless of the current application language.

When you save the library display, all strings that can be switched are saved in the current application language. Only the text strings for the current language will be displayed. Text strings in any other language will appear as undefined strings. The strings will have to be translated before they will display in other languages.

### To set up support for multiple languages in a graphic library shipped with FactoryTalk View SE:

1. In FactoryTalk View Studio, in the Explorer window, double-click the Libraries icon, right-click the graphic library you want to modify, and then select **Open**.
2. Right-click the library display, and then select **Display Settings**.
3. In the Properties tab, select the checkbox, **Support Multiple Languages**.
4. Click **OK**.

Test strings in the graphic library are saved in the current application language.

## Language support in new graphic libraries

By default, the new graphic libraries that you create for an application support multiple languages.

You can set up a new library to show text strings only in the current application language. If you do this, however, text strings in the new library that use some other language will be deleted when you save the change.

### To remove support for multiple languages in a new graphic library:

1. Right-click the library display, and then select **Display Settings**.
2. In the Properties tab, clear the checkbox, **Support Multiple Languages**.
3. Click **OK**.

Only text strings for the current application language are saved as language-neutral strings.

## Using the graphic libraries in a multi-language application

To use the contents of graphic libraries in an application, you can add an entire library into the Displays folder, or you can copy one object or more from a library into an existing graphic display.

If the graphic library you want to use supports multiple languages, and you add the entire library into the Displays folder, all text strings are copied with the library.

If you copy an object from the library into a graphic display, only text strings that use an application language are copied with the object. Text strings that use a language the application does not support are deleted.

# Setting up FactoryTalk system availability

This chapter describes:

- What it means to maximize availability in a FactoryTalk system.
- Monitoring the status of application servers.
- Monitoring the status of the FactoryTalk Network Directory server.
- Redundancy as part of a system availability strategy.
- Setting up a redundant HMI server.
- Determining the Active HMI server in a redundant pair.
- Switching the Active and Standby HMI servers manually.
- What happens when the primary HMI server fails.
- Modifying HMI tags at runtime.
- Managing HMI data in an online redundant system.
- Monitoring network client and server connections.

## FactoryTalk features that maximize system availability

An automation and control system that uses FactoryTalk services, and integrates FactoryTalk products and components, is known as a FactoryTalk system.

A complete FactoryTalk system consists of all the networks, devices, and software applications you have deployed to monitor and control your plant or process.

Helping to ensure that the system can provide data in a secure and predictable fashion depends on a number of variables.

To minimize data loss and down time, and to help ensure that critical parts of your system are always available to connected clients, FactoryTalk View Site Edition provides these health monitoring and redundancy features:

- **Server status monitoring** of non-redundant and redundant application servers. For more information, see [Monitoring the status of application servers on page 233](#).
- **Disconnected operation.** For example, connected clients can continue to run when the FactoryTalk Directory becomes unavailable. For more information, see [Monitoring the status of the Network Directory server on page 236](#).
- **Redundant application servers.** In a network distributed application, you can set up redundancy for the following types of application servers:
  - FactoryTalk View SE Servers (also called HMI servers). For information about setting up redundant HMI servers, see [Setting up a redundant HMI server pair on page 241](#).
  - FactoryTalk Alarms and Events servers. For information about setting up redundant FactoryTalk Alarms and Events servers, see the product documentation. For information about setting up redundant FactoryTalk Linx servers, see the product documentation.
  - OPC data servers, including RSLink Classic. For information about setting up redundant OPC servers, see the product documentation.
- **Support for online changes** to HMI tag and alarm properties. For information about this feature, see [Modifying HMI tag and alarm properties at runtime on page 248](#).
- **Replication of HMI server changes** from primary to secondary HMI servers. For information about this feature, see [Replicate changes to the secondary HMI server on page 242](#).

- **Network connection monitoring** on each computer (clients and servers) in the system. For information about this feature, see [Monitoring network client and server connections on page 251](#).



**Tip:** Many of the availability features built into FactoryTalk View SE support local station, network station, and network distributed applications. However, you can only set up redundant servers in a network distributed application. For more information about Station, see [Working with network station applications on page 131](#), and [Working with local station applications on page 142](#).

## Finding more information about system availability

This manual contains information about the availability features built into FactoryTalk View SE, and how they can help protect your application against software failures.

This manual does not describe how to protect the hardware, the control system, the information network, the operating system, or any other supporting software that your FactoryTalk View SE applications might use.

For information about these components, see the product documentation. For additional information about FactoryTalk availability features, see the FactoryTalk Help.

## Monitoring the status of application servers

In FactoryTalk View Studio, or in the FactoryTalk View SE Administration Console, you can monitor the operational status of servers in a FactoryTalk View SE application, to determine whether the servers are ready to provide service, and to troubleshoot problems.

In the Server Status dialog box, you can:

- Check the status of non-redundant and redundant FactoryTalk View SE Servers, Rockwell Automation Device Servers (FactoryTalk Linx), and FactoryTalk Tag Alarm and Event Servers.
- Specify switchover options. For example, you can specify whether to always switch back to the primary server, if it's available.
- Switch the Active and Standby servers manually. For more information, see [Switching the Active and Standby servers manually on page 245](#).

### To open the Server Status dialog box:

In FactoryTalk View Studio, in the Explorer window, right-click the server's icon, and then select **Server Status**.



**Tip:** The illustration is of the Server Status dialog box for the primary HMI server in a redundant pair.

For a non-redundant server, the dialog box shows only the primary computer name and status, and the switchover options are unavailable.



## States for non-redundant and redundant servers

The **Server Status** dialog box will show any of the following states for non-redundant and redundant application servers. The following table shows the characteristics for a server state.

A server with this status	Has these characteristics
Loading	The server is loading into memory. You can set up an HMI server to load when the operating system starts, or on demand (when the first client connects to the server). An HMI server can be made redundant only if it is set up to load when the operating system starts.
Starting	The server is initializing with data. For example, the HMI server is loading its HMI project, or uploading names from a controller.
Ready to provide service	The server is finished initializing data and synchronizing with its partner, and is ready to provide service to clients. Redundant servers are either ready to be active, or ready to be standby. For more information, see "States for redundant servers," next.
Active	The Active server is ready to accept client connections and provide services to clients. For redundant servers, if the Active server's partner fails, the status will change to Active (Can't reach partner). For more information, see "States for redundant servers," next.
<Host> unreachable	A client computer cannot contact the computer hosting the server. For example, this state applies if communications fail between the client and the computer hosting the Network Directory server or the HMI server.
Failed	The server can no longer provide service.

A server with this status	Has these characteristics
	This would occur, for example, if a supporting service is unhealthy. For information about what happens when an HMI server fails, see <a href="#">What happens if a non-redundant HMI server fails on page 236</a> and <a href="#">What happens when the primary HMI server fails on page 246</a> .
Not loaded	The server is not loaded.  For example, an HMI server set up to start on demand does not load until a client connects to it.

## States for redundant servers only

The **Server Status** dialog box will show any of the following states for redundant application servers only, as long as a partner is defined for the server.

A server with this status	Has these characteristics
Synchronizing with standby	The server becoming the Active server is synchronizing with its Standby partner.  After restarting, for example, an Active server will try to get a copy of the Standby server's state information, before going into service. This ensures that the latest state information is used.  If the Standby server is unavailable, or if it fails during synchronization, the Active server will go into service without its partner's latest state information.
Synchronized - ready to be Active	The server becoming the Active server has finished synchronizing with its partner, and is ready to become active.
Synchronizing with active	The server becoming the Standby server is synchronizing with its Active partner.  After restarting, for example, a Standby server will try to get a copy of the Active server's state information, before becoming standby. This ensures that the latest state information is used.  If the Active server is unavailable, or if it fails during synchronization, the Standby server will become standby without its partner's latest state information.
Synchronized - ready to be Standby	The server becoming the Standby server has finished synchronizing with its partner, and is ready to become standby.
Active (Can't reach partner)	The server is ready to provide service to clients, but its partner is not ready or cannot be located.  This state is also shown if the Active server's partner has not been defined.
Standby	The Standby server is ready to accept client connections and provide services to clients, in the event that the Active server fails.

A server with this status	Has these characteristics
	If the Active server fails, the Standby server's status will change to Active (Can't reach partner), and the Standby server will provide service to connected clients.
Not configured	A secondary server is not configured.

## What happens if a non-redundant HMI server fails

An Active, non-redundant HMI server might fail due to any of the following conditions:

- The HMI server is shut down with its host computer, from the Start menu.
- The HMI server is disconnected from the network. For information about how the system monitors network connections, see [Monitoring network client and server connections on page 251](#).
- The HMI server is found to be unhealthy, because an essential service has stopped functioning.
- The HMI server's host computer has a power failure.

When an HMI server fails, if the server is not set up with redundancy, it is taken out of service. Within 5 seconds, Diagnostic messages are logged at connected client computers, notifying the operators of the failure.

Clients that were connected when the failure occurred will continue to run, using a local cache of directory information.

However, any data or services the failed HMI server provides, such as HMI tag values, derived tags, command processing, or data logging, will be unavailable.

For example, graphic objects connected to HMI tags in the server's database will show in outline form.

If connected clients are receiving data from other HMI or data servers, that service will not be interrupted.

When the failed HMI server is restored, all connected clients in the system can resume receiving data from the server, without having to be restarted.

## Monitoring the status of the Network Directory server

FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed applications, and network station applications. All of the client and server computers participating in a given network distributed application, or network station application must point at the same Network Directory.

In the **Log On to FactoryTalk** dialog box, you can view the current status of the active Network Directory server:

- **(Connected)** means all FactoryTalk products and components participating in a FactoryTalk system on the current computer are connected to and communicating with the Network Directory server computer.
- **(Read-only)** means FactoryTalk system participants on the current computer are disconnected from the Network Directory server and are retrieving information from a local cache.
- **(Unknown)** means that the connection status is temporarily unknown, for example, because the system is starting up and waiting to determine which server is active, or is unable to determine the current status.

## To check the status of the Network Directory server:

1. In the notification area, click the **FactoryTalk Directory** icon.
2. The notification shows the connection state.

## What happens if the Network Directory server is unavailable

The Network Directory server cannot be made redundant. Instead, if the Network Directory becomes unavailable while client computers are connected to an application, the clients continue to run, using a local cache of directory information.

There is no need to restart previously connected clients; they will continue to resolve tag addresses, read and write tag values, acknowledge alarms, and open graphic displays, even if the tags and displays the clients require were never used before.

While the Network Directory is unavailable, you cannot modify the structure of any dependent application.

For example, you cannot add areas or servers to the application, create security accounts, or change system security policies.

When the Network Directory is available again, all dependent clients in the system will resume using the directory automatically.

## Redundancy as part of a system availability strategy

The more protection you provide, the more dependable the entire control system will be in the event any hardware or software component fails.

The level of system availability increases with the number of redundant system components. To protect against:

- **Programmable controller failures**, make programmable controllers redundant. For example, you can set up redundancy for Studio 5000 controllers.
- **Programmable controller network failures**, install redundant programmable controller networks, such as a Rockwell Automation's ControlNet network.
- **Information network failures**, install a redundant Ethernet network with a backup domain controller, network cables, and network interface cards.
- **Host computer hardware failures**, provide backup computers to host application software such as FactoryTalk View SE.
- **FactoryTalk software failures**, set up redundant FactoryTalk View SE and FactoryTalk Linx application servers.

## Planning the layout of a redundant system

The servers you can use in a FactoryTalk View SE network distributed application are separate entities, which means they can be hosted on any computer on the network.

However, when designing a redundant control system, you should keep in mind the system requirements that apply to different types of servers.

For example, to support FactoryTalk Network Directory availability, it is recommended that you run the directory server on a computer that plan to keep running.

This could be the same computer used to host a FactoryTalk View SE Server, or a Rockwell Automation Device Server (FactoryTalk Linx).

**NOTE:** Do not run the FactoryTalk Network Directory on the same computer as a Windows domain controller.

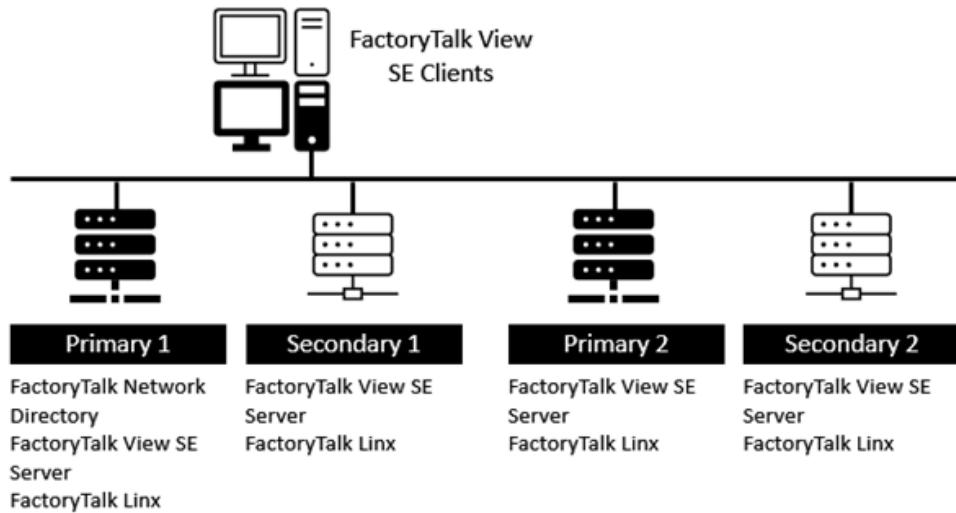
### Example: An application with two redundant HMI server pairs

In its simplest form, a redundant FactoryTalk View SE application might consist of one pair of redundant FactoryTalk View SE Servers (also called HMI servers), and one pair of redundant FactoryTalk Linx servers.

In this example, two redundant server computer pairs host primary and secondary HMI servers, and FactoryTalk Linx data servers.



**Tip:** The FactoryTalk Network Directory (on the first primary computer in the following illustration) cannot be made redundant. For information about what happens when the Network Directory fails, see [What happens if the Network Directory server is unavailable on page 75](#).



Only one application server of any type, whether it's a primary or secondary server, can run on a single host computer. When designing a control system, observe this and the other limits described in "About FactoryTalk View SE system limits," next.

The overall design of the control system will ultimately depend on the application's size and computing needs, and on the level of redundancy required to support the entire system.

For information about setting up redundancy for devices, networks, and other types of servers the system might use, see the documentation for the specific hardware or software.

## About FactoryTalk View SE system limits

In a network application, it is possible for a single-server computer that meets the recommended requirements to host all of the following components:

- One FactoryTalk View SE Server
- One FactoryTalk Linx data server (which can also be set up as the FactoryTalk Alarms and Events device-based alarm server, or also includes the secondary data server)
- One FactoryTalk Alarms and Events tag-based alarm server
- The FactoryTalk Network Directory

If a network application requires multiple HMI servers, data servers, or alarm servers, to distribute the load, it is recommended that you install the necessary software and run the servers on multiple host computers.

For information about setting up redundant FactoryTalk View SE servers, see *FactoryTalk View Site Edition User's Guide*.

For information about installing the FactoryTalk View SE software, and about system requirements that might affect the network design, see *FactoryTalk View Site Edition Installation Guide*.

## FactoryTalk View SE redundant component limits

For best results when running redundant FactoryTalk View SE components in a network distributed application, observe these limits:

- 120 is the maximum number of FactoryTalk View SE client sessions that can have simultaneous access to a network distributed application.
- 10 is the maximum number of redundant FactoryTalk View SE Servers that can be used in a network distributed application. (Each redundant server consists of a pair of primary and secondary servers.)
- 1 is the maximum number of redundant FactoryTalk View SE Servers that can be hosted on a single computer.

---

**NOTE:** You might use one computer to host multiple HMI servers while developing an application. When deploying the application, however, ensure the primary and secondary HMI servers in redundant pairs are hosted on separate computers.

---

For information about the limits and requirements that apply to redundant data servers (FactoryTalk Linx, RSLinx Classic, or other OPC data servers), see the product documentation.

## System sizing recommendations

When you design and set up a FactoryTalk system, we recommend using the following guidelines. If you find that your system needs to expand beyond these guidelines, contact your Rockwell Automation sales or distributor representative to discuss your application design.

Component	Recommended guidelines
<b>FactoryTalk Alarms and Events</b>	
FactoryTalk Alarms and Events servers per FactoryTalk Directory	10 non-redundant servers or 10 redundant server pairs* (1)
FactoryTalk tag-based alarms per server	20,000* (2)

Component	Recommended guidelines
Logix device-based alarms per server	10,000* (3)
Total alarms per FactoryTalk Directory	100,000* (4)
<b>FactoryTalk Linx</b>	
FactoryTalk Linx data servers per FactoryTalk Directory	10 non-redundant servers or 10 redundant server pairs* (1)
Total device tags per FactoryTalk Directory	1,000,000* (5)
Tags per FactoryTalk Linx data server	100,000* (5)
<b>FactoryTalk View SE</b>	
HMI servers per FactoryTalk Directory	10 non-redundant servers or 10 redundant server pairs* (1)
HMI tags per HMI server	40,000* (6)
Total data logged tags per HMI server (1 second scan rate)	DataLogPro 50,000 or data log model 5,000  <b>Note:</b> If you want to run both DataLogPro and data log model on the same HMI server, reduce the number of tags because the greater the number, the more powerful the system is required.
FactoryTalk View SE client sessions simultaneously connected to the application	120
FactoryTalk ViewPoint SE servers per application	4
FactoryTalk ViewPoint SE concurrent clients per ViewPoint server	50

1. Because multiple applications can run simultaneously within the same FactoryTalk Directory, the recommended number refers to the sum of servers across applications in one FactoryTalk Directory.

2. The number of alarm backing tags influences the maximum number of available alarms because they increase the resource requirements of the alarm server. When using backing tags for status and control, a good rule of thumb is to multiply the total number of backing tags by the total number of alarms, and ensure that the result is less than 20,000. PlantPAx® users should refer to the PlantPAx Reference Manual [PROCES-RM001](#) for alarm server sizing guidance for a PlantPAx application.

3. This number is the sum of both instruction alarms and Logix tag-based alarms across all the controller shortcuts in each FactoryTalk Linx server.

4. The total number of alarms per FactoryTalk Directory includes all FactoryTalk Alarms and Events tag-based and device-based alarms in the directory.

5. Refer to Knowledgebase Document ID: [IN7377 - Amounts of device tags that may have in FactoryTalk View SE application](#) for detailed information to determine system scalability for the total device tags per FactoryTalk Directory.

6. The maximum number of HMI tags that can be created on a single HMI server is 175,000. Extensive use of HMI tags has a significant performance impact on an HMI server, so Rockwell Automation recommends minimizing their use, and instead using direct-reference tags as much as possible.

For information about FactoryTalk View SE requirements that might affect the design of a redundant system, see *FactoryTalk View Site Edition Installation Guide*. For information about requirements that apply to redundant data servers such as FactoryTalk Linx, see the product documentation.

## Activating FactoryTalk View SE in a redundant system

To further maximize the availability of HMI data in a redundant FactoryTalk system, ensure that FactoryTalk View SE software components can obtain the necessary activations.

The FactoryTalk View SE Server software must be installed and activated on the primary and the secondary HMI server in a redundant pair.

To ensure that HMI servers and FactoryTalk View SE Clients can always obtain activation, it is recommended that activation be provided locally, on server and client host computers.

For more information about activating FactoryTalk View software, see the *FactoryTalk View Site Edition Installation Guide*.

For information about activating data servers such as FactoryTalk Linx, see the product documentation.



**Tip:** The FactoryTalk Network Directory is part of the integrated FactoryTalk architecture.

## Setting up redundant servers in FactoryTalk View SE

You can specify an alternate server for any FactoryTalk View SE Server (also called the HMI server), FactoryTalk Alarms and Events server, FactoryTalk Linx server, or OPC data server (including RSLinx Classic) in a network distributed application.

In FactoryTalk View SE, the alternate server in a redundant server pair is called the secondary server. In the event that the primary HMI server cannot provide service to application clients, the system switches to the secondary server, and FactoryTalk View SE Clients continue to function normally.

For more instructions, see [Setting up a redundant HMI server pair on page 241](#).

## Setting up a redundant HMI server pair

These are the steps involved in setting up a redundant FactoryTalk View SE Server (also called an HMI server), in a FactoryTalk View SE network distributed application:

1. On the secondary server computer:
  - Install and activate the FactoryTalk View SE Server software. For details, see the *FactoryTalk View Site Edition Installation Guide*.
  - Specify the location of the Network Directory for the application. For details, see [Setting up the FactoryTalk Directory on page 70](#).
2. On the primary sever computer:
  - Set up redundancy options, and specify the secondary server computer name. For details, see [Set up HMI server redundancy options on page 242](#).

## Specify the Network Directory on the secondary computer

After installing the FactoryTalk View SE Server software on the secondary server computer, specify the location of the FactoryTalk Network Directory for the application.

### To specify the Network Directory location:

1. Select **Start > Rockwell Software > Specify FactoryTalk Directory Location**.
2. Click the **Browse** button beside the field, **Computer hosting directory server**.

- 
3. Select **Remote computer**, type the name of the Network Directory server computer, and then click **OK**.
- 

**Tip:**

To use the FactoryTalk Directory Server Location Utility, you must sign in as a user with administrative privileges at the Network Directory and in Windows, on the computer where the utility is running.

---

## Set up HMI server redundancy options

In the primary HMI server's Properties dialog box, you can set up an HMI server to fail over to a secondary server when the primary server fails. You can also specify whether the system will switch back to the primary server automatically, or continue using the secondary server, after service is restored. Use this option to avoid unnecessary disruptions in the flow of data from the HMI server to clients.

Settings on the **Redundancy** tab are saved with the application, and apply to both the primary and the secondary server. No further setup or special programming is required on the secondary server computer, or on computers hosting FactoryTalk View SE Clients.

### To set up redundancy options:

1. In the FactoryTalk View Studio Explorer window, right-click the HMI server and select **Properties**.
  2. In the HMI Server **Properties** dialog box, on the **General** tab, select **Load and run startup components when operating system initializes**.
- 

**Tip:** Servers set up to start On demand cannot be made redundant.

When changing the redundancy setting (from non-redundant to redundant or vice versa), for the changes to take effect you must restart the computers hosting the redundant pair, and restart the computer hosting the FactoryTalk Network Directory Server.

---

3. On the **Redundancy** tab, select **Provide redundancy using a secondary server**.
4. Under **Computer hosting the server**, type the name of the secondary server.
5. Under **Switchover options**, select a switchover option.
6. When complete, click **OK**.
7. To automatically save the project online edits to both active and standby servers:
  - a. In FactoryTalk View Studio, from the menu, select **Tools > Options**.
  - b. On the **Settings** tab, make sure the **Save edits to both active and standby HMI servers** option is selected.
8. To keep the HMI memory tag values the same on both active and standby servers, on the **Settings** tab, select the **Auto synchronize HMI memory tag values to both active and standby HMI servers** option.

For details about options on the **Redundancy** tab, click **Help**.

## Manually replicate changes to the standby HMI server

We recommend you set up the option to automatically save the project online edits to both active and standby servers. For more information, see [Set up HMI server redundancy options on page 242](#).

If you do not set up the option, manually replicate the changes from the active server to the standby server. The replicate operation copies the current active server's configuration files, including settings in the Components tab, to the standby server.

For example, if an On Active macro and a Shutdown macro are selected for the Active server, the settings will be replicated to the Standby server. This means that the same macros specified for the Active server will run when the Standby server becomes active or shuts down. Information that is not included in the replicate operation includes data log files generated at runtime, the current value of HMI memory tags and retentive tags.

---

**NOTE:** Make sure to verify the current HMI server name and status before the replication. For information about checking server states, see [Monitoring the status of application servers on page 233](#).

---

### To manually replicate the changes:

1. In the FactoryTalk View Studio Explorer window, right-click the HMI server and select **Server Status**.
2. In the **Server Status** dialog box, verify that the current HMI server status is **Active** and the other is **Standby**.



**Tip:** To show the status of the HMI servers, FactoryTalk Alarms and Events servers, and data servers at runtime, add one of the following to the application:

- A **Server Status** display under the **Library** folder.
- A redundancy function, such as **PrimaryServerStatus** and **SecondaryServerStatus**.

3. In the HMI Server Properties dialog box, in the **Redundancy** tab, click **Replicate Active to Standby**.

## Determining the Active HMI server in a redundant pair

At runtime, either the primary or the secondary HMI server in a redundant pair can be in an active or a standby state:

- The **Active** server is actively providing data and services (for example, graphic displays and event processing) to connected clients.
- The **Standby** server is ready to provide data and services to connected clients, in the event that the Active server fails.

The primary and secondary servers communicate about their status, and determine which server should be active, and which should be standby, based on redundancy options set up in the primary server's Properties dialog box.

If you select the option, **Switch over to the primary server when it becomes active**, the redundant server pair will agree that, whenever it is ready, the primary server will always become the Active server.

If you select the option, **Continue to use the secondary server when the primary server becomes available again**, the redundant server pair will agree that the currently Active server will always remain active.

For information about setting up redundancy for an HMI server, see [Set up HMI server redundancy options on page 242](#). For information about other server states, see [Monitoring the status of application servers on page 233](#).

## When an HMI server is ready to be active or standby

Application servers become active or standby only when they are ready, that is, when they are capable of accepting and processing requests from connected clients.

An HMI server is considered to be ready when:

- The server is able to provide project components, such as graphic displays and macros.
- The server is able to receive and run commands.
- HMI tag services are started and completely initialized.

For information about when other application servers (FactoryTalk Linx data servers, for example) are ready to be active or standby, see the product documentation, or see Help.

## Specifying On Active and On Standby macros

In the primary HMI server's Properties dialog box, in the Components tab, you can select to run macros when the primary server becomes active or standby.



For example, you might specify an On Active macro that contains commands to run a certain event component and data log model, when the server becomes active.

Then, you might use the On standby macro to stop those components when the server becomes standby.

In the Components tab, you can also start and stop components manually, on both the primary and the secondary HMI server. For details about options in the Components tab, click **Help**.



**Tip:** If an On Active macro contains many commands (or other macros), it might not finish running before the HMI server is finished becoming active. This means that the macro might still be running when the server is ready to accept requests from clients.

### Ensuring that the macros finish running:

If the Active HMI server in a redundant pair fails while a command or macro is running, the command or macro stops running during the failover to the Standby server.

You might have to rerun the command or macro when the failover process finishes, and the Standby server becomes active.

## What happens if both servers become active

If a network disconnection breaks communications between the primary and secondary HMI server computers, it is possible for both servers to be in an active state when the network is restored.

In a partitioned network, for example, where a switch exists between the primary and secondary host computers, clients on the primary server side of the switch would remain connected to the Active primary server, when a network break occurs.

Meanwhile, on the secondary server side of the switch, the secondary server would detect that the primary server has failed, and become active. Clients on that side of the switch would connect to the newly Active secondary server.

When the network is restored, if redundancy options are set up to continue using the Active secondary server, then both the primary and the secondary HMI servers would remain active.

If that happens, the HMI servers would use the following criteria to determine whether the primary or the secondary server should be active:

- If one HMI server in the redundant pair has more clients connected to it, then the server with the most clients will become the Active server.  
The server with the least clients will become the Standby server, and the clients connected to this server will reconnect to the new, Active server.
- If the HMI servers have an equal number of clients connected to them, the primary server will become active, and the secondary server will go on standby.

## Switching the Active and Standby servers manually

When the Active HMI server in a redundant pair fails, the system fails over automatically to the Standby server, as long as it is ready and able to provide service.

If the system fails over to the secondary server, and the secondary server becomes active, when service is restored to the primary server, either the system will switch back automatically to the primary server, or the secondary server will remain active.

This depends on the switchback option that you have selected in the **Redundancy** tab of the primary HMI server's Properties dialog box. (You can also select a switchover option in the Server Status dialog box.)

If you select the option, **Continue using the secondary server even when the primary server becomes available again**, then the secondary server will remain active, unless you manually switch the Active and Standby servers.



**Tip:** If you select the switchover option, **Switch over to the primary server when it becomes available**, you cannot switch the Active and Standby servers manually.

### To switch the Active and Standby HMI servers:

1. In FactoryTalk View Studio, in the Explorer window, right-click the HMI server's icon, and then select **Server Status**.
2. In the Server Status dialog box, click **Switchover**.  
The Server Status display should reflect the switch for both servers.



**Tip:** If the server currently on standby is not ready to become active, then the manual switchover will not occur. For information about when an HMI server is ready, see [When an HMI server is ready to be active or standby on page 244](#).

You can also use the **Switchover** command to switch the Active server. For more information about FactoryTalk View commands, see FactoryTalk View SE Help.

## Set up FactoryTalk Security permissions

To switch the Active and Standby servers manually, users must have permission to perform the switch operation, under User Rights Assignments for the system.

### To set up the User Rights Assignment for switching servers:

1. In FactoryTalk View Studio, in the Explorer window, open the System folder.
2. In the System Policies folder, double-click User Rights Assignments.
3. Under Manage Servers, click Manual server switchover, and then click the Browse button in that row, beside Configure Security.
4. Select the Allow checkbox, beside users you want to give permission to switch servers.

For information about FactoryTalk Security, see [Setting up security on page 77](#).

## What happens when the primary HMI server fails

The following are some conditions that might cause the primary, Active HMI server in a redundant pair to fail:

- The HMI server is shut down with its host computer, from the **Start** menu.
- The HMI server is shut down using the FactoryTalk View SE Service Manager. For information about this tool, see the *FactoryTalk View Site Edition Installation Guide*.
- The HMI server is disconnected from the network. For information about how the system monitors network connections, see [Monitoring network client and server connections on page 251](#).
- The HMI server is found to be unhealthy, because an essential service has stopped functioning.
- The HMI server's host computer has a power failure.

Diagnostic messages are logged at connected client computers, notifying operators of the server failure. The time when diagnostic messages are logged upon HMI server failure is positively correlated to the **Health Monitoring Policy > Network failure detection interval** policy of FactoryTalk Services Platform.

## Failing over to the Standby secondary server

When the primary HMI server in a redundant pair fails, the system automatically fails over to the secondary server.

If the secondary server is on standby, and ready to become active, it becomes the Active server. If the secondary server is unavailable, the HMI server is taken out of service.

Clients that were connected at the time of the failure, start using the active secondary server upon it becoming active. The time of failing over to the active secondary server is positively correlated to the **Health Monitoring Policy > Network failure detection interval** policy of FactoryTalk Services Platform.

**Note:** In certain cases of a poor network or a large number of clients running at the same time, the failing over might be impacted. To reduce the impact, increase the value of the computer detection interval or network failure detection interval policy in FactoryTalk Services Platform. For more information on these policies, see *FactoryTalk Services Platform Help*.

Clients will only connect to an Active server.

If for some reason the secondary server cannot finish becoming active, then the server is taken out of service. For information about when an HMI server becomes active, see [Determining the Active HMI server in a redundant pair on page 243](#).



**Tip:** If both primary and secondary servers become active, then the servers decide which will be the Active server. For more information, see [What happens if both servers become active on page 244](#).

## Switching back to primary, or staying with the Active server

When service is restored at the primary server, either the system will switch back to the primary server automatically, or the secondary server will remain active.

This depends on the switchover option that you select in the **Redundancy** tab of the HMI server's Properties dialog box. For detailed options in the Properties dialog box, click **Help**.



**Tip:** You can also select a switchover option in the Server Status dialog box.

## Continuing to use the secondary server

If you select the switchover option, **Continue using the secondary server even when the primary server becomes available again**, the secondary server will remain the Active server, even if the primary server is ready.

Use this option if you want to be able to choose when to switch back to the primary server. When you are ready, you can manually change the Active and Standby servers. For more information, see [What happens if both servers become active on page 244](#).

Clients will remain connected to the healthy Active server, until you perform the manual switchover, or until the currently Active secondary server fails.

If the Active secondary server fails, as long as the primary server is on standby and ready to provide service, then the primary server will become the new, Active server.

## Switching back automatically to the primary server

If you select the switchover option, **Switch over to the primary server when it becomes available**, the system will switch back automatically, from an Active secondary to a restored primary server.

Connected clients will switch back to the Active primary server, as soon as it is ready.

Choosing to switch to the primary server means that the primary server is always preferred. If you select this option, you cannot manually change the Active and Standby servers.

If you choose not to switch back automatically to the primary server, the primary server will go on standby when service is restored, until the Active secondary server fails.

If that happens, the failover and switch-back cycle begins again.

## Notifying clients when switching back to the primary

During a switch back to the primary server, the server notifies connected clients when it is ready to provide service, and then waits for a specified time period for clients to respond.

After the time period expires, the server becomes active even if there are clients that have yet to respond.

The predefined amount of time the server waits before becoming active is two minutes. You can change the time period by modifying the system policy setting, Maximum delay before server is active.



**Tip:** The policy setting, **Maximum delay before server is active**, applies only during a switch back to the primary server. It does not apply when the system is failing over to the secondary server.

### To change the time a server waits becomes active:

1. In FactoryTalk View Studio, in the Explorer window, expand the System folder, double-click the Policies folder, and then double-click the System Policies folder.
2. Double-click Health Monitoring Policy, and then select **Maximum delay before server is active**.
3. Click the number of minutes. To select another number, click the up or down arrow beside the number.

## Modifying HMI tag properties at runtime

You can change the HMI tag properties while a FactoryTalk View SE network distributed application is running, and have those changes take effect without restarting clients.

This means that you don't have to stop HMI servers or the application from running, in order to make or deploy these changes. Critical application components can remain available to connected clients, while you are updating the HMI servers.

## Modifying HMI tag properties

In FactoryTalk View Studio, in the Tags editor, you can modify the properties of HMI tags while an application is running.

Some of the changes you can make will be visible immediately on connected clients. Other changes require some other event to occur before the change takes effect.

For example, if you change the Security Code assigned to an HMI tag, the new security check will not occur until there is an attempt to write to the tag. When that happens, the tag write will be validated against the new Security Code.

## When HMI tag property changes take effect

The following table lists the HMI tag properties that you can modify at runtime, and what needs to happen before each change takes effect.

Use the information in the table to ensure that the changes you make are reflected as expected, at connected clients.

If you change this property	For this type of HMI tag	The change takes effect when
Security code	Any HMI tag	An attempt is made to write to the tag, from a FactoryTalk View SE Client.

If you change this property	For this type of HMI tag	The change takes effect when
		The Security Code property is read using the VBA code.
Description	Any HMI tag	A graphic display using the Description property, in a tag label object or in a trend's pen, is test run in FactoryTalk View Studio or opened in a FactoryTalk View SE Client.
		The Description property is read using the VBA code.
		The Description property is shown in the Tag Browser.
Data Source Type	Any HMI tag	Immediately. The Value property is updated to use the new Data Source Type as soon as the property is modified. The Data Source Type cannot be modified if the tag is currently active.
Address	Any HMI tag	Immediately. The Value property is updated to use the new Address as soon as the property is modified.
Minimum or Maximum	HMI analog tags	An attempt is made to write to the tag, from a FactoryTalk View SE Client, FactoryTalk View Studio, or the HMI server.
		A graphic display using the Minimum or Maximum property, in a tag label object (Low EU or High EU) or in a trend's pen, is test run in FactoryTalk View Studio or opened in a FactoryTalk View SE Client.
		The Minimum or Maximum property is read using the VBA code.
Units	HMI analog tags	A graphic display using the Units property, in a tag label object (Engineering Units) or in a trend's pen, is test run in FactoryTalk View Studio or opened in a FactoryTalk View SE Client.
		The Units property is read using the VBA code.
Data Type	HMI analog tags	Immediately. The Value property is updated to use the new Read/Write Data Type, if the modification causes the tag's value to change.
		<i>Or when:</i>

If you change this property	For this type of HMI tag	The change takes effect when
		An attempt is made to write to the tag, from a FactoryTalk View SE Client, FactoryTalk View Studio, or the HMI server.
		The Canonical Data Type of the Tag object is read using the VBA code.
Scale	HMI analog tags with Device as data source	Immediately. The Value property is updated to use the new Scale value as soon as the property is modified.
Offset	HMI analog tags with Device as data source	Immediately. The Value property is updated to use the new Offset value as soon as the property is modified.
Initial Value	HMI analog tags with Memory as data source	The tag's Data Source Type is changed from Device to Memory.
		The Initial Value property is read using the VBA code.
		The HMI server is started.
Retentive	Any HMI tag with Memory as data source	Immediately, if the tag is not in use. Otherwise, the change takes effect when the HMI server is started.
Off Label or On Label	HMI digital tags	A graphic display using the Off Label or On Label property, in a tag label object (Contact Close Label or Contact Open Label) is opened in a FactoryTalk View SE Client.
		The Off Label or On Label property is read using the VBA code.
Length	HMI string tags	Immediately. The Value property is updated to use the new Length value as soon as the property is modified.
		<i>Or when:</i>
		An attempt is made to write to the tag, from a FactoryTalk View SE Client, FactoryTalk View Studio, or the HMI server.
		The Length property is read using the VBA code.



**Tip:** The Name and Tag Type properties of an HMI tag cannot be modified after the tag is created.

## Managing HMI data in an online redundant system

To help ensure that HMI data generated in an online redundant system is as accurate and accessible as possible, keep the following considerations in mind.

### Centralize storage of diagnostic data

Diagnostic log files are stored on every computer where system activity is generated.

For network distributed applications, it is highly recommended that you log diagnostic information to a central ODBC database, such as Microsoft SQL Server, even for HMI servers that are not redundant.

A central, system-wide ODBC log can be made secure and redundant through features of the database. Central logs also simplify troubleshooting, by letting you search all diagnostic information in one location.

For additional protection, it is also recommended that you set up FactoryTalk View SE local diagnostic to buffer logged data, in the event that communications with the ODBC database are lost.

For information about setting up a central ODBC database, see [Logging system activity on page 253](#).

### Determine which server will run events

Events that are triggered by an event detector are not synchronized specifically between primary and secondary HMI servers.

However, it is possible to manage which server is responsible for detecting and running events, so that only one server is active at a time.

Use an HMI server's On Active and On Standby macros, to run the **EventOn** command (starts event detection) when the HMI server becomes active, and to run the **EventOff** command (stops event detection) when the HMI server goes on standby.

This will automatically ensure that event detection is only run on the Active (primary or secondary) HMI server.

For information about On Active and On Standby macros, see [Specifying On Active and On Standby macros on page 244](#).

For information about creating macros, see [Adding logic and control on page 423](#).

### Synchronize derived tags and data log files

To keep derived tags and data logs synchronized, ensure that the same derived tags components and data log models are running on the primary and secondary computers.

You can also keep memory tags synchronized, if their values are the result of derived tags.

For information about replicating changes, see [Replicate changes to the secondary HMI server on page 242](#).

### Monitoring network client and server connections

The health monitoring system monitors network connections on all computers hosting application clients and servers in a network distributed application.

The system does the following connection monitoring:

- The **computer detection interval** sets how often the system attempts to detect whether a computer exists on the network. The default interval is two seconds.
- The **network failure detection interval** sets how often the system attempts to verify the health of the network connection to remote computers. The default interval is 2 seconds.
- The **maximum network glitch** sets the amount of time used to distinguish a temporary network disruption from an actual communications failure. For more information, see "About network glitches," next.
- The **maximum delay before server is active** sets the maximum amount of time during a switch back to the primary server, that the server will wait for clients to respond, before it becomes active. For more information, see [Notifying clients when switching back to the primary on page 248](#).

You can change the default settings in the Health Monitoring Policy Settings dialog box.

### To change Health Monitoring Policy Settings:

1. In FactoryTalk View Studio, in the Explorer window, expand the System folder, double-click the Policies folder, and then double-click the System Policies folder.
2. Double-click Health Monitoring Policy, and then select the policy setting you want to change.
3. Click the amount of time for the policy setting. To select a different time, click the up or down arrow beside the time.



**Tip:** Settings in the System folder, including the Health Monitoring Policy Settings, are stored at the FactoryTalk Network Directory, and apply to all application servers the directory manages.

## About network glitches

Sometimes communications across a network are temporarily disrupted, for fractions of seconds, by noise or brief disconnections.

When this happens, it is possible for the Standby server in a redundant pair to lose contact with its Active partner, and assume it must become the Active server.

To prevent the Standby server from becoming active before it is necessary, the health monitoring system distinguishes a temporary disconnection—called a network glitch—from an actual communications failure.

If the Standby server can re-establish contact with its Active partner within a set time period, then it remains on standby. If the time period expires before contact is re-established, then the Standby server becomes the Active server.

The default time period that defines a network glitch is 5 seconds. You can change the definition by modifying the policy setting, Maximum network glitch. For details, see [Monitoring network client and server connections on page 251](#).



**Tip:** In a partitioned network, if clients are connected to both partners in the redundant pair on either side of a network switch, it is possible for both the primary and the secondary server to become active. For more information, see [What happens if both servers become active on page 244](#).

# Logging system activity

This chapter describes:

- What FactoryTalk Diagnostics is.
- Key FactoryTalk Diagnostics concepts.
- Setting up FactoryTalk Diagnostics in FactoryTalk View.
- Tracking system events in the Diagnostics List.
- Viewing FactoryTalk Diagnostics logs.

## About FactoryTalk Diagnostics

FactoryTalk Diagnostics records information about system activity, including:

- Command and macro usage
- Operator comments
- System messages and errors
- Errors from the communication network
- Tag read and write activity

This information can be:

- Viewed in the Diagnostics List or the Diagnostics Viewer.
- Archived for future processing or analysis.
- Exported to ODBC format while online.

Exporting to ODBC format let you analyze the data in third-party, ODBC-compliant tools such as Microsoft Excel, and Business Objects Crystal Reports.

## Summary of steps for setting up Diagnostics

These are the tasks involved in setting up FactoryTalk Diagnostics for a FactoryTalk View SE application:

1. Decide which computers on the network need to log system activity.
2. On each computer where system activity will be logged, use the FactoryTalk Diagnostics Setup tool to set up destinations, and message routing.  
For more information, see [Setting up message routing on page 256](#).
3. To log system activity to a central, ODBC-compliant database, first set up the database, and then, in the FactoryTalk Diagnostics Setup tool, set up the ODBC Database destination.

## Finding more information about FactoryTalk Diagnostics

For additional information about FactoryTalk Diagnostics and for detailed setup instructions, see the FactoryTalk Diagnostics Help.

### To open the FactoryTalk Diagnostics Help:

1. Select **Start > Rockwell Software > Diagnostics Viewer**.
2. In the FactoryTalk Diagnostics Viewer window, select **Help > FactoryTalk Diagnostics Viewer Help**.

FactoryTalk View provides the Diagnostics Viewer object to view diagnostics on HMI displays.

## To open the FactoryTalk Diagnostics Help from FactoryTalk View Studio:

1. In the Graphic editor, select **Objects > Advanced > Diagnostics Viewer**.
2. Drag the mouse into position and draw a rectangle.
3. In the **FTDiagnosticsViewer Class Properties** dialog box, click **Help**.

## Key FactoryTalk Diagnostics concepts

This section presents some of the common terms and concepts used to describe FactoryTalk Diagnostics services.

### Destinations

FactoryTalk Diagnostics messages can be sent to various destinations, including the message log on the local computer, an ODBC-compliant database, and the Diagnostics List in FactoryTalk View Studio, or the FactoryTalk View SE Client.

By default, system activity is logged locally on each computer.

### Message routing

Message routing lets you specify which destinations receive messages of a certain severity, for a certain audience. This ensures that information is provided to the appropriate person and place. For example, you might decide to:

- Route messages that contain information about system activity, and warnings about things that might go wrong, to the local log.  
This would let a control systems engineer to analyze system activity and performance, and make corrections during scheduled maintenance times.
- Route errors that require immediate action to the FactoryTalk Diagnostics List, and to the local log.  
At runtime, if the FactoryTalk Diagnostics List is visible, an operator can alert the plant's control system engineers to problems such as tag errors, as they occur.  
During scheduled maintenance time, the engineer can use the errors, together with warning or information messages recorded in the local log, to analyze the operation of the system and make the necessary corrections.

For more information, see [Setting up message routing on page 256](#).

### Message categories

FactoryTalk Diagnostics messages are categorized by severity and audience.

To route messages, you specify a destination for messages of a particular severity and audience. For example, you can specify the FactoryTalk Diagnostics List as the destination for Errors that have Operator as the audience.



**Tip:** You cannot change the audience or the severity categories assigned to Diagnostics messages. For example, you cannot specify that a Developer receive all messages of the Error type.

## Message severity

FactoryTalk View SE uses four message severities:

- **Errors** indicate that a process or action has failed. For example, an error might indicate that a tag's value could not be written, or an ActiveX object is missing.
- **Warnings** indicate that a process or action might not function correctly, or might eventually fail if preventive action is not taken. For example, if an ActiveX object in a graphic display is a different version than the one installed at the FactoryTalk View SE Client, a warning is logged to indicate the mismatch. Mismatched ActiveX controls might not behave as expected at runtime.
- **Information** messages indicate that a process or action has completed successfully. For example, an information message is logged when a user signs in to the system.
- **Audit** messages indicate that the system configuration has been changed.



**Tip:** FactoryTalk View SE records an Audit message for all tag writes, and whenever a component is added, removed, or deleted from a multi-component editor, such as the Data Log Models, Graphics, or Macros editor.

## Message audience

FactoryTalk View SE uses three message audiences: Operator, Engineer, and Developer.

FactoryTalk View assigns the Operator audience for all messages that it generates, except for messages with the Audit severity. Audit messages, including tag write confirmations, are assigned to the Developer or Engineer audience.

The Secure audience is reserved for auditing tools, such as those required for US Government 21 CFR Part 11 compliance, to track system activity.

## How tag writes are categorized

The severities assigned to tag writes that succeed Audit and Information. Severities assigned to tag writes that fail are Audit, Information, and Error.

The audiences assigned to tag writes that succeed or fail are Developer and Secure.

## Setting up FactoryTalk Diagnostics in FactoryTalk View

Use the FactoryTalk Diagnostics Setup tool to set up message routing, logging to a central database, and the local log, on each computer where system activity is to be logged.



**Tip:** You must run the FactoryTalk Diagnostics Setup tool separately on each computer where system activity is to be logged. FactoryTalk Diagnostics settings apply to all the FactoryTalk products installed on a single computer.

## To open the FactoryTalk Diagnostics Setup tool:

In FactoryTalk View Studio or in the FactoryTalk View SE Administration Console, from the Tools menu, select **Diagnostics Setup**.

For details about options in the FactoryTalk Diagnostics Setup tool, click **Help**.

## Setting up message routing

When you set up message routing, you specify where messages associated with a particular audience and severity will be logged.

For example, you can specify that tag write messages logged for the Developer audience are sent to the local log, to the Diagnostics list, or to neither destination.

All messages are logged as Error, Warning, or Information, with the audience Operator. The exception is tag write messages, which are logged as Information and Audit, with the audiences Developer and Secure.

---

**NOTE:** If messages for a particular combination of audience and severity are not routed to a destination, they will not be logged.

---

For more information about audience and severity, see [Message categories on page 254](#).

For details about message routing options, click **Help** in the Diagnostics Setup tool.

## Logging to a central database

In addition to logging information locally on each computer, you can set up FactoryTalk Diagnostics to log to a central, ODBC-compliant database.

Central logging works by exporting the contents of the local log periodically to an ODBC-compliant database such as Microsoft SQL Server (fully tested), Microsoft Access, or Oracle.

Logging to a central database provides redundancy for Diagnostics logging. If you lose communications with the ODBC-compliant database, you can use the local log to buffer information.

For details about central logging options, click **Help** in the Diagnostics Setup tool. For information about the ODBC tables, see the FactoryTalk Diagnostics Help.



**Tip:** If you have set up FactoryTalk Diagnostics to overwrite events in the local log, make sure that messages are logged to the ODBC-compliant database, before the oldest events are deleted.

---

## Setting up message buffering

FactoryTalk Diagnostics messages are stored in the computer's local log and exported to an ODBC-compliant database at the interval that you specify.

To buffer messages, in the ODBC Database Destination Setup window, specify how long messages will remain in the local log after they have been exported.

This is useful in the event that the ODBC-compliant database becomes unavailable, for example, due to a network failure. In such cases, messages will remain in the local log until the buffer time expires.

If the ODBC-compliant database becomes available before the buffer time expires, the buffered messages will be exported to the database.

## Tracking system events in the Diagnostics List

To track what is happening when an application is running, use the Diagnostics List in FactoryTalk View Studio, the FactoryTalk View SE Administration Console, or the FactoryTalk View SE Client.

To include system activity messages in the Diagnostics List, they must be routed to the list. For information about routing messages, see [Setting up message routing on page 256](#).



**Tip:** For the FactoryTalk View SE Client, the option to show, hide or undock the Diagnostics List is set up when you create a client file. For details, click **Help** in the FactoryTalk View SE Client Wizard.

## Working with the Diagnostics List

When FactoryTalk View Studio first starts, the Diagnostics List is visible and is docked above the status bar in the FactoryTalk View main window.

### Showing or hiding the Diagnostics List

To show or hide the Diagnostics List in FactoryTalk View Studio (or the FactoryTalk View SE Administration Console), from the **View** menu, select **Diagnostics List**. When the Diagnostics List has a check mark beside it, the list is visible.

### Moving the Diagnostics List

You can detach (undock) the Diagnostics List from the main window, and then move the Diagnostics List anywhere on the screen.

To undock the Diagnostics List, drag the grab bars at the bottom left of the Diagnostics List. If you can't see the grab bars, drag the top edge of the Diagnostics List to make it larger.

To prevent the Diagnostics List from docking automatically while you move it, hold down the Ctrl key on the keyboard.

### Resizing the Diagnostics List

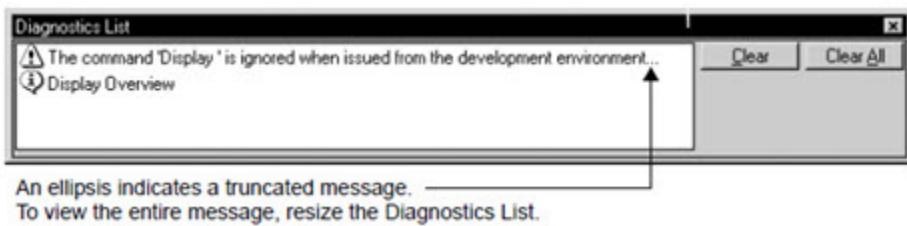
When the Diagnostics List is undocked, you can make it any size you want. For example, to view more than one message at a time, make the list larger.

To resize the Diagnostics List, drag an edge or corner until the list is the desired size.

## Viewing messages in the Diagnostics List

Messages in the Diagnostics List are preceded by a blue, yellow, gray, or red icon: blue indicates information, yellow indicates a warning, gray indicates an audit message, and red indicates an error.

The following illustration shows a warning and an information message:



The types of messages logged to the Diagnostics List depend on how message routing is set up in the Diagnostics Setup tool.

### Clearing messages in the Diagnostics List

To clear messages in the Diagnostics List, use one of these methods:

- To clear the most recent message or the selected message, click **Clear**.
- To clear all messages in the list, click **Clear All**.

## Viewing FactoryTalk Diagnostics logs

Use the Diagnostics Viewer object or the FactoryTalk Diagnostics Viewer tool to view the contents of Diagnostics logs.

### To add the Diagnostics Viewer object to a display:

1. In the **Toolbox**, select **Advanced > Diagnostics Viewer**.
2. Drag the mouse into position and draw a rectangle.
3. In the **FTDiagnosticsViewer Class Properties** dialog box, set up the properties as needed.

### To open the FactoryTalk Diagnostics Viewer tool, do one of the following:

- Select **Start > Rockwell Software > Diagnostics Viewer**.
- In FactoryTalk View Studio or in FactoryTalk View SE Administration Console, select **Tools > Diagnostics Viewer**.

### To open the FactoryTalk Diagnostics Viewer tool at runtime:

1. In the Graphic editor, create a button that an operator uses to open the tool.
2. In the **Button Properties** dialog box, for the press action, type the following command:

```
AppStart "C:\Program Files\Rockwell Software\RSView Enterprise  
\ActivityLogViewer.exe"
```



**Tip:** At runtime, due to operating system rules, the Diagnostics viewer might open behind the FactoryTalk View SE Client window. If you are unaware of this, and try to open the viewer again, another instance will open. This could result in multiple viewers being open at the same time. To avoid this, you can bring the viewer to the front manually, and close it when it is no longer required. For information on how to use the programmatic solution, see Knowledgebase Document ID: [BF15995 - Use Windows to Move External Application Window to Foreground](#).

# Creating graphic displays

This chapter describes:

- What graphic displays are.
- Parts of the Graphics editor.
- Techniques for working in graphic displays.
- Techniques for working with graphic objects.
- Testing graphic displays.
- Creating and working with global object displays.
- Adding Application Libraries to an application.
- Using objects from the graphic libraries.
- Importing graphic files from third-party applications.
- Using placeholders to specify tag values.
- Setting up tag placeholders for global objects.
- Setting up the appearance and behavior of graphic displays.
- Docking displays to the FactoryTalk View SE Client window.
- Printing graphic displays at runtime.

## About graphic displays

A graphic display represents a runtime operator's view of plant activity.

A graphic display can show system or process data, and provide an operator with ways to write values to external devices such as programmable controllers. To create a visual record of tag values, the operator can print the display at runtime.

A graphic display can contain up to a total of 3000 connections, whether they originate from the expressions or the tags.

- Each expression associated with an object is counted as one connection regardless of the number of tags in the expression.
- Each animation of an object (except the Touch animation) is counted as one connection.
- Each connection in the Connections property of an object that is linked to a tag is counted as one connection.
- Each pen configured in a Trend object is counted as one connection.
- Duplicate references of the same expression or tag connection are counted as the additional connections. For example, one display can contain up to 3000 numeric inputs, even if all numeric input objects refer to the same tag.



**Tip:** Tags associated with embedded variables do not count towards the limit.

Graphic displays are made up of graphic objects, which can be:

- Created in the Graphics editor.
- Dragged and dropped from a graphic library.

- ActiveX objects embedded in the graphic display.
- Created by another Windows application, then copied and pasted into a display, or inserted using OLE (Object Linking and Embedding).

## Importing and exporting graphic display XML files

The information a graphic display contains is stored in a file called *displayname.gfx* (where *displayname* is the name you give to the graphic display).

In FactoryTalk View Studio, you can use the Graphics Import Export Wizard to export graphic display information to an XML file, or to import a graphic display XML file into an application.

You can edit the XML files before importing them back into an application, to modify existing graphic objects, or to add new objects to a display.

For more information about importing and exporting graphic display files, see [Importing and exporting XML files on page 479](#).

## About global object displays

FactoryTalk View global objects let you link the appearance and behavior of one graphic object to multiple copies of the object in the same application.

Global objects are created on global object displays. In FactoryTalk View Studio, you create global object displays in the Global Objects folder, the same way you create standard graphic displays in the Displays folder.

All of the objects and groups of objects on a global object display are global objects. Any graphic object that you can create in FactoryTalk View can be a global object, except for ActiveX controls and OLE objects.

## Techniques for working in graphic displays

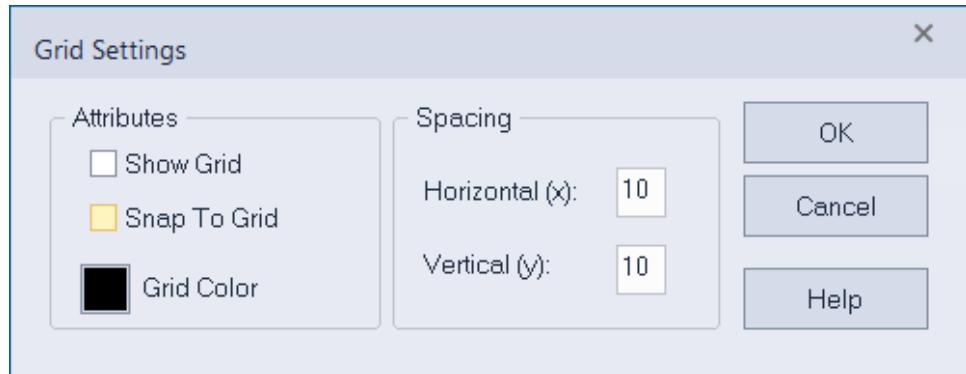
When working with graphic displays, you will use certain actions and techniques frequently. Knowing how to perform these actions can save time.

### Zooming in and out

To magnify or reduce the view of a graphic display, use the Zoom In or Zoom Out option on the **View** menu. Zoom In magnifies objects, while Zoom Out reduces magnification.

## Setting up a display grid

To size and position objects precisely, use the grid items on the View menu. You can change the grid settings anytime during the drawing process.



Use the grid to simplify aligning and sizing objects. When the grid is on, all objects you draw or place are aligned to the grid automatically.

## Techniques for working with graphic objects

When creating and modifying graphic objects in a display, you use certain actions and techniques frequently.

Knowing how to perform these actions can save time.

For specific information about different objects, see [Creating graphic objects on page 288](#).

## Copying objects

To copy objects, you can drag objects between displays, or from a global object display to a standard display.

When an object is copied, any animation attached to the object is also copied. If a group is copied, the new copy of the group can be ungrouped to individual objects, just like the original.

## Copying objects with multiple languages

If an object has multiple language strings set up, copying the object copies all the languages. You have two options for pasting an object with multiple language strings into an application:

- If you use the **Paste** command, the object is pasted into an application with different languages, only the strings for languages that are used by the application are pasted. If the new application has languages that are not set up for the object, those language strings are undefined and will be shown with single question marks.
- If you use the **Paste without localized strings** command, the object is pasted with only the current localized language. The **Paste without localized strings** command removes all other language strings from the object and sets the language strings to Undefined.

For more information about setting up multiple languages, see [Setting up language switching on page 210](#).

### To copy objects in the same display:

1. Select one or more objects.
2. Drag the object, and then press **Ctrl**.

When you press **Ctrl**, a plus sign is added to the cursor.

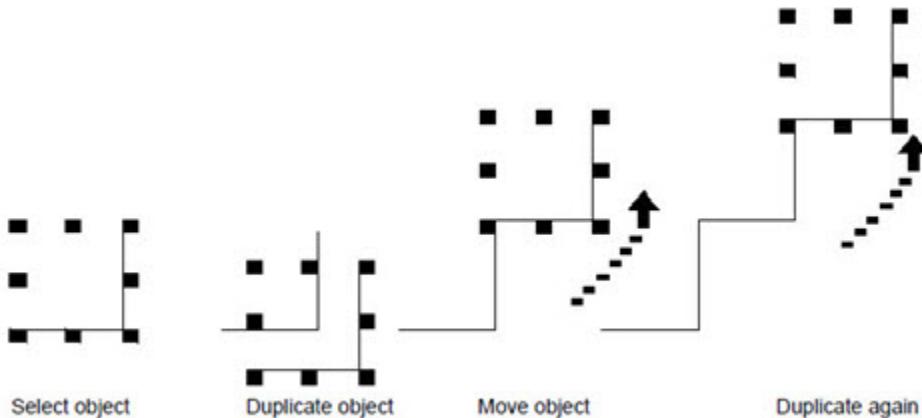
- When the object is where you want it, release the mouse button, and then the Ctrl key.

A new copy of the object is created.

If you selected several objects, dragging one of the objects copies all the selected objects. The objects maintain their position relative to each other.

## Duplicating objects

When you duplicate graphic objects in a display, you also duplicate actions. For example, if you duplicate an object, move it, and then duplicate it again, the second duplicate action copies and moves the object, in one step. This is useful for creating a series of objects with an equal distance between them.



**To duplicate multiple copies of a numeric or string object:**

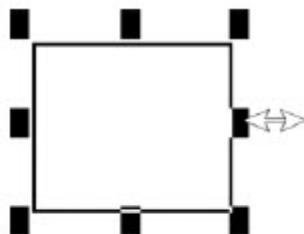
Right-click the object in a display, select **Duplicate to Grid**, and specify the columns, rows and spacing.

The following illustration shows an example that duplicates multiple copies of a numeric display object with three columns and six rows.

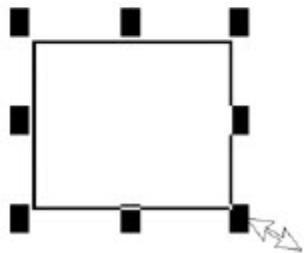
## Resizing and reshaping objects

When you select an object, handles appear around it. You can then click and drag the handle, to resize and reshape the object.

You can also press **Shift** and the arrow keys on the keyboard, to resize and reshape objects in small increments.



Drag a side handle to change either the width or height.



Drag a corner handle to change both the width and height.

You can reshape some objects using the object's editing tool. To do this, right-click the object, and then select **Edit**. The tool varies depending on the object that has been selected.

### To maintain an object's proportions while resizing:

Click a corner handle, press **Shift**, and then drag the mouse.



**Tip:** If you attempt to resize a global reference object with its LinkSize property set to True, the object will snap back to its original size. For more information about the LinkSize property, see [Setting up the link properties of reference objects on page 305](#).

## Arranging objects

You can arrange objects in a graphic display in a number of ways, using the **Arrange** menu or buttons on the toolbar.

You can:

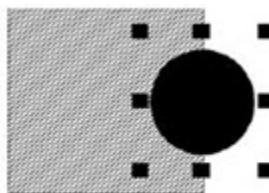
- Stack objects by moving them in front of or behind other objects.
- Align objects with each other.
- Space objects horizontally or vertically.
- Flip objects horizontally or vertically.
- Rotate objects around an anchor point.
- Combine several objects into a group that behaves as a single object.
- Separate a grouped object into its component objects.

## Stacking objects

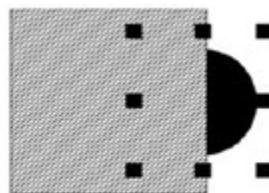
Objects in a graphic display are stacked in the order that they are created, with the most recently created object on top.

To change the order, move objects to the front or back of the stack.

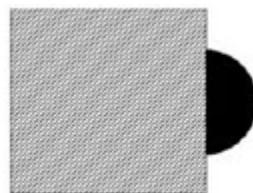
Use  to move the selected object to the bottom of the stack.



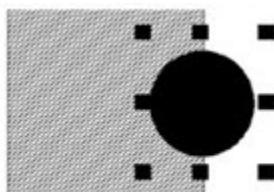
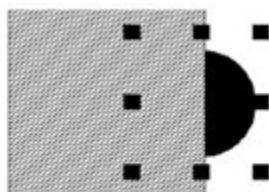
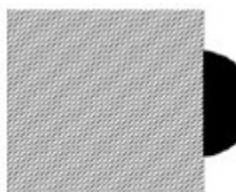
Select the object.



Send the object to back.



Use  to move the selected object to the top of the stack.



Select the object from behind. Bring the object to front.

### To select the object at the back:

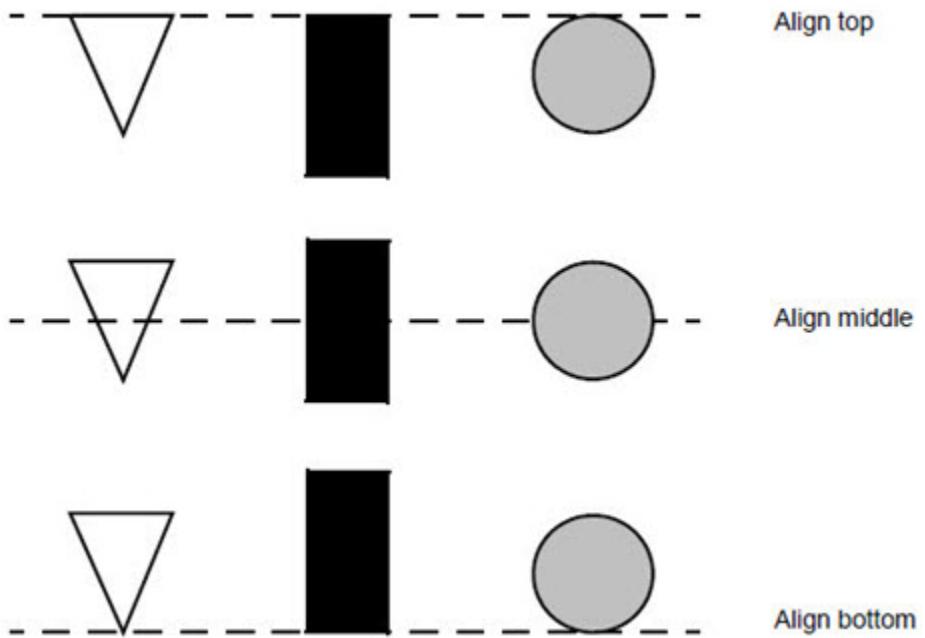
Click the top object once, pause, and then click again. Do not double-click, and do not move the mouse.

## Aligning objects

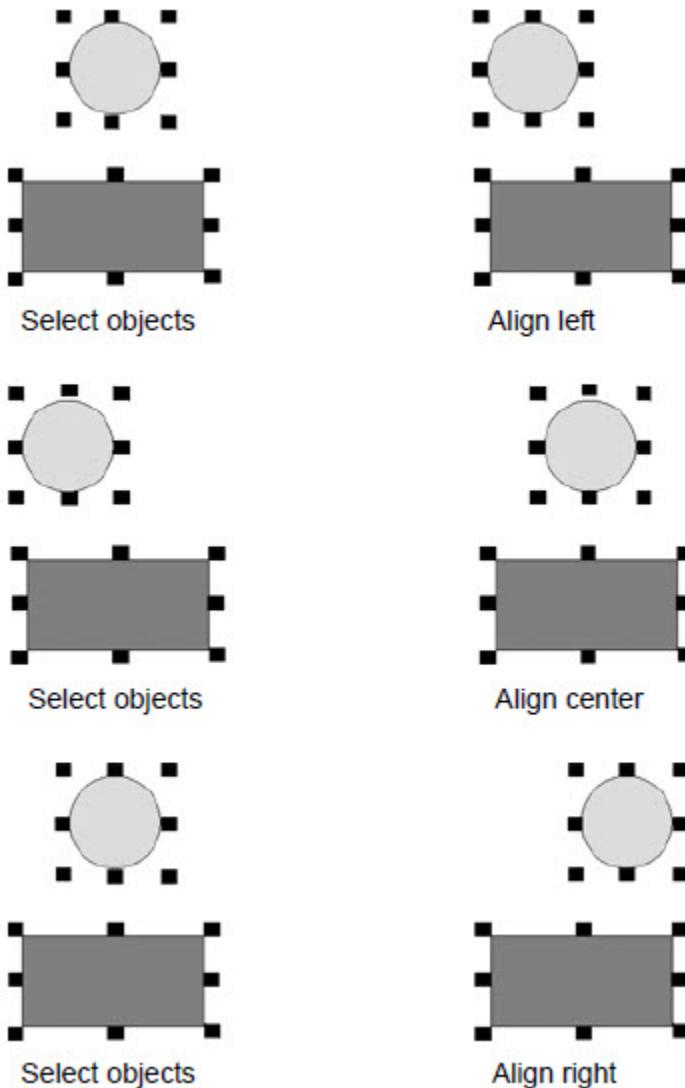
To line up the tops, bottoms, or sides of objects in a display, align the objects with each other, or with the display grid.

To do this	Click this button or menu item
Align the selected object (or objects) with the leftmost selected object.	 Align Left
Align the selected object (or objects) with the right-most selected object.	 Align Right
Align the selected object (or objects) with the horizontal center of the largest selected object.	 Align Center
Align the selected object (or objects) with the top-most selected object.	 Align Top
Align the selected object (or objects) with the bottom-most selected object.	 Align Bottom
Align the selected object (or objects) with the vertical center of the largest selected object.	 Align Middle
Align the selected object (or objects) with the center of all selected objects.	 Align Center Points
Align the selected object (or objects) with the grid.	 Align to Grid

## Aligning objects top, middle, and bottom



## Aligning objects left, right, and center

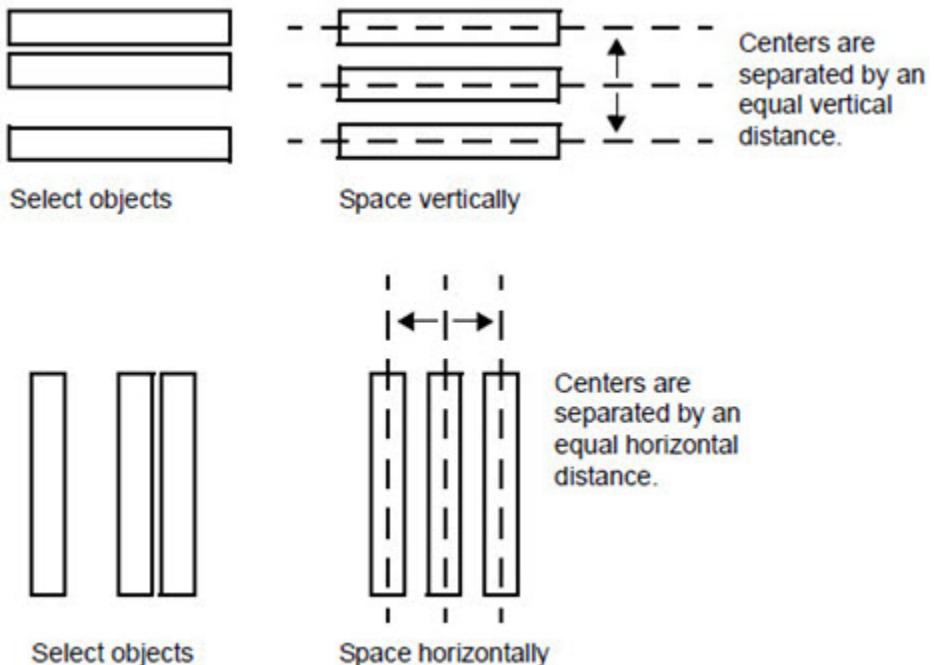


## Spacing objects

To create an equal amount of space between the *center points* of objects in a graphic display, space the objects vertically or horizontally.

To do this	Click this button or menu item
Place the centers of the selected objects an equal distance apart vertically.	Space Vertical
Place the centers of the selected objects an equal distance apart horizontally.	Space Horizontal

## Spacing objects vertically and horizontally

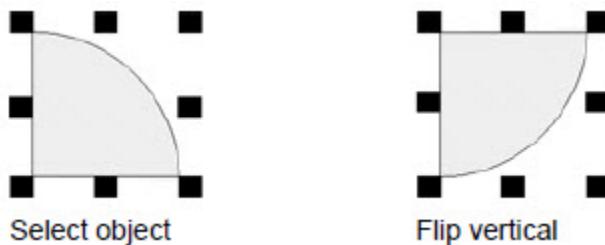


## Flipping objects

To move an object in a graphic display to a position that is a mirror image of its original position, flip the object vertically or horizontally.

To do this	Click this button or menu item
Flip the selected object (or objects) top to bottom (upside-down).	Flip Vertical
Flip the selected object (or objects) left to right	Flip Horizontal

### Flipping objects vertically

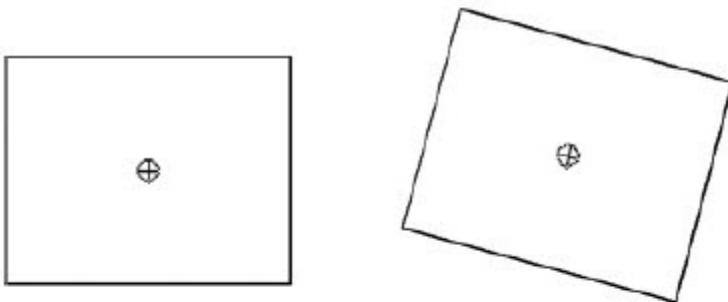


### Flipping objects horizontally

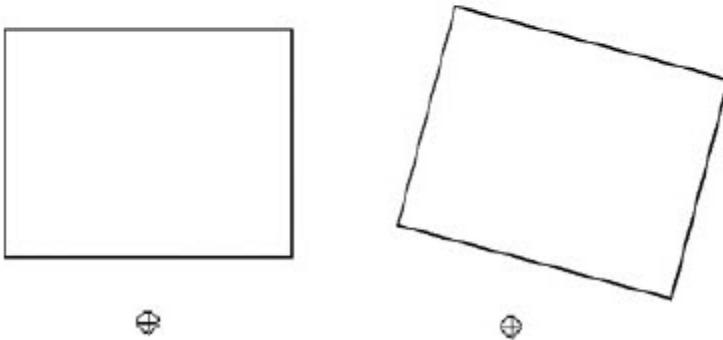


## Rotating objects

To rotate an object or group of objects around an anchor point, use . The anchor point is represented by a crosshair, as shown in the following illustrations:



You can place the crosshair inside an object.



You can place the crosshair outside an object.

You can also use the Rotate tool when attaching rotation animation to a graphic object.

---

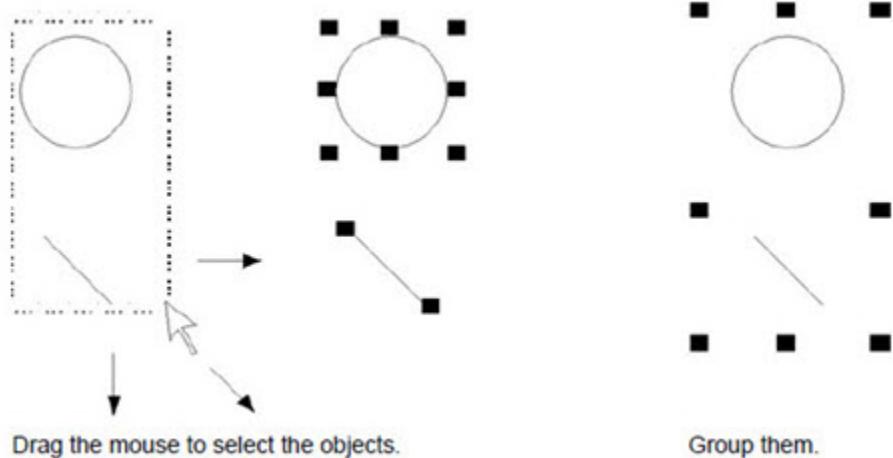
**NOTE:** You cannot rotate OLE objects, ActiveX objects, bitmaps, text, or panel objects.

---

## Grouping objects

Grouping combines several objects into one so you can manipulate them as a single object. Grouping is useful for keeping objects in the same position relative to each other.

You can cut, copy, and paste groups, arrange the group as a single object relative to other objects, and apply the same properties to all the members of the group at once.



You can attach animation to a group and preserve any animation attached to the objects that make up the group.

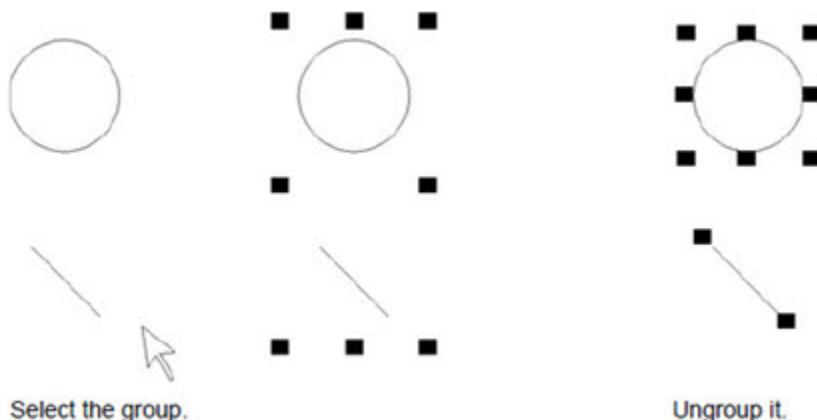
Group animation generally takes precedence over the animation attached to individual objects. For more information, see [Applying animation to object groups on page 316](#).

Deleting a group deletes all individual objects in the group. Changing the color or pattern style of the group changes the color or pattern style of all individual objects in the group.

## Ungrouping objects

When you disconnect a group of objects, the individual objects in the group are selected in the graphic display.

Ungrouping deletes the animation attached to the group, because the group no longer exists. However, any animation attached to the individual objects remains active.



## Modifying grouped objects

You can modify a group of objects without breaking up the group. This is particularly useful when an animation is attached to the group, because ungrouping deletes the animation.

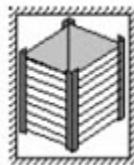
Modify grouped objects using:

- The Property Panel, to apply shared properties to all members of the group at the same time. For example, change the line width of all objects in the group to two pixels.
- Toolbars, to apply the same pattern style, background style, foreground colors, and background colors to all members of the group.
- The States toolbar, to cycle through the states and apply the same properties to them, for all members of the group at the same time.

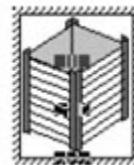


**Tip:** When you select a group containing objects with states, only the states shared by all objects in the group are shown in the toolbar. For information about using the States toolbar, see [Testing the appearance of objects in different states on page 273](#).

To modify the objects within a group, double-click the group. A rope-like border indicates that the group is in edit mode.



Double-click a group to place it in edit mode.



Click inside the box to select objects within the group.

Click inside the box to select individual objects or other groups within the group. You can also add new objects to the group.

To see which objects are selected, look in the Object Explorer or the FactoryTalk View Studio status bar.

To cancel group edit mode, click outside the group border.

## Changing line properties

The line styles are:

**Solid**



**Dash**



**Dot**



**Dash-Dot**



**Dash-Dot-Dot**



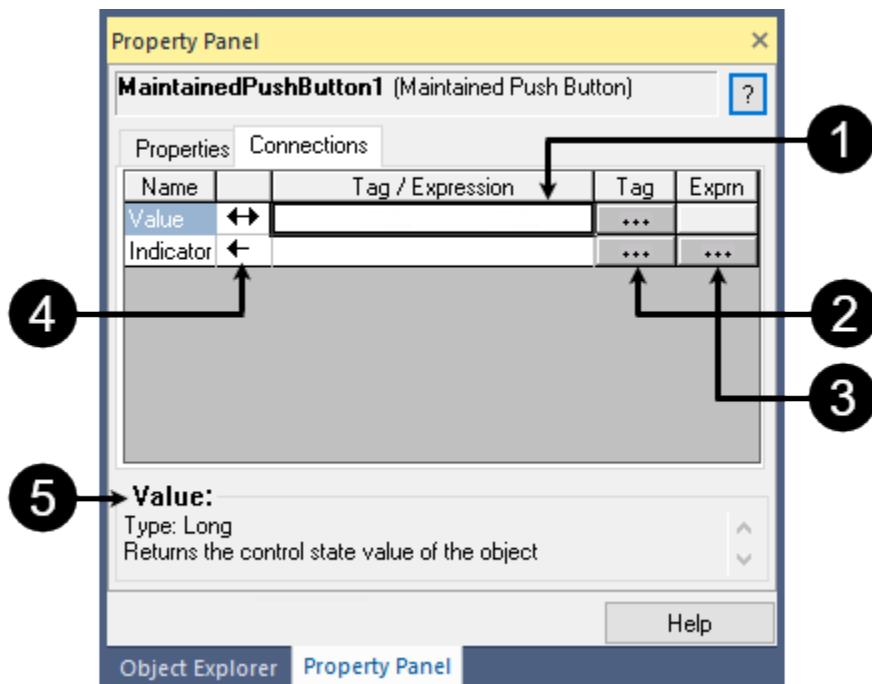
Line style uses both the foreground color and background color attributes. Foreground color applies to the line, and background color applies to the spaces in the line.

For example, to obtain the dash-dot line, choose black as the foreground color, and choose white as the background color. Black is applied to the dots and dashes and white is applied to the spaces between the dots and dashes.

## Assigning tags and expressions to objects

To assign tags or expressions to an individual object, use the Connections tab in the object's Properties dialog box, or in the Property Panel.

The following illustration is of the Property Panel for a maintained push button. In the Connections tab, you can set up a Value tag and an Indicator tag or expression



Items	Description
1	Type a tag name or expression in this column.
2	Click this button to open the Tag Browser.
3	Click this button to open the Expression editor.
4	The arrows show the direction in which data flows between the object and the tag or expression.
5	This box describes the selected connection, and the type of data it uses.



**Tip:** The Connections tab is blank, if multiple objects are selected. You can only assign a tag or expression to one object at a time.

The arrows beside the connection names indicate the direction data flows between the tag or expression and the object:

- A right arrow indicates that data flows from the object to the tag or expression only. In other words, the object can write to the tag or expression.
- A left arrow indicates that data flows from the tag or expression to the object only. In other words, the object can read from the tag or expression.
- A double arrow indicates that data flows in both directions. In other words, the object can write to, or read from, the tag or expression.

## Using tag substitution to replace text strings

Use tag substitution to replace the text strings in graphic objects or embedded variables in a graphic display.

You can find and replace text strings in tag names, expressions, and FactoryTalk View commands.

You cannot use tag substitution to replace text created using the Text tool.

## Creating a background for a display

You can create a background for a graphic display by converting objects to wallpaper. For example, you can import photographs of a machine or process, convert the images to wallpaper, and then overlay the wallpaper with animated objects.

When graphic objects are converted to wallpaper, they are locked into position and become a static background for other objects in the display. You cannot select, modify, or animate wallpaper objects.

To reactivate converted objects, and to restore any animation attached to the objects, unlock the wallpaper.



**Tip:** Objects in a global object display cannot be converted to wallpaper.

If a graphic display contains bitmaps that do not need to be selected or animated, to reduce the time it takes to open the display, convert the bitmaps to wallpaper.

## Testing graphic displays

You can test the objects in a graphic display quickly, by switching to the test display mode in the Graphics editor.

When you are finished testing, to continue working on the display, switch back to edit display mode . To switch between test and edit modes, use the buttons on the toolbar or the items on the View menu.

Testing a graphic display in FactoryTalk View Studio is not the same as running the display in the FactoryTalk View SE Client.

Test display mode does not change the appearance or position of the display, as set up in the Display Settings dialog box, and you cannot switch between open displays.

In addition, some FactoryTalk View commands are ignored when run in test display mode. For a complete list of these commands, see [FactoryTalk View commands on page 442](#).



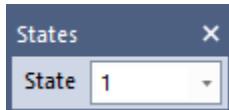
**Tip:** If objects in a graphic display are connected to tags in devices, to fully test the display you must set up communication with the devices or data servers.

Before you deploy an application, it is recommended that you test it in the FactoryTalk View SE Client, to verify that everything works as intended.

## Testing the appearance of objects in different states

Some objects have multiple states. You can set up each state differently, so the object's appearance changes whenever the state changes.

To make sure the different states for an object are set up correctly, view each state using the States toolbar or the Property Panel.



## Creating global object displays

To create a global object display, create a display in the Global Objects folder, or add a standard display or graphic library into the Global Objects folder.

When you add a standard display into the Global Objects folder, the graphic objects in the display convert to global objects, except for ActiveX controls and OLE objects, and reference objects with broken links. These objects are deleted.

### About global object display files

Global object display files are saved with the extension .ggfx on the development computer. The default path is:

C:\Users\Public\Documents\RSView Enterprise\SE\HMI projects\<HMI Project Name>\Global Objects.

where <HMI Project Name> is the name of the HMI server in the Explorer window.



**Tip:** The number of global object displays in an application does not count toward the maximum number of licensed displays the application can contain. For information about activation and licensing, see the *FactoryTalk View Site Edition Installation Guide*.

## Adding standard displays that contain reference objects

In a network distributed application, if you add a standard display or graphic library containing linked reference objects into the Global Objects folder, the linked reference objects:

- Convert to global objects, if added within the same HMI server
- Are deleted, if added to a different HMI server



**Tip:** Existing reference objects with broken links are always deleted, whether you add them within the same HMI server, or to a different one.

## About global object displays at runtime

When a graphic display containing global reference objects is run, the global object display (or displays) containing the linked base objects also runs in the background.

When you modify the base object in a global object display, all linked reference objects are updated with the changes, the next time their host displays are opened or refreshed.

Since global object displays run in the background, you cannot select or specify one:

- As a parameter for the **Display** command.
- From the Component Browser, when creating a display list selector object.
- As the initial display, when creating a client file in the FactoryTalk View SE Client Wizard.

However, you can run a global object display in test display mode, when developing an application in FactoryTalk View Studio.

The runtime performance of applications that leverage a large quantity of global objects can be impacted. To optimize the runtime performance of the FactoryTalk View SE client, click  on the toolbar to compile global objects. An asterisk shows on , indicating that there are pending changes in the system that require compilation.

## Adding Application Libraries to an application

When you add an HMI server to a network distributed or station application, you have the option to add one or multiple Application Libraries that are installed with FactoryTalk View SE.

Application Libraries provide visualization elements such as global objects and faceplates that let you quickly build applications. The Application Libraries that are installed along with FactoryTalk View SE include Rockwell Automation Device Object Libraries and Rockwell Automation Library of Process Objects. These libraries are stored in C:\Users\Public\Documents\RSView Enterprise\SE\Application Libraries. You can add Application Libraries to a new or an existing HMI server.

The installed Application Libraries and their versions are:

- Device Object Libraries
  - Condition Monitoring Device Library - <version>
  - IO Device Library - <version>
  - IO Link Device Library - <version>
  - Network Device Library - <version>

- Power Device Library - <version>
- Safety Device Library - <version>
- Library of Process Objects
  - Process Library - <version>
  - Process Library - <version>

<version> is a placeholder, representing the library version.

To download the latest versions of Device Object Libraries and Library of Process Objects, go to the [Product Compatibility and Download Center](#) and search "Device Object Libraries" and "Process Library". To add the libraries, you can place them in the **Application Libraries** folder and import them together, or you can import them separately.

### Remark

- Adding the Application Libraries affects the license counts because it adds extra displays to the application. Each added faceplate (.gfx file) counts as one display for activation purposes. The corresponding global object display (.ggfx file) does not add to the total license count.
- In a network system where the HMI server and FactoryTalk View Studio are on two computers, to add Application Libraries, the library files must exist on both computers.

## Working with faceplates in the Graphics editor

The faceplate displays you selected are added to the **Graphics** folder, under the **Displays** and **Global Objects** icons.

Like the graphics libraries installed with FactoryTalk View SE, you can use the faceplate displays in an application as they are, or you can copy and then paste the faceplates into existing standard and global objects displays.

## Using objects from the graphic libraries

FactoryTalk View comes with libraries that contain graphic objects and displays. Many of the objects are preconfigured with animation. Use the objects as they are, or change them to suit your needs. You can:

- Look at the objects and displays to get ideas for the application you are creating.
- Drag objects from the libraries into displays.

## Location of library files

Library files are stored in the folder **C:\Users\Public\Public Documents\RSView Enterprise\SE\Libraries** on the local computer.

For information about changing the Libraries location, see the FactoryTalk View Site Edition Help.

## Using placeholders to specify tag values

Tag placeholders can save time spent developing and maintaining applications, by providing a way to use a single graphic display for several similar operations.

For example, to create displays for a plant that uses the same machinery to can corn and beans, instead of specifying corn-related tags in one display and bean-related tags in another, you can create one display and use tag placeholders where tags are required.

At runtime, the placeholders must be replaced with the actual tag names for the different corn and bean processes. To do this, you would specify the actual tag names in parameter files or parameter lists, and then load the appropriate file or list with the display.

Parameter files and parameter lists are loaded using the **Display** command parameters */P* and */T*, respectively. For details, see the examples on [Replacing tag placeholders using parameter lists on page 277](#).

## Creating a tag placeholder

In the following illustration, the tag placeholder **#1** is the value assigned to a numeric input object. In this case, the tag placeholder stands for the tag's full name, which will be provided at runtime or at test display mode.



You can also use one or more tag placeholders to specify parts of a complete tag name. In **#1\PV**, for example, the tag placeholder **#1** stands for the name of the folder that contains an HMI tag named **PV**.

To specify the full tag name at runtime, you would only have to provide the folder name in a parameter file or parameter list.

## Replacing tag placeholders using parameter files

One way to replace tag placeholders in a graphic display with actual names at runtime is to load a parameter file with the display.

The parameter file should contain one entry for each unique tag placeholder in the display. For example, to replace **#1** with a tag named **corn/weight**, you would create a parameter file that specifies **#1 = corn\weight**.

## Loading a parameter file with the initial client display

You can specify a parameter file to load with the graphic display that opens initially, when the FactoryTalk View SE Client starts up.

To do this, in the FactoryTalk View SE Client Wizard, select the display in the Initial display list. Then, in the **Display parameters** box, type */P* followed by the name of the parameter file to load with the display (for example, */PBeans*).

### Example: Replacing tag placeholders using a parameter file

To open a graphic display named Canning and replace the tag placeholders in the display with the names specified in a parameter file named Beans, you would issue this command:

```
Display Canning/PBeans
```

## Replacing tag placeholders using parameter lists

Instead of using a parameter file to replace tag placeholders in a graphic display, you can load a parameter list, using the /T parameter with the Display command.

### Example 1: Replacing tag placeholders with a list of tag names

To run a graphic display named Canning, with the tags Pea\_Weight, Pea\_Level, and Pea\_Temp, type:

```
Display Canning/TPea_Weight, Pea_Level, Pea_Temp
```

### Example 2: Replacing the tag placeholder with a list of folder names

The tag database contains these tags:

Corn\Weight	Bean\Weight
Corn\Level	Bean\Level
Corn\Temp	Bean\Temp

Wherever the tags are needed, the placeholder #1 is used for the folder name, as follows:

```
#1\Weight, #1\Level, #1\Temp
```

To run the display named Canning with the tags in the Corn folder, you would type:

```
Display Canning /TCorn
```

To run the display named Canning with the tags in the Bean folder, you would type:

```
Display Canning /TBean
```

## Setting up the appearance and behavior of a graphic display

Use the Display Settings dialog box to set up the appearance and behavior of a graphic display. You can modify display settings at any time, while you are setting up the contents of the display.

## Specifying the display type

You can set up the following display types in a FactoryTalk View application:

**Replace** is the default display type. If you want the graphic display to replace other open displays when it opens, use the **Replace** option. A display of Replace type closes displays that it overlaps. You don't need to run separate commands to close other open displays.

**Overlay** graphic displays will layer with other displays, overlapping some and being overlapped by others as the focus changes between open displays. If the display doesn't need to replace other displays or appear on top at all times, use the **Overlay** option.

For overlay displays, if you want the display always at the back, you can select the checkbox, **Keep at Back**. However, it is recommended that you use the On Top display type to control the layering of displays.

Overlay displays always appear behind On Top displays, and are replaced by Replace displays. Use the Overlay type with care; keeping multiple displays open at runtime can affect system performance.

**On Top** graphic displays will stay on top at all times, even if another display has focus. If you want the display to always appear on top, use the **On Top** option.

If more than one graphic display of the On Top type is open at once, the display that appears on top is the one that has focus, or the one that had focus most recently.

You can use the **PullForward**, **PushBack**, and **SetFocus** commands to cycle through multiple On Top and Overlay displays at runtime. For more information about these commands, see the FactoryTalk View SE Help.

## Allowing multiple running copies

Use this option with graphic displays of the type Overlay or On Top, to allow multiple copies of the display to run simultaneously.

For example, you can open two copies of the same display in different parts of a FactoryTalk View SE Client window, by issuing the following commands:

```
Display PID /Q1
```

```
Display PID /Q2
```

PID is the name of a graphic display: the */Q1* parameter positions the first copy of the display at the top-right corner of the window; the */Q2* parameter positions the second copy at the top-left corner of the window.

If the **Display** command specifies a display that does not allow multiple copies, and the display is already running, it is brought to the foreground only.

If multiple copies of a display (or several separate displays) are running, use the **SetFocus** command to bring any hidden displays to the foreground. A display of the type On Top is always at the front, regardless of which display has focus.

For more information about commands used to navigate between displays at runtime, see the FactoryTalk View Site Edition Help.



**Tip:** To run multiple copies of displays without using the **Allow Multiple Running Copies** option, use the **Display** command with different parameter files, for each copy of the display you want to run. For more information, see [Replacing tag placeholders using parameter files on page 276](#).

## Caching displays

Placing a graphic display in the cache makes the display appear more quickly at runtime, every subsequent time it is opened, because it does not have to be read from disk.

You can have up to 40 graphic displays in the cache. Place large or complex displays in the cache, to minimize the use of system resources.

The Always Updating option keeps a cached display up to date, even when the display is not visible. Use this option to update trend data continuously, or to run VBA code in the background.

The Always Updating option affects the behavior of the display's startup and shutdown commands. For more information, see [Specifying startup and shutdown commands on page 283](#).



**Tip:** Cached displays consume memory. Always updating a cached display can add to communications overhead, as data is retrieved for tags whose values might not be needed.

## Setting up the title bar and other display attributes

You can set up a graphic display to have a title bar, so that operators can grab and move the display at runtime.

If you provide a title for the display, the title will appear in the title bar instead of the component name. You can also insert variables into the title bar text. For more information, see [Creating embedded variables on page 365](#).

The following illustration shows the window style options that are available when you select the **Title Bar** checkbox. To add or remove an item from the title bar, select or clear the appropriate checkbox.



## Scaling the graphic display

You can set up a graphic display to resize automatically when it starts running, so it fits the size of the FactoryTalk View SE Client main window. When resized, the display is panned or scaled, depending on which option you selected.

FactoryTalk View graphics are resolution independent. This means that no matter what resolution you use to create graphic displays, they are resized automatically to suit the monitor on which they are shown at runtime.

## Showing the last known values of HMI tags

You can set up a graphic display to show the last known value for each HMI tag in the display, until current values arrive from the programmable controller.

In many applications, selecting this option will help to show graphics more quickly.

At runtime, if a display is not set up to show the last known value of HMI tags, objects with values that have not yet been updated appear in outline form. The outline indicates that data is not current, or is in error.

Graphic objects might appear in outline form the first time a display starts, if the HMI tags used in the display are not initialized.

Once the tags are initialized and the values arrive, the objects appear in their normal form. However, if data is unavailable or in error, the objects will remain in outline form.



**Tip:** The last acquired value can be shown for HMI tags only. For other kinds of tags, the last value cannot be retained.

## Setting the update rate for tags

The maximum tag update rate is the fastest rate at which data servers can send tag value changes to the graphic display.

Set the update rate as fast as, or faster than, the rate at which the values of tags used in the expressions change, unless it is desirable to miss changes in tag values.

The default update rate is one second. This means that data servers will not send tag value updates faster than once every second.

## Viewing screen statistics of a display

In the graphic editor, you can view the statistic information of a display, including:

- Total tags referenced

This includes all tags referenced in expressions, commands and embedded variables of the current display.

Duplicated references of the same tag in an expression are counted as one tag.

Each global object parameter used in an object is counted as one tag.

- Unique tags referenced

This includes all unique tags used in expressions, commands and embedded variables of the current display.

- Number of Expressions used

- Number of Alarm functions used

This includes all FactoryTalk alarm and event functions.

- Number of Global Objects referenced

- Number of ActiveX Controls used

- Number of Security functions used

This includes all references to the security functions **CurrentUserHasGroup** and

**CurrentComputerHasGroup**.

- Number of referenced images

This includes all references to images that are listed under the Images folder of the FactoryTalk View Explorer window.

- Number of imported images

This includes all references to imported bitmap or icon files for the button appearance.

## Specifying the size of the graphic display

You can use the current size of the display or specify the width and height in pixels. Click **Use Current Size** to specify that the size of the display in the edit window will be its *natural* size at runtime. Click **Specify Size in Pixels** to specify a width and height in pixels for the display. For more information, refer to the FactoryTalk View Site Edition Help.

## Preventing scroll bars on the main window

If a graphic display is larger than the FactoryTalk View SE Client main window, scroll bars will appear on the window when the display is started.

The scroll bars will remain on the window, even if subsequent displays are small enough that scroll bars are not needed.

To minimize the need for scroll bars, the FactoryTalk View SE Client tries to position all graphic displays within its main window, unless this placement is explicitly overridden by the /X and /Y parameters of the **Display** command.

To prevent scroll bars, create all graphic displays smaller than the working area in the FactoryTalk View SE Client main window.

Keep in mind that the size of the working area depends on several factors: the monitor's display resolution, whether the display has a title bar, the size of the client window, and whether the client shows the Diagnostics List.

To ensure that the FactoryTalk View SE Client main window never has scroll bars, for large displays, select the option, **Size to Main Window at Runtime**.



**Tip:** For smaller graphic displays not meant to fill the client window, do not use **Size to Main Window at Runtime**, to avoid scroll bars. Instead, position smaller displays so that they are completely visible within the client window.

## Specifying the display position

Instead of setting up a graphic display to use its current size and position, you can specify an exact size and position for the display at runtime.

If you do this, you can override the display settings by using the size and position parameters with the **Display** command. For details, see the FactoryTalk View Site Edition Help.

## Securing graphic display

To restrict access to a graphic display, select a security code. For more information about securing displays, see [Setting up security on page 77](#).

## Background color

Select the background color for the graphic display from the color palette.

## Using gradient style

If you want the background to have a gradient look, select the Use gradient style checkbox. When selected, clicking the background color shows the gradient fill effects dialog box.

## Tracking screens for navigation

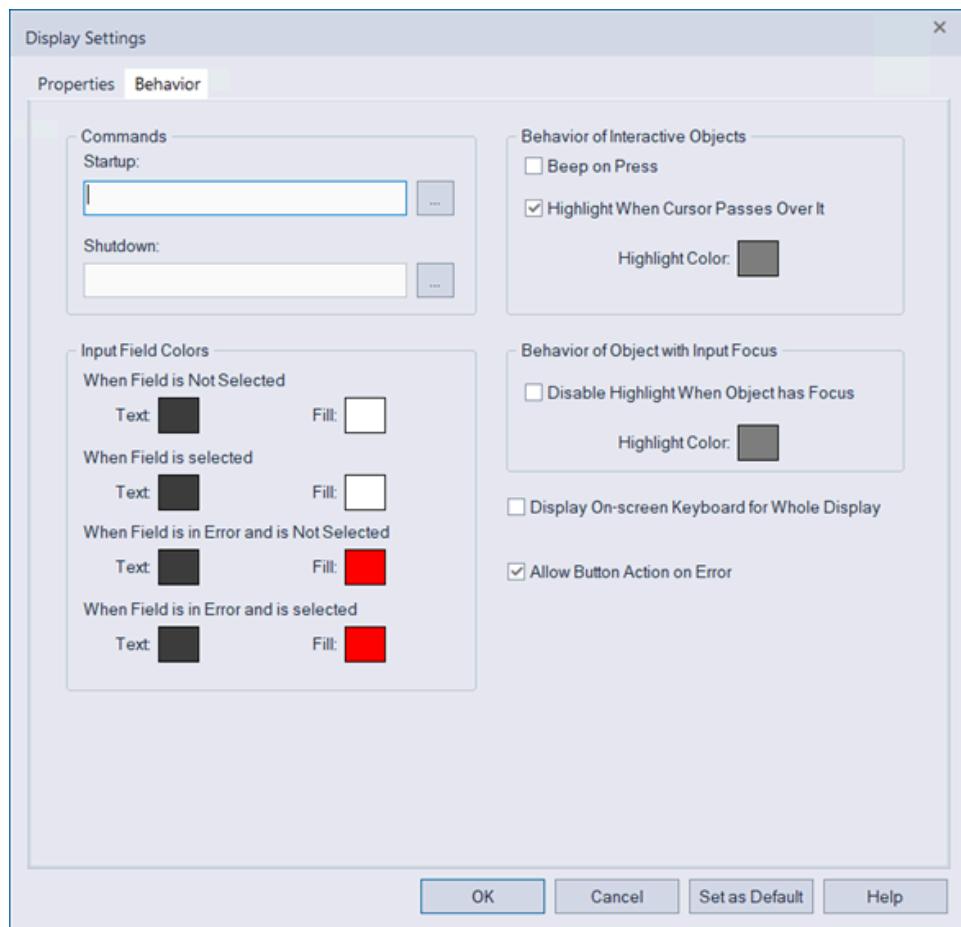
You can include displays in the navigation history by selecting the **Track Screen for Navigation** checkbox. When you select this checkbox, a text box is enabled where you can input an operator-friendly screen name to be shown within the navigation history. If you do not specify a name, the actual display name is used in the navigation history.

If you do not select this checkbox, the display will not be added to the navigation history.

## Setting up the runtime behavior of a graphic display

In the Behavior tab of the Display Settings dialog box, you can specify:

- Startup and shutdown commands.
- Colors for input fields.
- The behavior of interactive objects, such as push buttons.
- The behavior of objects with input focus.
- Whether an on-screen keyboard is available, for systems that do not have a hardware keyboard attached to them at runtime.
- Whether any actions defined for a Button push button object will continue to work even if the defined animation for the object is in an error state.



## Specifying startup and shutdown commands

To run commands when the graphic display starts or stops running, specify display startup and shutdown commands, or macros.

If you use the **Always Updating** option with the **Cache After Displaying** option, the startup command is run when the display is loaded into the cache.

The shutdown command is run only when the cache is flushed. This happens when:

- The **FlushCache** command is run.
- A user logs off the FactoryTalk View SE Client.
- The FactoryTalk View SE Client is closed.

For details about the **FlushCache** command, see the FactoryTalk View Site Edition Help.



**Tip:** The startup command runs before the display opens, so commands such as UploadAll or RecipeRestore will not work as startup commands or in a startup macro.

The shutdown command runs after the display closes, so commands such as DownloadAll or RecipeSave will not work as shutdown commands or in a shutdown macro.

## Specifying colors for input objects

At runtime, an operator can use input objects to read values from or write values to programmable controllers and other devices.

You can select the color an input object will have when selected, when not selected, and when there is an operator input error.

Text color is the color of the text in the input object. Fill color is the background color of the input object. To select a color, click the colored box to open a palette, and then click a color to apply.

Choose colors for input objects that will stand out from the background color of the display. Also choose different colors for selected objects, so that an operator can tell when an object is selected, and when it is highlighted (see the next sections).

## Specifying the behavior of interactive objects

Interactive objects are objects that an operator can interact with at runtime, using a mouse, keyboard, or touch screen. A button with a press action is an example of an interactive object.

You can specify whether interactive objects in a graphic display beep when pressed, whether they have a highlight box when the mouse passes over them, and what color the highlight will be.

## Specifying the behavior of objects with input focus

Objects with input focus are ready to accept keyboard or mouse input.

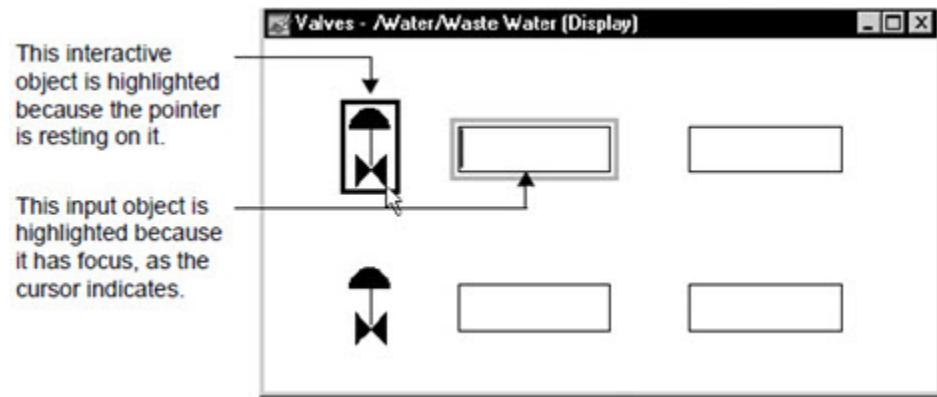
You can specify whether objects with input focus in a graphic display have a highlight box, and what color the highlight will be.

## Using both types of highlight in the same display

You can choose to highlight only interactive objects, only objects with input focus, or both types of object.

An interactive object is highlighted when the mouse passes over it. An object that can take input focus is highlighted when it has focus. It is possible for one object to show both types of highlight at the same time.

The following illustration shows what the two types of highlight might look like in a graphic display.



When selecting highlight colors, be sure to pick colors that stand out from the background of the display.

## Showing the on-screen keyboard

To show a keyboard at runtime, to interact with numeric input, string input, and recipe objects in the graphic display, select the checkbox, **Display On-screen Keyboard**.

The on-screen keyboard is typically used with systems that do not have a hardware keyboard, such as systems that only use touch screens.

When the operator clicks or touches the selected object, or presses Enter on a hardware keyboard, the on-screen keyboard is presented.

The keyboard for string input and recipe fields accepts alphanumeric characters. The keypad for numeric input fields accepts numeric characters only. For more information, see [Parts of the on-screen keyboard on page 297](#).

## Setting up displays to open more quickly

If graphic displays are stored in the display cache, they open more quickly. You can place up to 40 graphic displays in the cache.

To cache a display, use one of these methods:

- Use the **Cache After Displaying** option in the Display Settings dialog box. For details, click **Help** in the dialog box.
- Use the **[cache]** parameter with the **Display** command.

The [cache] parameter loads a graphic display into the cache without displaying it. The parameter has two options:

- /Z loads the graphic display into the cache without opening it. When the display is called up subsequently, it opens quickly.
- /ZA loads the display into the cache and continually updates the values in the display, even when the display is not visible.

For details about the **Display** command, see the FactoryTalk View Site Edition Help.

## Removing displays from the cache

To remove all graphic displays from the display cache, run the **FlushCache** command, or close the FactoryTalk View SE Client.

To remove a specific graphic display from the cache, run **FlushCache <file>**, where **<file>** is the name of the display you want to remove. For details, see the FactoryTalk View Site Edition Help.

If a graphic display uses the **Always Updating** option with the **Cache After Displaying** option, the display's shutdown command is run when the **FlushCache** command is run, or when you close the application.

## Changing the default display settings

When you begin using the Graphics editor to create graphic displays for an application, the values already specified in the Display Settings dialog box are used as default values.

For example, the Display Type property is set to Replace, the Title Bar property is selected (meaning the display will have a title bar), and the Background Color property is set to white.

All the new displays you create will have these initial default settings.

## Docking displays to the FactoryTalk View SE Client window

At runtime, graphic displays can be docked to an edge of the FactoryTalk View SE Client window, so that operators can gain access to the displays at all times.

For example, you might consider docking:

- Navigational menus that provide operators with ways to move among displays in an application.
- Headers or banners that provide specific information to the operator, such as the current user's name and area, or information about alarms.
- Control panels that contain standard buttons for special purposes, such as changing users, closing open windows, or sending information to a maintenance team.

## Command parameters for docking displays

To dock graphic displays, run the **Display** command.

### Example: Using the **Display** command to dock a display after the FactoryTalk View SE Client starts up

To dock a display named **Menu** to the top edge of the SE Client window after the client starts up, create a button in the initial display that uses this command as its press action:

Display Menu /DT

When the operator presses the button, the display will attach to the top of the window.

## About the appearance and behavior of docked displays

When you open a graphic display using one of the docking parameters, certain display settings are ignored. For example, regardless of what is specified in the Display Settings dialog box, a docked display:

- Has no borders or scroll bars.
- Has no title bar (and therefore no minimize, maximize, or close buttons).
- Cannot be undocked or moved.
- Cannot be resized, independent of the FactoryTalk View SE Client window.



**Tip:** Changing the width of the client window changes the width of displays docked at the top or bottom of the window. Changing the height of the window changes the height of displays docked at the left or right.

If a graphic display is set up to pan or scale, to allow multiple running copies, or to cache after displaying, these settings will still apply when the display is docked.

## Docking multiple displays in the same position

You can also run more than one docked display in the same position, if the additional displays are the Overlay or On Top type.

For example, if you use /DT to dock a menu to the top of the client window, and then use /DT to open another display, as long as the second display is of the type Overlay or On Top type, it will run directly beneath the first.

If you dock a Replace type of display, it will close any other displays running in the same position. For more about display type, see [Specifying the display type on page 277](#).

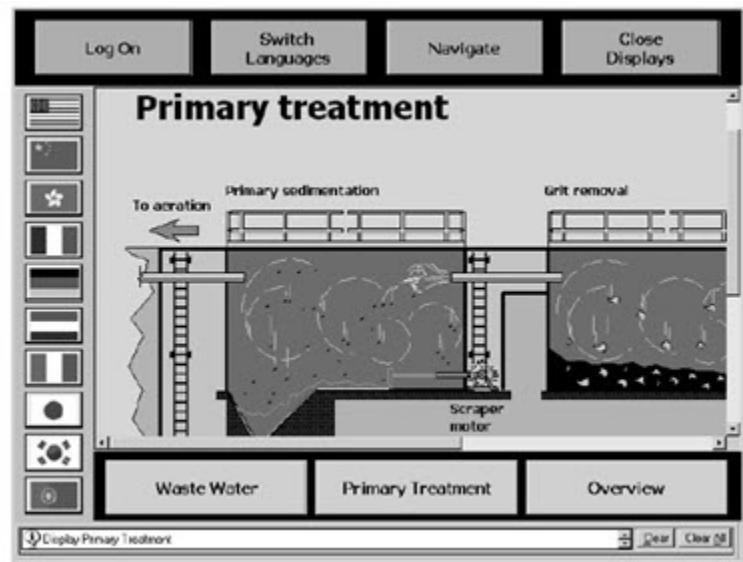
## Running other displays in the available client area

The size and position of docked graphic displays in the FactoryTalk View SE Client window determines where in the window other graphic displays can run.

For example, if you dock a display named Menu at the top of the SE Client window, and then run a display named Line1, it will run in the area not occupied by the docked display.

If Line1 is larger than the available client area, scroll bars will be provided for gaining access to the hidden portion of the display. If you opened Line1 first, and then ran the Menu display with a docking parameter, Line1 would be repositioned in the client window, to accommodate the Menu display.

In the following illustration, the FactoryTalk View SE Client window contains three docked menus (top, left, and bottom), and a graphic display named Primary Treatment is running in the available client area.



## Closing docked displays

Use the **Abort** command to close docked displays individually, or to close all docked displays, and any other displays running in the FactoryTalk View SE Client window.

### Example: Closing an individual docked display

To provide the operator with a way to close a docked display named Menu, create a button in the display that uses this command as its press action:

```
Abort Overview
```

When the operator presses the button, the Menu display will close.

### Example: Closing all docked displays

To provide the operator with a way to close all docked displays, along with any other displays running in the FactoryTalk View SE Client window, create a client key that uses this command as its press action:

```
Abort * /D
```

When the operator presses the key, all running displays will close.



**Tip:** The **Abort** command's **/D** parameter, which closes any docked displays, is valid only when used with **Abort \***.

# Creating graphic objects

This chapter describes the graphic objects available in FactoryTalk View Site Edition.

For information about using the Graphics editor, and creating and setting up graphic displays, see [Creating graphic displays on page 259](#).

## Types of graphic objects

In FactoryTalk View Studio, in the Graphics editor, you can create the following types of graphic objects:

- **Drawing** objects are geometric and freehand objects, images, panels, and text. Drawing objects are the only FactoryTalk View graphic objects that do not use data connections. For more information, see [Creating the different types of drawing objects on page 289](#).
- **Push button** objects are typically used to start or stop processes or actions. For more information, see [Different types of push buttons on page 293](#).
- **Numeric and string** objects are used for entering or displaying numbers or text. For more information, see [Different types of data display and input objects on page 294](#).
- **Indicator** objects show the status of a process or operation by showing colors, captions, images, or options to indicate different states. For more information, see [Different types of indicators on page 298](#).
- **Gauge and graph** objects show a range of values, and relationships between variables. For more information, see [Different types of gauges and graphs on page 299](#).
- **Key** objects represent keys on a keyboard, and are for use with touch screens. For more information, see [Using key objects to simulate keyboard functions on page 301](#).
- **Trending** objects, including Trend and TrendPro, provides operators with a way to track plant activity as it is happening. For information about creating trend objects, see [Setting up trends on page 394](#).
- **Advanced** objects provide ways for operators to monitor, analyze, and interact with application data at runtime. For more information, see [Different types of advanced objects on page 302](#).
- **Alarm and Event objects** such as the alarm and event summary and the alarm status explorer, provide ways for FactoryTalk View SE users to monitor and control FactoryTalk Alarms and Events. For information about the Alarm and Event objects, see [Setting up FactoryTalk alarms on page 177](#).
- **OLE objects** such as spreadsheets, charts, or text, are produced by other Windows applications. The OLE objects that are available depend on the software installed on the system. For more information, see [Working with OLE objects on page 307](#).
- **ActiveX objects** such as gauges, or sliders, can be created using a tool like Visual Basic or purchased from a third-party vendor, and then set up to initiate an action in FactoryTalk View. For more information, see [Working with ActiveX object on page 307](#) and [Animating ActiveX objects on page 330](#).
- **.NET Control** provides built-in Windows Forms and WPF (Windows Presentation Foundation) controls and supports self-created controls or third-party libraries for more custom controls. For more information, see [Working with .NET Control objects on page 308](#).
- **Symbol Factory** is a library of graphic objects that can be imported into a Graphic Display. For more information, see [Working with Symbol Factory on page 309](#).
- **Import** objects such as graphics from the file.

## About global objects

You can create a FactoryTalk View global object to link the appearance and behavior of one graphic object to multiple copies of the object in the same application.

Any graphic object that you can create in FactoryTalk View can be a global object, except for ActiveX controls and OLE objects. For more information, see [Creating global objects on page 304](#).

For information about global object displays, see [Creating graphic displays on page 259](#).

## Creating different types of drawing objects

To draw simple objects in the Graphics editor, select drawing tools from the **Objects** menu or toolbar.

After selecting a drawing tool, draw the object by dragging the tool, or by clicking end points. You can draw rectangles, ellipses, and arcs only by dragging; you can draw polylines and polygons only by clicking end points.

## Adding images into graphic displays

To add an image into a graphic display, you can:

- Place an image to be used in several displays.
- Import images into graphic displays.
- Paste an image copied from another application.

## Placing images in graphic displays

Use the **Image** graphic object to place a bitmap, JPEG, PNG, or SVG image in a graphic display.

If the image is monochrome (1 bit per pixel, 2 colors), you can change its color, background color, and transparency, and you can make it blink at runtime.

For monochrome images, the background color becomes transparent when you change the Image back style to Transparent.

For color images, any area of the image that is black becomes transparent when you change the Image back style to Transparent.

## Adding images to an application

Images you add to an application can be inserted in graphic displays over and over again. If you modify the original image, all displays that use the image are updated automatically.

To add an image you want to use in several displays, add it to the application using the Image tool, or add the image to the Images folder in FactoryTalk View Studio, in the Explorer window.

When you add an image using the Image Browser, the image appears automatically in the Explorer window, and vice versa.

You can add an image to an application when you place it in a graphic display or before you place it in the display. For more information, click **Help** in the Image Browser.

## Importing images into graphic displays

FactoryTalk View can read BMP, PNG, and JPEG (.jpg, .jpeg, .jpe, .jfif, .jfif) files in their original format, without requiring conversion.

When you import an image, FactoryTalk View converts .gif, .tif, .pcx, and .jpg images to the Windows .bmp format. Vector images in .dxf and .wmf formats are converted to FactoryTalk View graphic objects.

You can import the following types of files into graphic displays from third-party applications using **Import** in the Toolbox.

File extension	Type of file
.wmf	Windows meta files
.bmp, .gif, .tif, .pcx	Bitmap files
.png	Portable network graphics
.jpg	JPEG files
.dxf	AutoCAD files *

\*FactoryTalk View does not import AutoCAD 13 or later .dxf files. To import a graphic file created in AutoCAD 13 or later, export the graphic file as a .wmf file in AutoCAD, and then open the .wmf file in FactoryTalk View.

Before importing the files, set up the computer's display properties to display more than 256 colors. This will ensure that the imported objects are the same colors as the original objects.

When you import objects, you can convert them to FactoryTalk View objects. This offers the following advantages:

- Graphic display files are smaller.
- Objects are groups of objects rather than a single object. This means you can modify the individual parts of the object, including attaching animation to individual parts.

Symbol Factory, when used with the Image Browser, allows you to locate a high-quality graphic and not only import it into your Graphic Display, but save it in the images folder for additional usages as a graphic object.

## Using bitmaps in a FactoryTalk View application

Bitmaps consume Windows resources. When using bitmaps, consider the following guidelines.

### Use device-dependent bitmaps

Device-dependent bitmaps (.bmp files) show faster than device-independent bitmaps (.dib files) because the FactoryTalk View Graphics editor is optimized for device-dependent bitmaps.

Also, you can modify device-dependent bitmaps in place, using the Microsoft Paint program.

### Avoid unnecessary color depth

Create bitmaps in the lowest color depth possible. The more colors that you use, the more memory that is consumed:

- 16-color bitmaps consume 4 bits per pixel ( $\frac{1}{2}$  byte per pixel)
- 256-color bitmaps consume 8 bits per pixel (1 byte per pixel)
- 24-bit bitmaps consume 24 bits per pixel (3 bytes per pixel)

If possible, use 16-color bitmaps. To change a higher-resolution bitmap to 16 colors, open the bitmap in the Microsoft Paint program and save the bitmap as a 16-color bitmap.

## Match palettes in 256-color systems

In a 256-color system, if bitmaps use two different color palettes, Windows must recalculate and redraw all bitmaps when window focus changes.

Redrawing bitmaps causes delays and can make a scanned image or photograph sparkle or appear as a negative.

To match palettes, use a bitmap-oriented graphical tool that lets you match palettes.

Palette matching is an issue only for 256-color video adapters. 24-bit color systems do not match palettes and 16-color systems dither colors (that is, alternate pixels of different colors to approximate another color).

## Do not use the Scale option

The Scale option in the Display Settings dialog box causes the contents of a graphic display to change size to suit the size of the graphic display's window.

To speed up the display of a graphic containing bitmaps, choose Pan rather than Scale. Bitmaps take longer to draw when they are scaled to a size different from their original size.

An OLE object can be a bitmap or a bitmap wrapped in a metafile. These types of OLE objects will also draw more slowly when scaled.

## Avoid large bitmaps

Graphic displays that contain large bitmaps consume memory and can be very slow to show because of the delay in loading the bitmaps from disk.

## Avoid many bitmaps

Whenever possible, create graphic objects using the FactoryTalk View drawing objects.

You can also change a bitmap to a FactoryTalk View object by converting the bitmap to wallpaper, tracing over the bitmap with FactoryTalk View drawing objects, and then deleting the bitmap.

## When to use a bitmap

Bitmaps generally make graphic displays slower. However, objects with large amounts of detail, such as subtle shading, might draw more quickly if converted to a bitmap because bitmaps take the same amount of time to draw regardless of their complexity.

## Pasting images into graphic displays

You can also add an image to a graphic display by copying the image to the Windows clipboard from another application, and then pasting it into the display.

To paste an image from the clipboard, open the display in the Graphics editor, and then, from the Edit menu, select **Paste Special**.

## Techniques for working with objects that use data

Graphic objects that an operator can interact with at runtime, using a mouse, keyboard, or touch screen, are called interactive objects.

Examples of interactive objects are a numeric input object, a button that has a press action, and a rectangle object with touch animation.

## Using the keyboard to select objects that can take focus

If a mouse or touch screen is not connected to the computer at runtime, the operator can use the keys on a keyboard or keypad to select (give focus to) these objects:

- Push button objects
- Numeric and string input objects
- Control list selectors, piloted control list selectors, and display list selectors
- Trends
- FactoryTalk Alarm and Event objects, including alarm and event summaries, banners, status explorers, and the log viewer.
- Third-party ActiveX input objects

Use the following keys to move to and select a different object:

To do this	Press
Move from the upper left to the lower right	Tab
Move from the lower right to the upper left	Shift + Tab
Move left, right, up, or down	Ctrl + arrow key

When a graphic display opens, of the objects that can have input focus, the object that has a tab index of 1 is selected initially.



**Tip:** If all of the objects in a display are continuously updating input objects, none of the objects will receive initial focus. Instead, the operator must select an input object to give it focus. For more information, see [Updating tag values continuously on page 295](#).

## Removing objects from the tab sequence

By default, you can use the Tab and arrow keys to navigate to objects that can have input focus in a graphic display.

You can also turn off key navigation for objects that take input focus, except for push buttons, recipe objects and input objects.

When an object's key navigation is turned off, an operator can still select the object using a mouse or touch screen, if one is available.

### Turning off key navigation:

To turn off key navigation, use one of these methods:

- For display list and control list selectors, open the object's Properties dialog box, click the General tab, and then clear the checkbox, **Key navigation**.
- For ActiveX objects, trends, and Alarm and Event objects, open the object's Properties dialog box, click the Common tab, and then clear the checkbox, **Key navigation**.
- In the Property Panel for the object, click the Properties tab, and then set the KeyNavigation property to **False**.

## Different types of push buttons

Push buttons start or stop processes or actions and change tag values. You can create the following types of push buttons, depending on the needs of the application:

- **Button** objects can change tag values or run FactoryTalk View commands. You can also attach different types of animation to a button, except for fill, rotation, and touch animation.
- **Momentary** push buttons change a tag to one value when the button is pressed, and another value when the button is released.

You can specify values for the press and release actions, or the momentary push button can have these states:

- **Normally open** means that the released state of the button is off: when the button is pressed, the tag's value is set to 1; when the button is released, the tag's value is set to zero. Pressing the button completes the circuit.
- **Normally closed** means that the released state of the button is on: when the button is pressed, the tag's value is set to 0; when the button is released, the tag's value is set to 1. Pressing the button breaks the circuit.

For example, momentary push buttons work like the **Pulse** button on a food processor, but with an indicator light to signal whether the motor is running or stopped. The machine is on only while the button is held down. When the button is released, the machine turns off.

- **Maintained** push buttons switch between two values.

When first pressed, the maintained push button changes a tag to one value. When pressed and released a second time, the button changes the tag to another value.

This type of button is useful for changing a setting within a machine or process, but not for starting or stopping the machine or process. For example, use the maintained push button for changing modes, such as Auto to Manual, or Metric to Imperial.

- **Latched** push buttons lock in the on position, and must be unlocked by another button or process to return to the off position. This type of button is useful for starting a machine or process.

The Handshake tag must be set back to zero before the operator can press the latched push button again.

- **Multistate** push buttons let an operator cycle through multiple options consecutively, using a single button that displays the current state of a process or operation. Each state can be represented by a different color, caption, or image.

Each time the operator presses the button, the tag changes to the value for the next state. If the button is in its last state when the operator presses it, the button returns to its first state.

- **Interlocked** push buttons work in groups, and share the same tag. The buttons work together like the preset station selector buttons on a car radio: Pressing one button cancels another. Although the interlocked push buttons work as a group, you add them to the display one at a time.

When the operator presses one of the interlocked push buttons, the button's Value tag changes to one value.

When the operator presses a different interlocked button, the button's Value tag changes to another value. All the buttons share the same Value tag.

- **Ramp** push buttons increase or decrease the value of a tag by either an integer or floating point value. You can use two ramp buttons together to create an increase/decrease control, for example for the speed of a motor.

Each time the operator presses the button, the tag changes to the next highest or next lowest value, depending on how you set up the button. Some data servers do not impose a minimum or maximum limit on the value of a tag. Set the upper and lower limits of the ramp push button to limit the range of values the operator can send to the data server.

- **Navigation** push buttons can be configured to let the operator navigate between the opened graphic displays.

A navigation push button can perform three types of actions:

- DisplayPreviousScreen opens the previous graphic display in the navigation history when the button is clicked.
- DisplayNextScreen opens the next graphic display in the navigation history when the button is clicked.
- DisplayNavigationHistory shows a list of previously opened screens when the button is clicked.

For details about setting up a push button, click **Help** in the object's Properties dialog box.

---

**NOTE:** Never use push buttons for emergency stops. Always hard-wire emergency stop buttons.

---

## About the runtime error state

Momentary, maintained, latched, and multistate push buttons display an error state when:

- The Value tag is unassigned.
- The display opens if the Value tag does not match one of the specified state values.
- The Indicator tag or expression does not match one of the specified state values.

For details about push button states, click **Help** in the Properties dialog boxes.

## Different types of data display and input objects

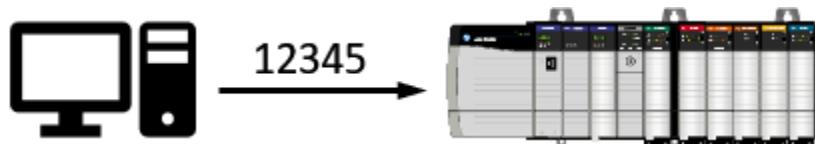
Operators can use data display and input objects to view or change tag data. You can create the following types of data display or input objects:

- **Numeric and string display objects** to let operators view tag or expression values in a graphic display at runtime. Use display objects to show numeric values or text that the operator does not need to modify.
- **Numeric and string input objects** to let operators read or write tag values at runtime. Use input objects to download or upload tag data. You can set up whether to show a confirmation or electronic signature dialog box before writing a value to a tag.

An input object can also be set up to display a tag's current value, eliminating the need to create separate input and display objects. For details, see [Updating tag values continuously on page 295](#).

## Using input objects to retrieve and send data

An operator can use numeric and string input objects to retrieve data from and send data to the value table, so it can be used by FactoryTalk Live Data servers (such as FactoryTalk Linx and Tag Alarm and Event Server) and OPC data servers.



To download data, the operator types a number into the input object, and then presses **Page down** or **Enter**.



To upload data to the input object, the operator presses **Page up**.

If a graphic display contains several input objects, the operator can select a single object and then read or write a value, or read or write values for all objects in the display at once.

## Tag-related limits

A graphic display can contain up to a total of 3000 connections, whether they originate from the expressions or the tags.

- Each expression associated with an object is counted as one connection regardless of the number of tags in the expression.
  - Each animation of an object (except the Touch animation) is counted as one connection.
  - Each connection in the Connections property of an object that is linked to a tag is counted as one connection.
  - Each pen configured in a Trend object is counted as one connection.
  - Duplicate references of the same expression or tag connection are counted as the additional connections.
- For example, one display can contain up to 3000 numeric inputs, even if all numeric input objects refer to the same tag.



**Tip:** Tags associated with embedded variables do not count towards the limit.

## Updating tag values continuously

You can set up a numeric or string input object to show a tag's value and update the value continuously. At runtime, the appearance of a continuously updating input object changes to reflect which mode it is in.

When the input object is showing a value from the programmable controller or server, it has a dotted border:



This is called display mode.

When a value is entered in the input object, but the value is yet to be downloaded, it has a solid border:



This is called pending write mode.

When the input object is ready to receive input, it has a solid border surrounded by a highlight box:



This is called input mode (or input focus).

At runtime, when a graphic display containing input objects is opened, the first non-updating object in the index sequence receives focus.

If all the input objects in the display are set to update continuously, none of the objects will receive initial focus. Instead, the operator must select an object to give it focus.

After selecting an input object, the operator can upload a value or restore a recipe into the object. If an upload fails because of a communication error, the input object appears in outline form.

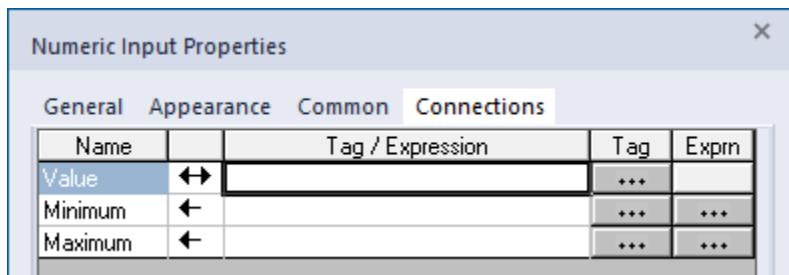
To return to the input object to display mode, press **ESC**.

## Validating operator input

To validate operator input, you can specify minimum and maximum values for numeric input objects.

At runtime, if the operator attempts to download a value outside the valid range, the input object changes color, an error message is logged to the Diagnostics List, and the download is canceled.

To define the minimum and maximum, in the Connections tab of the Numeric Input Properties dialog box, specify a tag, expression, or number for each value.



In the Displays Settings dialog box, you can select the colors the input objects will become when operator input errors occur. For more information, see [Setting up the runtime behavior of a graphic display on page 282](#).

Alternatively, you can use the graphic Display object's event BeforeInputFieldDownload, to validate input objects before downloads occur. For information about VBA and the Display object, see the FactoryTalk View Site Edition Help.

## Shortcut keys for retrieving and sending data

An operator can use the following keys to retrieve data from and send data to the value table. You can re-assign these actions to other keys, or assign them to button objects.

- **PgDn** downloads the contents of all input objects that are in pending write mode (in the active graphic display) to the value table.
- **Ctrl+PgDn** downloads the contents of the selected input object to the value table.
- **Enter** downloads the contents of the selected input object to the value table

If the graphic display is set up to use the on-screen keyboard, pressing Enter brings up the on-screen keyboard.



**Tip:** To prevent an operator from using Enter to download values, or to open the on-screen keyboard, use the /E parameter with the **Display** command to open the graphic display. This turns off the Enter key.

- **PgUp** uploads all values from the value table and then displays them in the input objects in the graphic display.
- **Ctrl+PgUp** uploads a value from the value table for the selected input object.
- **Tab** moves among input objects.

## FactoryTalk View commands for retrieving and sending data

An operator can use the following commands to retrieve data from and send data to the value table:

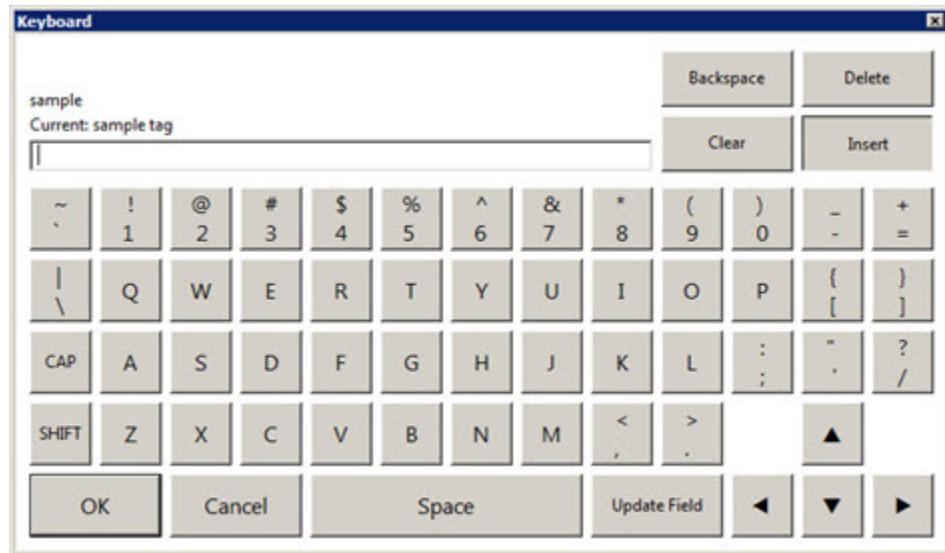
- **Download** downloads the contents of the selected input object to the value table.
- **DownloadAll** downloads the contents of all input objects that are in pending write mode to the value table.
- **Upload** uploads a value from the value table and displays it in the selected input object.
- **UploadAll** uploads all the values from the value table and displays them in the input objects.

To let an operator use these commands, assign them to FactoryTalk View button objects in the graphic display. For more information, see the FactoryTalk View Site Edition Help.

## Parts of the on-screen keyboard

You can set up graphic displays so that the operator can use an on-screen keyboard for input entry in recipe, string, or numeric input objects. You can configure the on-screen keyboard at the client level, display level, or input entry level.

The keyboard that opens for string input and recipe objects (shown in the following illustration) accepts alphanumeric characters. The keypad that opens for numeric input objects accepts numeric characters only.



Characters typed in the on-screen keyboard are transferred to the selected input object when the operator presses Update Field or Download in the on-screen keyboard, or presses Enter on a hardware keyboard.

To do this	Press
Close the on-screen keyboard and store the new value in the input object for a subsequent download.	Update Field
For numeric and string input objects, close the on-screen keyboard and download the value or text the operator typed. For recipe objects, close the on-screen keyboard, open the Recipe dialog box, and insert the text the operator typed into the Recipe File box.	Download
Close the on-screen keyboard and discard the operator's changes.	Cancel

For more information about the on-screen keyboard, see [Setting up the runtime behavior of a graphic display on page 282](#).

## Different types of indicators

Indicators show the status of processes or operations by showing different colors, captions, images, or options to reflect different states.

You can create the following types of indicators, depending on the needs of the application:

- **Multistate** indicators show the current state of a process or operation by showing a different color, caption, or image to reflect different states.  
At runtime, the multistate indicator shows the state whose value matches the Indicator tag or expression's value.
- **Symbol** indicators show a symbol that changes to match the state of a process or operation. This lets the operator see the status of a process or operation at a glance.

At runtime, the symbol indicator shows the state whose value matches the Indicator tag or expression's value.

- **List** indicators show a list of states for a process or operation, and highlight the current state. Each state is represented by a caption in the list.

This lets the operator view the current state and also see the other possible states. For sequential processes, the list can inform the operator about what happens next.

At runtime, the list indicator highlights the state whose value matches the Indicator tag or expression's value.

For details about setting up an indicator object, click **Help** in the object's Properties dialog box.

## Setting up states for indicators

Indicators change their appearance based on their states. Most indicators have several states, and an error state. The error state occurs when the indicator is receiving invalid data.



**Tip:** The List indicator has no error state. If the value of the Indicator tag does not match an available state, none of the states is highlighted.

## Using the Least Significant bit to trigger states

The Least Significant Bit (LSB) trigger type changes the indicator's state based on the position of the lowest bit in the tag's value. Any higher bit positions are ignored.

Each bit position in the tag's value corresponds to one of the indicator's states: For example, position 1 triggers state 1.

The number of states a tag's value can trigger depends on the tag's data type. For example, a tag of type long integer can be used to change up to 33 of an indicator's states (32 bit positions plus zero).

You can use a programmable controller to set these bits individually.

## Setting up connections for indicators

To connect with a device such as a programmable controller, indicators use a tag or expression called the Indicator tag or expression.

The Indicator tag is similar to a pilot light on a hard-wired panel. The tag or expression changes the indicator's appearance for each of its states, providing visual feedback to the operator. For example, the Indicator tag can show that a process is running or stopped.

Set up the Indicator tag or expression in the Connections tab of the Properties dialog box.

## Different types of gauges and graphs

Gauges and graphs provide graphical representations of numeric values.

You can create the following types of gauges and graphs, depending on the needs of the application:

- **Bar graphs** represent a numeric value by filling and emptying as the value rises and falls.

At runtime, the graph shows the value of the tag or expression in relation to the graph's minimum and maximum values.

- **Gauges** represent a numeric value using a needle on a dial.  
At runtime, the gauge indicates the value of the tag or expression in relation to the gauge's minimum and maximum values.
- **Scales** represent a scale for a bar graph.  
To place values on the scale as a legend, use text objects. Because the scale does not change at runtime, you don't need to connect it to a tag.

## Using gauges to show limits

Gauges show numeric values in dial format. Gauges are useful for showing a value in relation to its lower and upper limits.

For example, a temperature gauge shows the current temperature in relation to its minimum and maximum extremes. By looking at the position of the needle on the gauge (pointing left, up, or right), the operator can tell at a glance whether the temperature is nearer its lower or upper limit, or nearer the middle.

Gauges are used instead of numeric displays when it's important for the operator to recognize an abnormal condition immediately, either from far away (when the scale on the gauge isn't visible), or before it is possible to make an exact reading on the gauge.

## Using graphs to compare values

Graphs display numeric values in bar graph format.

Graphs are useful for comparing multiple values, or for representing the fill levels of tanks that suit readings on a vertical scale.

Use graphs instead of numeric displays when it's important for an operator to analyze the relationships between numeric values.

It's easier for the operator to see that one graph is at a lower level than the other, or that one graph's fill is green and the other's is red, than it is to subtract one numeric value from another.

For example, one bar graph can show the required level of a tank of ingredients for a recipe, and a second bar graph can show the actual level of the tank.

The first graph changes to represent the required level for each recipe, and the second graph changes as the actual level in the tank rises or drops.

## Showing limits using scales with bar graphs

Unlike gauges, bar graphs do not have integrated scales.

Instead, you can show values on a bar graph using a scale and text. Scales consist of major ticks, represented by long lines, and minor ticks, represented by short lines. To indicate the values of major or minor ticks, use text objects.

## Using key objects to simulate keyboard functions

In the FactoryTalk View Graphics editor, the term key can mean any of the following:

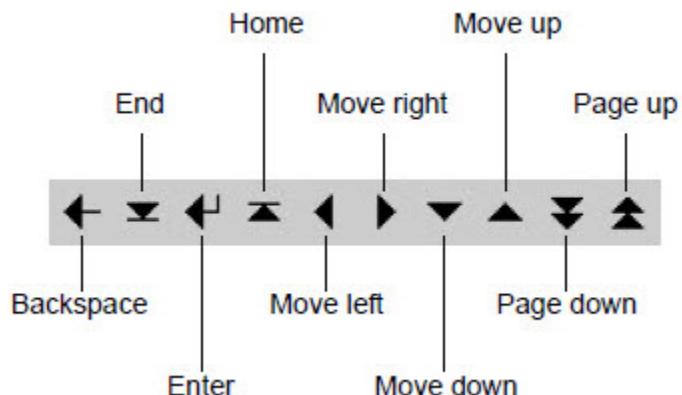
- **Key animation** links a graphic object or display to a keyboard key or mouse button, so that an operator can perform an action by pressing the key or mouse button. For more information, see [Associating objects and displays with keys on page 326](#).
  - The **on-screen keyboard** lets touch screen users type numbers or text in input objects without the need for a hardware keyboard.
- To make the on-screen keyboard available at runtime, in the Behavior tab of the Display Settings dialog box, select the checkbox, **Display on-screen keyboard**. For details, see [Setting up the runtime behavior of a graphic display on page 282](#).
- **Keys** are graphic objects that you place on a display to simulate the functions of keyboard keys. This type of key can only be used with control list selectors, piloted control list selectors, display list selectors, and Trend objects.

## Creating different types of key objects

For control list selectors, display list selectors, piloted control list selectors, or trends, you can create the following types of keys, depending on the needs of the application:

- **Backspace** moves the cursor back to the highlighted item.
- **End** moves to the bottom item of the page that is currently shown. For trends, pressing End resumes trend scaling and moves to the current or latest data in the trend.
- **Enter** selects the item that is currently highlighted.
- **Home** moves to the top item of the page that is currently shown. For trends, pressing Home pauses the trend and moves to the earliest data in the trend.
- **Move left** pauses the trend and scrolls to the left.
- **Move right** pauses the trend and scrolls to the right.
- **Move down** moves down one item in the list. For trends, pressing Move down scrolls down to display lower values on the vertical scale.
- **Move up** moves up one item in the list. For trends, pressing Move up scrolls up to display higher values on the vertical scale.
- **Page down** moves down one page in the list.
- **Page up** moves up one page in the list.

To create the different key objects, use the tools shown in this illustration:



In the Properties dialog box, the various features of the object are organized in tabs. For details about options in each tab, click **Help**.

## Setting up the auto-repeat function for selected keys

All of the key objects have similar features, except that the move left, move right, move up, move down, page up and page down keys can be set up to auto-repeat.

In the Timing tab, set up whether the key press repeats automatically when the operator presses and holds down the key. You can also set up the rate at which the key repeats. For details, click **Help**.

## Using the same set of keys with different graphic objects

If a graphic display contains more than one control list selector, piloted control list selector, display list selector, or trend, you can use one set of keys to control all the objects. You don't have to set up a separate set of keys for each object.

To set up all the objects to use the same set of keys, create the key objects, and then, in the General tab of each key's Properties dialog box, select **Object with Focus** for the Send press to option. For more information, click **Help** in the Properties dialog box.

## Different types of advanced objects

The FactoryTalk View advanced objects have various functions. You can create the following types of advanced objects, depending on the needs of the application:

- **Arrow** objects track tag or expression values in a vertical or horizontal path.
- **Control list Selector** objects let an operator select from a list of states, and write values associated with those states.
- **Piloted Control List Selector** objects have the same function as control list selectors. However, they can be controlled directly by an operator, or remotely by a device such as a programmable controller.
- **Display List Selector** objects let an operator navigate between graphic displays in a list.
- **Tag Label** objects show information about a tag's properties, such as the description, engineering units, and minimum and maximum values.
- **Local Message Display** objects provide an operator with information, or prompts about what to do next.
- **Time and Date** shows the current time and date.
- **Web Browser** objects provide the system with the ability to show webpages from an HMI application.
- **Diagnostics Viewer** objects show FactoryTalk diagnostics messages on the display.
- **Signature** objects provide a way to secure operations by verifying the identity of the operator before an action can occur.
- **Legacy Recipe** objects let an operator restore data from a recipe file to input objects in a graphic display.
- **Data Grid** objects show the historical data stored in a SQL Server database, FactoryTalk View SE data logs, and FactoryTalk Historian.
- **Radar Chart** objects are a graph used to display multivariate data in a radial format.

For details about setting up any of the advanced objects, click **Help** in the object's Properties dialog box.

## Creating piloted control list selectors

Piloted control list selectors let an operator (or a remote device) scroll through and select from a list of states for a process. A highlight in the list indicates the current state.

### Piloted control list features

Piloted control list selectors have features that control list selectors do not:

- States can be selected by an operator, by a remote network device, or by both an operator and a network device.  
Control list selectors can be controlled by the operator or a network device, but not both.
- Individual states can be turned off, to prevent them from being selected.  
You cannot turn off states in a control list selector.
- The state values of all the items that are visible in the list can be written to the Visible States tag when the list scrolls. The Visible States tag must be a data server tag that supports arrays. You cannot use an HMI tag.  
Control list selectors have no Visible States tag.
- The state value of the item at the top of the list can be written to the Top Position tag when the list scrolls.  
Control list selectors have no Top Position tag.

## Selecting states in a piloted control list selector

A piloted control list selector can show several states at the same time, but only one state can be selected at a time.

You can set up a piloted control list selector to be operator controlled or remote controlled, by assigning tags or expressions in the Connections tab.

You can also set up individual states to be operator controlled, remote controlled, operator and remote controlled, or none of those. If none is selected, the state is turned off and cannot be selected.

If the piloted control list selector is set up to be operator controlled, and the operator selects a remote controlled state, or one that is turned off, a hollow cursor is shown, as shown on the right in the following illustration.



The filled cursor means the operator can select this state.



The hollow cursor means the operator cannot select this state.

## Creating the Alarm and Event objects

The FactoryTalk Alarm and Event objects provide runtime displays of FactoryTalk device-based and tag-based alarm data, sources, and activity.

You can create the following Alarm and Event objects:

- The **Alarm and Event Summary** shows detailed alarm information from selected Rockwell Automation Device Servers (FactoryTalk Linx) and Tag Alarm and Event Servers in a FactoryTalk View SE application. You can use the Alarm and Event Summary to view, acknowledge, suppress, and disable the alarms on display.
- The **Alarm and Event Banner** shows up to five of the highest priority, most severe, and most recent alarms in the system. You can use the Alarm and Event Banner to monitor and acknowledge the most serious FactoryTalk alarms in the system.
- The **Alarm Status Explorer** shows the areas in an application that contain alarm sources, and the tags being monitored for alarms. You can use the Alarm Status Explorer to view alarm sources, suppress or unsuppress, and enable or disable alarms.
- The **Alarm and Event Log Viewer** shows information stored in the logs generated by the FactoryTalk Alarms and Events Historian. You can use the Alarm and Event Log Viewer to view, sort, filter, and export historical alarm information.
- The **Automatic Diagnostic Event Summary** shows diagnostic events enabled by the Automatic Diagnostics feature of Studio 5000 Logix Designer. You can use the Automatic Diagnostic Event Summary to suppress, unsuppress, filter, and sort diagnostic events at runtime.

For information about creating the FactoryTalk Alarm and Event objects, and using them to monitor and interact with alarms at runtime, see [Setting up FactoryTalk alarms on page 177](#).

## Creating global objects

You create global objects in displays in the Global Objects folder, in the same way you create graphic objects in displays in the Displays folder.

All of the objects and groups of objects created in a global object display are global objects.

For more information about creating global object displays, see [Creating graphic displays on page 259](#).

## Creating reference objects

After creating a global object (or group of objects), you can copy it for use in standard displays throughout an application.

When you copy a global object onto a standard display in the Displays folder, the copy is called a reference object. The original global object in the Global Objects folder becomes its base object.

Reference objects have special properties that link them to the original, base objects. When you modify properties of a base object, the changes are copied to all reference objects linked to the base object.

You can produce an unlimited number of reference objects from a single base object. However, reference objects can only link to a base object within the same application or, in a network distributed application, within the same HMI server.

## Using placeholders to specify values

A tag placeholder is the crosshatch character (#) followed by a number from 1 to 500. You can use tag placeholders to specify a value for a graphic object, instead of (or as part of) a tag name, expression command, or embedded variable.

Tag placeholders used in standard graphic objects let you reuse one graphic display for several purposes, by substituting tag values for the placeholders at runtime.

You can also set up tag placeholders for a global object, to assign different runtime values to the reference objects linked to the global base object.

To do this, define the tag placeholder at the global base object, and then assign the runtime values to the linked reference objects.

For more information, see [Setting up tag placeholders for global objects on page 306](#).

## Adding global objects into the graphic libraries

Displays in the **Libraries** folder cannot contain reference objects. If you add:

- A global object display into the Libraries folder, the global objects are converted into standard graphic objects.
- A standard graphic display containing reference objects into the Libraries folder, the reference objects are deleted when the new library is saved.

## Setting up the link properties of reference objects

A reference object has properties that link it to its base object. You can set these properties when you copy or drag a global object into a standard graphic display, or you can modify them later.

The Link properties have default values when the reference object is created. You can modify these properties in the Property Panel for the object, or in the Global Object Defaults dialog box.

The values of the Link properties determine whether aspects of the reference object's appearance and behavior are defined by the base object, or by the reference object itself. For details, see the property descriptions that follow.

### To open the Global Object Defaults dialog box:

- Right-click the reference object that you want to modify, and then select **Global Object Defaults**.  
For details about options in the Global Object Defaults dialog box, click **Help**.



**Tip:** If you change the properties of a linked reference object, you might lose the original settings.

For example, if you change a reference object's LinkConnections property from **False** to **True**, any connections previously assigned for the reference object are deleted. To restore the original connections, from the **Edit** menu, select **Undo**.

## Setting up tag placeholders for global objects

Using tag placeholders in a global object lets you assign unique values to the different reference objects linked to the global base object.

To set this up, first you define the tag placeholder for the global base object, and then you assign runtime values to the linked reference objects.

When displays containing the reference objects are run, the tag placeholder is replaced with the value specified for each reference object.

For grouped reference objects, the value specified for the top-level object is applied to every object comprising the group that uses the same tag placeholder.

If you do not assign a value to a tag placeholder at design time, you can provide the value at runtime using either a parameter file, or the /T parameter with the **Display** command. For information about using these methods, see [Replacing tag placeholders using parameter files on page 276](#) and [Replacing tag placeholders using parameter lists on page 277](#).

## Defining tag placeholders for use in reference objects

You can use tag placeholders instead of (or as part of) a tag name, expression, command, or embedded variable associated with a global object or group of objects.

These are the steps involved in setting up tag placeholders for global objects:

1. Define one or more tag placeholders at the global base object.
2. Create one or more reference objects linked to the base object.  
For information about creating reference objects, see [Creating global objects on page 304](#).
3. For each reference object, assign a value to the tag placeholder (or placeholders) defined at the base object.  
You can assign a unique value to the same tag placeholder, for each different reference object.

You can define multiple tag placeholders for a single global object or group.

For grouped objects, the placeholder defined for the top-level object applies to all objects in the group.

## Modifying global objects that use tag placeholders

Modifying a global object that uses tag placeholders has various effects, depending on the type of modification.

### Modifying grouped base objects:

Tag placeholders defined for a grouped base object are deleted, if the grouped object is:

- Ungrouped.
- Regrouped with another object or group, to form a new grouped object.
- Copied and then pasted into another grouped object.

### Modifying tag placeholders defined for the base object:

Changes made to tag placeholder definitions at the global base object are copied to linked reference objects, the next time the displays containing the reference objects are opened or refreshed.

For example, when you add, delete, or change the description of a tag placeholder for a base global object, the same change will take effect for any linked reference objects, the next time the reference objects are updated.

## Working with OLE objects

OLE is an acronym for Object Linking and Embedding.

OLE objects are objects that were created in other Windows applications and linked to or embedded in a FactoryTalk View graphic display.

The main difference between linking and embedding is where data is stored:

- **Linked OLE objects** are stored in the source file. The graphic display stores the location of the source file and displays a representation of the linked data. This means that if you modify the source file, the object in the graphic display also changes.  
When you double-click a linked OLE object to modify it, the object's source file becomes active in the application that it is associated with.
- **Embedded OLE objects** become part of the graphic display. If you modify the source file, the object in the display is not affected.  
When you double-click an embedded OLE object to modify it, a border indicates that the object is ready to be modified, and a toolbar for the application associated with the object opens in the Graphic editor. This is called in-place editing.

## Converting OLE objects

Convert an OLE object to a FactoryTalk View graphic object, if you want to:

- Change an embedded, vector-based object into a polygon, to make it easier to manipulate.
- Reduce the size of a graphic display file.

Embedding an object in a graphic display increases the size of the graphic file, because the embedded object includes information about its source application. This lets you double-click the object and modify it using the source application.

If you convert the object, you can no longer modify it using its source application. However, you can use the editing tools in the Graphics editor to work with the object.

## Working with ActiveX objects

ActiveX objects are software components supplied by products independent from FactoryTalk View, such as Microsoft Office, Visual Basic, and many other third-party applications.

For example, you can embed the Microsoft Forms ActiveX objects in a FactoryTalk View graphic display.

By embedding an ActiveX object in a FactoryTalk View display, and then assigning properties or specifying handlers for the object's events, the object can interact with FactoryTalk View.

Data is passed between an ActiveX object and FactoryTalk View using tags.

If you attach a tag to an ActiveX object's Value property, the object's behavior changes as the tag's value changes.



**Tip:** FactoryTalk View supports the use of windowed ActiveX controls only. You cannot use windowless ActiveX controls in a FactoryTalk View graphic display.

### Exchanging data with tags:

To pass data between an ActiveX object and FactoryTalk View, the ActiveX object must supply information in the same format as the tags with which the data is exchanged.

For example, if the ActiveX object is connected to an HMI tag, the ActiveX object must provide information that is compatible with the analog, digital, or string format of the tag.

FactoryTalk View does not support pointer parameters in an ActiveX object.

## Attributes of ActiveX objects

An ActiveX object has three types of attributes that expose its features:

- **Properties** are named characteristics and values of an object, such as shape, color, position, or number.
  - **Events** are actions triggered by the ActiveX object in response to an external action on the object, such as a mouse click.
- In FactoryTalk View you can use events to change the value of a tag, or to run a FactoryTalk View command or macro. When the event occurs, the command or macro runs. To use the ActiveX object to change a tag's value, associate the tag with one of the object's event parameters.
- **Methods** are functions implemented in the ActiveX object that make it possible for external actions to change the object's appearance, behavior, or properties.

A call to a method might be made in response to events from other controls and could trigger other events.

You can use the FactoryTalk View **Invoke** command as the external event that calls a method. For more information about the **Invoke** command, see the FactoryTalk View Site Edition Help.

## Deploying ActiveX components automatically at runtime

FactoryTalk View SE clients can automatically install the correct versions of ActiveX controls used in graphic displays.

To deploy the ActiveX controls automatically, you need to create .cab files for the ActiveX controls, and then put the .cab files in the same folder where you installed FactoryTalk View SE.

To do this, run the program CABARC.exe, on the computer hosting the HMI server. CABARC.exe is in the folder **C:\Users\Public\Public Documents\RSView Enterprise\ActiveX Control Setup**.

For information about creating .cab files, see the text file CreatingCabFiles.txt in the ActiveX Control Setup folder. The text file contains examples for creating .cab files, and information about the naming conventions that must be used.

If you open a graphic display, containing ActiveX objects that are not installed, the graphic display runs, but a shaded rectangle appears in place of the ActiveX object.

## Working with .NET Control objects

A .NET control is a component that was developed by using the Microsoft .NET Framework. You can configure and run .NET controls on FactoryTalk View SE displays. The actual functions are contained within a .NET Control object, which provides the configuration dialog boxes.

The .NET Control object provides built-in Windows Forms and WPF (Windows Presentation Foundation) controls. You can also import self-created controls or third-party libraries for more custom controls. If the custom controls that are developed with .NET Framework 3.5 or earlier do not function properly, install .NET Framework 3.5. For more information, see the Microsoft website <https://docs.microsoft.com/en-us/dotnet/framework/migration-guide/version-compatibility>.

**IMPORTANT:** The runtime behavior of .NET controls depends on the vendor's implementation. Test objects thoroughly before you run an application to ensure they behave as expected.

## Attributes of .NET Control objects

A .NET control includes:

- **Properties:** Configure the .NET control or obtain information from it.
- **Methods:** Invoke methods for the .NET control to perform a specific action.
- **Events:** Trigger an action that runs every time the .NET control performs a specified event.

Unlike FactoryTalk View SE client object model, VBA scripting for .NET Control objects have their own rules and standards. The VBA sample code shows at the bottom of the dialog box.

## Working with Symbol Factory

Symbol Factory is a library of over 5,000 graphics for industrial automation, including pumps, pipes, valves, tanks, mixers, motors, ducts, electrical symbols, flowmeters, material handling, sensors, PLCs, transmitters, and ISA symbols.

In Symbol Factory, you can switch between two types of symbols:

- Basic symbols  
Includes a format of Windows Metafile (WMF) or bitmap image file (BMP).
- Enhanced symbols  
Includes a format of scalable vector graphics (SVG) and contains gradients.

Note that any Symbol Factory object, including Bitmap, DIB, PNG, SVG and metafile can be pasted into the Image Browser where it will be stored as a bitmap. An enhanced version of the library in Symbol Factory is available to provide the SVG format and gradients.

### To open the Symbol Factory:

Symbol Factory opens into a new window when launched, there are four ways to open Symbol Factory:

- From the **Explorer**, click the Symbol Factory object.
- When the **Objects menu** is available, click the Symbol Factory dropdown menu item.
- In the Toolbox, click the Symbol Factory icon .
- In the Image Browser, click the **Launch Library** button.

## Using electronic signatures to authorize runtime changes

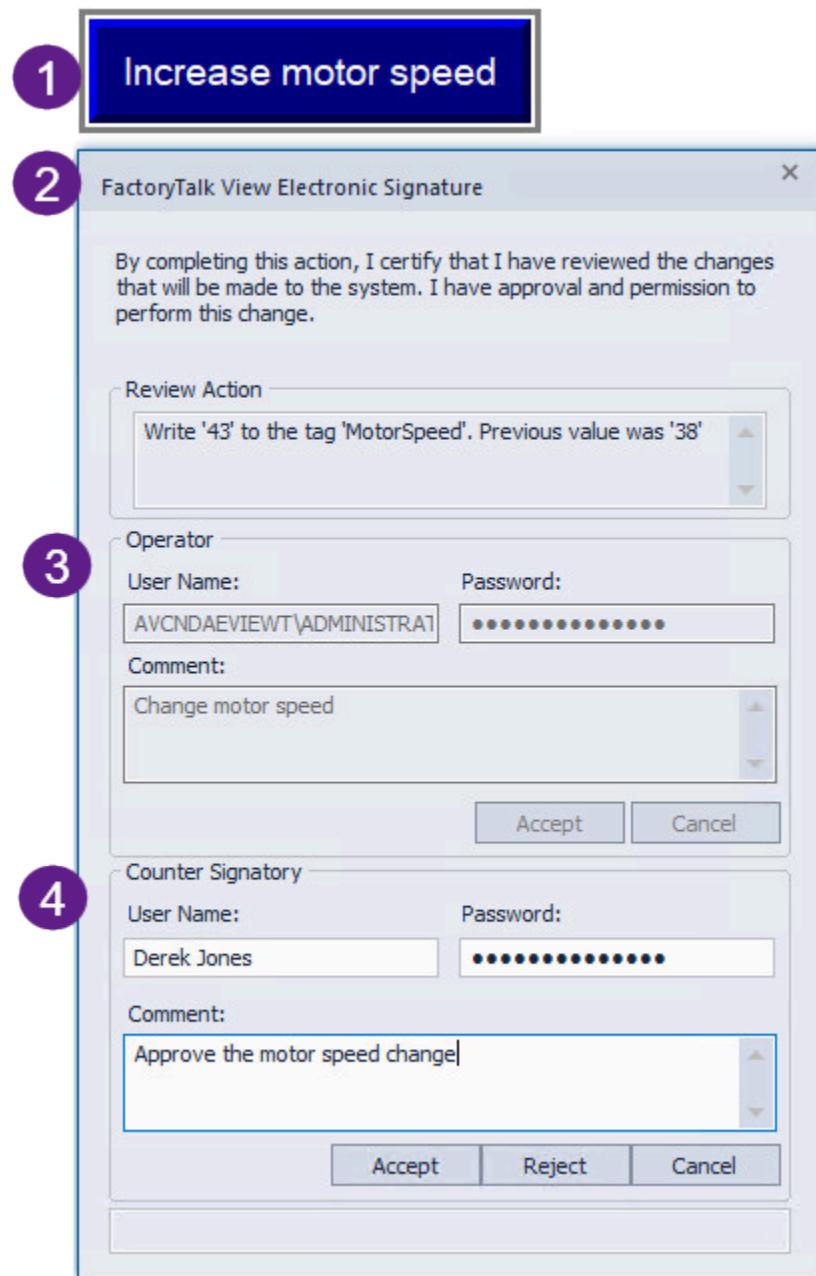
FactoryTalk View Site Edition's electronic signature feature provides a way to secure operations by verifying the identity of the operator before an action can occur.

Objects that provide the electronic signature feature are:

- **Push Button**
  - Button
  - Maintained Button
  - Latched Button
  - Multistate Button
  - Interlocked Button
  - Ramp Button
- **Numeric and String**
  - Numeric Input
  - String Input
- **Advanced**
  - Control List Selector
  - Piloted Control List Selector
  - Signature

Electronic signature and authorization control for operations verify the identity of the operator who is logged in and authorize the operation that is going to be performed. This feature, along with others such as FactoryTalk View SE's *headless server* architecture, which allows users to be logged in only at the client, help critical operations meet the security standards required for 21 CFR Part 11 compliance.

The following illustration shows a ramp button configured with an electronic signature. Both reauthentication and counter signature are enabled for authorizing a motor speed change from 38 to 43 after the button is pressed.



Action or behavior	Description
1	The operator clicks the button.
2	The electronic signature dialog box opens.
3	The operator needs to enter the password and comment. The password requirement is enabled by the reauthentication option.
4	After the operator enters the password and comment, the authentication by another person is required, including user name, password, and comment. This requirement is enabled by the counter signature option.



**Tip:** If the operator is a Windows-linked user that does not belong to the current Windows domain, include the domain name with the user name, using the syntax **domain\user name**. For more information about Windows-linked users, see [Setting up security on page 77](#).

## Tracking changes using FactoryTalk Diagnostics

The operations initiated by the signature button are logged to FactoryTalk Diagnostics, and can be viewed in the Diagnostics List or Diagnostics Viewer.

The logged information includes user name, old value, new value, operator's comments, and the name of the person approving the change.

For information about logging system activity, see [Logging system activity on page 253](#).

# Animating graphic objects

This chapter describes:

- What FactoryTalk View animation is.
- Setting up animation for FactoryTalk View graphic objects.
- Creating effects using the different types of animation.
- Using index numbers to navigate to objects in a display.
- Associating objects and displays with keys.
- Animating ActiveX objects.

## About animation in FactoryTalk View

In FactoryTalk View Site Edition, you can animate objects in graphic displays, to enhance how you visualize automation controls and processes in an application.

The following are some of the ways that you can animate objects in FactoryTalk View:

- Link a FactoryTalk View graphic object to a tag or an expression, so that the object's appearance changes when the value of the tag or expression changes.  
For more information, see [Setting up animation for FactoryTalk View graphic objects on page 313](#).
- Link a graphic object or display to a key or mouse button, to let an operator perform an action by pressing the key or mouse button.  
For more information, see [Associating objects and displays with keys on page 326](#).
- Animate an ActiveX object:
  - By assigning tags to the ActiveX object's properties, so that the tags' values change when the properties change or, in some cases, the properties change when tag values change.
  - By running commands based on the ActiveX object's events.
  - Mapping tags to the ActiveX object's event parameters.
  - Interacting with the ActiveX object using VBA code.

For more information about animating ActiveX objects, see [Animating ActiveX objects on page 330](#).

## Setting up animation for FactoryTalk View graphic objects

To animate FactoryTalk View objects in a graphic display, create the objects first, and then apply animation to them.

In the Graphics editor, use the Animation dialog box to apply animation to the selected object or group of objects.

For details about options in each Animation tab, click **Help**.



**Tip:** To set up a key animation, open the Object Key or the Display Key dialog box from the Graphics editor's **Edit** menu. For more information about key animation, see [Associating objects and displays with keys on page 326](#).

While working in a graphic display, you can leave the Animation dialog box open and move it around. This means you only have to open the dialog box once, to apply animation to different objects in the display.

It also means you can use the Object Smart Path to set the range of motion for an object, rather than needing to know and then to type the number of pixels you want the object to move. For more information, see [Defining the range of motion on page 315](#).

### To open the Animation dialog box:

1. Select the object or object group that you want to apply the animation to.
2. Select **View menu > Animation**, and then select the type of animation to apply.

Types of animation an object cannot have will not be available on the **Animation** menu.

## Linking animation to tag values

Most types of animation can be linked to the value of a tag. To link an animation to a tag value, you have to specify a tag name or a tag placeholder.

### To set up an animation using a tag name:

In the Animation dialog box, in the Expression box, type a tag name, or click **Tag** to open the Tag Browser, to search for and select a tag.

You can use a tag that already exists in a device or in the HMI server's tag database, or you can use the name of a new tag that you plan to create later.

## Using tag placeholders

A tag placeholder is the crosshatch character (#) followed by a number from 1 to 500. A tag placeholder can also contain wildcard characters and folder names.

Tag placeholders used in graphic displays are replaced with tag names when the display is run. To apply an animation using a placeholder instead of a tag name, type the placeholder in the Expression box.

For more information about tag placeholders, see [Using placeholders to specify tag values on page 275](#).

## Linking animation to expressions

Many types of animation require an expression. You can build expressions using tag values, constants, mathematical equations, security functions, and if-then-else logic.

A tag name or tag placeholder can be included as part of an expression, or standalone as the entire expression.

For more information about expressions, see [Creating expressions on page 344](#).

### To set up an animation using an expression:

In the Animation dialog box, in the Expression box, type the expression, or click **Expression** to open the Expression editor, to build an expression.

## Linking animation to actions

Some types of animation, such as touch animation, require you to specify an action.

The action can be a FactoryTalk View command or macro. For example, to use touch animation to open a graphic display, specify the **Display** command as the press or release action.

For a complete list of FactoryTalk View commands and their syntax, see the FactoryTalk View Site Edition Help. For more information about macros, see [Adding logic and control on page 423](#).

## Determining start and end points for a range of motion

Many types of animation require an expression range, using minimum and maximum values to determine the start and end points for the object's range of motion.

When setting up an animation, to calculate minimum and maximum values, select one of the following options in the Animation dialog box:

- **Use tag's min and max property values** uses values taken from the minimum and maximum range defined for the first HMI tag used in the expression. This is the default option.
- **Use constant** uses numeric constants. Type the minimum and maximum values in the boxes.
- **Read from tags** uses the values of the tags that you specify. If you expect the minimum and maximum values to change, use the **Read from tags** option.



**Tip:** With the Read from tags option, the minimum and maximum values are read-only the first time animation is started for the object. For example, the values are read when the display containing the object is run.

## Defining the range of motion

To define a range of motion when setting up an animation for an object, you can:

- Move the object in the Graphics editor. This feature is called Object Smart Path. For an example, see "About Object Smart Path," next.
- Type values in the At minimum and At maximum boxes.

Motion is defined in pixels, degrees, or percent, depending on the type of animation.



**Tip:** Visibility, color, and touch animation do not use a range of motion, because these types of animation represent a change of state, not a range of values.

### About Object Smart Path

Object Smart Path lets you position objects in a display interactively, rather than by entering numbers in the dialog box. This means that you can specify an object's position by dragging it into place rather than trying to determine what the position is in pixels.

### Example: To use Object Smart Path to define a range of motion for a horizontal slider

1. In the Graphics editor, create a slider using a line and a rectangle, or copy a slider object from the Sliders library in the Graphics Libraries.
2. Right-click the slider, select **Animation > Horizontal Slider**.

3. In the Tag box, type a tag name. To find and select a tag, click the **Browse** button.
4. In the display, drag the slider to the position indicating the lowest number in the range.
5. In the Animation dialog box, set this position by selecting the **At minimum** checkbox.
6. In the display, drag the slider to the position indicating the highest number in the range.
7. In the Animation dialog box, set this position by selecting the **At maximum** checkbox.
8. To save the settings, click **Apply**.

When you finish setting up the animation, the slider returns to its starting position.

## Applying animation to object groups

You can apply animation to a FactoryTalk View graphic object, to a group of objects, and to the individual objects that comprise a group.

To animate a group of objects in a graphic display, group the objects first, and then apply animation to the group.

At runtime, animation set up for object groups generally takes precedence over animation set up for objects within the group, except for visibility, fill, and color animation.

Visibility and color animation applied to individual objects takes precedence over any group animation. Fill animation applied to a group is added to any animation applied to individual objects.

To apply animation to individual objects in a group, use the group edit feature. For details about this feature, see [Modifying grouped objects on page 270](#).



**Tip:** Combining different types of animation can produce unexpected results. Be sure to test the group animation, to ensure that the animation achieves the effect you want.

## Testing animation

You can test the objects in a graphic display quickly, by switching to the test display mode in the Graphics editor.

When you are finished testing, to continue working on the display, switch back to edit display mode . To switch between test and edit modes, use the buttons on the toolbar or the items on the View menu.

Testing a graphic display in FactoryTalk View Studio is not the same as running the display in the FactoryTalk View SE Client.

Test display mode does not change the appearance or position of the display, as set up in the Display Settings dialog box, and you cannot switch between open displays.

In addition, some FactoryTalk View commands are ignored when run in test display mode. For a complete list of these commands, see [FactoryTalk View commands on page 442](#).



**Tip:** If objects in a graphic display are connected to tags in devices, to fully test the display you must set up communication with the devices or data servers.

Before you deploy an application, it is recommended that you test it in the FactoryTalk View SE Client, to verify that everything works as intended.

## Viewing the animation applied to objects

To view the types of animation applied to an object or object group, use one of these methods:

- Right-click the object, and then select **Animation**. On the **Animation** menu, there is a check mark beside each type of animation the object has.
- Select the object, and then open the Animation dialog box. In the dialog box tabs, there is a check mark beside each type of animation the object has.
- In the Object Explorer, click **Settings**, select the animation types you want to find, and then click **OK**. You can also use the Object Explorer to check which objects have animation linked to a particular tag.
- To view animation applied to individual objects within a group, use the group edit feature. For details, see [Modifying grouped objects on page 270](#).

## Copying or duplicating objects with animation

When you copy or duplicate objects that have animation, the animation is copied or duplicated with the object.

If you copy or duplicate an object group, the copy can be ungrouped, just like the original object group.

You can also copy an animation without copying the original object, and then paste the animation onto another object. If the original object has more than one type of animation, all the types will be copied and pasted.

### To copy and paste only the animation:

1. Right-click the object that has the animation you want to copy, and select **Copy animation**.
2. Right-click the object that you want to paste the animation onto, and select **Paste animation**.

## About global objects and animation

You can create a FactoryTalk View global object to link the appearance and behavior of one graphic object to multiple copies of the object in the same application.

When you copy a global object into a standard graphic display in the Displays folder, the copy is called a reference object. The reference object has special properties that link it to its base object, which is the original object in the global object display.

For example, the value of the LinkAnimation property determines whether the reference object will use the animation set up for itself, or the animation set up for the base object.

For more information about the LinkAnimation property, see [Setting up the link properties of reference objects on page 305](#).

## Creating effects using the different types of animation

FactoryTalk View SE provides a range of animation types, to let you create different effects for the objects you use in graphic displays.

You can apply one type of animation to an object or group of objects, or you can combine animation types to achieve a particular effect.

For example, applying both width and height animation to a drawing object, such as a rectangle, gives it the appearance of moving into or out of the display, as it shrinks and expands.

The following sections provide an overview of the animation types available in the Animation dialog box. For details about each type, click **Help** in the dialog box.

For information about how to apply animation, see [Setting up animation for FactoryTalk View graphic objects on page 313](#).

## Showing and hiding objects

Use a visibility animation to make an object visible or invisible, based on a tag value or the result of a logical expression. If an object is invisible, no other animation applied to the object is evaluated, to prevent unnecessary processing.

You can apply visibility animation to all FactoryTalk View graphic objects. Visibility animation overrides an object's visibility property, if there is a conflict.

The following types of objects can *only* have visibility animation:

- Images and panels
- Push buttons (except the button type)
- Indicators
- Gauges and graphs
- Key objects
- Advanced objects (except arrows and labels)

For details about Visibility options in the Animation dialog box, click **Help**.

## Changing the color of an object

Use color animation to change an object's color, based on a tag value or the result of an expression. You can specify up to 16 color changes (A to P) for any object.

Choosing between Solid, Gradient, Original, and Shaded Fill Styles allows for more animation options for the blinking object.

**Solid** - Allows you to change the colors, and they will have a solid look to them.

**Gradient** - Allows you to change the colors, and they will have a gradient look to them. If you select Gradient as the Fill Style, the Fill color box changes to gradient. Click it to bring up the gradient fill dialog box to configure the gradient fill.

**Original** - Retains the colors and the shading of the object to what was originally imported. Both the line and fill color can be altered from the original using Blink.

**Shaded** - Uses a tight dot pattern to soften or give a shaded appearance. This fill style takes effect only on a group object that is composed of individual objects with dark to light colors, such as Symbol Factory objects.



#### Tip:

Some types of graphics are made up of many elements. An individual element can be selected and have color animation applied just to it.

The following example shows the effects of Shaded and Gradient fill styles.



For each color change, specify the value or threshold at which the color changes, and specify the colors you want the object to change to. At runtime, when the value reaches or crosses the threshold, the color of the object changes.



**Tip:** The color animation does not affect string input, numeric input, or Legacy Recipe objects. The color for these objects is defined in the Display Settings dialog box.

## Keeping the original color

Using the **Original Fill Style** allows for more variations on animating an object. Instead of a specific color for the object's non-blink state, the object's original color can be used. If **Original** is used as a **Fill Style** on any state, the color palette boxes will not allow a color change. By selecting the **Blink** option, a different color can be used for both the **Line** and the **Fill** color or individually.

## Controlling color shading

Using the **Shaded Fill Style** along with the **Blink** option allows control of the shading aspect of a display object by changing combinations of **Line** and **Fill** and different colors.

For details about Color options in the Animation dialog box, click **Help**.

## Example 1: Creating text that blinks

This example describes how to create text that blinks between two colors. In this example, the expression is simply a constant value that matches the value for the selected threshold.

1. Using the **Text** tool, create a text object.
2. In the Text Properties dialog box, type some text, and then click **OK**.
3. Select the text object, open the Animation dialog box, and then click the **Color** tab.
4. In the Expression box, type 0. (Zero is the default value for threshold A.)
5. In the list, click threshold A. Leave the value in the Value box as 0.
6. Next to the Line color, select the **Blink** checkbox. (Only line color affects text.)

7. Select the Blink color checkbox next to the **Blink** checkbox to open the color palette, and then select a color.
8. Click **Apply**, and then close the Animation dialog box.

Use the Test Display tool to test the animation in this example. The text will blink between the colors you selected.

## Example 2: Creating an object that changes color

This example describes how to create a rectangle object that changes color as the object's fill level increases. This example uses a tag called *Hopper\FlourLevel*, which has a range of 1 to 100. To test the animation, you must create the tag.

### To create the object:

1. Using the Rectangle tool, create a rectangle.
2. Right-click the rectangle, and then select **Properties**. In the Back Style list, click **Solid**.
3. Using the Foreground Color and Background Color palettes, make the rectangle gray.
4. With the rectangle selected, open the Animation dialog box, and then click the **Fill** tab.

### To apply the fill animation to the object:

1. In the Expression box, type *Hopper\FlourLevel* (the tag that monitors the fill level).
2. For Fill Direction, click **Up**.

### To apply color animation to the object:

1. Click the **Color** tab.
2. In the Expression box, type *Hopper\FlourLevel* (the same tag used in the **Fill** tab).
3. Set up Line and Fill colors for the normal state, and for the first and second warnings (see next tasks).
4. Click **Apply**, and then close the Animation dialog box.

#### To set up the color for the normal state:

- a. In the thresholds and colors list, select **A**, and leave the value in the Value box as 0.
- b. For line and fill colors, select the Solid Fill Style.
- c. For each of the colors, open the color palettes, and then click gray (the same gray used for the rectangle).

#### To set up the color for the first warning:

- a. In the thresholds and colors list, select **B**.
- b. In the Value box, type 80.
- c. For line and fill colors, select **Blink**, opening the blink palette.
- d. For the first blink color, select gray. For the second color, select yellow.
- e. Repeat step d for the fill blink colors.

#### To set up the color for the second warning:

- a. In the thresholds and colors list, select **C**.
- b. In the Value box, type 95.
- c. For line and fill colors, select **Blink**, opening the blink palette.
- d. For the first blink color, select gray. For the second color, select red.
- e. Repeat step d for the fill blink colors.

Use the Test Display tool to test the animation in this example.

When the flour level reaches 80, the rectangle will blink between gray and yellow, warning the operator that the hopper is nearly full. When the flour level reaches 95, the rectangle will blink between gray and red.

## Changing the level of fill in an object

Use the fill animation to change the level of fill in an object, based on the result of an expression or a tag value, in relation to the specified minimum and maximum values.

The object's fill level is proportional to the value of the expression. For example, if the value of the expression is halfway between the minimum and maximum values, the object will be half full.

Fill animation does not affect string input, numeric input, or Legacy Recipe objects, or push button objects. It also does not affect transparent objects or line objects, even if those objects are grouped into a single object.

For details about Fill options in the Animation dialog box, click [Help](#).

## Moving an object horizontally in a display

Use the horizontal position animation to move an object horizontally in a display, based on the result of an expression or a tag value, in relation to the specified minimum and maximum values.

The object's horizontal position is proportional to the value of the expression. For example, if the value of the expression is halfway between its minimum and maximum values, the object will be halfway between its minimum and maximum pixel offset.

For details about Horizontal Position options in the Animation dialog box, click [Help](#).

## Moving an object vertically in a display

Use the vertical position animation to move an object vertically in a display, based on the result of an expression or a tag value, in relation to the specified minimum and maximum values.

The object's vertical position is proportional to the value of the expression. For example, if the value of the expression is halfway between its minimum and maximum values, the object will be halfway between its minimum and maximum pixel offset.

For details about Vertical Position options in the Animation dialog box, click [Help](#).

## Rotating objects with animation

Use the rotation animation to rotate an object around an anchor point, based on the result of an expression or a tag value, in relation to the specified minimum and maximum values.

The degree of rotation is proportional to the value of the expression. For example, if the value of the expression is halfway between its minimum and maximum values, the object will rotate half the specified rotation range.

You can attach rotation animation to drawing objects only. The exceptions are the **Text**, **Image**, **Panel** and **Rounded Rectangle** objects.

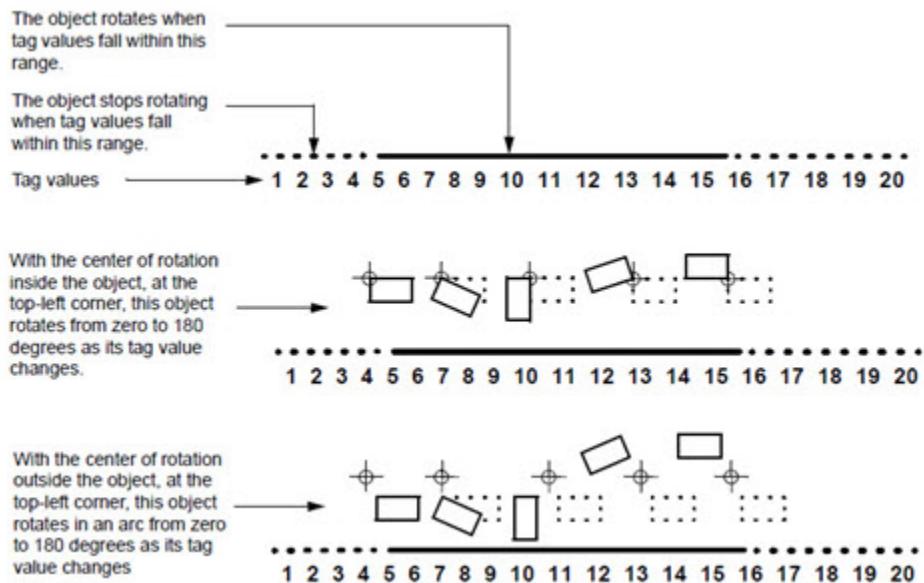
For details about Rotation options in the Animation dialog box, click [Help](#).

## Setting up rotation animation

To set up the rotation animation, you must specify:

- The tag or expression that will provide values to rotate the object.
- The range of values for the tag or expression (values outside the range will not be used to rotate the object.)
- The degree of rotation for the object.
- The center, or axis, of rotation (for example, to rotate the object around its center point, or around its top-left corner). The center of rotation can be inside or outside the object. If the center is outside the object, the object moves in an arc.

The following illustrations show how the rotation range and the center of rotation work.



Text objects always remain in their original orientation while rotating about their axis. With the center of rotation at the center of the object, text objects do not rotate at all unless part of a group of objects, to which you have applied rotation animation. To prevent a text object from rotating, exclude it from the group of objects you are animating.

## Changing the width of an object

Use the width animation to change an object's width, based on a tag value or the result of an expression or a tag value, in relation to the specified minimum and maximum values.

The object's width is proportional to the value of the expression. For example, if the value of the expression is halfway between its minimum and maximum values, the object will be half its full width.

For details about Width options in the Animation dialog box, click **Help**.

## Changing the height of an object

Use the height animation to change an object's height, based on the result of an expression or a tag value, in relation to the specified minimum and maximum values.

The object's height is proportional to the value of the expression. For example, if the value of the expression is halfway between its minimum and maximum values, the object will be half its full height.

For details about Height options in the Animation dialog box, click **Help**.

## Setting up touch zones

Use the touch animation to specify press, repeat, and release actions for an object that are triggered when an operator touches the object using a mouse or a touch screen.

To highlight touch objects, use the options in the Display Settings dialog box. You can turn on the highlight for interactive objects when the cursor passes over it at runtime. You can specify the touch area as rectangular or irregular style, and whether to show a confirmation dialog box before the release action is performed. You can also make the computer beep when a touch object is selected.

---

**NOTE:** Do not create momentary push buttons using drawing objects with touch animation. Instead, create a momentary push button object, or a button object with a momentary action. For more information about creating buttons, see [Creating the different types of push buttons on page 293](#).

---

For details about Touch options in the Animation dialog box, click **Help**.

## Creating a horizontal slider

Use the horizontal slider animation to create a graphic object that sets the value of a tag. To do this, define a path for the object and then use the mouse to move the object horizontally.

The pixel position of the object is translated into values that are written to the tag. If the tag value is changed externally, the position of the slider also changes. An object can have both vertical and horizontal slider animation.

For details about the Horizontal Slider options in the Animation dialog box, click **Help**.

## Creating a vertical slider

Use the vertical slider animation to create a graphic object that sets the value of a tag. To do this, define a path for the object and then use the mouse to move the object vertically.

The pixel position of the object is translated into values that are written to the tag. If the tag value is changed externally, the position of the slider also changes. An object can have both vertical and horizontal slider animation.

For details about Vertical Slider options in the Animation dialog box, click **Help**.

## Creating a hyperlink for an object

Use the hyperlink animation to specify a link for an object. When the object is clicked at runtime or Test Display mode, the linked destination opens. The destination can be a file like a PDF, a webpage, or a program. All graphic objects supporting the touch animation, except the numeric input and string input objects, support the hyperlink animation.

For details about **Hyperlink** options in the Animation dialog box, click **Help**.

## Animating OLE verbs

Use the OLE verb animation to set up OLE objects to perform certain actions.

When the expression evaluates to true—that is, when the expression does not equal 0—the specified OLE verb is activated. The verbs available depend on the OLE object. Typical verbs include open and edit.

For details about OLE Verb options in the Animation dialog box, click **Help**.

## Applying other types of animation to OLE objects

You can apply visibility, touch, and OLE verb animation to OLE objects. Like other objects, the type of animation you can apply depends on the object.

For example, you could apply visibility animation to a spreadsheet and then create a button that, when pressed, would display or hide the spreadsheet.

Do this to show or hide various shift reports or management summaries contained in embedded spreadsheets or database forms.

## Using index numbers to navigate to objects in a display

As you create certain FactoryTalk View graphic objects, they receive a tab index number automatically.

FactoryTalk View gives index numbers to the following types of objects:

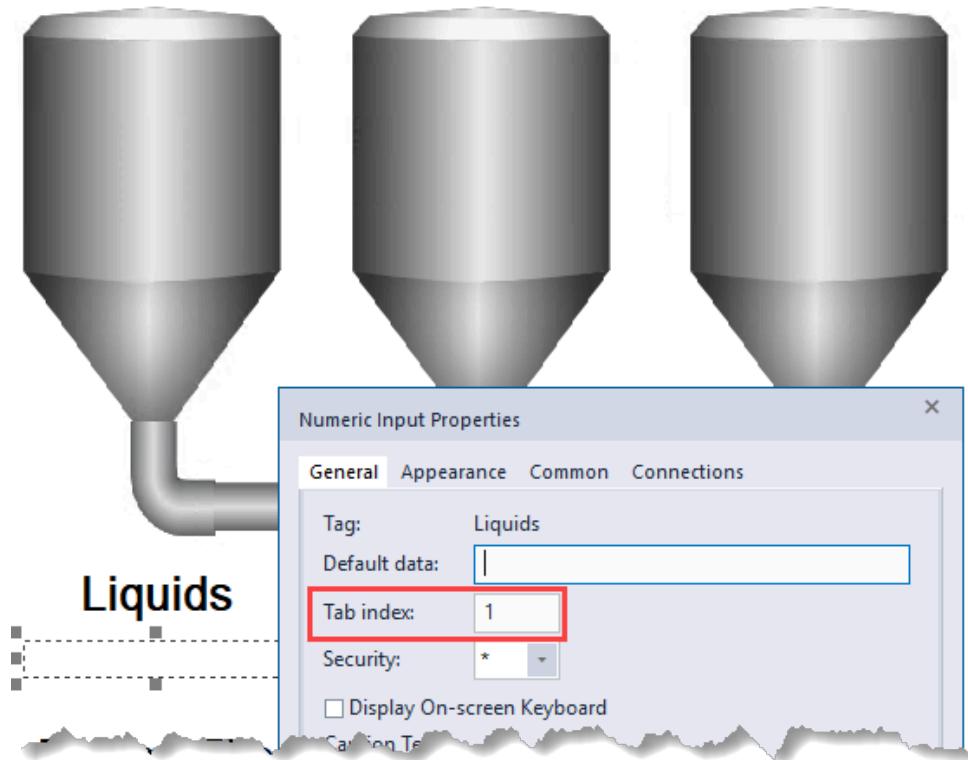
- Numeric and string input objects
- Button objects
- Objects with object key animation
- ActiveX objects

The index number increases each time you create another of these objects in the display. For example, if you create a numeric input object, followed by a button object and a string input object, the objects will have index numbers 1, 2, and 3.

FactoryTalk View also gives index numbers to objects that support key navigation, for example, display list selectors. If you want operators to be able to navigate to these objects using the keyboard, set the object's KeyNavigation property to True.

## Checking an object's index number

To check an object's index number, double-click the object to open its **Properties** dialog box, and then check the number in the **Tab** index box, as shown in this illustration. The Properties dialog box for the selected input object shows that the object has tab index number 1.



You can also check the tab index number using one of these methods:

- In the **Property Panel** of the object, check the value of the **TabIndex** property.
- For objects with object key animation, right-click the object, select **Object Keys**, and then check the number in the **Tab** index box.

## How tab index numbers work

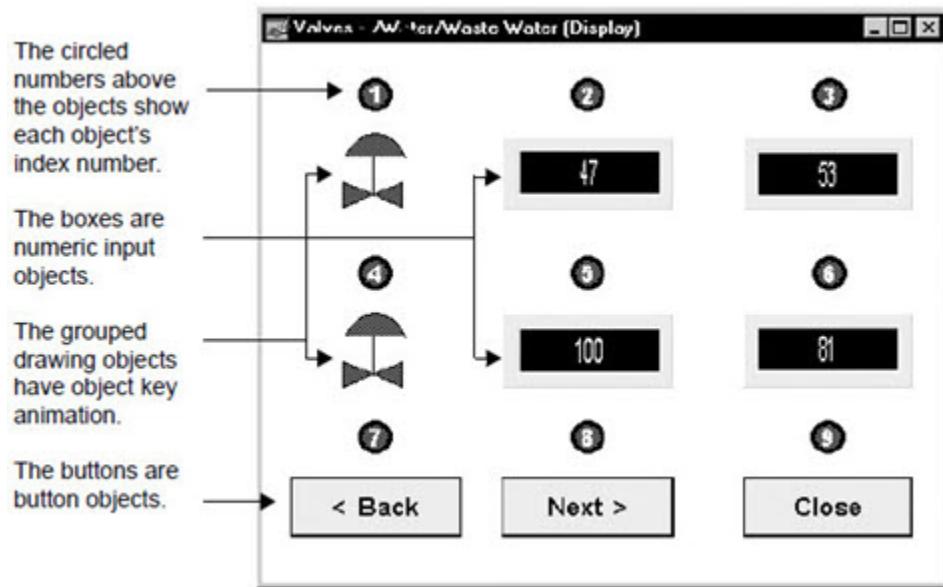
Tab index numbers are used to:

- Determine a tab sequence for objects that an operator can navigate to in a display.
- Move among objects using the **Position**, **NextPosition**, and **PrevPosition** commands at runtime. For details about these commands, see the FactoryTalk View Site Edition Help.
- Specify which tag values in a recipe file go into which numeric or string input objects.

## Creating a tab sequence

You can use index numbers to create a tab sequence. The tab sequence is the order in which users can move through a series of objects in a graphic display, using the Tab key.

In the following illustration, the tab order is from left to right:



### Moving through a tab sequence:

To move through the tab sequence in a display, use one of these methods:

- Press the **Tab** key to move through the objects from the lowest index number (1) to the highest index number.
- Press the **Shift-Tab** keys to move through the objects from the highest index number to the lowest index number.

## Changing index numbers

Once you have created two or more objects that have index numbers, you can change the index numbers.

For example, if you have created four input objects, you can modify the fourth object to have an index number 1. However, you cannot change the fourth object to have index number 5, because there are only four input objects in the display.

When you change an index number, other index numbers are adjusted automatically, so that no two objects in the display have the same index number, and so that there are no gaps in the numbering.

If you type an index number that is too high, FactoryTalk View changes it to the highest available number. If you type a number that is in use, FactoryTalk View renames other objects in the display, changing the tab sequence.

## Associating objects and displays with keys

You can associate FactoryTalk View commands with graphic objects in a display, or with the display itself, using object keys and display keys.

A key can be either a key on a keyboard or keypad, or a button on a touch screen connected to the computer or monitor an operator is using to interact with an application at runtime.

An operator can use keys to perform actions such as moving from one display to another, setting tag values, and so on.

Some keys are reserved for use by Windows and FactoryTalk View SE. For more information about reserved keys, see [Keyboard shortcuts on page 341](#).

### About client keys

Client keys are different from object and display keys. Client keys are always active at runtime. Object and display keys are only active while a specific object or display has focus.

For more information about client keys, see [Creating client keys on page 434](#). For information about precedence among object, display, and client keys, see [General rules governing precedence on page 340](#).

## Setting up object keys

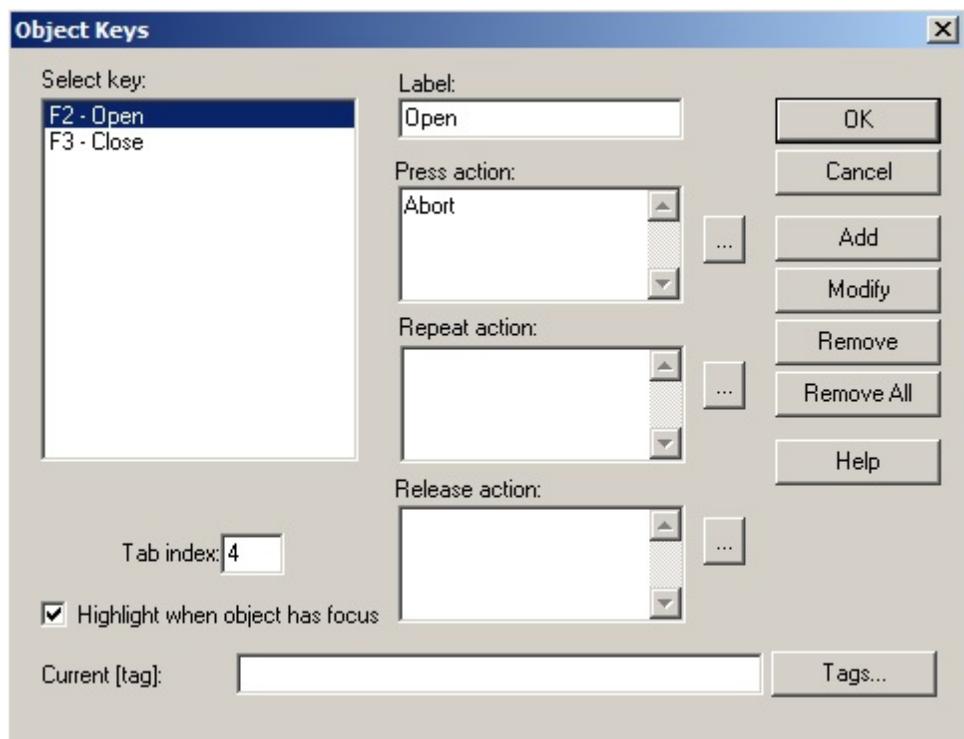
Use object keys to connect a graphic object with one or more keys.

For example, if you connect a rectangle to the F4 key, at runtime, when the rectangle has input focus, the operator can press **F4** to perform the key's press, repeat, or release action.

Use the Object Keys dialog box to set up keys, assign an index number to the object, and specify whether the object has a highlight box when it is selected at runtime.

### To open the Object Keys dialog box:

Right-click the object that you want to set up keys for, and then select **Object Keys**.



For details about options in the Object Keys dialog box, click **Help**.

### Example: Using object keys to open and close valves

A graphic display shows a tank with two valves, named Valve 1 and Valve 2.

The valves control the flow in and out of the tank. Both valves are set up with the same object key animation:

F2 = Open

F3 = Close

At runtime, the operator can select either Valve 1 or Valve 2, and press F2 to open the valve or F3 to close it.

## Setting up display keys

Use display keys to connect a graphic display with one or more keys.

Display keys are similar to object keys, except you don't have to select an object in the display for a display key to respond.

For example, you could connect a display to the F6 key. At runtime, when the display is open, the operator can press **F6** to perform the key's press, repeat, or release action.

### To open the Display Keys dialog box:

Right-click the display, and then select **Display Keys**.

For information about options in the Display Keys dialog box, click **Help**.

### Example: Using a display key to open a menu

Suppose you want to use the Home key to open a main menu display.

To do this, create a display key that redefines the Home key, by following these steps:

1. Right-click the display, and then select **Display Keys**.
2. In the Display Keys dialog box, assign a display key.
3. Type the following in the **Press Action** box:

Display "Main Menu"

Whenever the operator presses Home, the active graphic display closes and the Main Menu display opens.



**Tip:** If the Main Menu display is of the Overlay type, you must use the **Abort** command to close the active display.

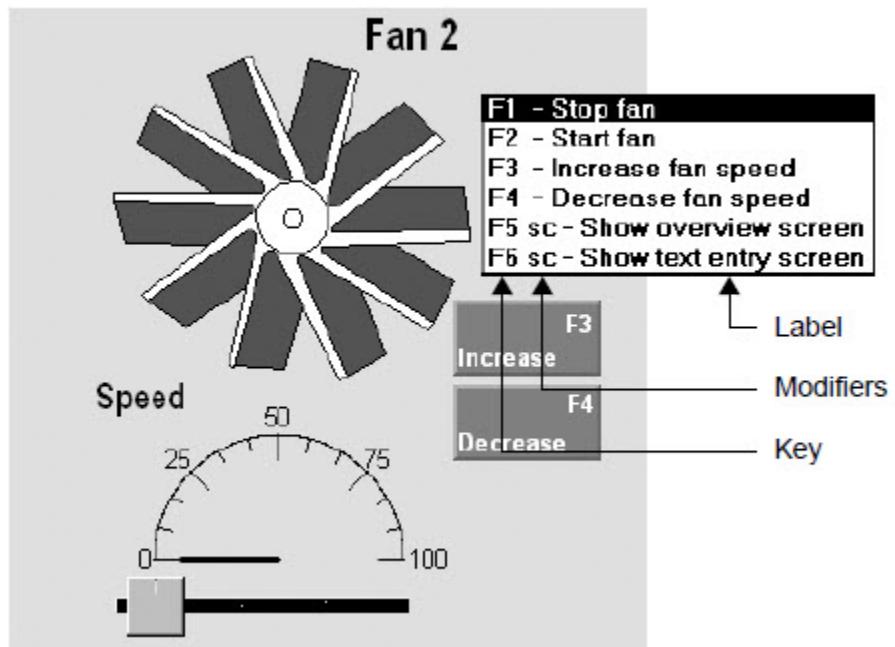
For details about how keys and other interactive objects work at runtime, see [Specifying the behavior of interactive objects](#) on page 283.

## Viewing the key list at runtime

An operator can open a key list at runtime, to see which keys are associated with an object and with the graphic display, and to see what actions the keys will perform.

### To show the key list:

Click an object that is associated with object keys.



The key list shows all the object keys set up for the selected object first, followed by all display keys. Key labels are as specified in the **Object Keys or Display Keys** dialog box.



**Tip:** The key list does not separate the two types of keys because this distinction does not matter to an operator. The operator only needs to know what will happen when a key is pressed.

### To show the key list:

- Use the /0 parameter with the **Display** command. For details, see Help about the **Display** command.

## Setting up object-specific commands using keys

You can set up object and display keys, using the [tag] parameter, so that selecting different objects in a display runs different FactoryTalk View commands.

To do this, you would create a display key first, using the [tag] parameter in the command specified for its press, repeat or release action, for example:

```
Set [tag] 1
```

Then, you would set up object keys for the different graphic objects in the display, using the Current [Tag] parameter as a placeholder for a tag name (or any character string).

At runtime, when an object is selected and the display key is pressed, the contents of the Current [tag] box replace [tag] in the FactoryTalk View command.

The following three examples show how you can use the Current [tag] parameter.

### Example 1: Creating display-wide keys to open and close valves

Imagine you have a graphic display containing 20 valves and you want the operator to be able to open and close all the valves with the same two keys, F2 and F3, respectively.

Instead of specifying the tag name for each valve, use the [tag] parameter with display keys. Set this up as follows:

1. In the **Display Keys** dialog box, assign the **F2** and **F3** keys.
2. For the **F2** key's press action, type **Set [tag] 1**.
3. For the **F3** key's press action, type **Set [tag] 0**.
4. For each valve object, open the **Object Keys** dialog box, and type the tag name in the **Current [tag]** box.

At runtime, the operator can select any valve object and press **F2** to open the valve, or press **F3** to close the valve.

### Example 2: Opening object-specific Help

Imagine you have a graphic display containing various graphic objects that represent plant-floor equipment.

If something goes wrong with a piece of equipment, you want the operator to be able to select the corresponding object in the display, and then open a Help file that suggests steps for correcting the situation.

1. In the Display Keys dialog box, create a display key. For details, click **Help**. In the **Press Action** box, type **Help [tag]**.
2. In the Object Keys dialog box, type the appropriate Help file name in the **Current [tag]** box for each equipment object.

At runtime, when the operator selects any equipment object and presses the appropriate Display key, the help file for that piece of equipment is shown.

## Animating ActiveX objects

To animate an ActiveX object in FactoryTalk View SE, connect the object's properties, methods, or events to tags, so it can receive or modify data at runtime. To use tag data to animate an ActiveX object, use one of these methods:

- In FactoryTalk View Studio, use the Property Panel, the Invoke command, or the Events and Methods editors. These methods are simpler, but less flexible.
- In the Visual Basic IDE (Integrated Development Environment), use VBA code. This method is more complex, but more flexible.

A graphic display is like a Visual Basic form. When you insert an ActiveX control in a graphic display, you can gain access to the ActiveX control's methods, properties, and events using VBA code.

For an overview of how VBA code works in FactoryTalk View SE, see [Using the SE Client object model and display code on page 475](#).

For detailed information and code examples, see the FactoryTalk View Site Edition Help.

You can also change an ActiveX object's properties without using tags. If you choose this method, the object's properties do not change dynamically at runtime.

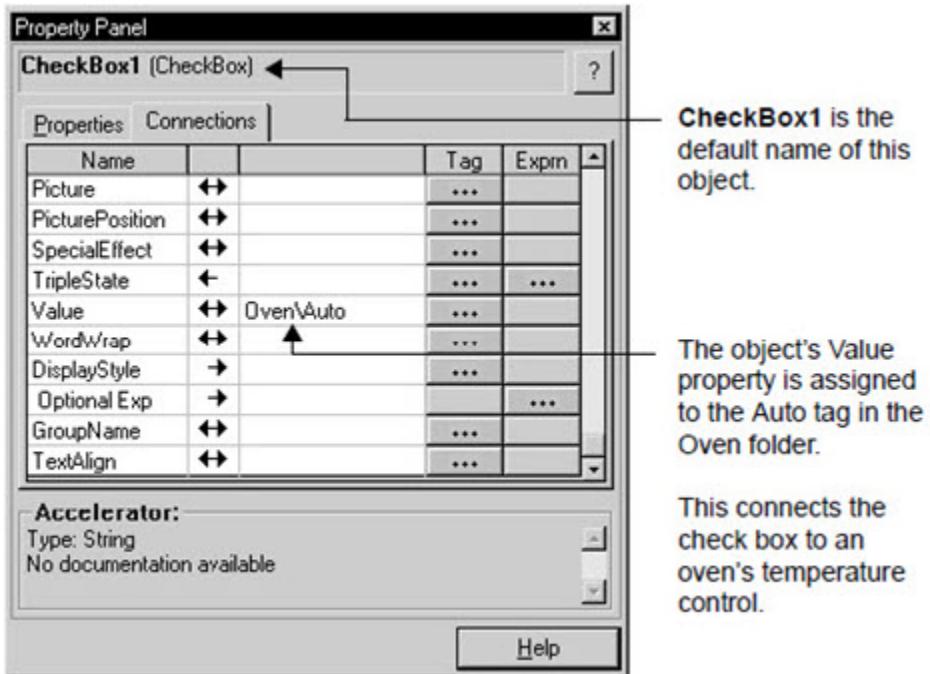
You make static changes to an ActiveX object's properties just as you do for any other graphic object—in the Properties tab in the Property Panel.

**NOTE:** The runtime behavior of ActiveX objects depends on the vendor's implementation. Before making your application available to users, test ActiveX objects thoroughly to ensure they behave as desired.

## Connecting tags to an ActiveX object's properties

ActiveX objects have sets of different properties. You can view an object's properties in the **Property Panel**. The content of the Property Panel is determined by the application that created the ActiveX object, not by FactoryTalk View. For information about a property, select the property, and then click **Help**.

In the following illustration, the Property Panel is for an ActiveX checkbox named **CheckBox1**. Clearing this checkbox changes an oven's temperature control from Automatic mode to Manual mode.



## About ActiveX object names

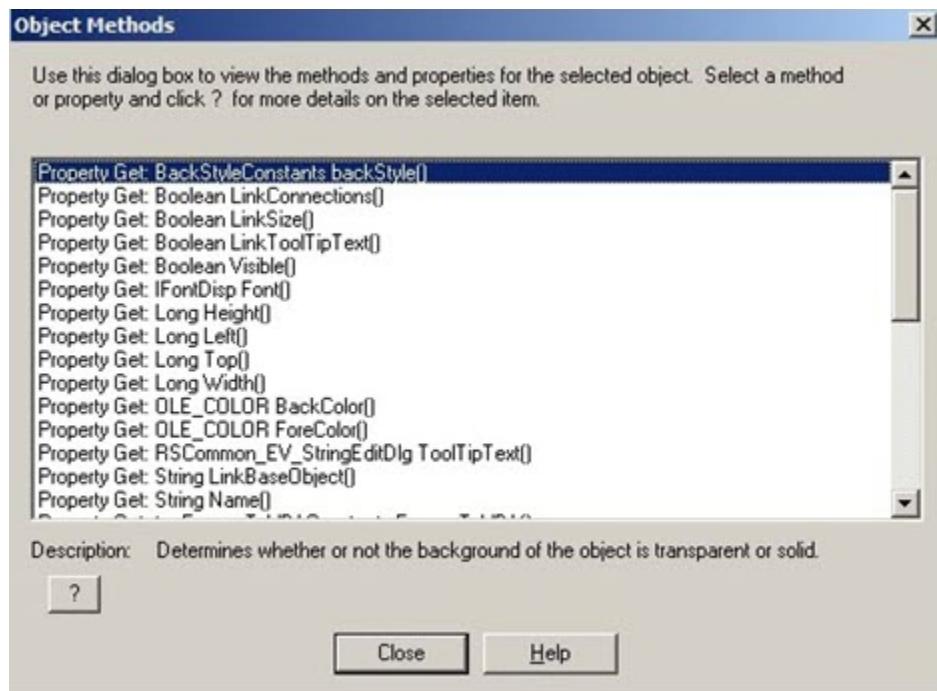
As shown in the previous illustration, ActiveX objects have a default name. The name is used:

- To identify the object when logging its events.  
For information about ActiveX events, see [Connecting tags to an ActiveX object's events on page 333](#).
- With commands.  
For example, when using the **Invoke** command to call a method, you must specify the name of the object in which the method is implemented. For more information, see [Connecting tags to an object's methods on page 332](#).

You can change the default name for any object.

## Viewing methods of an object

The **Object Methods** dialog box shows the methods implemented for the selected object. To view the object's methods, right-click the object, and then select **Methods**. For details about options in the **Object Methods** dialog box, click **Help**.



## Connecting tags to an object's methods

A method is an action the ActiveX object can perform.

Methods are part of the object. You can:

- View a list of the object's methods.
- Find documentation for the object's methods.
- Use the **Invoke** command to call the object's methods.

## Using the **Invoke** command to call a method

Use the FactoryTalk View **Invoke** command to call a method for the named ActiveX object. With the **Invoke** command you can:

- Assign the value returned by a method to a tag.
- Set an object's property to a tag value or a constant.
- Set a tag to the value of an object property.

To specify parameters for the **Invoke** command, use the **Command Wizard**. For details about using the **Invoke** command, see the FactoryTalk View Site Edition Help.



**Tip:** To use the **Invoke** command to interact with an ActiveX object, the object must have a name.

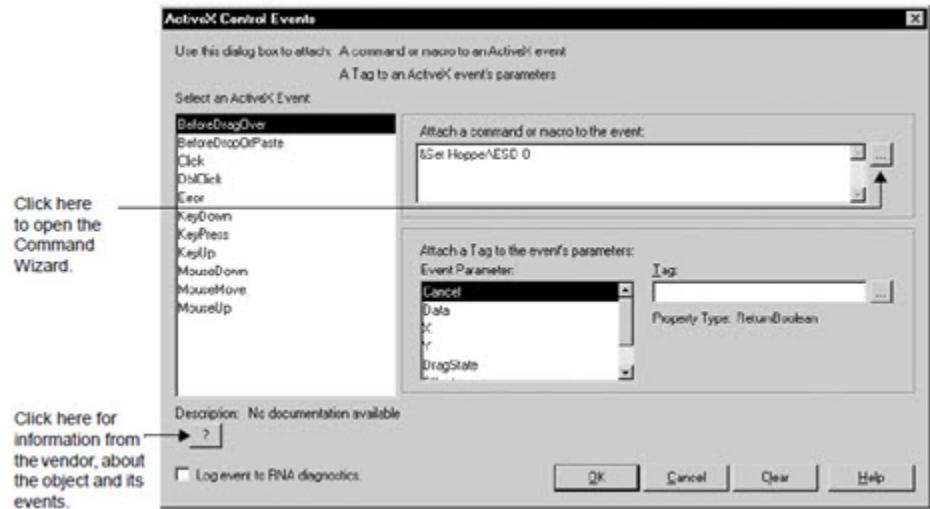
## Connecting tags to an ActiveX object's events

The **ActiveX Control Events** dialog box shows the events associated with the selected ActiveX object. In the dialog box, you can specify FactoryTalk View commands or macros that will run when an event occurs at runtime.

You can also use VBA code to make an ActiveX object's properties, methods, and events interact with FactoryTalk View SE. For VBA code examples, see the FactoryTalk View Site Edition Help.

### To view an ActiveX object's events:

Right-click the object, and then select **ActiveX Events**.



For details about options in the ActiveX Control Events dialog box, click **Help**.

## Setting up navigation

This chapter describes:

- What a display hierarchy is.
- Setting up ways to move among displays in an application.
- Setting up keys to run FactoryTalk View commands.
- Creating and running client key components.
- Creating and using navigation buttons.

### Designing a display hierarchy for an application

An important part of designing a complete operator interface is determining how operators will navigate through and interact with graphic displays in an application.

To direct an operator through the main parts of an application, set up a hierarchy (or series) of graphic displays that provides progressively more detail as the operator moves through different levels of information and data.

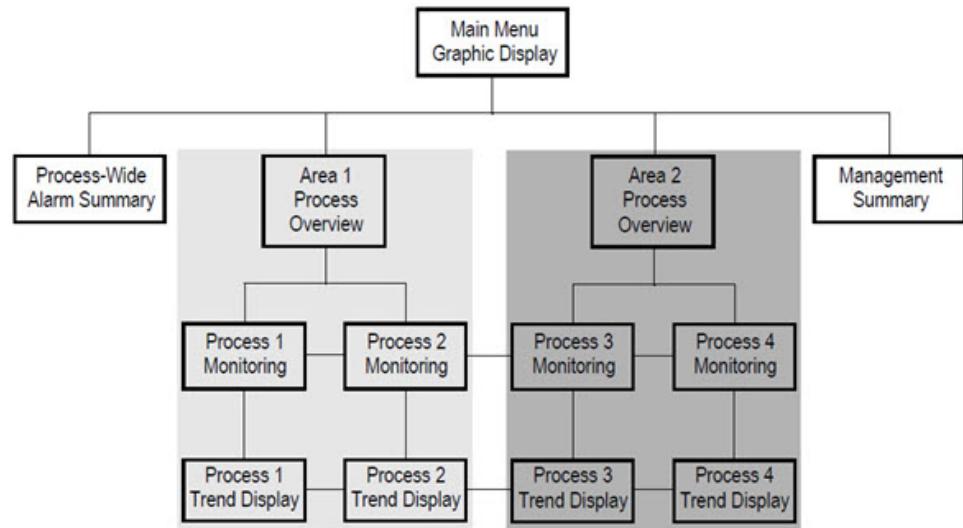
The display hierarchy can represent parts of a plant or process, as well as different types of data displays. For example, the top level might represent an area in the plant, and the bottom level might contain trends and alarm displays specific to each area.

Operators or supervisors with the necessary security permissions, might also be able to navigate between areas in the application, or gain access to displays that provide specific information, such as management summaries.

When designing a display hierarchy, consider the needs of the various application users, including managers, supervisors, and operators. A hierarchy might include:

- An initial graphic display that serves as a menu.
- An overview of the plant, including links to displays on FactoryTalk View SE Servers in areas around the plant.
- A comprehensive display of each process being monitored.
- Process-specific displays.
- Management summary displays.
- Trend displays of historical and real-time data.
- Alarm displays, for monitoring and responding to alarms.

The following illustration shows a simple display hierarchy design for a network distributed application that contains two areas:



**Tip:** For a live example of a display hierarchy that involves different navigation methods, run the FactoryTalk View SE InstantFizz application. To do this, open **C:\Users\Public\Documents\RSView Enterprise\SE\Client** and double-click a .cli file with the appropriate resolution to run the application.

## Setting up ways to move among displays

For operators to move among displays, create navigation with the following methods:

- Keyboard-based
- Touch screen-based

Although keyboard-based and touch screen-based methods look different to operators, they work similarly; that is, both involve the use of FactoryTalk View commands.

## Using commands to open, close, and switch displays

You can use the following FactoryTalk View commands to open, close, and switch between open displays at runtime.

Use the commands in macros, or as actions specified for touch zones, buttons, display keys, or object keys in a graphic display.

To do this	Run this command
Open the specified graphic display. If the specified display is already open and it allows multiple running copies (set up in the <b>Display Settings</b> dialog box), the <b>Display</b> command opens another copy of the display and makes it active. If the specified display is already open and it does not allow multiple running copies, the <b>Display</b> command makes the display active.	<b>Display</b>

To do this	Run this command
A display of type: <ul style="list-style-type: none"> <li>• <b>Replace</b> opens on top of other open displays, and closes the ones it overlaps.</li> <li>• <b>Overlay</b> opens on top of any open displays, but does not close them.</li> <li>• <b>On Top</b> opens on top of any open displays and remains in the foreground.</li> </ul>	
Close the active or specified graphic display. Use the Abort command when you cannot use a display of type Replace to close other running displays.	<b>Abort</b>
Pull the specified graphic display in front of other open displays.  If the specified graphic display is of the Replace or Overlay type, PullForward gives the specified display focus and positions it behind any On Top display that is open.	<b>PullForward</b>
Push the specified graphic display behind other open displays.  If the specified display is of type On Top, the PushBack command positions the display behind any other open On Top displays, but in front of any open Replace or Overlay displays.	<b>PushBack</b>



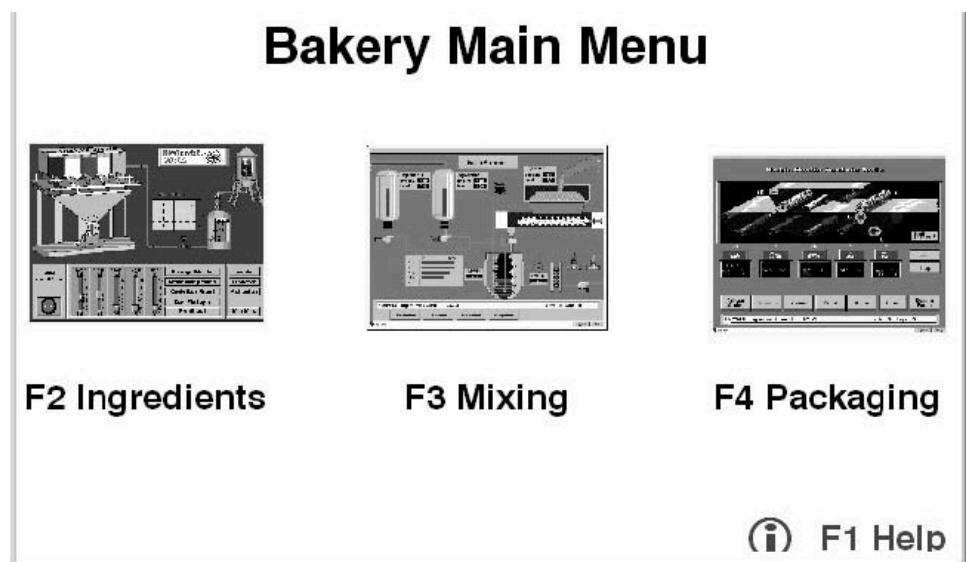
**Tip:** The **PullForward** and **PushBack** commands provide quick display changes because displays are already open. However, the more displays you have open, and the more complex the displays are, the more memory and CPU are used.

Commands for opening, closing, and switching displays run only at the FactoryTalk View SE Client. Attempting to run these commands at a FactoryTalk View SE Server (for example, in a server startup macro) or in FactoryTalk View Studio, will result in errors.

For more information about where commands run, see [FactoryTalk View commands on page 442](#).

### Example: Setting up navigation using keyboard operation

The graphic display in this example is designed to act as a menu, by providing keys that an operator can press to open graphic displays representing different processes.



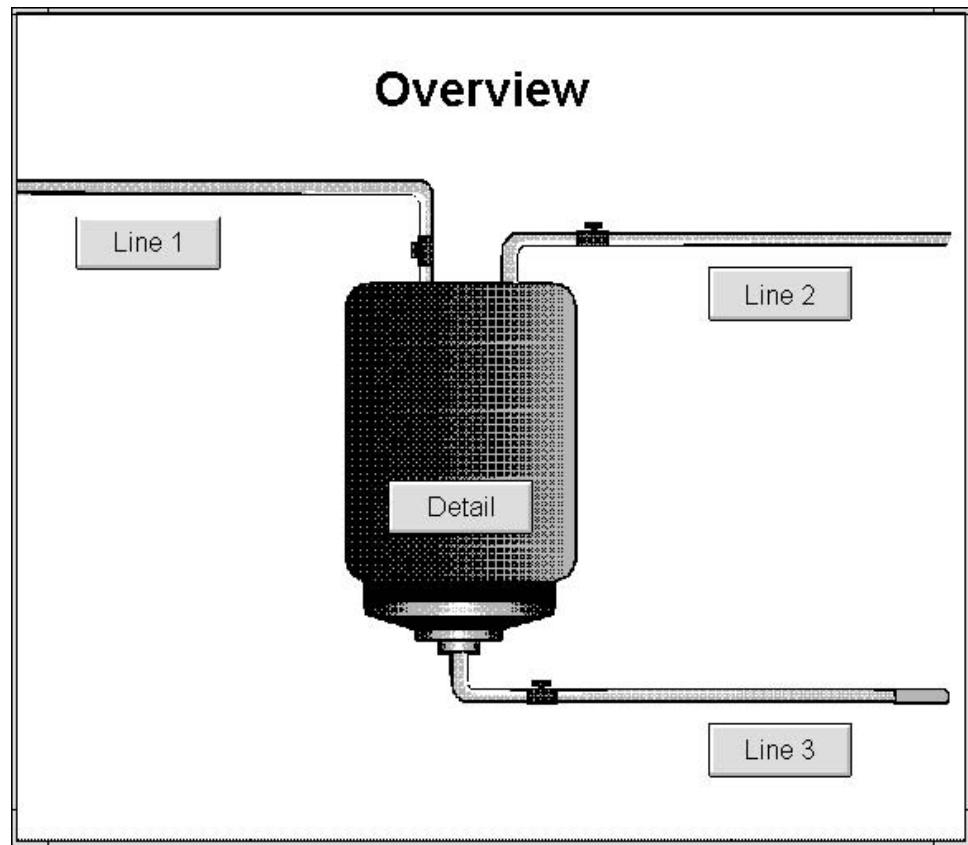
To create this display, the designer assigned various FactoryTalk View commands to keys using the three types of key definitions: object, display, and client. In all cases, keys (not mouse buttons) were defined to run commands.

Object keys and display keys are set up in the Graphics editor. For more information, see [Animating graphic objects on page 313](#).

Client keys are created in the Client Keys editor. For more information, see [Creating client keys on page 434](#).

### Example 2: Setting up navigation using mouse and touch screen operation

The graphic display in this example contains buttons that an operator can click using a mouse, or press a touch screen, to open detail displays. This display acts as a menu and presents information.



To create the buttons, the designer used the **Button drawing** tool in the Graphics editor. The buttons can be selected with a mouse or by pressing a touch screen. For information about creating buttons, see [Creating the different types of push buttons on page 293](#).

## Using the navigation menu to switch displays

The application-level runtime navigation menu provides runtime search capability and alarm indicators and does not need FactoryTalk View objects to switch between displays. You can create one or multiple navigation menus that can be shown distinctly on different display clients, multiple monitors, or one display client's different positions. Each navigation menu can be built according to the hierarchy of your application. The alarm and automatic diagnostic notifications ensure that operators are promptly informed. Additionally, you can use the runtime search to locate referenced tags across all displays configured in the display client's navigation menu and double-click an item to navigate to the associated tag's faceplate or graphic display.

To configure the navigation menu, perform these key tasks:

1. Configure the navigation menu appearance.
2. Add and organize menu items or folders.
3. Configure individual menu items or folders on the navigation menu, including name, icon, how a display shows, and alarm and automatic diagnostic event notifications.
4. Launch the navigation menu using the **DisplayNavigationMenu** command.

## Choosing display types with navigation in mind

When designing an application, the display types give you additional control over how an operator can navigate from one display to another.

For example, use the **On Top** option to keep a display on top at all times, even when another display has focus.

If you want a display to replace any open displays that it covers or touches when it opens, use the **Replace** option.

You select a type for a graphic display in the **Display Settings** dialog box. For more information, see [Specifying the display type on page 277](#).



**Tip:** Displays that you want to run in a fixed position, for example, menus or banners, can be docked to the inside of the FactoryTalk View SE Client window. For more information, see [Docking displays to the FactoryTalk View SE Client window on page 285](#).

## Reducing display call-up time

To reduce the time required to open a graphic display, load it into the display cache. You can load the display:

- Before it is shown by using the **Display** command with the /Z or /ZA parameter. For details, see the FactoryTalk View Site Edition Help.
- When it is shown for the first time by using the **Cache After Displaying** option in the **Display Settings** dialog box. For details, see [Caching displays on page 278](#).

The runtime performance of applications that leverage a large quantity of global objects can be impacted. To optimize the runtime performance of the FactoryTalk View SE client, click  on the toolbar to compile global objects. An asterisk shows on , indicating that there are pending changes in the system that require compilation.

## Setting up keys to run FactoryTalk View commands

You can associate FactoryTalk View commands with objects in a display, or with the entire display using object or display key animation.

You can also associate commands with keys that are independent of objects or displays, and are available at all times throughout the system by creating client keys. For more information, see [Creating client keys on page 434](#).

Operators can use keys to interact with the system, for example, to change displays or set tag values. When deciding what type of key to create, use the following table as a guide:

To do this	Set up	For details, see
Associate a key with a specific graphic object (object key)	Object key animation in the Graphics editor	<a href="#">Setting up object keys on page 327</a>
Associate a key with a specific graphic display (display key)	Display key animation in the Graphics editor	<a href="#">Setting up display keys on page 328</a>
Create a key that works everywhere on a FactoryTalk View client (client key)	A key definition component in the Client Keys editor	<a href="#">Creating client keys on page 434</a>

## General rules governing precedence

You can assign a single key to one or more of the three types of key definitions – object, display, or client.

For example, you could assign the F2 key to open a valve when the valve object has input focus, close a pop-up display that has focus, and, as a client key, to open a graphic display containing a process overview.

When a single key has more than one definition, the following rules of precedence apply:

- When a graphic display is active and an object has input focus, object keys have precedence over display keys and client keys.
- When a graphic display is active, display keys have precedence over client keys.

For example, if you assign the F2 key as a display key in some graphic displays in an application, and you assign F2 as a client key in the same application, F2 will only work as a client key if the active display does not also use F2 as a display key.

When designing an application, pay particular attention to the keys used by embedded objects.

Object keys and display keys generally have precedence over keys used by embedded objects (for example, ActiveX, or OLE objects).

However, keys used by OLE objects that are not part of FactoryTalk View (for example, an Excel worksheet), have precedence over object or display keys. For details, see the pages that follow.

## Precedence and the F1 key

When you are developing an application in FactoryTalk View Studio, the F1 key is reserved for opening context-sensitive Help.

At runtime, if a graphic display has focus and a press, release, or repeat action has been defined for the F1 key, F1 acts as a display, object, or client key instead of opening Help.

## Precedence and embedded ActiveX objects

When a graphic display is active and an embedded ActiveX object has input focus, a key that triggers an action in the embedded object will not trigger that action, if the same key is also defined as an object or display key.

Instead, when you press the key, the action associated with the object key or display key will be triggered.

For example, an ActiveX slider object controls the speed of a motor by using the F2 key to increase the speed and the F3 key to decrease the speed. If F2 is also defined as an object key to jog the motor's position, pressing F2 when the slider has input focus will always jog the motor's position, instead of increasing the motor's speed.

If a key that triggers an action in an embedded ActiveX or OLE object is also defined as a client key, pressing that key will trigger both the action defined for the embedded object and the action defined for the client key.

For example, if the F2 key for an ActiveX gauge object increases a motor's speed, and F2 is also defined as a client key to print the current graphic display, each time the operator presses F2, the motor's speed will be increased, and the graphic display will be printed.

## Precedence and embedded OLE objects

For embedded OLE objects, a key that triggers an action in the embedded object will trigger only that action, even if the key is also defined as an object or display key. In this case, the action defined for the object or display key will not be triggered at all.

If a key that triggers an action in an embedded ActiveX or OLE object is also defined as a client key, pressing that key will trigger both the action defined for the embedded object and the action defined for the client key.

## Keyboard shortcuts

The following keyboard shortcuts are normally reserved for use by Windows and FactoryTalk View SE.

To do this	Press this key
Move focus to the object with the next highest index number.	<b>Tab</b>
Move focus to the object with the next lowest index number.	<b>Shift+Tab</b>
Move focus to the next object, in the direction the arrow key points.	<b>Ctrl+Up Arrow,</b> <b>Ctrl+Left Arrow,</b> <b>Ctrl+Down Arrow,</b> <b>Ctrl+Right Arrow</b>
Move focus to the next window.	<b>Ctrl+F6</b>
Move focus to the previous window.	<b>Ctrl+Shift+F6</b>
Close the active window.	<b>Ctrl+F4</b> or  <b>Ctrl+Shift+F4</b>
Perform the press and release actions for the button object that has focus.  Download the value in the input object that has focus.  Open the Recipe dialog box when a Legacy Recipe object has focus. If Ctrl+W was pressed previously, the recipe is saved. If Ctrl+R was pressed previously, the recipe is restored.  Open the on-screen keyboard, if the input or Legacy Recipe object with focus is set up to display the keyboard.	<b>Enter</b>
Upload data into all input objects in the display.	<b>PgUp</b>
Upload data into the input object that has focus.	<b>Ctrl+PgUp</b>
Download data from all input objects in the display.	<b>PgDn</b>
Download data from the input object that has focus.	<b>Ctrl+PgDn</b>
Delete the contents of the input object.	<b>Home+Shift+End+Del</b>
Move the input focus to the Legacy Recipe object, and prepare for a recipe restore.	<b>Ctrl+R</b>
Move the input focus to the Legacy Recipe object, and prepare for a recipe save.	<b>Ctrl+W</b>
Open the Recipe dialog box.	+ on the numeric keypad

To do this	Press this key
If Ctrl+W was pressed previously, the recipe is saved. If Ctrl+R was pressed previously, the recipe is restored.	
Move the selection bar on the Object Key menu.	<b>Up Arrow, Down Arrow</b>
Close the Object Key menu, or exit input mode for the updating input object that has focus.	<b>Esc</b>
Move the cursor one position left or right.	<b>Left Arrow, Right Arrow</b>
Delete the character to the left of the cursor.	<b>Backspace</b>
Delete the character to the right of the cursor.	<b>Del</b>
Delete all characters from the cursor position to the end of the line.	<b>Shift+End+Del</b>
Copy the selected items to the clipboard.	<b>Ctrl+C or Ctrl+Ins</b>
Cut the selected items and place them in the clipboard.	<b>Ctrl+X, or Shift+Del</b>
Paste the contents of the clipboard at the current cursor position.	<b>Ctrl+V, or Shift+Ins</b>
Position the cursor at the beginning of the data entry object.	<b>Home</b>



**Tip:** The arrow keys perform different actions when a trend graphic object has focus.

## Precedence and reserved keys

If you assign a reserved key to an object or display key, the object or display key function takes precedence, and the default, reserved function of that key is disabled.

However, if you use a reserved key or key combination as a client key, the key will perform both the actions of the client key and the action of the reserved key. For that reason, using reserved keys to define client keys is not recommended.

For more information about client keys, see [Creating client keys on page 434](#).

## About navigation buttons

You can create up to three types of navigation buttons:

Button type	Button function
DisplayPreviousScreen	Opens the previous graphic display in the navigation history when the button is clicked.
DisplayNextScreen	Opens the next graphic display in the navigation history when the button is clicked.
DisplayNavigationHistory	Opens a list of previously opened displays when the button is clicked. From the list, you can select a display to view it.

You can use the navigation button object to create buttons to navigate between previously viewed graphic displays. FactoryTalk View SE can maintain a navigation history of the displays that are viewed, and the navigation buttons let you browse back to a previous display and forward to the next.

Each navigation button performs a single function at runtime, with the option to display the previous screen, the next screen, or the navigation history. Multiple navigation buttons can be added in a group to mimic the functionality of navigation buttons in a web browser, letting you move back, move forward, and view history.

Navigation history is optional, and is configured for each client. To track displays, you have to do the following:

- Activate the display navigation functionality using the FactoryTalk View SE Client.
- Specify what displays to track using the Properties tab in the Display Settings for each display.

This provides flexibility in tracking only the displays you specify.

For additional details, click Help in the FactoryTalk View SE Client wizard.

## How navigation buttons work

When the button is active, clicking it starts the corresponding action that is set on the **Navigation Button Properties** dialog box. If the button is inactive, clicking it causes no response.

Whether the button is active or inactive depends both on the configured action of the button and the position of the current opened graphic display in the navigation history list. For example, if the button is configured to show the next graphic display and the current shown graphic display is the first one in the navigation history list, the button is inactive. When the current shown graphic display is not the first one in the navigation history list, the button is active.

# Creating expressions

This chapter describes:

- What expressions are and where you can use them.
- Working in the **Expression** editor.
- Using tag names and tag placeholders in expressions.
- Using constants in expressions.
- Using operators to modify values.
- Using built-in functions.
- Using if-then-else logic.

## About expressions

Sometimes the data you gather from devices is only meaningful when you compare it to other values, combine it with other values, or create a cause-effect relationship with other values. You can use expressions to create mathematical or logical combinations of data that return more meaningful values.

An expression can be a tag name, a tag placeholder, a mathematical or logical equation, or a function that returns a numeric, string, or true or false value. For expressions that return true and false values, 1 and other nonzero values signify True, and zero signifies False. Expressions let you manipulate the data gathered from devices and make it more meaningful by comparing it to other values, combining it with other values, creating cause and effect relationships between values, or displaying it.

## Where you can use expressions in FactoryTalk View SE

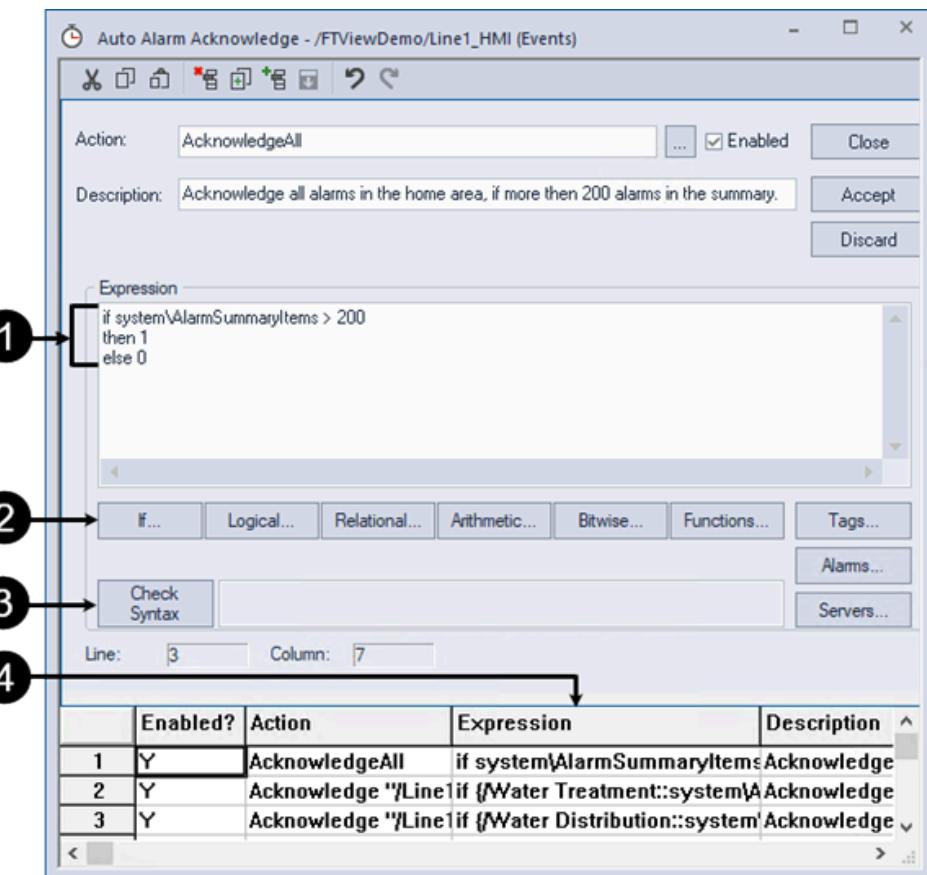
In FactoryTalk View Studio, you can set up expressions in the following editors:

- In the **Graphics editor**, use an expression to control the appearance of a graphic object, or to display the value of an expression in a numeric or a string display object.  
You can also use expressions to set the value of a tag, using the **If** or **Set** command. For details, see the FactoryTalk View Site Edition Help.
- In the **Derived Tags editor**, specify the name of a tag that will store the result of an expression.
- In the **Events editor**, associate an expression with an action (a command or macro). When the expression changes from *false* to *true* (from 0 to any nonzero value), the action runs.
- In the **Data Log Models editor**, when setting up file management for a data log model, use an expression to determine when log files are created.

## Working in the Expression editor

Editors that can use expressions contain an Expression box. The editors might also contain expression buttons, and an Expression column.

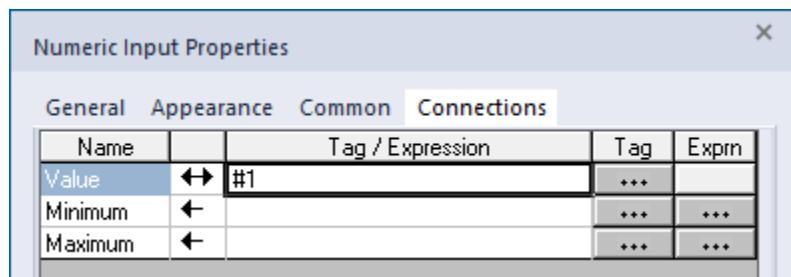
For example, the following illustration shows the **Expression** box, buttons, and column in the **Events** editor.



Items	Description
1	Type the expression in the Expression box.
2	Select from lists of components to help build the expression.
3	Check the syntax of the expression.
4	View the expression in the Expression column.

Dialog boxes or tabs, in which you can type expressions, contain an **Expression** button for opening the Expression editor.

For example, the following illustration shows the **Expression** button in the **Connections** tab of the **Numeric Input Properties** dialog box.



## Expression components

Expressions can contain:

- Tags, tag placeholders, arithmetic, and bitwise operators, and mathematical functions such as SQRT (square root) that return numeric values.
- Relational and logical operators, and functions such as ALMFAULT (alarm fault) that return true/false values.
- Non-mathematical functions, such as the CurrentLanguage function that returns the RFC17766 code for the current language of an application.
- If-Then-Else logic that returns numeric or true and false values, depending on how it's structured. These are called conditional expressions because the result of the expression depends on whether the If statement is true or false. When the If statement evaluates to True, the result is defined by the Then statement. When the If statement is False, the result is defined by the Else

To select components, click the following buttons in the **Expression** box or editor.

To open this list	Click
If - then - else operators	If...
Logical operators	Logical...
Relational operators	Relational...
Arithmetic operators	Arithmetic...
Bitwise operators	Bitwise...
Built-in functions	Functions...
Tags (in the Tag Browser)	Tags...
Alarm sources (in the Alarm Source Browser)	Alarms...
Servers in the current open application	Servers...

## Checking the syntax of an expression (Expression editor)

In the Derived Tags editor, to verify that the expression you typed uses correct syntax, click **Check Syntax**. If the syntax is invalid, an error is shown in the Check Syntax box.

You can check the syntax of an expression at any time while the Derived Tags editor is open. The syntax is also checked automatically when you click **Accept** or **OK** in the editor.

## Cutting, copying, and pasting expressions

You can cut, copy, or paste an expression or parts of an expression. How you perform these actions depends on whether the expression is in a window or in a dialog box.

When you are working in a window, you can use the commands on the **Edit** menu, the toolbar, or the keyboard.

When you are working in a dialog box, you can only use the keyboard because there is no access to the menu bar or toolbar.

When you cut or copy an expression to the clipboard, you can paste it into any other **Expression** box. You can also paste it into a command line.

## Formatting expressions

You can format expressions using tabs, line returns, and multiple spaces so they are easier to read. However, do not let tag names, key words, function names, or function arguments span more than one line.

### Example: Formatting an expression

To format the if - then - else statement, you can align the Else with the appropriate If, so the logic is easy to understand:

```
If (tag1 > tag2) Then 0  
  
Else If (tag1 > tag3) Then 2  
  
Else 4
```

Or you can condense the statement:

```
If (tag1 > tag2) Then 0 Else If (tag1 > tag3) Then 2 Else 4
```

## Using tag names and tag placeholders in expressions

You can use a tag name as part of an expression, or the tag can stand alone as the entire expression.

To supply a tag name, type it in the **Expressions** box, or click **Tags** to browse for and select a tag in the **Tag Browser**. Enclose tag names that contain dashes or start with a number in brackets {} when you use them in an expression.

Use brackets when using wildcard characters (\* or ?) to represent multiple tags in an expression. For more information about these built-in functions, see [Tag functions on page 354](#).

This wildcard character	Does this
?	Matches any single character.
*	Matches any number of characters, including the backslash (\) character.

For more information about working with tags, see [Working with tags on page 159](#).

## Specifying the area with a tag name

In a FactoryTalk View SE network distributed application, when referring to a tag in a different area, you can use an absolute or relative reference to the tag.

For more information, see [About tag references on page 164](#).

## Using tag placeholders to specify tag values

In the **Graphics** editor, you can use tag placeholders instead of tag names, to specify tag values in expressions.

A tag placeholder is the crosshatch character (#) followed by a number from 1 to 500. A tag placeholder can also contain wildcard characters and folder names.

At runtime, tag placeholders in a display are replaced with the values of tags specified in the parameter file or parameter list that is loaded with the display.

For more information about tag placeholders, see [Using placeholders to specify tag values on page 275](#).

## Using constants in expressions

A constant can be one of the following:

- Integer (for example, **123**)
- Floating point value (for example, **123.45**)
- string (for example, "**this is a character string**")
- **Pi** string (to represent the pi symbol)

FactoryTalk View replaces the string with its numeric value.

## Using operators in expressions

To modify the values returned to an expression, use arithmetic, relational, logical, or bitwise operators.

### Arithmetic operators

Arithmetic operators calculate values based on two or more numeric values.

Be sure that any tag value you use as a divisor cannot at some point have a value of zero. Expressions that attempt to divide a number by zero produce an error at runtime.

The following table lists the arithmetic operators.

Symbol	Operator	Example
+	addition	tag_1 + tag_2
-	subtraction	tag_1 - tag_2
*	multiplication	tag_1 * tag_2
/	division	tag_1 / tag_2
<b>MOD, %</b>	modulus (remainder)	tag_1 % tag_2 The modulus operator is the remainder of one number divided by another. For example, the remainder of 13 divided by 5 is 3; so 13 % 5 = 3 This operator is for integers only, not floating point numbers.
<b>**</b>	exponent	tag_1 ** 2

## String operands

The + operator can be used to join string operands. For example, the expression `hello + world` returns: `helloworld`.

## Relational operators

Relational operators compare two values to provide a true or false result. If the statement is True, a value of 1 is returned. If False, 0 is returned. There are six relational operators; each has two different symbols.

The following table lists the relational operators.

Symbol	Operator	Example
<b>EQ, ==</b>	equal	<code>tag_1 == tag_2</code>
<b>NE, &lt;&gt;</b>	not equal	<code>tag_1 &lt;&gt; tag_2</code>
<b>LT, &lt;</b>	less than	<code>tag_1 &lt; tag_2</code>
<b>GT, &gt;</b>	greater than	<code>tag_1 &gt; tag_2</code>
<b>LE, &lt;=</b>	less than or equal to	<code>tag_1 &lt;= tag_2</code>
<b>GE, &gt;=</b>	greater than or equal to	<code>tag_1 &gt;= tag_2</code>

### How string operands are evaluated

String operands are evaluated by case and by alphabetical order. Lower case letters are greater than upper case

letters. For example, `h` is greater than `H`. Letters later in the alphabet are greater than those earlier in the alphabet.

For example, `B` is greater than `A`.

## Logical operators

Logical operators determine the validity of one or more statements. The operators return a value of 1 (or another nonzero value) if the expression is True, or a value of 0 if it is False. Any statement that evaluates to a nonzero value is True.

The parentheses are essential in these expressions. For more information, see [Evaluation order of operators on page 350](#).

The following table lists the logical operators.

Symbol	Operator	Example
<b>AND, &amp;&amp;</b>	and	<code>(tag_1 &lt; tag_2) AND (tag_1 = 5)</code>
<b>OR,   </b>	or	<code>(tag_1 = 5) OR (tag_1 = 10)</code>
<b>NOT</b>	negation	<code>NOT(tag_1 &gt; tag_2)</code>

## Bitwise operators

Bitwise operators let you examine and manipulate individual bits within a value. These operators must only be applied to integers, not floating point numbers.

The following table lists the bitwise operators.

Symbol	Operator	Example
&	AND	tag_1 & 07
	inclusive OR	tag_2   tag_1
^	exclusive OR (XOR)	tag_1 ^ 01
>>	right shift	tag_1 >> 1
<<	left shift	tag_1 << 2
~	complement	~ tag_1

Example: Bitwise operators

For these examples **tag1** = 5 (binary 0000 0000 0000 0101) and **tag2** = 2 (binary 0000 0000 0000 0010)

`tag1 & tag2`

Returns 0 (binary 0000 0000 0000 0000)

`tag1 | tag2`

Returns 7 (binary 0000 0000 0000 0111)

`tag1 ^ tag2`

Returns 7 (binary 0000 0000 0000 0111)

`tag1 >> 1`

Returns 2 (binary 0000 0000 0000 0010)

`tag1 << 1`

Returns 10 (binary 0000 0000 0000 1010)

`~ tag1`

Returns -6 (binary 1111 1111 1111 1010)

## Evaluation order of operators

In expressions with more than one operator, the evaluation order is determined as follows:

- An operator in a statement enclosed in parentheses takes precedence over operators that are not.
- Operators of unequal precedence are evaluated according to precedence, from highest to lowest.
- Operators of the same precedence are evaluated from left to right.

Operators not enclosed in parentheses are evaluated as follows:

Precedence Level	Name	Symbols
1(Highest)	Parentheses	( )
2	logical negation bitwise complement	NOT ~
3	Multiplication	*

Precedence Level	Name	Symbols
	division modulus (remainder) exponent logical and bitwise and bitwise right shift bitwise left shift	/ MOD, % ** AND, && & >> <<
4	Addition subtraction logical or bitwise inclusive or bitwise exclusive or functions	+
		- OR,      ^
5 (lowest)	Equal not equal less than greater than less than or equal to greater than or equal to	EQ, == NE, <> LT, < GT, > LE, <=
		GE, >=

Examples: Evaluation order

In the following examples, **tag1** = 5, **tag2** = 7, and **tag3** = 10.

#### Example 1:

`(tag1 > tag2) AND (tag1 < tag3)`

is evaluated in this sequence:

1. `tag1 > tag2` = 0
2. `tag1 < tag3` = 1
3. `0 AND 1` = 0

The expression evaluates to 0 (false).

#### Example 2:

`tag1 > tag2 AND tag3`

is evaluated in this sequence:

1. `tag2 AND tag3` = 1
2. `tag1 > 1` = 1

The expression evaluates to 1 (true).

#### Example 3:

`NOT tag1 AND tag2 > tag3 ** 2`

is evaluated in this sequence:

1. NOT tag1 = 0
2. 0 AND tag2 = 0
3. tag3 \*\* 2 = 100
4. 0 > 100 = 0

The expression evaluates to 0 (false).

## Using built-in functions in expressions

The following sections describe the built-in functions that you can use in FactoryTalk View SE expressions.

Many of the built-in functions check for specific true and false conditions. If the condition is true, the expression returns a value of 1. If the condition is false, the expression returns a value of 0.

## Math functions

Use the built-in math functions to calculate the square root, log (natural or base 10), or trigonometry ratios (in radians or degrees) of a tag or expression.

The following table lists math functions:

This function	Returns this value
<b>SQRT(expression)</b>	The square root of the expression
<b>LOG(expression)</b>	The natural log of the expression
<b>LOG10(expression)</b>	The base ten log of the expression
<b>SIN(expression)</b>	The sine of the expression in radians
<b>COS(expression)</b>	The cosine of the expression in radians
<b>TAN(expression)</b>	The tangent of the expression in radians
<b>ARCSIN(expression)</b>	The arc sine of the expression in radians
<b>ARCCOS(expression)</b>	The arc cosine of the expression in radians
<b>ARCTAN(expression)</b>	The arc tangent of the expression in radians
<b>SIND(expression)</b>	The sine of the expression in degrees
<b>COSD(expression)</b>	The cosine of the expression in degrees
<b>TAND(expression)</b>	The tangent of the expression in degrees
<b>ARCSIND(expression)</b>	The arc sine of the expression in degrees
<b>ARCCOSD(expression)</b>	The arc cosine of the expression in degrees
<b>ARCTAND(expression)</b>	The arc tangent of the expression in degrees

## File functions

Use the built-in file functions to check whether a file exists, or to check the amount of free disk space.

The file parameter is the complete path name, surrounded by quotes. The drive parameter is the letter of the drive.

The following table lists file functions:

This function	Returns this value
<b>FILE_EXISTS("file")</b>	1(True) if the specified file exists.
<b>FREE_BYTES(drive)</b>	The number of bytes free on the specified drive. The maximum value the Free_Bytes function can return is the largest number a signed 32-bit value can provide. If Free_Bytes is used on a drive that has more than 2.1 GB, it will return 2.1 GB (approximately 2,147,483,647 bytes).

To have an expression containing these functions evaluated more than once, assign the expression to an event rather than to an object in a graphic display.

## Time functions

Use the built-in time functions to examine system time.

These functions use the time or interval parameters. The time and interval parameters must be enclosed in quotes.

The following table lists time functions:

This function	Returns this value
<b>TIME("time")</b>	1(True) if the time specified is the current time.
<b>BEFORE_TIME("time")</b>	1(True) if the expression is evaluated before the specified time.
<b>AFTER_TIME("time")</b>	1(True) if the expression is evaluated after the specified time.
<b>INTERVAL("interval")</b>	1(True) if the specified time interval has elapsed.

The *time* parameter can include the following options:

- Day of week[Sun, Mon, Tue, Wed, Thu, Fri, or Sat]
- Month[Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, or Dec]
- Date[1 to 31]
- Year[1997 to 2100]
- Hour of day[00: to 23:]
- Minute[:00 to :59]
- Second[:00 to :59]

It does not matter in what order the options are listed. You can include any or all of these options; the more you include, the more specific the time becomes.

### Example 1: Specific time parameters

The following all represent the same date and time, and are valid time parameters:

- "mon aug 22 2005 17:00"
- "mon aug 22 2005 17: :00"
- ":00 aug 22 mon 2005 17:"



**Tip:** The validity of the date is not checked. For example, if **Aug 22 2005** is not a Monday, this error is not detected.

## Example 2: Less specific time parameters

The following are valid examples of time parameters:

- "17:00" means any day at 5:00 ap.m.
- ".:30" means any hour, on the half hour
- "mon 17:" means 5:00 p.m. each Monday

The *interval* parameter has the format

`<number> <units>`

where *<units>* is one of:

- Mil (millisecond)
- Sec (second)
- Min (minute)
- Hou (hour)
- Day (day)
- Wee (week)
- Mon (month)
- Yea (year)

## Example 3: Time and interval functions

`TIME( "sun aug 21 2005 14:30" )`

If it is exactly 2:30 p.m. and 0 seconds, on Sunday, August 21, 2005, it returns 1(true); otherwise, it returns 0(false).

`AFTER_TIME( "sun aug 21 2005 14:30" )`

Returns 1(true) when the first time the expression is evaluated after 2:30 p.m. on Sunday, August 21, 2005.

`BEFORE_TIME( "aug 21 2005" )`

Returns 1(true) the first time the expression is evaluated before August 21, 2005.

`INTERVAL( "1 min" )`

Returns 1(true) if a minute has elapsed since the expression last returned a 1.

`(tag1 > 500) and INTERVAL( "30 sec" )`

Returns 1(true) when tag1 > 500 on some 30-second interval since the event file started running. (It does not mean 30 seconds after tag1 > 500.)

## Tag functions

Use the built-in tag functions to examine the status of a tag.

The following table lists tag functions:

This tag function	Returns this value
<b>COMM_ERR(tag)</b>	1(True) if a read or write operation for the specified tag produced a communication failure. You cannot examine the status of more than one tag with this function.

In a network distributed application, you can specify an absolute or a relative reference to a tag. For more information, see [About tag references on page 164](#).

To examine multiple tags at once, use a wildcard character in the expression argument.

This wildcard character	Does this
?	Matches any single character.
*	Matches any number of characters, including the backslash (\ ) character.

## Security functions

Use the built-in security functions to control access to an application, based on a user's identity or security permissions.

The following table lists security functions:

This function	Returns this value
<b>CurrentComputerHasGroup(GroupName)</b>	1(True) if the current computer is assigned to the specified group. The argument can be a literal string, string tag, or tag placeholder. If you use a string, it must be enclosed in double quotes, for example, CurrentComputerHasGroup("GroupName"). The argument can also be a combination of a literal string and string tag in the format of "Literal string" + String tag, for example, CurrentComputerHasGroup("GroupName" + [[Controller]Main.GroupName]). When using the function, note that: <ul style="list-style-type: none"><li>• The function only works in the Network Station or Network Distributed application. It always returns 1(True) if you use it in a Local Station application.</li><li>• In the FactoryTalk security policies, if you use the names of the terminal client to identify terminal server clients, make sure to add the terminal client account to the FactoryTalk network directory. Otherwise, it cannot get the current terminal client computer name and will return errors. For more information about FactoryTalk security policies, see FactoryTalk Help.</li></ul>
<b>CurrentUserHasCode(SecurityCode)</b>	1(True) if the current user has the specified security code. The security code is a letter from A through P.

This function	Returns this value
	The argument can be a security code, tag name, or tag placeholder. If you use a tag name or tag placeholder, it must be enclosed in curly brackets ({}), for example, CurrentUserCode({#1}).  If checking multiple security codes, do not type a space between the security code letters, for example, CurrentUserCode(ABP).
<b>CurrentUserHasGroup(<i>UserGroupName</i>)</b>	1(True) if the current user is assigned to the specified group.  The argument can be a literal string, string tag, or tag placeholder. If you use a string, it must be enclosed in double quotes, for example, CurrentUserHasGroup ("UserGroupName").  The argument can also be a combination of a literal string and string tag in the format of "Liter string" + String tag, for example, CurrentUserHasGroup ("UserGroupName"+[Controller]Main.UserGroupName).  <b>IMPORTANT:</b> <b>CurrentUserHasGroup</b> only works for Azure Active Directory (Azure AD) groups that are added to the FactoryTalk Directory, provided that the Azure AD group will not be updated on the Azure side. Anytime the Azure AD group is updated on Azure, you must add the group to the FactoryTalk Directory again for it to get the latest Azure AD group information.
<b>CurrentUserName( )</b>	The current user name.

## Language function

Use the built-in language function to return the name of the current runtime language. An expression containing this function is evaluated whenever a language switch occurs.

The following table lists language functions:

This function	Returns this value
<b>CurrentLanguage()</b>	RFC1766 code of the current runtime language.

## FactoryTalk alarm functions

Use the built-in FactoryTalk alarm functions to examine the status of one or more FactoryTalk alarms.

The following table lists Alarm and Event functions:

This function	Returns this value
<b>AE_ActiveEvents(<i>EventName</i>)</b>	A long integer from 0 to the number of server-based events that are active.
<b>AE_ActiveHighAlarms(<i>AlarmName</i>)</b>	A long integer from 0 to the number of FactoryTalk active, high-priority alarms that are <b>In Alarm</b> and <b>Acknowledged</b> or <b>In Alarm</b> and <b>Unacknowledged</b> .

This function	Returns this value
<b>AE_ActiveHighUnackedAlarms(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk active, high-priority alarms that are <b>In Alarm</b> and <b>Unacknowledged</b> .
<b>AE_ActiveLowAlarms(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk active, low-priority alarms that are <b>In Alarm</b> and <b>Acknowledged</b> or <b>In Alarm</b> and <b>Unacknowledged</b> .
<b>AE_ActiveLowUnackedAlarms(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk active, low-priority alarms that are <b>In Alarm</b> and <b>Unacknowledged</b> .
<b>AE_ActiveMediumAlarms(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk active, medium-priority alarms that are <b>In Alarm</b> and <b>Acknowledged</b> or <b>In Alarm</b> and <b>Unacknowledged</b> .
<b>AE_ActiveMediumUnackedAlarms(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk active, medium-priority alarms that are <b>In Alarms</b> and <b>Unacknowledged</b> .
<b>AE_ActiveUrgentAlarms(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk active, urgent-priority alarms that are <b>In Alarm</b> and <b>Acknowledged</b> or <b>In Alarm</b> and <b>Unacknowledged</b> .
<b>AE_ActiveUrgentUnackedAlarms(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk active, urgent-priority alarms that are <b>In Alarm</b> and <b>Unacknowledged</b> .
<b>AE_DisabledCount(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk alarms that are disabled.
<b>AE_FaultedAlarmsExist(Alarm\ngly)</b>	A Boolean representing whether any alarm exists in the <b>Faulted</b> state with the specified alarms, area path, or group path. Returns 0 if no alarm exists and 1 if one or more alarms exist.
<b>AE_HighPriorityActive(AlarmName)</b>	An integer from 1 to 4 representing the highest priority value associated with the specified alarms, whose states are <b>In Alarm</b> . <ul style="list-style-type: none"> <li>• 1: Low</li> <li>• 2: Medium</li> <li>• 3: High</li> <li>• 4: Urgent</li> </ul> If 0 is returned, it means that none of the specified alarms were <b>In Alarm</b> .
<b>AE_HighSeverityAcked(AlarmName)</b>	An integer from 1 to 1000 representing the highest severity value associated with the specified alarms, whose states are <b>In Alarm</b> and <b>Acknowledged</b> . If 0 is returned, it means that none of the specified alarms were both <b>In Alarm</b> and <b>Acknowledged</b> .
<b>AE_HighSeverityUnacked(AlarmName)</b>	An integer from 1 to 1000 representing the highest severity value associated with the specified alarms, whose states are <b>In Alarm</b> and <b>Unacknowledged</b> . If 0 is returned, it means that none of the specified alarms were both <b>In Alarm</b> and <b>Unacknowledged</b> .

This function	Returns this value
<b>AE_HistoryStatus(ServerName)</b>	An integer from 0 to 8 representing the connection status of the alarm history database. <ul style="list-style-type: none"><li>• 0: Initializing the connection to the database</li><li>• 1: Connected to the database</li><li>• 2: Disconnected to the database</li><li>• 3: Started logging to the database</li><li>• 4: Database definition deleted</li><li>• 5: Failed to log to the database</li><li>• 6: Stopping logging to the database</li><li>• 7: Stopped logging to the database</li><li>• 8: Alarm history not configured</li></ul>
<b>AE_InAlmAckedCount(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk alarms that are <b>In Alarm</b> and <b>Acknowledged</b> .
<b>AE_InAlmShelvedCount(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk alarms that are <b>In Alarm</b> and <b>Shelved</b> .
<b>AE_InAlmSuppressedCount(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk alarms that are <b>In Alarm</b> and <b>Suppressed</b> .
<b>AE_InAlmUnackedCount(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk alarms that are <b>In Alarm</b> and <b>Unacknowledged</b> .
<b>AE_NormalShelvedCount(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk alarms that are <b>Normal</b> and <b>Shelved</b> .
<b>AE_NormalSuppressedCount(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk alarms that are <b>Normal</b> and <b>Suppressed</b> .
<b>AE_NormalUnackedCount(AlarmName)</b>	A long integer from 0 to the number of FactoryTalk alarms that are <b>Normal</b> and <b>Unacknowledged</b> .
<b>AE_ShelvedAlarmsExist(AlarmName)</b>	A Boolean representing whether any alarm exists in the <b>Shelved</b> state with the specified alarms, area path, or group path. Returns 0 if no alarm exists and 1 if one or more alarms exist.
<b>AE_UnackedAlarmsExist(AlarmName)</b>	A Boolean representing whether any alarm exists in <b>In Alarm</b> and <b>Unacknowledged</b> state with the specified alarms, area path, or group path. Returns 0 if no alarm exists and 1 if one or more alarms exist.

You can specify an absolute or a relative reference to an alarm source.

To examine multiple alarms at once (a set of alarms for a machine in a particular area, for example) use a wildcard character in the expression argument.

Wildcard character	Description
?	Matches any single character.
*	Matches any number of characters, including the backslash (\ ) character.

For more information about using FactoryTalk alarm functions in expressions, and for examples, see [Setting up FactoryTalk alarms on page 177](#).

## Automatic Diagnostics functions

Use the Automatic Diagnostics functions to retrieve information about automatic diagnostic events from an individual device or multiple devices with similar names. These functions are available wherever expressions are supported in a FactoryTalk View SE application.

The following table lists Automatic Diagnostics functions.

This function	Returns this value
<b>ADDA_ActiveSuppressedCount(DevicePath)</b>	A long integer from 0 to the number of automatic diagnostic events that are <b>Active</b> and <b>Suppressed</b> .
<b>ADDA_ActiveUnsuppressedCount(DevicePath)</b>	A long integer from 0 to the number of automatic diagnostic events that are <b>Active</b> and <b>Unsuppressed</b> .
<b>ADDA_InactiveSuppressedCount(DevicePath)</b>	A long integer from 0 to the number of automatic diagnostic events that are <b>Inactive</b> and <b>Suppressed</b> .

To retrieve automatic diagnostic events from multiple devices at once (all devices in the device serve or in the same area as the display, for example) use a wildcard character in the expression argument.

Wildcard character	Description
?	Matches any single character.
*	Matches any number of characters, including the backslash (\ ) character.

## FactoryTalk alarm names and device path in expressions

To supply an alarm name or device path for an expression, type it in the **Expressions** box, or click the **Alarms** button to browse or areas, groups, alarm sources, device servers, or shortcuts.

For the *AlarmName* parameter, you can specify a single alarm name, or use the wildcard characters ? and \* (question mark and asterisk) to specify multiple alarms.

For the *DeviceName* parameter, you can specify a single device name, or use the wildcard characters ? (question mark) and \* (asterisk) to specify multiple devices.

You can also use tag placeholders in alarm names and device path, to substitute values for the placeholders at runtime. For more information about tag placeholders, see [Using placeholders to specify tag values on page 275](#).

## Redundancy functions

Use the redundancy functions to retrieve the server status in a redundant pair.

The status value is returned as a constant integer in decimal or hexadecimal. You can then use these values to reflect the different status of your servers. For example, you can create color animations or multistate indicators to show different colors, captions, or images.

The following table lists redundancy functions:

This function	Returns this value
<b>PrimaryServerStatus(ServerName)</b>	<p>Returns a constant integer that presents the status of the primary server.</p> <p>The argument is an absolute reference to a server. It can only be a literal string with the format <code>/AreaName/ServerName</code>, where <code>AreaName</code> is the name of the area and <code>ServerName</code> is the name of the server. The argument must be enclosed in double quotes, for example, <code>PrimaryServerStatus("/Area1/HMI_Demo")</code>.</p>
<b>SecondaryServerStatus(ServerName)</b>	<p>Returns a constant integer that presents the status of the secondary server.</p> <p>The argument is an absolute reference to a server. It can only be a literal string with the format <code>/AreaName/ServerName</code>, where <code>AreaName</code> is the name of the area and <code>ServerName</code> is the name of the server. The argument must be enclosed in double quotes, for example, <code>SecondaryServerStatus("/Area1/HMI_Demo")</code>.</p>

**Note:** In certain cases of a poor network or a large number of clients running at the same time, the returned constant may be empty or not reflect the true server status. To solve this problem, in FactoryTalk Administration Console, go to **System > Policies > System Policies > Health Monitoring Policy Properties** to increase the value of computer detection interval or network failure detection interval. For more information on these policies, see *FactoryTalk Services Platform Help*.

## ThinManager functions

Use the ThinManager functions to show status and monitoring information of the ThinManager terminal.



**Tip:** The TermMon ActiveX Control communicates to a ThinManager terminal over TCP port 2031. For the ThinManager functions to get the correct information, enable this port on both thin server and thin client.

This function	Returns this value
<b>TM_ConnectionState()</b>	<p>An integer from 1 through 3 representing the connection state to the ThinManager terminal.</p> <ul style="list-style-type: none"> <li>• 1: Failed Failed means that a network connection was established to the ThinManager terminal, but the connection was rejected. For more information, see the ThinManager user manual.</li> <li>• 2: Disconnected Disconnected means that the connection is not enabled or unable to establish a network connection to the ThinManager terminal.</li> </ul>

This function	Returns this value
	<ul style="list-style-type: none"> <li>• <b>3:</b> Connected Connected means that the connection to the ThinManager terminal is successful.</li> </ul>
TM_DisplayClientCurrent()	The ThinManager Display Client that is currently being displayed on the ThinManager terminal.
TM_RelevanceLocationName()	The full path and name of the current Relevance Location.
TM_RelevanceUsername()	The user name of the current Relevance User.
TM_TerminalIP()	The IP address of the ThinManager terminal.
TM_TerminalName()	The name of the ThinManager terminal.

## DataLogPro functions

Use the DataLogPro functions to show the active HMI server's connection state to the InfluxDB database.

The following table lists the DataLogPro functions.

This function	Returns this value
<b>DBConnectionState</b> ( <i>ServerName</i> )	<p>An integer representing the active HMI server's connection state to the InfluxDB database.</p> <ul style="list-style-type: none"> <li>• <b>0:</b> The connection to the database is lost.</li> <li>• <b>1:</b> The data is being logged to the database.</li> <li>• <b>2:</b> The data logging service has stopped.</li> <li>• <b>3:</b> The data logging has lost its required configuration, for example, the database is deleted.</li> </ul>

## Using if-then-else logic in expressions

Use the if-then-else logic in an expression to carry out an action conditionally, or to branch actions depending on the statements in the expression. You can also use if-then-else statements to repeat actions until a condition changes.



**Tip:** Do not confuse the if-then-else expression with the If command. For details about the If command, see the FactoryTalk View Site Edition Help.

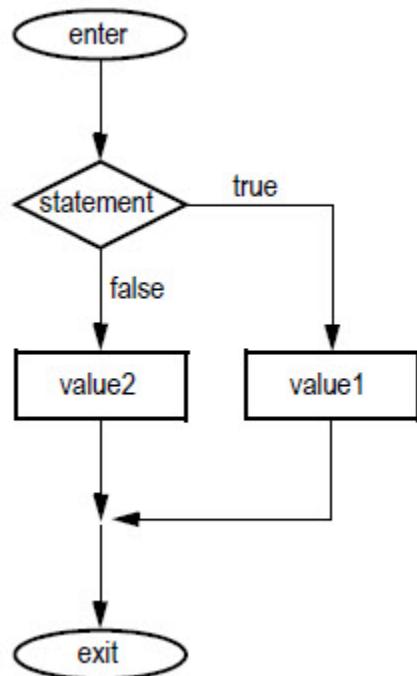
To build conditional expressions, use the relational operators and the logical operators for the statement, and values.

The if- then - else structure is:

```
If statement Then value1 Else value2
```

If the statement is true, then the expression returns *value1*; if the statement is false then the expression returns *value2*. Keep in mind that the *statement* is a mathematical equation and true means a nonzero value, and false means zero.

The following is an illustration of the if-then-else structure:



## Nested if - then - else

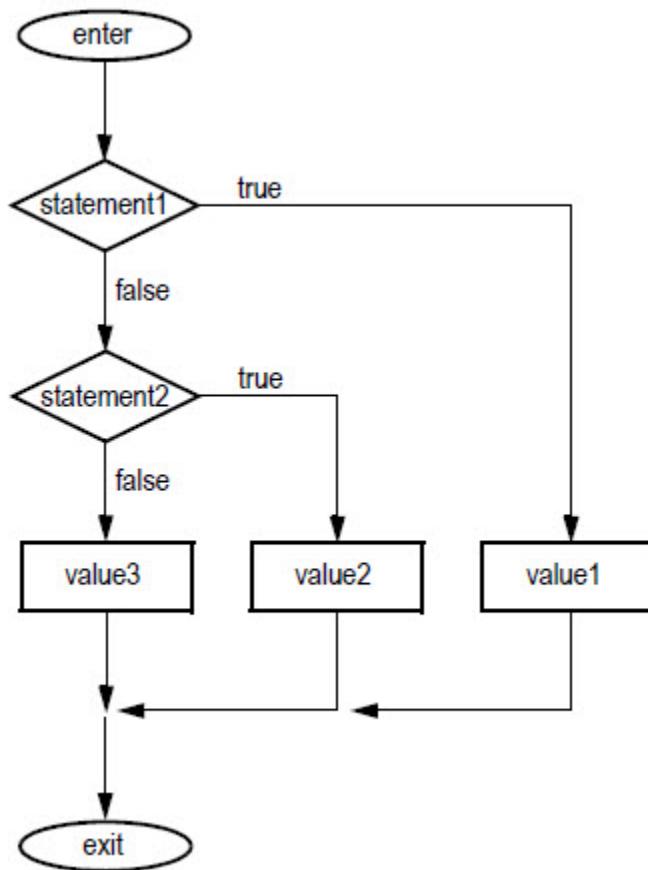
It is common to nest an if- then - else structure inside the then or else part of an if - then - else structure.

### Example 1: Nested if - then- else

This expression:

```
If statement1 Then value1  
Else if statement2 Then value2  
Else value3
```

is interpreted in this way:

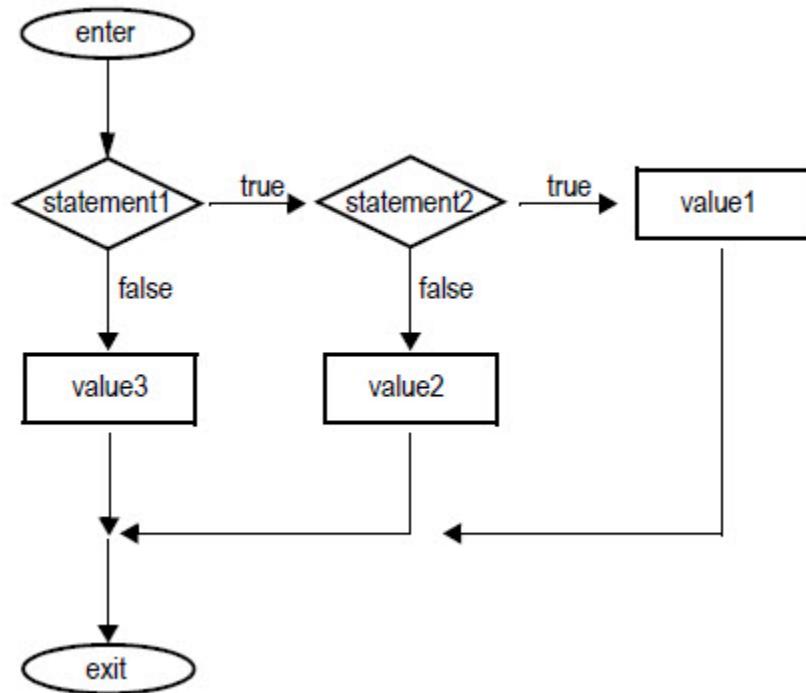


### Example 2: Nested if - then - else

This expression:

```
If statement1 Then  
  If statement2 Then value1  
    Else value2  
  Else value3
```

is interpreted in this way:



## Creating embedded variables

This chapter describes:

- What embedded variables are.
- Where you can insert embedded variables.
- How to create numeric embedded variables.
- How to create string embedded variables.
- How to create time and date embedded variables.
- How embedded variables are shown at runtime.
- How embedded variables are updated at runtime.

## About embedded variables

Use embedded variables to provide information in a graphic display that updates dynamically at runtime.

Embedded variables can contain the following elements:

- Tags
- Tag placeholders
- Literal numbers or strings
- The time and date

You can insert one or more embedded variables into the strings for captions, tool tips, display titles, local messages, and electronic signature's confirmation messages.

For example, if you embed a tag value and a time variable in a local message, when the local message is shown at runtime, it will show the tag's current value and the current time. The variables will update whenever the tag's value and the time change.

For more information about how embedded variables update at runtime, see [How embedded variables are updated at runtime on page 371](#).

You can also insert variables in the alarm messages you create for FactoryTalk alarm definitions. For more information, see [Setting up FactoryTalk alarms on page 177](#).

## Inserting embedded variables

You can insert one or more embedded variables in:

- Captions you create for graphic objects.

For objects with multiple states, you can insert different embedded variables in each state's caption. For information about creating specific graphic objects, see [Creating graphic objects on page 288](#).

- Local messages you create in the Local Messages editor.
- Tooltips you create for graphic objects.



**Tip:** Tooltips can consist of a single line of text, or multiple lines.

- Electronic signature's confirmation messages for graphic objects.

- Title bar text you create for graphic displays.



**Tip:** Embedded variables are case sensitive and must use specific syntax to work. Otherwise, the entire embedded variable is treated as a string of text.

To avoid this when inserting embedded variables, use the **Numeric Variable**, **String Variable**, and **Time and Date Variable** dialog boxes, instead of typing the variables manually. For more information about embedded variable syntax, see subsequent sections in this chapter.

## Creating numeric embedded variables

Insert numeric embedded variables to show analog or digital tag values or literal numbers.

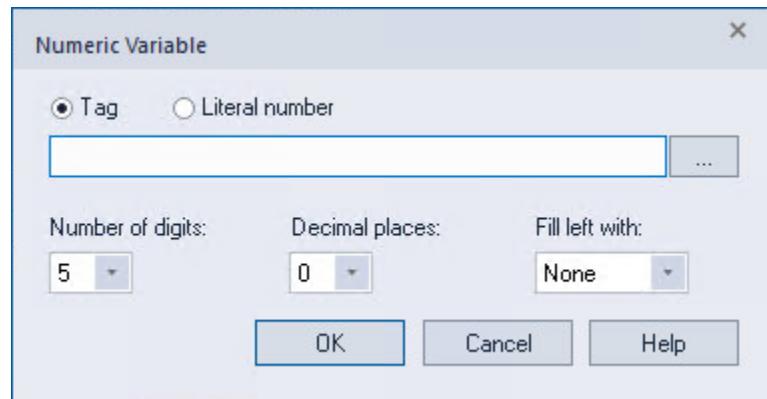
In the **Numeric Variable** dialog box, select the **Tag** option to show the value of the specified tag (or tag placeholder) in the embedded variable at runtime.

Select the **Literal number** option, to show the number you type in the dialog box, or the parameter value that replaces the tag placeholder.

Formatting options (**Number of digits**, **Decimal places**, **Fill left with**) apply to either a tag value or a literal number. For example, if you type the literal number 9999, and select 2 decimal places, at runtime the number will show as 99.99.

For details about options in the **Numeric Variable** dialog box, click **Help**.

For information about how embedded variables are updated at runtime, see [How embedded variables are updated at runtime on page 371](#).



## Syntax for numeric embedded variables that use a tag value

Numeric embedded variables that use a tag value use this syntax:

```
/*N:# Tag_name Fill_character DP:# */
```

where

- *N* indicates that it is a numeric embedded variable.
- *#* indicates the number of digits.
- *Tag\_name* specifies which numeric tag's value to display.

You can specify a tag placeholder instead of a tag name.

- *Fill\_character* is the fill character: NOFILL, ZEROFILL, or SPACEFILL.
- # indicates the number of decimal places.

### Example: Numeric embedded variable using a tag

At runtime, the numeric embedded variable

```
/*N:3 Oven_temp NOFILL DP:0 */
```

would show the current value of an analog tag called `Oven_temp`, with three digits, no decimal places, and no fill characters.

### Syntax for numeric embedded variables that use a literal number

Numeric embedded variables that use literal numbers use this syntax:

```
/*LN:# ### Fill_character DP:# */
```

where

- *L* indicates that the embedded variable will contain a literal number.
- *N* indicates that it is a numeric embedded variable.
- # indicates the number of digits.
- ### is the literal number (or tag placeholder).
- *Fill\_character* is the fill character: NOFILL, ZEROFILL, or SPACEFILL.
- # indicates the number of decimal places.

### Example: Numeric embedded variable using a literal number

At runtime, the numeric embedded variable

```
/*LN:3 9999 NOFILL DP:0 */
```

would show the number

```
9999
```

with no decimal places, and no fill characters.

## Creating string embedded variables

Insert string embedded variables to display string tag values or literal text strings.

In the **String Variable** dialog box, select the **Tag** option to show the value of the specified tag (or tag placeholder) in the embedded variable at runtime.

Select the **Literal string** option, to show the text string you type in the dialog box, or the parameter value that replaces the tag placeholder.

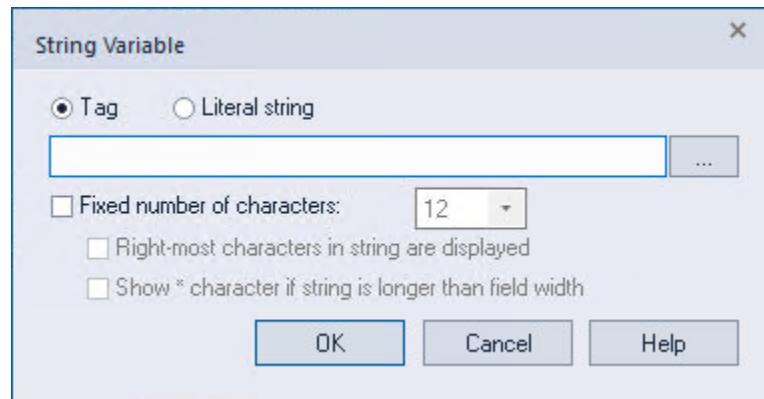


**Tip:** The literal string that you type (or specify in a parameter file) cannot contain spaces.

Formatting options for a fixed number of characters apply to either a tag value or a literal string. For example, if you type the literal string *September*, and select **3** for **Fixed number of characters** (but no other options), at runtime the string will show as *Sep*.

For details about options in the **String Variable** dialog box, click **Help**.

For information about how embedded variables are updated at runtime, see [How embedded variables are updated at runtime on page 371](#).



## Syntax for string embedded variables that use a tag value

String embedded variables that use a tag value use this syntax:

```
/*S:# Tag_name*/
```

where

- *S* indicates that it is a string embedded variable.
- *#* indicates the fixed number of characters to display. In the **String Variable** dialog box, if you don't want to use a fixed number, clear the check box, **Fixed number of characters**.
- *Tag\_name* specifies which string tag's value to show. You can specify a tag placeholder instead of a tag name.

### Example: String embedded variable using a tag

At runtime, the string embedded variable

```
/*S:20 Blower_status*/
```

would display the current value of the string tag named *Blower\_status*, with a fixed length of 20 characters.

## Syntax for string embedded variables that use a literal string

String embedded variables that use a text string use this syntax:

```
/*LS:# SSS*/
```

where

- *L* indicates that the embedded variable will contain a literal string.
- *S* indicates that it is a string embedded variable.

- # indicates the fixed number of characters to display. In the **String Variable** dialog box, if you don't want to use a fixed number, clear the check box, **Fixed number of characters**.
- SSS is the literal string (or tag placeholder).

### Example: String embedded variable using a literal string

At runtime, the string embedded variable

```
/*LS:0 September28*/
```

would show the text string

September28

with no fixed number of characters.



**Tip:** A literal string embedded variable can contain numbers. However, if you want to format the number (with decimal places, for example), insert a literal number instead. For details, see [Creating numeric embedded variables on page 366](#).

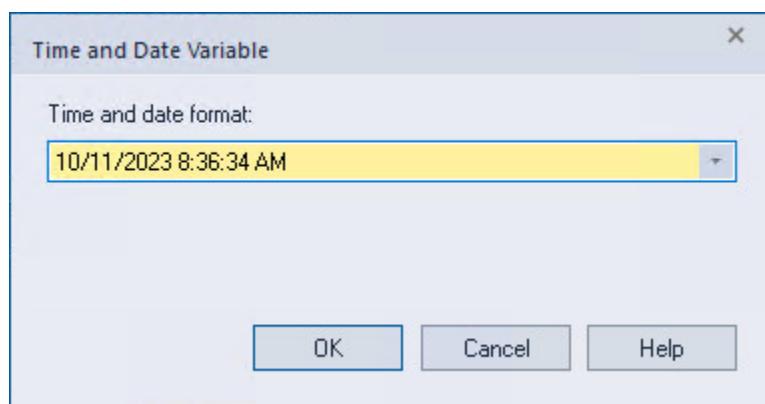
## Creating time and date embedded variables

Insert time and date embedded variables to show the current time and date.

In the **Time and Date Variable** dialog box, select the format, in which you want the current time and date shown at runtime.

For details about options in the **Time and Date Variable** dialog box, click **Help**.

For information about how embedded variables are updated at runtime, see [How embedded variables are updated at runtime on page 371](#).



## Syntax for time and date embedded variables

Time and date embedded variables use this syntax:

```
/*Time_date_format */
```

The following table shows the character sequences that *Time\_date\_format* uses to specify the format:

Character sequences	Description	Example
SD	Short date	11/16/15
LD	Long date	Monday, November 16, 2015
SDT	Short date and time	11/16/15 03:22:01 PM
LDT	Long date and time	Monday, November 16, 2015 03:22:01 PM
T	Time	03:22:01 PM
TSD	Time and short date	03:22:01 PM 11/16/15
TLD	Time and long date	03:22:01 PM Monday, November 16, 2015

### Example: Time and date embedded variable

At runtime, the time and date embedded variable

```
/*SDT*/
```

would show the short date followed by the time, for example:

```
9/28/2006 1:12:25 PM
```

A space is inserted between the time and date at runtime.

### How embedded variables are shown at runtime

If there is no valid data available for an embedded variable, it will show question marks (?).

This might occur when a display first opens and the data has not arrived yet, or when there is a problem that prevents communication with the data source.

### Numeric embedded variables

For numeric embedded variables that use tag values, how the value is shown depends on whether the data type is integer or floating point: Integer values are shown as they are; floating point values are rounded to fit the number of digits specified for the variable.

For example, if 6 is the number of digits specified, the floating point value 1234.56 is rounded to 1234.6, and the value 1234.44 is rounded to 1234.4. The decimal point counts as one of the digits. If the value is negative, the minus sign also counts as a digit.

If the floating point value, including a decimal point and a minus sign, contains more digits than specified for the variable, the numeric variable is replaced with asterisks (\*).

### Number formats

Numeric embedded variables use the number format that you set up on the runtime computer. For example, if the runtime computer is set up to use a comma for the decimal symbol, the numeric variable will also use a comma for the decimal symbol.

## String embedded variables

For string embedded variables that do not use a fixed number of characters, the entire tag value or literal string is shown, unless a null character (?) is read. Nothing after a null character is shown.

For string embedded variables that do use a fixed number of characters, the entire tag value or literal string is shown up to the number of characters specified, unless a null character (?) is read. Nothing after a null character is shown.

If necessary, spaces are used to make up the specified number of characters.



**Tip:** Null characters have a hex value of 0, and indicate the end of a string. A null character does not add to the actual length of the string.

By default, string variables that use a fixed number of characters show the characters from left to right. To show characters from right to left, in the **String Variable** dialog box, select the checkbox, **Right-most characters in string are shown**.

If the string is longer than the width of the field it is embedded in, you can also specify whether asterisks (\*) are shown at runtime.

## Time and date embedded variables

For embedded variables that show both the time and the date, a space is placed between the time and date when the embedded variable is shown at runtime.

### Time and date formats

Time and date embedded variables use the time and date formats that you set up on the runtime computer. For example, if the time and date variable shows the short date, at runtime the variable will use the short date format set up for the runtime computer.

## How embedded variables are updated at runtime

How an embedded variable is updated at runtime depends on the type of variable and where it is embedded:

- In caption text, title bar text, and local messages, numeric or string embedded variables that use tag values are updated whenever a new tag value is read from the data source.  
Embedded variables that use tags are also updated when the graphic display is loaded into the cache, using the **Always Updating** option. For information about setting up display caching, see [Setting up displays to open more quickly on page 284](#).
- In tooltips on graphic objects, numeric or string embedded variables that use tag values are updated only when the graphic display containing the object is opened. Embedded variables that use tags in tooltips will not update dynamically while the display is running.  
In the electronic signature's confirmation messages on graphic objects, numeric or string embedded variables that use tag values are updated only when the confirmation message containing the object shows. Embedded variables that use tags in the confirmation message will not update dynamically while the confirmation message is opened.
- Time and date embedded variables are updated as the system time and date change.

# Setting up data logging

This chapter describes:

- What data logging is.
- DataLogPro and Data Log Models.
- Data log storage formats.
- Creating a data log model.
- Working with ODBC data sources.
- Creating files at runtime.
- Switching log paths at runtime.
- Logging on demand.
- Modifying existing data log models.
- Starting and stopping data logging.

## About data logging

The HMI tag database does not store actual tag values. Instead, it defines which values are to be collected. The specified HMI server stores the actual values. When this HMI server is turned off, the value table is cleared. To store the historical tag values, use the FactoryTalk View components:

- DataLogPro
- Data Log Models

The two data logging components provide some similar functions, for example, historical records stored in both components can be viewed with trends or data grid objects, and both components provide periodical, on-change, and on-demand options to trigger logging.

In addition to the common functions, DataLogPro provides a redesigned editor with more intuitive configurations and has better data logging and accessing performance. Different from Data Log Models that stores historical data in file sets or an ODBC-compliant database, DataLogPro stores historical data in the built-in or external InfluxDB, which is more efficient to manage time series data.

## DataLogPro

In DataLogPro, you can drag multiple tags for logging, and specify log properties per tag or per group. By default, data logging starts automatically without having to trigger it by command. A log group or an individual tag can be configured to start logging periodically, on change, or on demand. You can drag an individual tag into a log group and that tag inherits the log properties of that group. The configuration of tags and groups can be exported and imported for batch editing.

The logged data is stored in a built-in InfluxDB or an external InfluxDB. The built-in InfluxDB comes pre-installed, and DataLogPro's historical data is stored in it by default. The external InfluxDB refers to an existing InfluxDB that you may want to use for storing DataLogPro's historical data. To use an external InfluxDB, make sure to create a database connection in **FactoryTalk View Studio > System > Connections > Databases**. You can view the data in data grid or TrendPro objects, or use third-party tools to analyze and create reports from the data.

**Automatic switch to built-in InfluxDB if external InfluxDB disconnected**

- If the HMI server loses connection to the external InfluxDB, it starts logging data to the built-in InfluxDB. The built-in InfluxDB uses the same storage duration as the external InfluxDB. Once the connection to the external InfluxDB is restored, all the data logged to the built-in InfluxDB will be synchronized to the external InfluxDB.

**HMI server redundancy support**

DataLogPro supports HMI server redundancy that helps prevent data loss.

- Using built-in InfluxDB only

In a redundant pair of HMI servers, DataLogPro logs data in both active and standby servers. If one server is disconnected, DataLogPro still logs data on that server as long as the DataLogPro service still runs. Only if the DataLogPro service stops, it stops logging data on that server. Once the DataLogPro service starts again, the data that is missing during that period is automatically synchronized between the two servers.

- Using external InfluxDB

In a redundant pair of HMI servers, DataLogPro on the active server logs data to the external InfluxDB while DataLogPro on the standby server logs data to the built-in InfluxDB. The data logged in the built-in InfluxDB by the standby server provides a one-minute cache of data. This cache supplements any missing data during the brief period of a switchover.

**DataLogPro editor**

In the **DataLogPro** editor, you can configure:

- Log groups and tag values to be logged.
- How each log group and individual tags' data logging is triggered.
- Data log storage duration.

**Start and stop logging with command**

DataLogPro starts logging automatically when the HMI server starts, but you can use the context menu or the **DataLogOn** and **DataLogOff** commands to start and stop data logging.

You can run this command from anywhere you can trigger the command, and the command is run on the HMI server side.

**Log historical data to an external InfluxDB**

If you want to store the historical data in an external, centralized InfluxDB, do the following in order:

- Create a database connection to the external InfluxDB.
- Configure the database connection within DataLogPro.

**Customize data logging location**

By default, the DataLogPro's historical data is stored in C:\Users\Public\Documents\RSView Enterprise\SE\...\DataLogProData\engine. If you want to have the data logged in a different folder, modify the file **Config.json** in C:\Program Files(x86)\Rockwell Software\RSView Enterprise\3rdParty\InfluxData\InfluxDB. Because the DataLogPro data storage service **Rockwell DataLogProPlus Service** is signed in as **Local Service**, the specified folder must be granted read and write permissions to the **Local Service** account.

**NOTE:** You must use slash (/), not backslash (\) in the path.

After customizing the folder, restart the **Rockwell DataLogProPlus Service** or your computer.

## Migrate data log model tags to DataLogPro

Migrate the legacy data log models to DataLogPro to take advantage of the DataLogPro functions and centralize the historical data in InfluxDB. The migration essentially means migrating the list of tags in the data log model to DataLogPro and these tags' historical data to InfluxDB. You can only migrate historical data stored in file sets or SQL Server.

To migrate the list of tags in the data log model to DataLogPro, you can add them one by one in DataLogPro or use a CSV file to add a number of tags at once. To migrate the tags' historical data to InfluxDB, the operations vary depending on whether the historical data is stored in file sets or SQL Server. However, unlike migrating historical data in file sets, which is supported by FactoryTalk View Studio's built-in function, migrating data in SQL Server requires knowledge of database schemas and Influx language programming.

To migrate historical data in file sets:

1. Add the list of tags to DataLogPro.
2. Migrate the historical data in file sets with the built-in **Migrate** function on the DataLogPro menu bar.

To migrate historical data in SQL Server:

1. Add the list of tags to DataLogPro.
2. Migrate the historical data in SQL Server with InfluxDB commands.

### Remarks

- When migrating historical data from data log models to DataLogPro:
  - New tags' historical data will be added to the InfluxDB database.
  - If the same tags already exist in DataLogPro or were firstly merged into the InfluxDB database during an earlier migration, their historical data will be merged.  
For example, if a tag's previous logging frequency is every 10 seconds, and the logging frequency of the same tag to be migrated is every 5 seconds, these additional logging data will be merged.
- If your application is a network distributed application, and you are logging to a local InfluxDB database, the historical data will be migrated to the active HMI server only. You must manually migrate the historical data to the standby HMI server by first switching to the standby HMI server, and then migrating the data to that server.

### Add the list of tags to DataLogPro

Tags in the data log model can be exported, and then imported into DataLogPro. This operation is a prerequisite if you want to migrate data log model's historical data to DataLogPro. You can add them one by one in DataLogPro or use a CSV file to add a number of tags at once.

To add tags with a CSV file, copy the tag list from the data log model's export CSV file, paste it to the DataLogPro's import CSV file, and then import it to DataLogPro.

When pasting the tag list to the DataLogPro's import CSV file, the CSV file content must be correctly formatted and contain the required fields. To ensure that the CSV file is intact and valid, we recommend that you create an export CSV file within DataLogPro, use that file to add the tag list and edit properties, and then import it back into DataLogPro.

Especially, the **Tag Type** column must be manually specified according to your data type in your device.

The valid types in the CSV file are BOOL, SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL, and STRING. The different types of tags in your device should fall into those types with the following matching rules. For example, if your tag type is INT8, specify its tag type as SINT in the CSV file.

- BOOL: Boolean
- SINT: INT8
- INT: INT16
- DINT: INT32
- LINT: INT64
- USINT: UINT8
- UINT: UINT16
- UDINT: UINT32
- ULINT: UINT64
- REAL: Float
- LREAL: Double
- STRING: String

## Migrate the historical data in file sets

If the historical data of data log models is stored in file sets, use the built-in **Migrate** function on the DataLogPro menu bar to migrate the historical data to InfluxDB. Make sure that you have added the list of tags to DataLogPro first.

## Migrate the historical data in SQL Server

If the historical data of data log models is stored in SQL Server, use InfluxDB commands and follow the database schema to migrate the historical data. To plot the data with FactoryTalk View SE controls, such as TrendPro or XY Plot, you must add the list of tags to DataLogPro first.

Before you begin, you must understand:

- SQL Server storage format.
- InfluxDB storage format.
- How data entries are constructed in InfluxDB.

## SQL Server storage format

Three tables are created in SQL Server for storing the data log model's historical data: Tag table, Float table, and String table. The string type data is stored in the String table while other types of data is stored in the Float table. Some of the tables' columns will be needed to establish the mapping with the InfluxDB storage format.

## InfluxDB storage format

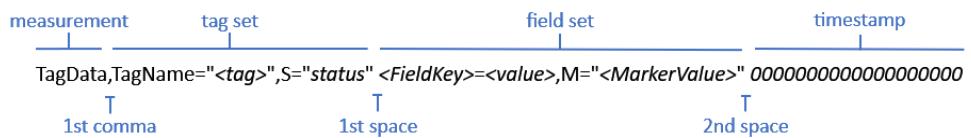
The InfluxDB database organizes data into buckets and measurements. The bucket name is the name of your HMI server. The InfluxDB's measurement is **TagData**, which includes **Tag** and **Field** elements. TagData, Tag and Field, and Tag and Field's keys will be needed to establish the data entry in InfluxDB.

## How data entries are constructed in InfluxDB

The construction of data entries follows the line protocol element parsing standard:

- Measurement
- Tag set
- Field set
- Timestamp
- Lines are separated by the newline character (`\n`). Line protocol is space sensitive.

For more information on the InfluxDB line protocol, see InfluxDB OSS documentation [Line protocol](#) and [Load data from sources in the InfluxDB user interface \(UI\)](#) or refer to your own InfluxDB version's documentation.



where

- **TagData**: The measurement in InfluxDB.
- **TagName**: The tag key in InfluxDB. In SQL Server, it's a column of the Tag table.
- **<tag>**: A tag's name. In SQL Server, it shows in the **TagName** column of the Tag table.
- **S**: The tag key in InfluxDB. It represents the **Status** column of the Float table or String table in SQL Server.
- **status**: The status (**C**, **D**, **E**, **S**, or **U**) in the **Status** column of the Float table or String table in SQL Server.
- **<FieldKey>**: The field key that you define for InfluxDB, such as **V\_B**, **V\_I**, **V\_D**, or **V\_S**, which represents Boolean, Integer, Double, or String value respectively.
- **<value>**: A tag's value. In SQL Server, it shows in the **Val** column of the Float table or String table in SQL Server. If **<FieldKey>** is **V\_I**, meaning the value is integer, then the value should follow this format **<value>i**. For example, **V\_I=8i**.
- **M**: The marker. In SQL Server, it's a column of the Float table or String table. If there is no marker, M and **<MarkerValue>** can be omitted.
- **<MarkerValue>**: The value (**B**, **E**, **S**, or blank) in the **Marker** column of the Float table or String table in SQL Server.
- **<Timestamp>**: The timestamp for a data point in Unix time in InfluxDB. To construct the timestamp, use the **DateAndTime** and **Millitm** columns of the Float table or String table in SQL Server.

### Example

```
TagData,TagName="system\Second",S="C" V_I=8i,M="B" 1465839830100400200
```

## Migrate historical data to InfluxDB

With the understanding of SQL Server and InfluxDB storage formats, use the `sql.from()` Influx function and script in the InfluxDB UI. To do so, visit **ComputerName:8086** in your browser, navigate to **Data Explorer**, enter your script (see example below), and then click **Run**. The script will read the data from SQL Server, transform the data to match the InfluxDB schema, and then write the data to InfluxDB. For more information, see [sql.from\(\) function](#).

Only users with administrative permissions to InfluxDB can perform the migration. For more information, see InfluxDB OSS documentation, [Create a token](#) or refer to your own InfluxDB version's documentation.

Overall, the migration of historical data includes migrating the tags' data and migrating the marker, which you can perform separately or together. You can migrate the data of multiple tags using one script as long as they are of the same type, which is Boolean (V\_B), Integer (V\_I), Double (V\_D), or String (V\_S). Also, you can migrate all the markers from one table, which is the Float table or the String table.

InfluxDB data type (field key)	Corresponding SQL Server table
Boolean (V_B), Integer (V_I), and Double (V_D)	Float table
String (V_S)	String table

The following is an example of migrating the data and marker together. Some of the parameters are necessary while others are editable. You can copy the script and modify the editable parameters per your own needs.

## Example

```

import "sql"

sql.from(
    driverName: "sqlserver",
    dataSourceName: "sqlserver://sa:password123@10.224.1.1:1433?database=mydb",
    query: "Select 'TagData' as _measurement,
FORMAT(DATEADD(SECOND,DATEDIFF(SECOND,GETDATE(),GETUTCDATE()),DATEADD(MILLISECOND,Millitm,CAST(DateA
ndTime as datetime2))), 'yyyy-MM-ddTHH:mm:ss.ffffZ') as _time, dbo.TagTableTagName as TagName,
COALESCE(NULLIF(dbo.FloatTable.Status,''),'C') as S,'V_I' as _field, dbo.FloatTable.Val as _value
        from dbo.TagTable
        Join dbo.FloatTable on dbo.FloatTable.TagIndex = dbo.TagTable.TagIndex
        where dbo.TagTableTagName = 'system\\Second''')

|> map(fn: (r) =>({r with _time: time(v:r._time)}))

|> map(fn: (r) =>({r with _value: int(v:r._value)}))

|>to(
    bucket: "HMIServer",
    org:"Rockwell")

sql.from(
    driverName: "sqlserver",
    dataSourceName: "sqlserver://sa:password123@10.224.1.1:1433?database=mydb",
    query: "Select 'TagData' as _measurement,
FORMAT(DATEADD(SECOND,DATEDIFF(SECOND,GETDATE(),GETUTCDATE()),DATEADD(MILLISECOND,Millitm,CAST(DateA
ndTime as datetime2))), 'yyyy-MM-ddTHH:mm:ss.ffffZ') as _time, dbo.TagTableTagName as TagName,
COALESCE(NULLIF(dbo.FloatTable.Status,''),'C') as S,'M' as _field, dbo.FloatTable.Marker as _value
        from dbo.TagTable
        Join dbo.FloatTable on dbo.FloatTable.TagIndex = dbo.TagTable.TagIndex
        where dbo.FloatTable.Marker <> '')

|> map(fn: (r) =>({r with _time: time(v:r._time)}))

|>to(
    bucket: "HMIServer",
    org:"MyOrganization")

```

### Explanation of the example

The first part of the script migrates the integer data (`system\Second`) in SQL Server database's `Float` table and the second part migrates all the markers in the `Float` table.

- **import "sql":** Imports the `sql` package. The `sql` package provides tools for working with data in SQL databases.
- **sql.from():** Retrieves data from a SQL data source. It has the following parameters:
  - **driverName:** Driver to use to connect to the SQL database. For SQL Server, it's `sqlserver`.
  - **dataSourceName:** Data source name (DNS) or connection string used to connect to the SQL database. It includes SQL Server's user name, password, host address, port, and database name, all of which should be edited according to your own needs.
  - **query:** Query executed on the SQL Server database.

The query in the first part of the script selects several fields from the `Tag` table and `Float` table where the **TagName** is `system\Second` and the field is `V_I`. The selected fields are then renamed to match the InfluxDB schema.

You should edit the sample script's Tag table name, Float or String table name, field key (**V\_B**, **V\_I**, **V\_D**, or **V\_S**), and tag name according to your own needs. You can add multiple tags of the same type separated with "**or**", for example: `dbo.TagTable.TagName = 'tag1' or dbo.TagTable.TagName = 'tag2'`.

The query in the second part of the scripts is similar to the first one, except that it selects the field **M** from the `Float` table.

You should edit the sample script's table type (Float table or String table) according to your own needs.

- **map:** Function to transform the `_time` field in each record from a string to a time value or transform the `_value` field in each record from a string to an integer.  
You should edit the sample script's value type (**int**, **bool**, **float**, or **string**) according to your own needs.
- **to:** Function to write the data to an InfluxDB bucket **HMIserver** in the organization **MyOrganization**.  
You should edit the sample script by entering your own bucket and organization. To migrate to the built-in InfluxDB, the organization name must be **Rockwell**; to migrate to an external InfluxDB, the organization name can be customized.

## InfluxDB

InfluxDB is an open-source time series database. FactoryTalk View SE installs InfluxDB for storing the logged data by DataLogPro. The installed version is InfluxDB OSS 2.7. The historical data in InfluxDB is stored in `C:\Users\Public\Documents\RSView Enterprise\SE\DataLogProData`.

Use the Influx command line interface (CLI) to interact with and manage your InfluxDB, such as creating users, assigning a token to users, and so on. The command line interface (CLI) is in `C:\Program Files (x86)\Rockwell Software\RSView Enterprise\3rdParty\InfluxData\InfluxCli`.

Use the InfluxDB UI to manage and access data from your InfluxDB. To access the UI, with InfluxDB running, visit **ComputerName:8086** in your browser. If it's on your current computer, visit **localhost:8086**.

## InfluxDB storage format and schemas

An organization is a workspace for a group of users. All dashboards, tasks, buckets, members, and so on, belong to an organization. The default organization in the built-in InfluxDB is **Rockwell**, which cannot be changed. If you have an external InfluxDB, you can use its existing organization.

The InfluxDB data model organizes time series data into buckets and measurements. A bucket is a named location where time series data is stored and it can contain multiple measurements. The bucket name is the name of your HMI server. To view your bucket list, use the command `influx bucket list`. You can also view the bucket list and historical logs with the InfluxDB UI. Measurements contain multiple tags and fields.

For more information, see InfluxDB OSS documentation, [Line protocol](#) and [InfluxDB schema design](#) or refer to your own InfluxDB version's documentation.

Measurement	InfluxDB Data Element	Key or Column	Description
TagData	Tag	TagName	The name of the tag in the FactoryTalk system.
		S	Status: <ul style="list-style-type: none"> <li>• <b>C:</b> Common</li> <li>• <b>D:</b> Tag disabled</li> <li>• <b>E:</b> Error</li> <li>• <b>S:</b> Stale data</li> <li>• <b>U:</b> Uninitialized data</li> </ul>
	Field	V_B	Boolean value
		V_I	Integer value The integers are logged in InfluxDB with a value range of INT64. FactoryTalk View tags of the integer type such as ULINT, UDINT, and so on are logged with this type, adhering to the INT64 value range.
		V_D	Double value FactoryTalk View tags of the REAL and LREAL types are logged in InfluxDB with this type.
		V_S	String value
		M	Marker value: <ul style="list-style-type: none"> <li>• <b>B:</b> Begin</li> <li>• <b>E:</b> End</li> <li>• <b>S:</b> Snapshot</li> <li>• Blank: Common data</li> </ul>

The following measurements are used for data processing purposes. Do not modify them.

- **LastRecordInfo:** Records the time of the last record.
- **SyncTask:** Records the status of sync tasks.
- **SnapshotTask:** Records the status of snapshot tasks.

## Set up InfluxDB user and permission

To access and manage InfluxDB data for your own purposes, create extra users and API tokens for specific users that identify InfluxDB permissions within the user's organization.

To grant the user permission to access data, add them as a member of an organization and provide them with an API token. Only the user with the administrative right that installed FactoryTalk View can perform these tasks. Use the Influx command line interface (CLI), which is in C:\Program Files (x86)\Rockwell Software\RSView Enterprise\3rdParty\InfluxData\InfluxCli.

For more information, see InfluxDB OSS documentation, [Create a user](#) and [Create a token](#) or refer to your own InfluxDB version's documentation.

### To set up InfluxDB user and permission

1. Create a user with the command line.

The following command will create a user in the organization.

```
influx user create -n johndoe -p password -o example-org, where example-org is  
the organization for data logging. The default is Rockwell.
```

2. Provide the user with a token.

The following command will assign permissions to the user.

```
influx auth create --org example-org --user johndoe --[permission flag]
```

For example, to grant a user all permissions in the Rockwell organization, the command is as below:

```
influx auth create --org Rockwell --user johndoe --all-access
```

For more information on the available flags, see InfluxDB OSS documentation, [Flags](#) or refer to your own InfluxDB version's documentation.

## Delete InfluxDB data

To delete data from an InfluxDB bucket, you can use the Influx command line interface (CLI), which is in C:\Program Files (x86)\Rockwell Software\RSView Enterprise\3rdParty\InfluxData\InfluxCli. The default bucket is the name of your HMI server name. To view your bucket list, use the command `influx bucket list`.

This section introduces an example to delete all points in a specified time range. For more information, see InfluxDB OSS documentation, [Delete data](#) or refer to your own InfluxDB version's documentation.

### To delete all points in a specified time range

- Use the following command:

```
influx delete --bucket example-bucket --start 2023-10-01T00:00:00Z --stop  
2023-10-02T00:00:00Z, where example-bucket is the name of your HMI server name, and the time  
range follows the format of year-month-dateThour:minute:secondZ.
```

---

**IMPORTANT:** The time stamp for the logging data is Coordinated Universal Time (UTC).

Therefore, when specifying the start and stop time range, make sure to use the UTC time.

---

## Migrate InfluxDB data

Logging data by DataLogPro can be migrated from the built-in InfluxDB to other InfluxDB instances for your own centralized management or analysis.

For more information, see InfluxDB OSS documentation, [Migrate data from InfluxDB OSS to other InfluxDB instances](#) or refer to your own InfluxDB version's documentation.

## Back up and restore InfluxDB data

Backing up or restoring an HMI server doesn't include the DataLogPro logging data stored in InfluxDB. To have those data moved from one production environment to another, you must manually back up and store the InfluxDB data.

The historical data in InfluxDB is stored in C:\Users\Public\Documents\RSView Enterprise\SE\DataLogProData. To back up and store the data, use the InfluxDB commands. During the operation, you can back up data from and restore to a specific bucket. To view your bucket list, use the command `influx bucket list`.

For more information, see InfluxDB OSS documentation, [Back up and restore data](#) or refer to your own InfluxDB version's documentation.

## Data Log Models

Data log is a FactoryTalk View component that collects and stores tag values. Logged data can be stored in an internal file set, or an ODBC-compliant database. The logged data can be shown in trend or on a data grid object, or archived for future use.

### Gathering tag data in data log models

A data log model defines which tags to log data for, when to log the data, and where to log the data.

In the **Data Log Models** editor, you also specify the format of the log files (file set, or ODBC), and when to create and delete the files (also called records).

A single data log model can log data for up to 10,000 tags.

### Using multiple data log models

At runtime, up to 20 data log models can run simultaneously on each FactoryTalk View SE Server.

Use multiple data log models to:

- Store-related information in separate file sets.
- Log groups of tags at different rates.
- Log groups of tags based on events.

### Data log storage formats

Each data log file set includes files for storing numeric values (either integers or floating point values), text, and tag names.

Internal file sets store tag values in proprietary-format files. You can view the contents of the file sets in trends, on a data grid object, or in FactoryTalk View File Viewer.

If you log tag values to an ODBC database, you can use third-party, ODBC-compliant tools to analyze and create reports from the data. If the ODBC database becomes unavailable, FactoryTalk View logs the data to backup files in proprietary format.

To use data in multiple formats, define multiple data log models for the same set of tags.

## How log file sets are named

When you create a log file set, the set is given a name automatically. The name is based on the date that the file was created, and the type of data it contains. The maximum length of a log file set's name, including its path, is 200 characters.

File sets use the following file name format:

```
YYYY MM DD NNNN <Log File Identifier String> <(type)>.dat
```

where

- *YYYY* is the year.
- *MM* is the month.
- *DD* is the day.
- *NNNN* is the sequential file identifier.

This number indicates that the sequence files were created in. The number can be up to 9999 file sets per day. At midnight, the sequence starts again at 0000.

- *<Log File Identifier String>* is a text string that you can specify to help identify the log file. The maximum string length is 20 characters.
- *<(type)>* is the file type.

The type is enclosed in parentheses. There are three file types: Tagname, Float (for analog and digital tag values), and String.

### Example: Parts of a data log long file name

The data log file named

```
2006 10 30 0004 Oven Temperatures (Float).dat
```

was created October 30 2006:

- *0004* indicates that this is the fourth file set created that day.
- *Oven Temperatures* is the log file identifier string, which the user defined to help identify the data.
- *(Float)* indicates that this file contains analog or digital tag values.

## ODBC database storage format and schemas

Use the ODBC database format to store data using an ODBC data source such as Microsoft SQL Server (fully tested), Microsoft Access, or Oracle.

The ODBC format stores data in up to three tables:

- **Float table** stores analog and digital tag values.
- **String table** (optional) stores string tag values.
- **Tag table** (optional) stores tag names in an index so that they can be referenced using a 2-byte or 4-byte numeric field (rather than a 40-byte character field) in the float and string tables.



**Tip:** ODBC backup files are stored as binary files with the extension .obf. You cannot view the contents of .obf files.

## How ODBC tables are named

The ODBC tables are created with the default names *TagTable*, *FloatTable*, and *StringTable*. In the **Data Log Models** editor, you can modify these names before creating the tables, or you can specify the names of different tables.

If you specify a backup path for an ODBC model, FactoryTalk View names the backup log files as described on [How log file sets are named on page 382](#).

## Format for ODBC float and string tables

Floating point and string data stored in ODBC format uses the following table: the float table contains analog and digital tag data; the string table contains string tag data.

This column	Contains	SQL data type	Length
DateAndTime	The date and time the tag values were logged.	SQL_DATETIME	Driver dependent
Millitm	The millisecond time the tag values were logged.	SQL_SMALLINT, or SQL_INTEGER	4
TagIndex, or TagName	The index number for the tag. The tag name that corresponds to this number is listed in the tag name table. If the tag name table is not used, the tag name appears in this column	SQL_SMALLINT, or SQL_INTEGER SQL_VARCHAR, or SQL_CHAR	2 or 4 255
Val	The analog or digital value of the tag.	SQL_DOUBLE, or SQL_INTEGER, or SQL_SMALLINT	8, or 4, or 2
	The string value of the tag.	SQL_VARCHAR, or SQL_CHAR	82
Status	Blank if communications are working properly.	SQL_VARCHAR, or SQL_CHAR	1

This column	Contains	SQL data type	Length
	E if a communication error occurred while FactoryTalk View was trying to acquire the tag value. S if the tag value is stale; that is, if the value has been acquired before, but has not been updated. U if the tag is uninitialized.		
Marker	Internal tag marker.	SQL_VARCHAR, or SQL_CHAR	

## Format for ODBC tag table

The following table shows the format for the ODBC tag table.

This column	Contains	SQL data type	Length
TagName	The tag name.	SQL_VARCHAR, or SQL_CHAR	255
TagIndex	The index number assigned to the tag name.	SQL_SMALLINT, or SQL_INTEGER	2 or 4
TagType	The tag type: 2 for Analog tag 3 for Digital tag 4 for String tag	SQL_SMALLINT	1
TagDataType	The tag data type: -1 for Integer 0 for Long 1 for Float 2 for String	SQL_SMALLINT	2

## Creating data log models

In the **Data Log Models** editor, you can create data log models that specify:

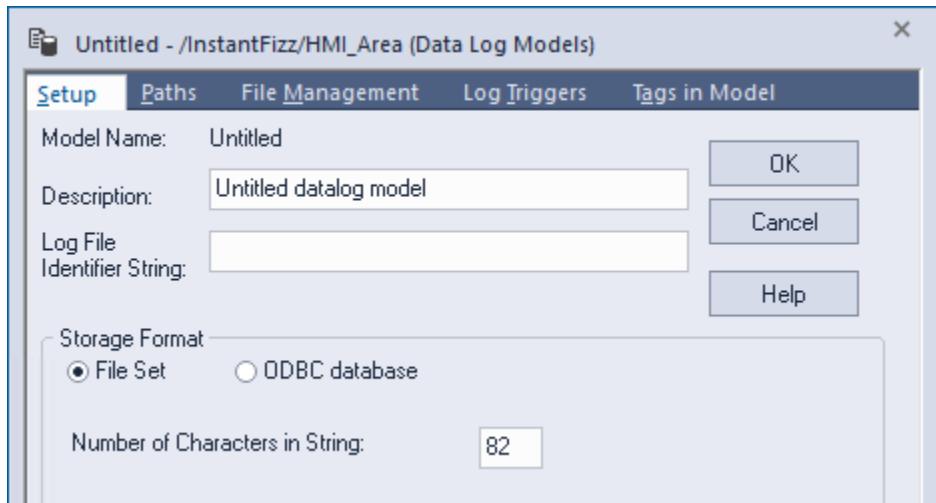
- Which log file format to use.
- Where to store data log file sets (primary or secondary paths).
- When to create and delete log files.
- Which actions will trigger logging.
- Which tag values will be logged.

## To open the Data Log Models editor:

1. In FactoryTalk View Studio, in the **Explorer** window, open the **Data Log** folder.
2. Right-click the **Data Log Models** icon, and then select **New**.



**Tip:** You can also open the **Data Log Models** editor by dragging its icon into the FactoryTalk View Studio workspace, or by double-clicking an existing Data Log Models component.



For details about options in the Data Log Models editor, click **Help**.

## Specifying the storage format

In the Data Log Models editor, use the **Setup** tab to specify a log file identifier string, storage format, and description for the data log model.

The log file identifier string becomes part of the file name. The description is for information only.

If you select **File Set** as the storage format and you plan to log string tags, you can also specify the number of characters to be logged.

For details about options in the **Setup** tab, click **Help**.

## Selecting the ODBC database format

When you select the ODBC database as the storage format, type the path and file name of the ODBC data source, and specify the ODBC tables that will be used, or create tables.

To create tables for a new data source, click **Create Tables**. If FactoryTalk View cannot create the tables automatically, you will have to open the database using its editing tool, and create the tables manually.

To select an existing table from the specified database, click the **Browse** button, and then select a table in the **Select ODBC Table** dialog box. To view the order, type, length, and precision of the fields in the table, click the table to select it and then click **Details**.

For more information, see [Logging data to an existing ODBC data source on page 387](#).

## Setting up log paths

In the **Data Log Models** editor, use the **Paths** tab to set up logging paths.

If the storage format is file set, specify the primary path, and an optional secondary file path in case the primary path becomes unavailable. If the storage format is ODBC, you can specify a backup file path for the ODBC database.

Setting up a secondary or backup logging path prevents loss of data. For example, if the disk space for the primary path becomes full, data logging will switch to the secondary path, if available, until space becomes available on the primary path. For more information, see [Switching log paths at runtime on page 389](#).

If you do not set up a secondary path, when the primary path runs out of disk space, data logging stops and no more log files are created.

**IMPORTANT:** If you specify a customized path, add the user **LOCAL SERVICE** to the security of that folder on both the primary and secondary HMI servers (or any other computer that folder resides on) and give the user full control. For more instructions, see Knowledgebase Document ID: [BF22544 - HMI Server Stuck at Starting or Not Loaded](#).

For details about options in the Paths tab, click **Help**.

## Setting up and managing data log files

In the **Data Log Models** editor, use the **File Management** tab to specify when to start new files and when to delete old files.

## Creating file sets

If the data log model uses file sets, you can set up data logging to create file sets:

- Periodically.
- At specified times.
- When a particular event occurs.
- Never. This means that all data will be logged to one file set.

Log files are saved in the folder that you specify in the Paths tab.



**Tip:** If the data log model uses the ODBC format, data is added continually to the same database, so this section does not apply.

## Deleting file sets and ODBC database records

If a data log model uses file sets, you can set up file management to delete file sets after a specified period or once a specified number of file sets has been created. To never delete file sets, clear the checkboxes under **Delete Oldest Files**.

Data log file sets are deleted only when a new file set is created. If the application creates a file set each day and deletes the oldest file set every third day, there will be file sets for the three previous days' data, as well as a file set for the current day.

If a data log model uses the ODBC database format, you can purge old records from the database using standard relational database tools or SQL queries. You can also set up FactoryTalk View to delete (purge) records in the ODBC database after a specified time.

For details about options in the File Management tab, click **Help**.

## Specifying when to log data

In the **Data Log Models** editor, use the **Log Triggers** tab to specify what will trigger the data to be logged, and when or how frequently this will occur.

You can set up data logging to log tag values:

- Periodically. This is called periodic logging.
- Only when a tag's value changes. This is called on-change logging.
- When a particular event occurs and triggers the **DataLogSnapshot** command. This is called on-demand logging. For more information, see [Using the DataLogSnapshot command on page 391](#).
- You can also combine types of logging. For more information, see [Combining logging on page 391](#).

For details about options in the **Log Triggers** tab, click **Help**.

## Choosing the data to be logged

In the **Data Log Models** editor, use the **Tags in Model** tab to specify which tags will be scanned for data. A data log model can contain up to 10,000 tags.

For details about options in the Tags in Model tab, click **Help**.

## Logging data to ODBC data sources

You can store logged data in an existing ODBC data source, or you can create an ODBC data source.

### Logging data to an existing ODBC data source

You can log data to an existing ODBC data source, but its tables must contain fields that are formatted for the FactoryTalk View data log.

There are two ways to prepare tables for data logging to an existing database:

- Set up the FactoryTalk View data log fields in the existing ODBC tables before setting up a data log model.
- Create tables in the existing database automatically or manually, from within the **Data Log Models** editor.

#### To use the tables in an existing ODBC data source:

1. Add fields for the data log data to the tables in the ODBC database.

For information about modifying the ODBC database tables, see the documentation for the ODBC database you are using.

2. Set up a data log model, as described in Help. Specify the existing ODBC database as the ODBC Data Source, and then specify the tables you edited.

## To create tables in an existing ODBC data source:

1. Specify the name of the existing ODBC data source in the **ODBC Data Source** field of the **Data Log Models** editor.
2. Click **Create Tables**. FactoryTalk View automatically creates data log tables in the existing database. You can also create tables manually.
3. Set up the rest of the data log model, as described in Help.

## Creating an ODBC data source

To create an ODBC data source, click the browse (...) button next to the **ODBC Data Source** text box, or use the ODBC Data Source Administrator in the **Windows Control Panel**. Click **Help** in the **ODBC Data Source Administrator** window for more details about creating an ODBC data source.

You can create the tables for a new data source using either of the methods described in [Logging data to an existing ODBC data source on page 387](#).

## Setting up security to log data to a remote computer

If you are logging data to an ODBC database that is not on the computer where the data is being generated, you must change certain DCOM security settings. For details, see the FactoryTalk View Site Edition Help.

## Creating files at runtime

To create file sets or ODBC backup files at runtime, use the FactoryTalk View command **DataLogNewFile**.

If a data log model uses the ODBC format, and if FactoryTalk View is logging to the backup path, running the **DataLogNewFile** command creates a set of backup files.

If FactoryTalk View is logging to the ODBC database, an End Snapshot and then a Begin Snapshot are logged when this command is run.

If a data log model uses file sets, **DataLogNewFile** creates a file set regardless of how the data log model is set up to start new files. The new file set is created in the same location that FactoryTalk View is currently logging to.

**DataLogNewFile** records two snapshots of data: an End Record in the old file, and a Begin Record in the new file. It is not necessary to use the **DataLogSnapshot** command in this case.

## Using the **DataLogNewFile** command

You can use the **DataLogNewFile** command with either of these parameters:

- **DataLogNewFile <component>** (where <component> is a data log model name) creates a file set for the specified data log model.
- **DataLogNewFile \*** creates a file set for all data log models that are currently running.

You can use the **DataLogNewFile** command anywhere you can type a FactoryTalk View command or macro.

To provide operators with a way to create files at runtime, create a button object in a graphic display, or a display key or client key that uses **DataLogNewFile** as the press action.

## Switching log paths at runtime

FactoryTalk View lets you specify a secondary or backup path to be used in the event the primary path for file sets, or the ODBC database, becomes unavailable. This could happen due to a network failure, or lack of disk space on the computer where the data is being logged.

If the primary data log location becomes unavailable, FactoryTalk View begins to store the data in a buffer.

The buffer can hold up to 64 Kb of data. If the primary location is still unavailable when the buffer fills, or when the maximum amount of time to buffer data has elapsed, FactoryTalk View switches to the secondary path.

FactoryTalk View checks periodically to determine whether the primary file path has become available again. Once it is available, FactoryTalk View switches back automatically.

If the secondary path becomes unavailable, FactoryTalk View also checks the status of the primary path, and will switch back if possible. If both paths are unavailable, FactoryTalk View buffers the data.

If the buffer fills and both paths are still unavailable, FactoryTalk View empties the buffer (the data in the buffer is lost) and begins storing new data in the buffer. FactoryTalk View continues checking both paths until one becomes available.



**Tip:** For data log models that use the file set format, FactoryTalk View creates a file set each time the logging path changes.

## Setting up switching options

In the **Advance Configuration** dialog box, you can specify the length of time data will be buffered before switching to the secondary path, and when data logging will switch back to the primary path.

For example, if the data log file is locked, data is buffered for the maximum time (in minutes) you specify to buffer data before attempting a switchover. When the buffer time elapses, a new set of files is created on the primary path.

If the model is logging to the secondary path and the file is locked, the behavior is the same: the data is buffered, and then a new file set is created when the specified time period has elapsed.

To open the **Advance Configuration** dialog box, in the **Data Log Models** editor, select the **Paths** tab, and then click **Advanced**. For details about options in the Paths dialog box, click **Help**.



**Tip:** If a secondary path is not set up, the data is buffered for 10 minutes (the default value for maximum time). If the maximum time is set to zero, a new file set is started immediately.

## Switching back manually to the primary path

To switch back to the primary path or ODBC database manually, use the FactoryTalk View command **DataLogSwitchBack**.

You can only use the command to switch back to the primary path or database. You cannot use it to switch from the primary path or ODBC database to the secondary or backup path.

The switchback is performed only if the specified data log model is running. FactoryTalk View is logging data to the secondary or backup path, and the primary path or ODBC database is available.

For data log models that use file sets, FactoryTalk View creates a set of files when it switches back to the primary path.



**Tip:** Data logged to the secondary path is not moved automatically, when a switchback to the primary occurs. To move the data manually, use the **DataLogMergeToPrimary** command. For details, see [Moving data from the secondary path on page 390](#).

## Using the **DataLogSwitchBack** command

You can use the **DataLogSwitchBack** command with either of these parameters:

- **DataLogSwitchBack <component>** (where <component> is a data log model name) switches logging for the specified data log model.
- **DataLogSwitchBack \*** switches logging for all data log models that are currently running.

You can use the **DataLogSwitchBack** command anywhere you can type a FactoryTalk View command or macro.

To provide operators with a way to switch manually to the primary path or ODBC database at runtime, create a button object in a graphic display, or a display key or client key that uses DataLogSwitchBack as the press action.

## Moving data from the secondary path

When you begin writing to or reading from a data log model, if there are files in the secondary (or backup) path, FactoryTalk View sends a message to the FactoryTalk Diagnostics log.

You must move these files manually to the primary path or ODBC database. To do this, use the FactoryTalk View command **DataLogMergeToPrimary**.

If a data log model is running when the command is run, FactoryTalk View performs a switchback to the primary path or ODBC database for the specified model, or all running models.

If the model uses file sets, FactoryTalk View moves all files on the secondary path (including the current file set) to the primary path, begins a new file set on the primary path, and then continues logging to the new file set.

If the model uses the ODBC format, FactoryTalk View merges the data in the ODBC backup files into the ODBC database, and then continues logging to the ODBC database.

## Using the **DataLogMergeToPrimary** command

You can use the **DataLogMergeToPrimary** command with either of these parameters:

- **DataLogMergeToPrimary <component>** (where <component> is a data log model name) moves data for the specified data log model, no matter the model is running or not.
- **DataLogMergeToPrimary \*** moves data for all data log models, no matter they are running or not.

You can use the **DataLogMergeToPrimary** command anywhere you can type a FactoryTalk View command or macro.

To provide operators with a way to restore backup data to the primary path or ODBC database at runtime, create a button object in a graphic display, or a display key or client key that uses DataLogMergeToPrimary as the press action.

## Logging on demand

To record the values of tags in a data log model on demand, use the FactoryTalk View command **DataLogSnapshot**. If you specified the **On Demand** trigger for the data log model, logging will occur only when the **DataLogSnapshot** command is run.

## Using the DataLogSnapshot command

You can use the **DataLogSnapshot** command with either of these parameters:

- **DataLogSnapshot <component>** (where <component> is a data log model name) records tag values for all tags in the specified model at the instant the command is run.
- **DataLogSnapshot \*** records tag values for all tags in all running models at the instant the command is run.

You can use the **DataLogSnapshot** command anywhere you can type a FactoryTalk View command or macro. For example, you can use the command as the action for an event.

To provide operators with a way to log on demand at runtime, create a button object in a graphic display, or a display key or client key that uses DataLogSnapshot as the press action.

### Example: Creating an event for on-demand logging

To create an event that will trigger logging when an alarm occurs:

1. In the **Data Log Models** editor, in the **Log Triggers** tab, select the option **On Demand**.
2. Open the **Event** editor. Create an expression such as:  

```
If alm_in_alarm(motor_fault) and new_batch_started then 1 else 0
```
3. In the **Event** editor's **Action** field, type *DataLogSnapshot <component>* where *component* is the name of the data log model.

When the tag called *motor\_fault* goes into alarm, and the tag called *new\_batch\_started* is 1, the **DataLogSnapshot <component>** command runs. All tags in the model will then be logged (not just the tag in alarm).

## Combining logging

You can combine periodic or on-change logging with on-demand logging, to capture data at particular times and when a particular event occurs.

### To combine logging:

1. In the **Data Log Models** editor, in the **Log Triggers** tab, select the **Periodic** or **On Change** option.
2. Type the **DataLogSnapshot** command with the <component> or \* (asterisk) parameter, anywhere you can use a macro or command.

## Modifying existing data log models

You can modify a data log model during development or at runtime, using either FactoryTalk View Studio or the FactoryTalk View SE Administration Console.

If you change a model at runtime, the changes will not take effect until you stop and then restart data logging. For information about starting and stopping data logging, see [Starting and stopping data logging on page 392](#).



**Tip:** If a tag that you delete from a data log model is used in a trend object, be sure to remove all references to the deleted tag from the trend.

## Changing log paths

To change the log paths for a data log model after deploying a FactoryTalk View SE application, you can use FactoryTalk View Studio or the FactoryTalk View SE Administration Console.



**Tip:** You can launch the FactoryTalk View SE Administration Console from the following location:

- (32-bit operating systems)  
C:\Program Files\Rockwell Software\RSView Enterprise\Admin Console.exe
- (64-bit operating systems)  
C:\Program Files (x86)\Rockwell Software\RSView Enterprise\Admin Console.exe

You can change the primary and secondary paths for file sets, and the backup path for ODBC data log models. You cannot use the Administration Console to change the ODBC database.

## Making runtime changes without modifying the model

At runtime, you can also change the following data log parameters, without modifying the data log model:

- The log rate for periodic logging (use the **DataLogChangeRate** command)
- The log file identifier string (use the **DataLogRenameFile** command)

Use these commands anywhere you can enter a FactoryTalk View command or macro. For example, in FactoryTalk View Studio, or the FactoryTalk View SE Administration Console, type the command directly at the command line.

These changes take effect immediately, but only apply to the current logging session. When you stop and restart logging, FactoryTalk View uses the logging rate and log file identifier specified for the data log model.

For details about the **DataLogChangeRate** and **DataLogRenameFile** commands, see the FactoryTalk View Site Edition Help.

## Starting and stopping data logging

There are many ways to start and stop data logging. Choose the way that works best for the application.

To provide operators with a way to start and stop data logging, use FactoryTalk View commands and macros in conjunction with graphic objects. For details about the FactoryTalk View command syntax, see the FactoryTalk View Site Edition Help.

In the following methods, *<component>* represents the name of a data log model.

## Ways to start data logging

To start data logging, use one of these methods:

- In the **HMI Server Properties** dialog box, click the **Components** tab, select the **Data Logging** checkbox, and then specify a data log model.  
The data log model will run the next time the HMI server runs, or when the HMI server's components are started manually.
- In the **Macros** editor, create a macro that contains the command, **DataLogOn <component>**. In the **HMI Server Properties** dialog box, click the **Components** tab, select the **Macro** checkbox, and then specify the macro.  
The specified data log model will run when the HMI server runs. To start more than one data log model, specify multiple **DataLogOn <component>** commands in the macro.
- In the **Graphics** editor, open the **Display Settings** dialog box, and then click the **Behavior** tab. In the **Startup** box, type **DataLogOn <component>**.
- In the **Graphics** editor, create a button object and then specify **DataLogOn <component>** as the button's press action. When the button is pressed, the command runs.
- In the **Graphics** editor, create a graphic object and attach a touch animation to it, using **DataLogOn <component>** in the **Action** box. When the object is touched, data logging starts for the specified model.
- In the **Events** editor, type the **DataLogOn <component>** command in the **Action** box for an event.
- At the command line in FactoryTalk View Studio, or the FactoryTalk View SE Administration Console, type **DataLogOn <component>** and then press **Enter**.

## Ways to stop data logging

To stop data logging only, use one of these methods:

- Use the **DataLogOff <component>** command (where <component> is the name of the data log model) to stop a single model. To stop all data log models, use the **DataLogOff \*** command.
- In the Graphics editor, create a button object, and then specify **DataLogOff <component>** or **DataLogOff \*** as the button's press action. When the button is pressed, the command runs.
- In the **Graphics** editor, create a graphic object and attach touch animation to it, using **DataLogOff \* <component>** or **DataLogOff** in the Action box. When the object is touched, data logging is stopped for the specified model.
- In the **Events** editor, specify **DataLogOff <component>** or **DataLogOff \*** as the action for an event.
- At the command line in FactoryTalk View Studio, or the FactoryTalk View SE Administration Console, type **DataLogOff <component>** or type **DataLogOff \*** and then press **Enter**.

You can also stop data logging by stopping all HMI server components running on the computer. For details, see [Starting and stopping HMI server components manually on page 139](#).

## Setting up trends

This chapter describes:

- What trends are.
- How to create trends.
- The parts of a trend.
- Trend chart styles.
- Choosing colors, fonts, and lines for a trend.
- Using templates for trend objects.
- Working with trends at runtime.

### About trends

A trend is a visual representation, or chart, of tag values over a historical period, a pre-set period, and in real-time. A trend provides an operator with a way to track plant activity as it is happening.

In a trend, you can:

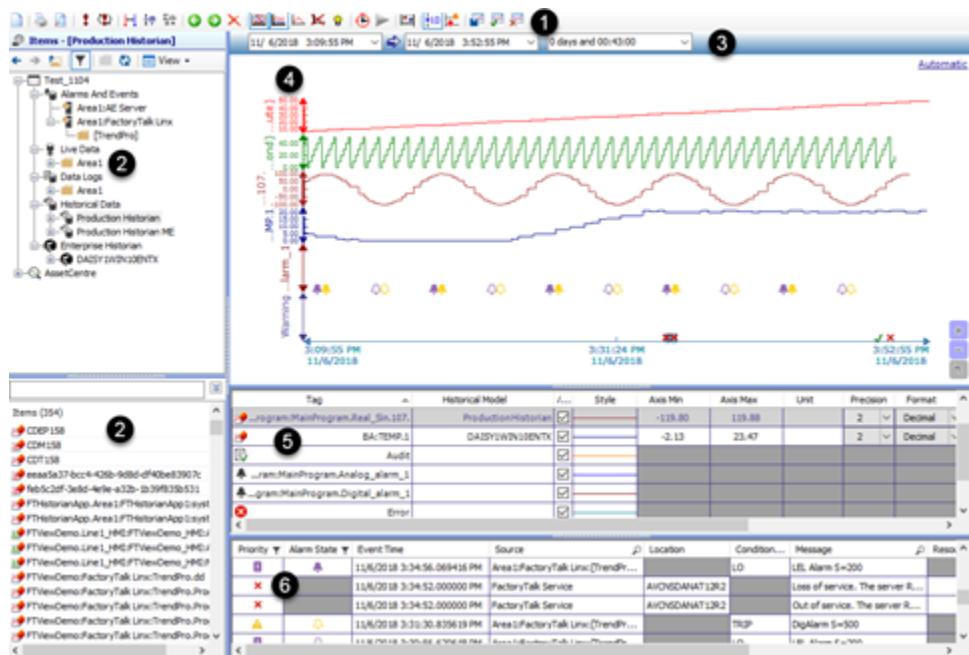
- Identify the boundaries that define "normal" conditions for a trace, by adding user-defined shapes.
- Determine whether the value of a tag is outside of a defined boundary.
- Display and compare multiple tags or traces concurrently.
- Manipulate the data in a variety of ways, including panning, zooming, and scaling after you have added tags.
- Perform easy runtime configuration, drag ad-hoc trending, and statistical and sampling calculations.

The data sources that TrendPro supports include live data, FactoryTalk View SE data logs, FactoryTalk Historian, OSIsoft PI Server, FactoryTalk AssetCentre, and FactoryTalk Alarms and Events.

For more information about each trending object, see FactoryTalk View Site Edition Help.

## TrendPro interface

The following illustration shows an example of the TrendPro interface.



Item	Name	Description
①	Toolbar	Displays the commonly used commands such as <b>Live Mode</b> , <b>Automatic Scales</b> , <b>Trace Highlighting</b> , and so on.
②	Tag Explorer	Displays the hierarchy of folders in a tree view that represents the areas and tag folders of the data sources. When you select a folder, its tags are listed in an alphabetically sortable list on the <b>Items</b> pane at the bottom. Double-click a tag to add it to the chart. At runtime, you can also drag one or multiple tags to the chart.
③	Time Bar	Provides options to select a time period using <i>absolute</i> or <i>relative</i> time. When using absolute time, you select specific dates and times for the start and end of the time period. When using relative time, you select a typical time period, such as the Last 5 minutes or Last week.
④	Chart	Displays data from one or more tags, also known as traces, over a period.

Item	Name	Description
⑤	<b>Tag List</b>	Displays the list of traces that are currently on the chart.
⑥	<b>Event List</b>	Displays the list of operator actions and alarm events that have occurred within the current trend time period. <b>Note:</b> Some events may not be visible on the chart because they are drawn behind higher priority events.

## Creating TrendPro objects

Create a TrendPro object to show real-time data, historical data, and data from FactoryTalk View SE data logs.

### To create a TrendPro object:

1. In the Toolbox, select **TrendPro**.
2. Drag the mouse to create a box approximately the size the trend chart is to be, and the **TrendPro Properties** dialog box opens.

## Setting up TrendPro properties

In the **TrendPro Properties** dialog box, you can set up the trend's data, appearance, and behavior.

- In the **General** tab, set up which trend options are available to operators at runtime.
- In the **Common** tab, set up the trend's size and position in the graphic display.

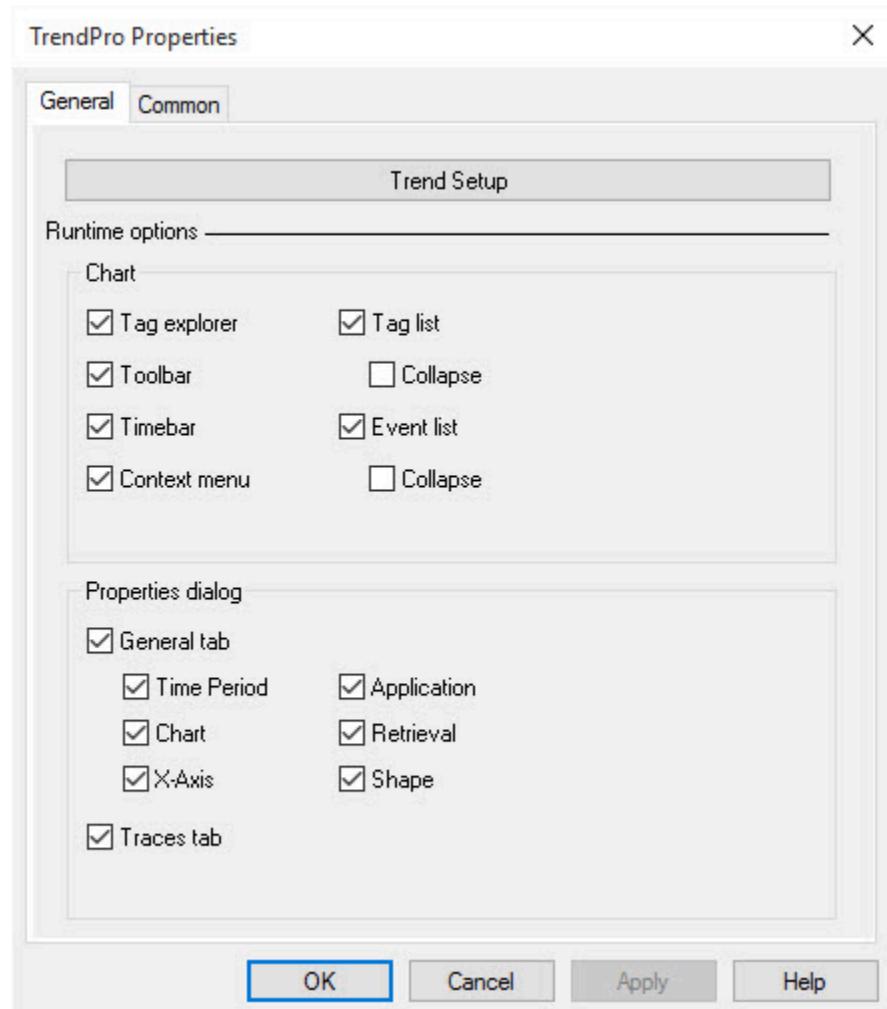
In the Trend Setup **Properties** dialog box, you can configure the detailed trend appearances and behaviors at runtime.

- In the **Time Period** page of the **General** tab, set up the trend's update period, update mode, and time zone.
- In the **Chart** page of the **General** tab, set up the trend's chart title, information that displays in the tooltips, zoom, and other options of the trend display.
- In the **X-Axis** page of the **General** tab, set up the trend's horizontal axis.
- In the **Application** page of the **General** tab, set up the trend's line styles and trace dimming.
- In the **Live Data** page of the **General** tab, set up the live data storage.
- In the **Retrieval** page of the **General** tab, set up the retrieval mode and retrieval information displayed on the trend.
- In the **Trace** page of the **Traces** tab, set up the appearance and other options for the selected trace.
- In the **Y-Axis** page of the **Traces** tab, set up the vertical axis for the selected trace.

Once you have set up the trend, you can modify it as any other graphic object. You can move it, resize it, attach animation to it, and so on. You can also copy a trend object by dragging it from one graphic display to another. For more information about working with graphic displays and objects, see [Creating graphic displays on page 259](#), and [Creating graphic objects on page 288](#).

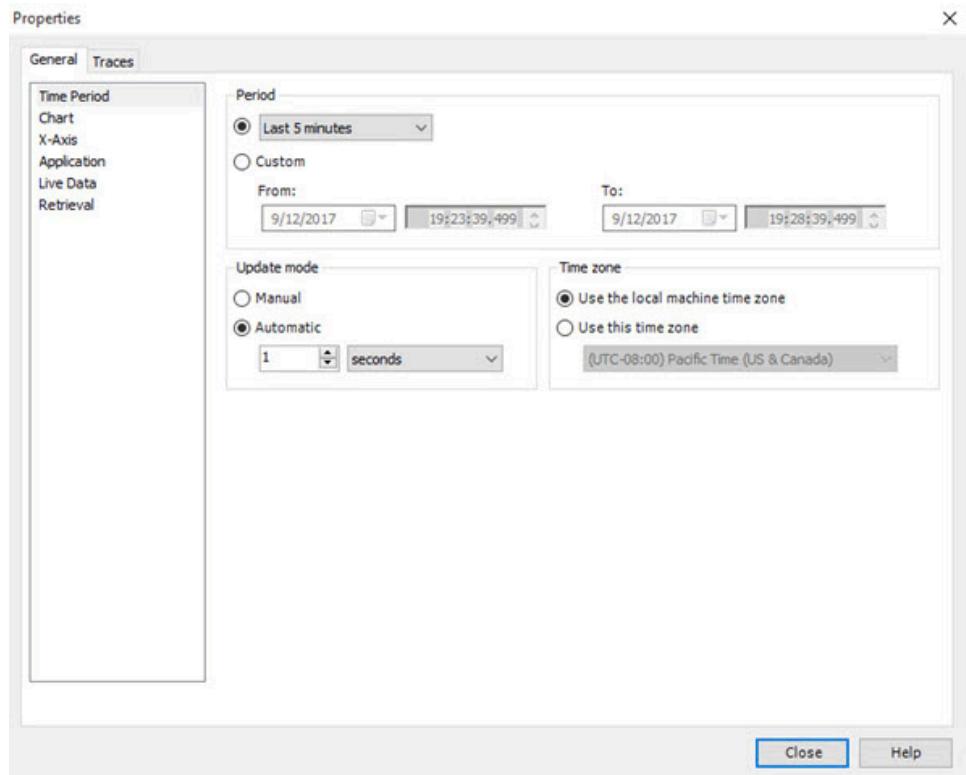
### To open the TrendPro Properties dialog box:

Double-click the trend object. You can also right-click the object and select **Properties**. For details about options in the dialog box, click **Help**.



To open the TrendPro Setup Properties dialog box:

In the **TrendPro Properties** dialog box, click **Trend Setup**. If the **Context menu** option is enabled, at runtime, you can also right-click the trend and select **Properties**. For details about options in the dialog box, click **Help**.



## Trace List (trend)

The **Tag List** displays details of the current traces and other data applicable to the selected tags. In the **Tag List**, each row represents a trace, and multiple traces are possible.

The following table describes the data provided for each trace:

Column	Description
<b>Tag</b>	Fully qualified name of the tag.
<b>Historical Model</b>	FactoryTalk View SE Data log models, DataLogPro, or FactoryTalk Historian SE or ME servers.
<b>Visible</b>	Indicates whether the trace is visible on the trend. Clear this checkbox to hide a selected trace without removing it from the list of traces.
<b>Style</b>	Color, width, and style of the trace line. Double-click a line in this column to change its appearance.
<b>Axis Min</b>	Minimum Y-axis value for the tag (in engineering units).
<b>Axis Max</b>	Maximum Y-axis value for the tag (in engineering units).
<b>Unit</b>	Engineering unit for the tag value.
<b>Precision</b>	Decimal precision for the tag value. Click the down arrow to change the precision.

Column	Description
<b>Format</b>	Scientific or decimal number format. Click the down arrow to change the format.
<b>Tag Min</b>	Minimum value of the raw acquired value of the tag. <b>Note:</b> This column only supports Historian tags.
<b>Tag Max</b>	Maximum value of the raw acquired value of the tag. <b>Note:</b> This column only supports Historian tags.

## Event list

Displays the list of operator actions and alarm events that have occurred within the current trend time period. Some events may not be visible on the chart because they are drawn behind higher priority events.

---

**NOTE:** The columns shown may vary depending on whether you are using the local or network application.

---

The following table describes columns for the events.

Column	Description
<b>Priority</b>	An icon that represents the urgency of the event.
<b>Alarm State</b>	An icon that represents the change in alarm state caused by an alarm event.
<b>Event Time</b>	The time at which the diagnostic information occurred or the alarm condition changed into the currently displayed state. You can click this column to sort the list.
<b>Source</b>	The name of the alarm, or the software product that generated the diagnostic message.
<b>Location</b>	The computer name from which the message generated
<b>Condition Name</b>	A name that describes the alarm condition. Condition names vary by event category. For example: <ul style="list-style-type: none"> <li>• Level: LO, LOLO, HI, HIHI</li> <li>• Discrete: TRIP, TRIP_L (Latched)</li> <li>• Deviation: DEV_LO, DEV_HI, DEV_HI_LO</li> <li>• Rate of Change: ROC_POS, ROC_NEG</li> </ul>
<b>Message</b>	The text that describes the operator action or alarm event.
<b>Resource</b>	The name of the hardware or software object that is the subject of the diagnostic message.
<b>User Name</b>	The ID of the user logged on to the computer when the message was generated.
<b>Current Value</b>	The value of the tag that is being monitored for alarms when the alarm condition changed its state.
<b>Limit Value</b>	An integer value that specifies the limit value of the alarms.

## Testing a trend

In FactoryTalk View Studio, in the Graphics editor, you can use **Test Display** mode to test a trend while you are setting it up. If you have set up communications and there is data for the tags, the traces will plot values.

When you are finished testing, switch back to **Edit Display** mode to continue setting up trend properties.

### To switch between Test and Edit Display modes:

In the **Graphics** editor, do any of the following:

- From the **View** menu, select **Test Display** or **Edit Display**.
- On the toolbar, click **Test Display**  or **Edit Display** .

## Changing the trend appearance

You can set properties that are immediately applied to the current trend from the **Properties** dialog box.

## Time Period page (trend)

Time period properties allow you to set the update rate for data, determine the time zone to use for data retrieval, and set a relative or absolute time period. The following properties are applied immediately to the current trend.

---

**NOTE:** Each time when TrendPro reads diagnostics from the FactoryTalk AssetCentre server, the maximum number of diagnostics is 5,000 for each type of severity.

---

The following options are available:

Options	Description
<b>Period</b>	Determines whether a relative or absolute time period is used. Select one of the following options: <ul style="list-style-type: none"> <li>• <b>Last 5 minutes.</b> Select this option, and then select a relative time period such as <i>Last week</i>.</li> <li>• <b>Custom.</b> Select this option, and then specify a date and time for the start (From) and end time (To).</li> </ul>
<b>Update mode</b>	Determines the method for refreshing data. Select one of the following options: <ul style="list-style-type: none"> <li>• <b>Manual.</b> Select this option to manually update the data.</li> <li>• <b>Automatic.</b> Select this option, and then specify the number of milliseconds, seconds, or minutes when data will be updated. When using Automatic Retrieval Mode, this value may be overridden if the Sampling Interval is slower than the rate configured here. When this occurs, the word "optimized" displays in the top-left corner of the plot surface beside the update rate and the effective update rate.</li> </ul>

Options	Description
<b>Time zone</b>	<p>Retrieves a data set for a particular time period regardless of the time zone. Select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Use the local machine time zone.</b> Select this option to use a local time zone when retrieving or displaying data.</li> <li>• <b>Use this time zone.</b> Select this option to use a fixed time zone when retrieving or displaying data. In this case, you can select a specific time zone from a list of all time zones defined by Windows.</li> </ul>

## Chart page (trend)

Chart properties allow you to set the title, information that displays in tooltips, zoom, and other options that display in the current trend.

**NOTE:** The following properties are applied immediately to the current trend.

The following options are available:

Property	Description
<b>Title</b>	<p>The title to use for the trend. Select one of the following options:</p> <ul style="list-style-type: none"> <li>• Click the <b>Use current trace label</b> checkbox to change the title based on the trace that currently has focus. As different tags are selected, the title updates. The tag description displays along with the tag name.</li> <li>• Type a title for the chart in the field below the <b>Use current trace label</b> checkbox. This title will remain regardless of the trace that has focus.</li> </ul> <p>You can set the title's font, style, size, and effects by clicking <b>Font</b>.</p>
<b>Margins</b>	<p>The amount of white space around the trend on the plot surface (in pixels).</p>
<b>Tooltips</b>	<p>Select the content of the tooltip from the following options:</p> <ul style="list-style-type: none"> <li>• Click the <b>Show value</b> checkbox to display the tag's value when hovering over a line in the chart.</li> <li>• Click the <b>Show date and time</b> checkbox to display the tag's time stamp data when hovering over a line in the chart.</li> <li>• Click the <b>Show quality</b> checkbox to display the tag's quality when hovering over a line in the chart.</li> </ul>

Property	Description
	<ul style="list-style-type: none"> <li>Click the <b>Show caption</b> checkbox to display the tag's caption instead of the tag name when hovering over a line in the chart.</li> </ul> <p><b>Note:</b> You must select <b>Caption</b> under <b>Legend</b> on the <b>Traces</b> tab for the caption to display in the tooltip.</p>
<b>Zoom and Pan</b>	<p>The display of the plot surface.</p> <p>Select from the following options:</p> <ul style="list-style-type: none"> <li>In the <b>Zoom by</b> box, type the percentage that you want to increase when using zooming or rubber band zooming to zoom in or out of a chart.</li> <li>For the <b>Display overlay panel</b> checkbox, select whether to display a toolbar in the bottom right of the plot surface to increase, decrease, or reset the zoom.</li> <li>For the <b>Enable pan</b> checkbox, select whether to allow panning the data by moving the mouse.</li> <li>For the <b>Pinch to zoom</b> checkbox, select whether to allow using two fingers to zoom in or out of a canvas on a touch screen.</li> </ul>
<b>Automatic scaling</b>	<p>Determines the method used to automatically scale data based on the following:</p> <ul style="list-style-type: none"> <li><b>From each individual trace.</b> Uses the lowest and highest value for each trace in the chart. Values are separated into their own axis. For example, if you have a tag with values from 0-100, and another tag with values from 500-900, when you select the first tag, the Y-axis displays 0-100. When you select the second tag, the Y-axis displays a value of 500-900.</li> <li><b>From all traces in the chart.</b> Uses the lowest and highest values from all traces in the chart for the Y-axis. For example, if you have a tag with values from 0-100, and another tag with values from 500-900, the Y-axis would have a value of 0-900 to include values from both traces. This is useful, for example, for comparing two production lines side by side that have the same target. If you scaled them individually, both would look fine, but if you compared them side by side using the same scale, you might notice that the downtime on the second line was 20% higher than the first line.</li> </ul> <p><b>Note:</b> Traces that have logarithmic mode enabled or have automatic scaling will be in a separate group and not combined with other traces in the chart.</p>

Property	Description
<b>Background</b>	<p>Determines the background color or image. Select from the following options:</p> <ul style="list-style-type: none"> <li>• To change the background color of the chart area, click the <b>Color</b> button beside <b>Window Color</b> and select a color.</li> <li>• To change the plot area's background color, click the <b>Color</b> button beside <b>Canvas Color</b> and select a color.</li> <li>• To change the plot area's background image, click the <b>Browse</b> button beside <b>Canvas Image</b> and select an image.</li> </ul>
<b>Cursors</b>	<p>Determines how the cursors display data when enabled. Select from the following options:</p> <ul style="list-style-type: none"> <li>• Click the <b>Time</b> checkbox to display the time difference between the two cursors.</li> <li>• Click the <b>Value differences</b> checkbox to display the difference between the value at the start of the cursor and the end of the cursor.</li> </ul>
<b>Trace layout</b>	<p>Displays each trace in its own grid or zone.</p> <ul style="list-style-type: none"> <li>• Click the <b>Stack Y-axis</b> checkbox to display each trace in a separate grid.</li> </ul>
<b>Annotations</b>	<p>Once enabled, it allows you to view, add, edit, or delete annotations for the FactoryTalk Historian SE trace from the runtime context menu.</p> <p><b>Note:</b> To view all the annotations, choose the raw retrieval mode via <b>General &gt; Retrieval &gt; Raw</b>. The other retrieval modes (automatic or custom) filter tags, thus not all the annotations will show on the trend.</p>

## X-Axis page

X-axis properties allow you to define how the X-axis displays time in the current trend. The properties are applied immediately to the current trend.

The following options are available:

Property	Description
<b>Label</b>	<p>The option to include in the axis label.</p> <p>Click the <b>Use default text</b> checkbox to display the default time period as the label. Enter custom text after clearing the <b>Use default text</b> checkbox.</p>
<b>Axis</b>	<p>The display options for the axis. Select from the following options:</p> <ul style="list-style-type: none"> <li>• Click the <b>Visible</b> checkbox to show the axis.</li> <li>• Click the <b>Show grid lines</b> checkbox to show a set of grid lines at each point on the axis.</li> </ul>

Property	Description
	<ul style="list-style-type: none"> <li>Click <b>Font</b> to modify the font style, size, and other font-related effects.</li> <li>Click <b>Line Style</b> to modify the line color, style, and width.</li> </ul>
<b>Time axis mode</b>	<p>The mode used to display time periods. Select one of the following options:</p> <ul style="list-style-type: none"> <li>Click <b>Absolute</b> for X-axis divisions to reflect the times shown on the <b>Time</b> bar.</li> <li>Click <b>Relative</b> for X-axis divisions to reflect the period displayed in the <b>Relative Mode</b> box on the <b>Time</b> bar. The application attempts to create smart divisions depending on the relative time selected.</li> </ul>
<b>Tick marks</b>	<p>Specifies short lines along the axis line that specify value divisions. Select one of the following options:</p> <ul style="list-style-type: none"> <li>In the <b>Major\Major every</b> box, type the number of major tick marks that display on the time axis for absolute time, or type the number of seconds between each tick mark. If you selected <i>Absolute</i> for the <b>Time axis mode</b>, entering zero results in the chart displaying three major tick marks. If you selected <i>Relative</i> for the <b>Time axis mode</b>, entering zero results in the chart displaying a major tick mark for every second.</li> <li>In the <b>Minor</b> box, type the number of unlabeled value divisions along the time axis line. If you selected <i>Relative</i> for the <b>Time axis mode</b>, entering zero results in the chart displaying a minor tick mark between each major tick mark. Note that this field is not available if you selected <i>Absolute</i> for the <b>Time axis mode</b>.</li> </ul>

## Application page (trend)

Application properties allow you to create customized trends that emphasize important data effectively.

The following options are available:

Properties	Description
<b>Line styles</b>	<p>The color, format style, and weight of trace lines in the chart. The number of the trace corresponds to the trace order as it is displayed in the <b>Trace List</b> below the plot surface. Click anywhere in the line style to change it.</p>
<b>Highlighted</b>	<p>Adjusts the color, format, and line weight of a highlighted trace.</p>
<b>Trace dimming</b>	<p>The brightness for traces that do not have focus when multiple time periods are used.</p>

## Retrieval page (trend)

Trend can retrieve and plot data using the properties on this page.

The following options are available:

Options	Description
<b>Retrieval mode</b>	<p>Select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Automatic.</b> Applies an optimal retrieval strategy based on a data source's plotting method. If the data source does not have a plotting method, the raw data points will be filtered based on the size of the trend surface and the currently selected time period. Automatic retrieval mode is designed to minimize the amount of data requested from the server.</li> <li>• <b>Note:</b> The sampling interval time does not apply to alarm events or diagnostics. TrendPro retrieves these data from the Alarm and Event Historian or FactoryTalk AssetCentre database.</li> <li>• <b>Custom.</b> Defines custom retrieval settings such as deadbands, quality, and sampling options. The first time you select <b>Custom</b>, the <b>Change Settings</b> dialog box opens.</li> <li>• <b>Raw.</b> Returns all data points.</li> <li>• <b>Effective settings.</b> Displays any settings applied to the retrieval mode that you selected.</li> </ul>
<b>Display</b>	<p>Click the <b>Show retrieval information on the trend</b> checkbox. When selected, the current retrieval setting displays as a link in the upper right corner of the plot surface.</p>

## Using templates for trend objects

A trend template contains a set of trend settings. A template can be applied to all trends in an HMI server to create a consistent appearance. When you apply a template, the property settings selected for the template are applied to the current trend and will be evident on the runtime chart.

Use a trend template:

- to apply consistent settings to multiple trend charts.
- to build a set of different views for the same data.
- as a quick way to return to a standard display after experimenting with options.

## Create a template (trend)

Create a TrendPro template at runtime. The new template is saved immediately to the HMI server and be available to all other users.

**To create a template:**

1. At runtime, on the TrendPro toolbar, click the **Save Template** icon.
2. In the **Save Template** dialog box, type a component name for the template and click **Save**.

## Apply a template (trend)

Apply a template for a TrendPro object at runtime.

**To apply a template:**

1. In FactoryTalk View Studio, open a display that contains a TrendPro object.
2. At runtime, on the TrendPro toolbar, click the **Apply Template** icon.
3. In the **Apply Template** dialog box, select a template and click **Apply**.

## Working with trends at runtime

You can modify the trends at runtime, including adding traces, viewing specific data, specifying time periods, creating shapes for data analysis, controlling the axes behavior and more.

### Add or delete traces at runtime (trend)

At runtime, you can add a trace as a tag to the trend. Traces added at runtime are not saved.

For each trace on a TrendPro object, samples are plotted using a horizontal X-axis representing time and a vertical Y-axis representing the value.

To add traces to a trend at runtime, you can drag one or more tags from the **Tag Explorer** to the chart or add from the **Properties** dialog box.

To delete traces from the chart, in the **Trace List**, right-click a trace and select **Delete Trace** or select a trace and select **Delete trace**  on the **TrendPro** toolbar.

### View data of traces (trend)

#### View live data

You can view data in real-time using live mode by clicking **Live mode**  on the **TrendPro** toolbar.

#### View trace data

On the plot surface, right-click a trace and select **Trace Data**. In the **Trace Data** dialog box opens, you can view trace data for a specific period, and can organize it for printing.

#### View trace statistics

Trend calculates statistics (including averages and time weighted averages) on the data being plotted on the graph. Statistics calculations are either performed on all the data, or across the time range represented by the two X-axis cursors. On the **TrendPro** toolbar, click **Statistics** . In the **Statistics** dialog box, select a time period to view.

## Use cursors (trend)

### X-axis cursors

When the X cursor is enabled, two vertical cursors that slide along the X-axis display. Each cursor can be operated individually. You can also use the X-axis cursor with the Y-axis cursor. This feature is useful when you stack the Y-axis so you can quickly see the values for a point in time for all traces at once and compare those values.

Cursors:

- Can be used while zooming in on data.
- Allow panning of data.
- Display values of data points that touch the cursor.
- Calculate the time difference between cursor data points.

To user X-axis cursors, on the **TrendPro** toolbar, click **X-axis Cursors** . Two vertical cursors display on the trend. Drag each cursor left or right on the plot surface.

### Y-axis cursors

When the Y cursor is enabled, two horizontal cursors that slide along the Y-axis display. Each cursor can be operated individually. You can also use the X-axis cursor with the Y-axis cursor.

Cursors:

- Can be used while zooming in on data.
- Allow panning of data.
- Display values of data points that touch the cursor.

To use Y-axis cursors, on the **TrendPro** toolbar, click **Y-axis Cursors** . Two horizontal cursors display on the trend. Drag each cursor up or down on the plot surface.

## Create time periods (trend)

Use time periods to analyze the same data over different time periods. When creating a trend, you usually select tags that are related: pump flow, speed, temperature, and pressure over a specific period. When plant operation is not consistent, it is useful to compare the operation of a tag, such as a pump, during normal operations against its operation during a suboptimal period.

### To create a time period:

1. Add one or more tags to the plot surface.
  2. On the **TrendPro** toolbar, click **New Time Period** . The **Time** bar changes from blue to red to indicate that you can create a time period.
  3. On the **Time** bar, do one of the following:
    - Click the down arrow beside the first date and time entry and select the start date and time. Select an end date and time and then click **Refresh**  between the two dates to refresh the trace.
    - Click the down arrow beside the Last 5 minutes entry and select a relative time period.
- Trend creates a trace definition using the same tags as the first time period.

## Use shapes (trend)

Shapes are a powerful analyzing tool that allows quick detection of normal or abnormal data behavior.

### Create shapes from existing traces

When creating a shape from an existing trace, the trend analyzes the underlying data and makes its best effort to mimic the trace data. The resulting shape will usually follow the shape of the data represented by the trace, but occasionally some data points may not be included in the shape. If this occurs, you can refine your shape.

To create a shape from an existing trace, right-click the plot surface and select **Shape > New from Trace**. A shape is added to the trend.

### Create shapes

When you create a shape that is not based on an existing trace, a rectangular shape displays on the plot surface and you can manually change the shape.

#### To create a shape:

1. Right-click the plot surface and select **Shape > New Default**. A rectangular shape is added to the plot and you are placed in edit mode.
2. Use the cursor to drag the square boxes on the corners of the shape to change the shape.
3. Right-click the plot surface and select **Exit Edit** when finished.

## Zoom and pan data (trend)

You can zoom in on or pan data on a trend for more detailed analysis.

#### To zoom in on a trend, do any of the following:

- Double-click an area of the plot surface that you want to zoom in on.
- Press CTRL-plus on the keyboard.
- At the bottom right of the plot surface, click .
- Scroll the mouse wheel up. Note that if you use the mouse wheel, the trend zooms relative to the location of the cursor on the plot surface.

#### To zoom out of a trend, do any of the following:

- Shift-click an area of the plot surface that you want to zoom out of.
- Press CTRL-minus on the keyboard.
- At the bottom right of the plot surface, click .
- Scroll the mouse wheel down. Note that if you use the mouse wheel, the trend zooms relative to the location of the cursor on the plot surface.

#### To pan data, select one of the following directions:

- **Previous data.** Press and hold the left mouse button and drag right.
- **More recent data.** Press and hold the left mouse button and drag left.
- **Up.** Press and hold the left mouse button and drag up.
- **Down.** Press and hold the left mouse button and drag down.

## Setting up XY Plot

This chapter describes:

- What XY Plot is.
- How to create XY Plot.
- The parts of XY Plot.
- XY Plot chart styles.
- Choosing colors, fonts, and lines for XY Plot.
- Using templates for XY Plot objects.
- Working with XY Plot at runtime.

### About XY Plot

An XY Plot is a specific type of trend that plots one set of tag values versus another set of tag values. This allows an operator to see the relationship between the two sets of data over time.

You can use XY Plot to query data and plot it on a graphical display.

Use the XY Plot objects to:

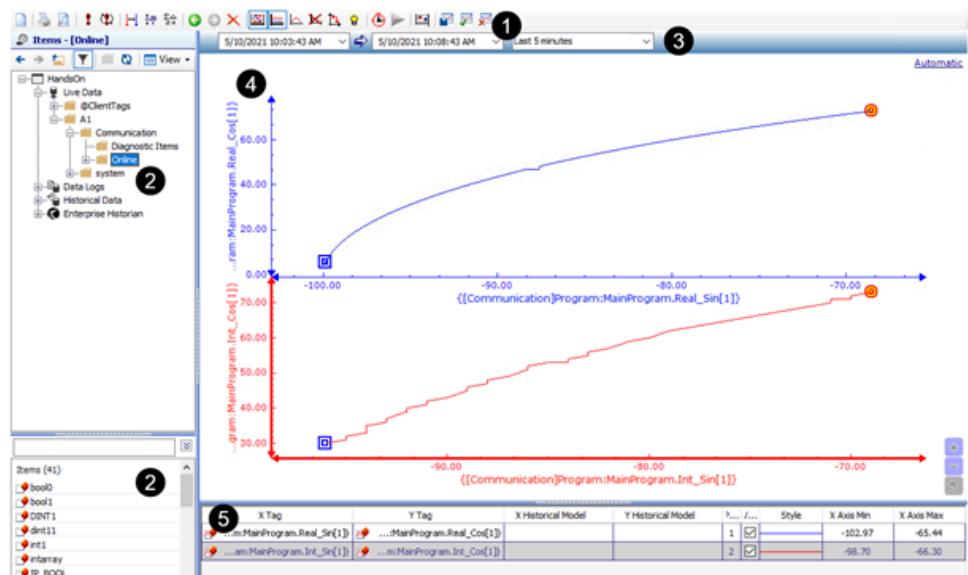
- See the relationship between the values of two tags (tag pair) over a historical period, a pre-set period, and in real-time. When plotted, this relationship is referred to as a trace.
- Easily identify the boundaries that define "normal" conditions for a trace, by adding user-defined shapes.
- Determine whether the value of a tag is outside of a defined boundary.
- Display and compare multiple tags or traces concurrently.

Once you have added tags, you can manipulate the data in a variety of ways, including panning, zooming, and scaling.

For more information about the XY Plot object, see FactoryTalk View Site Edition Help.

### XY Plot interface

The following is an overview of the XY Plot interface:



Item	Name	Description
①	<b>Toolbar</b>	Displays the commonly used commands such as <b>Live Mode</b> , <b>Automatic Scales</b> , <b>Trace Highlighting</b> and so on.
②	<b>Tag Explorer</b>	Displays the hierarchy of folders in a tree view that represents the areas and tag folders of the data sources. When you select a folder, its tags are listed in an alphabetically sortable list on the <b>Items</b> pane at the bottom. Double-click a tag to add it to the X-axis, and then double-click another tag to add it to the Y-axis. At runtime, you can also drag one or multiple tags to the chart.
③	<b>Time Bar</b>	Provides options to select a time period using absolute or relative time. When using absolute time, select specific dates and times for the start and end of the time period. When using relative time, select a typical time period, such as the last 5 minutes or last week.
④	<b>Chart</b>	Displays data from one or more pairs of tags, also known as traces, over a period.
⑤	<b>Trace List</b>	Displays the list of traces that are currently on the chart.

## Creating XY Plot objects

Create an XY Plot object to show one set of tag values versus another set of tag values. This allows an operator to see the relationship between the two sets of data over time.

### To create an XY Plot object:

1. In the Toolbox, select **XY Plot**.
2. Drag the mouse to create a box approximately the size the trend chart is to be, and the **XY Plot Properties** dialog box opens.

## Setting up XY Plot properties

In the **XY Plot Properties** dialog box, you can set up the data, appearance, and behavior.

- In the **General** tab, set up which XY Plot options are available to operators at runtime.
- In the **Common** tab, set up XY Plot's size and position in the graphic display.

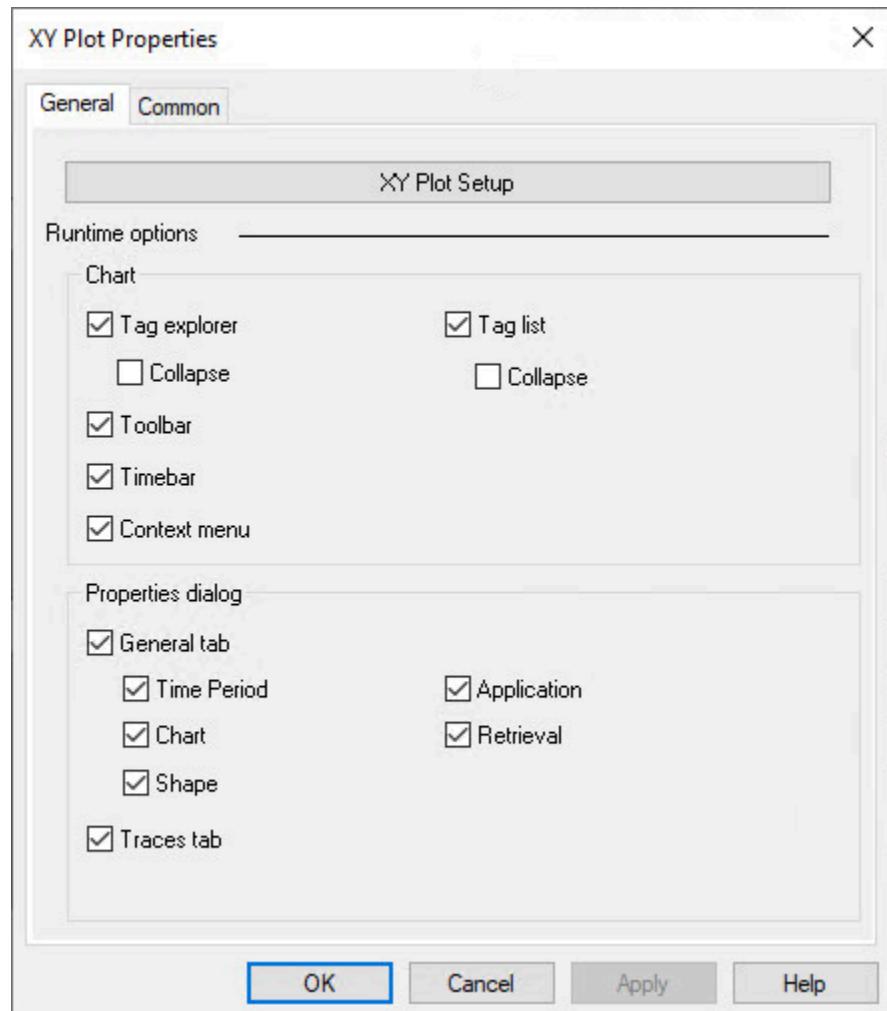
In the XY Plot Setup **Properties** dialog box, you can configure the detailed XY Plot appearances and behaviors at runtime.

- In the **Time Period** page of the **General** tab, set up XY Plot's update period, update mode, and time zone.
- In the **Chart** page of the **General** tab, set up XY Plot's chart title, information that display in tooltips, zoom and pan, and other options of the XY Plot display.
- In the **Application** page of the **General** tab, set up XY Plot's line styles and trace dimming.
- In the **Live Data** page of the **General** tab, set up the live data storage.
- In the **Retrieval** page of the **General** tab, set up the retrieval mode and retrieval information displayed on XY Plot.
- In the **Trace** page of the **Traces** tab, set up the appearance and other options for the selected trace.
- In the **X-Axis** page of the **Traces** tab, set up the horizontal axis for the selected trace.
- In the **Y-Axis** page of the **Traces** tab, set up the vertical axis for the selected trace.

Once you have set up XY Plot, you can modify it as any other graphic object. You can move it, resize it, attach animation to it, and so on. You can also copy an XY Plot object by dragging it from one graphic display to another. For more information about working with graphic displays and objects, see [Creating graphic displays on page 259](#), and [Creating graphic objects on page 288](#).

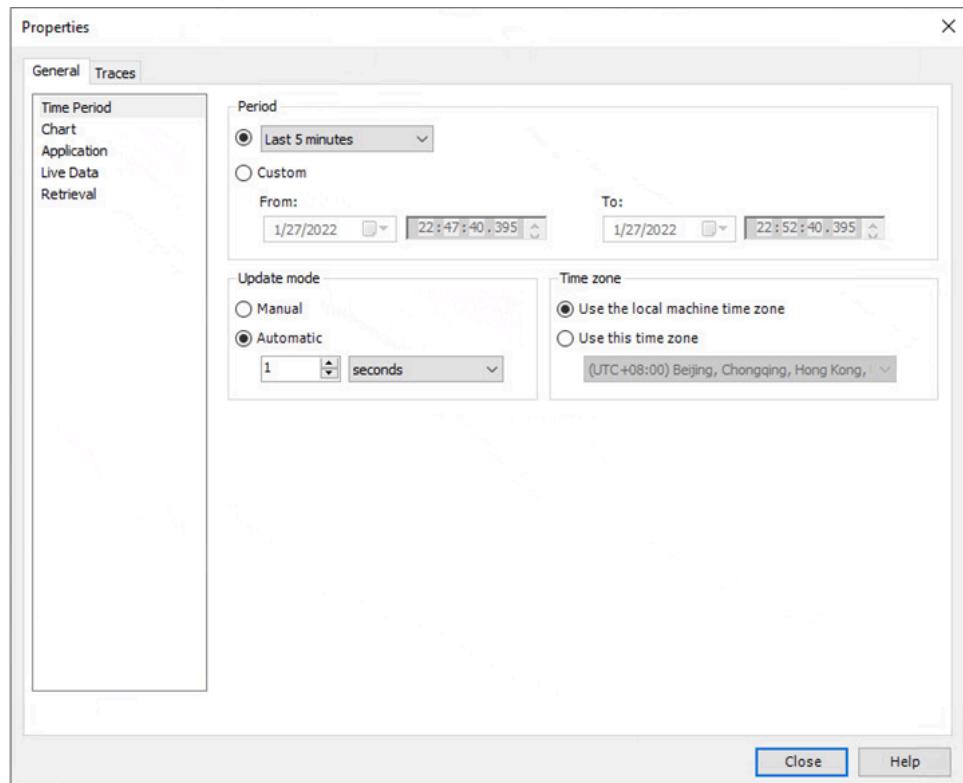
### To open the XY Plot Properties dialog box:

Double-click the XY Plot object. You can also right-click the object and select **Properties**. For details about options in the dialog box, click **Help**.



**To open the XY Plot Setup Properties dialog box:**

In the **XY Plot Properties** dialog box, click **XY Plot Setup**. If the **Context menu** option is enabled, at runtime, you can also right-click XY Plot and select **Properties**. For details about options in the dialog box, click **Help**.



## Trace List (XY Plot)

The **Trace List** displays details of the current traces and other data applicable to the selected tags. In the **Trace List**, each row represents a trace, and multiple traces are possible.



**Tip:** Right-click the list to show the context menu for configuring column settings.

The following table describes the data provided for each trace.

Column	Description
<b>X Tag</b>	Fully qualified name of the x tag.
<b>Y Tag</b>	Fully qualified name of the y tag.
<b>X Historical Model</b>	FactoryTalk View SE Data log models or FactoryTalk Historian SE or ME servers for X-axis.
<b>Y Historical Model</b>	FactoryTalk View SE Data log models or FactoryTalk Historian SE or ME servers for Y-axis.
<b>Visible</b>	Whether the trace is visible on the chart. Clear this checkbox to hide a selected trace without removing it from the list of traces.
<b>Style</b>	Color, width, and style of the trace line. Double-click a line in this column to change its appearance.
<b>X Axis Min</b>	Minimum X-axis value for the tag (in engineering units).

Column	Description
X Axis Max	Maximum X-axis value for the tag (in engineering units).
Y Axis Min	Minimum Y-axis value for the tag (in engineering units).
Y Axis Max	Maximum Y-axis value for the tag (in engineering units).
X Unit	Engineering unit for the x tag value.
Y Unit	Engineering unit for the y tag value.
Precision	Decimal precision for the tag value. Click the down arrow to change the precision.
Format	Scientific or decimal number format. Click the down arrow to change the format.
X Tag Min	Minimum value of the raw acquired value of the x tag. <b>Note:</b> This column only supports Historian tags.
X Tag Max	Maximum value of the raw acquired value of the x tag. <b>Note:</b> This column only supports Historian tags.
Y Tag Min	Minimum value of the raw acquired value of the y tag. <b>Note:</b> This column only supports Historian tags.
Y Tag Max	Maximum value of the raw acquired value of the y tag. <b>Note:</b> This column only supports Historian tags.

## Testing XY Plot

In FactoryTalk View Studio, in the Graphics editor, you can use **Test Display** mode to test XY Plot while you are setting it up. If you have set up communications and there is data for the tags, the traces will plot values.

When you are finished testing, switch back to **Edit Display** mode to continue setting up XY Plot properties.

### To switch between Test and Edit Display modes:

In the **Graphics** editor, do any of the following:

- From the **View** menu, select **Test Display** or **Edit Display**.
- On the toolbar, click **Test Display**  or **Edit Display** .

## Changing the XY Plot appearance

You can set properties that are immediately applied to the current XY Plot from the **Properties** dialog box.

## Time Period page (XY Plot)

Time period properties allow you to set the update rate for data, determine the time zone to use for data retrieval, and set a relative or absolute time period.

---

**NOTE:** The following properties are applied immediately to the current XY Plot.

---

The following options are available:

Options	Description
<b>Period</b>	<p>Determines whether a relative or absolute time period is used. Select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Last 5 minutes.</b> Select this option, and then select a relative time period such as <i>Last week</i>.</li> <li>• <b>Custom.</b> Select this option, and then specify a date and time for the start (From) and end time (To).</li> </ul>
<b>Update mode</b>	<p>Determines the method for refreshing data. Select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Manual.</b> Select this option to manually update the data.</li> <li>• <b>Automatic.</b> Select this option, and then specify the number of milliseconds, seconds, or minutes when data will be updated. When using Automatic Retrieval Mode, this value may be overridden if the Sampling Interval is slower than the rate configured here. When this occurs, the word "optimized" displays in the top-left corner of the plot surface beside the update rate and the effective update rate.</li> </ul>
<b>Time zone</b>	<p>Retrieves a data set for a particular time period regardless of the time zone. Select one of the following options:</p> <ul style="list-style-type: none"> <li>• <b>Use the local machine time zone.</b> Select this option to use a local time zone when retrieving or displaying data.</li> <li>• <b>Use this time zone.</b> Select this option to use a fixed time zone when retrieving or displaying data. In this case, you can select a specific time zone from a list of all time zones defined by Windows.</li> </ul>

## Chart page (XY Plot)

Chart properties allow you to set the title, information that displays in tooltips, zoom, and other options that display in the current XY Plot.

---

**NOTE:** The following properties are applied immediately to the current XY Plot.

---

The following options are available:

Property	Description
<b>Title</b>	<p>The title to use for the XY Plot. Select one of the following options:</p> <ul style="list-style-type: none"> <li>Click the <b>Use current trace label</b> checkbox to change the title based on the trace that currently has focus. As different tags are selected, the title updates. The tag description displays along with the tag name.</li> <li>Type a title for the chart in the field below the <b>Use current trace label</b> checkbox. This title will remain regardless of the trace that has focus.</li> </ul> <p>You can set the title's font, style, size, and effects by clicking <b>Font</b>.</p>
<b>Margins</b>	<p>The amount of white space around the XY Plot on the plot surface (in pixels).</p>
<b>Tooltips</b>	<p>Select the content of the tooltip from the following options:</p> <ul style="list-style-type: none"> <li>Click the <b>Show value</b> checkbox to display the tag's value when hovering over a line in the chart.</li> <li>Click the <b>Show date and time</b> checkbox to display the tag's time stamp data when hovering over a line in the chart.</li> <li>Click the <b>Show quality</b> checkbox to display the tag's quality when hovering over a line in the chart.</li> <li>Click the <b>Show caption</b> checkbox to display the tag's caption instead of the tag name when hovering over a line in the chart.</li> </ul> <p><b>Note:</b> You must select <b>Caption</b> under <b>Legend</b> on the <b>Traces</b> tab for the caption to display in the tooltip.</p>
<b>Zoom and Pan</b>	<p>The display of the plot surface.</p> <p>Select from the following options:</p> <ul style="list-style-type: none"> <li>In the <b>Zoom by</b> box, type the percentage that you want to increase when using zooming or rubber band zooming to zoom in or out of a chart.</li> <li>For the <b>Display overlay panel</b> checkbox, select whether to display a toolbar in the bottom right of the plot surface to increase, decrease, or reset the zoom.</li> <li>For the <b>Enable pan</b> checkbox, select whether to allow panning the data by moving the mouse.</li> <li>For the <b>Pinch to zoom</b> checkbox, select whether to allow using two fingers to zoom in or out of a canvas on a touch screen.</li> </ul>

Property	Description
<b>Automatic scaling</b>	<p>Determines the method used to automatically scale data based on the following:</p> <ul style="list-style-type: none"> <li>• <b>From each individual trace.</b> Uses the lowest and highest value for each trace in the chart. Values are separated into their own axis. For example, if you have a tag with values from 0-100, and another tag with values from 500-900, when you select the first tag, the Y-axis displays 0-100. When you select the second tag, the Y-axis displays a value of 500-900.</li> <li>• <b>From all traces in the chart.</b> Uses the lowest and highest values from all traces in the chart for the Y-axis. For example, if you have a tag with values from 0-100, and another tag with values from 500-900, the Y-axis would have a value of 0-900 to include values from both traces. This is useful, for example, for comparing two production lines side by side that have the same target. If you scaled them individually, both would look fine, but if you compared them side by side using the same scale, you might notice that the downtime on the second line was 20% higher than the first line.</li> </ul> <p><b>Note:</b> Traces that have logarithmic mode enabled or have automatic scaling will be in a separate group and not combined with other traces in the chart.</p>
<b>Background</b>	<p>Determines the background color or image. Select from the following options:</p> <ul style="list-style-type: none"> <li>• To change the background color of the chart area, click the <b>Color</b> button beside <b>Window Color</b> and select a color.</li> <li>• To change the plot area's background color, click the <b>Color</b> button beside <b>Canvas Color</b> and select a color.</li> <li>• To change the plot area's background image, click the <b>Browse</b> button beside <b>Canvas Image</b> and select an image.</li> </ul>
<b>Cursors</b>	<p>Determines how the cursors display data when enabled. Click the <b>Value differences</b> checkbox to display the difference between the value at the start of the cursor and the end of the cursor.</p>
<b>Trace layout</b>	<p>Displays each trace in its own grid or zone.</p> <ul style="list-style-type: none"> <li>• Click the <b>Stack Y-axis</b> checkbox to display each trace in a separate grid.</li> </ul>

## Application page (XY Plot)

Application properties allow you to customize XY Plot traces that emphasize important data effectively.

---

**NOTE:** The following properties are applied immediately to the current XY Plot.

---

The following options are available:

Properties	Description
<b>Line styles</b>	The color, format style, and weight of trace lines in the chart. The number of the trace corresponds to the trace order as it is displayed in the <b>Trace List</b> below the plot surface. Click anywhere in the line style to change it.
<b>Highlighted</b>	Adjusts the color, format, and line weight of a highlighted trace.
<b>Trace dimming</b>	The brightness for traces that do not have focus when multiple time periods are used.

## Retrieval page (XY Plot)

XY Plot can retrieve and plot data using the properties on this page.

---

**NOTE:** The following properties are applied immediately to the current XY Plot.

---

The following options are available:

Options	Description
<b>Retrieval mode</b>	Select one of the following options: <ul style="list-style-type: none"> <li><b>Automatic.</b> Applies an optimal retrieval strategy based on a data source's plotting method. If the data source does not have a plotting method, the raw data points will be filtered based on the size of the plot surface and the currently selected time period. Automatic retrieval mode is designed to minimize the amount of data requested from the server.</li> <li><b>Custom.</b> Defines custom retrieval settings such as deadbands, quality, and sampling options. The first time you select <b>Custom</b>, the <b>Change Settings</b> dialog box opens.</li> <li><b>Raw.</b> Returns all data points.</li> <li><b>Effective settings.</b> Displays any settings applied to the retrieval mode that you selected.</li> </ul>
<b>Display</b>	Click the <b>Show retrieval information</b> checkbox. When selected, the current retrieval setting displays as a link in the upper right corner of the plot surface.

## Using templates for XY Plot objects

An XY Plot template contains a set of settings. A template can be applied to all XY Plots in an HMI server to create a consistent appearance. When you apply a template, the property settings selected for the template are applied to the current XY Plot and will be evident on the runtime chart.

Use an XY Plot template:

- to apply consistent settings to multiple XY Plot charts.
- to build a set of different views for the same data.
- as a quick way to return to a standard display after experimenting with options.

The XY Plot templates are typically created at runtime and then available for use to all users.

## Create a template (XY Plot)

Create an XY Plot template at runtime. The new template is saved immediately to the HMI server and available to all other users.

### To create a template

1. At runtime, on the XY Plot toolbar, click **Save Template** .
2. In the **Save Template** dialog box, type a component name for the template and click **Save**.

## Apply a template (XY Plot)

Apply a template for an XY Plot object at runtime.

### To apply a template

1. At runtime, on the XY Plot toolbar, click **Apply Template** .
2. In the **Apply Template** dialog box, select a template and click **Apply**.

## Working with XY Plot at runtime

You can modify XY Plot at runtime, including adding traces, viewing specific data, specifying time periods, creating shapes for data analysis, controlling the axes behavior and more.

## Add or delete traces at runtime (XY Plot)

At runtime, you can add a trace with a pair of tags to the XY Plot. Traces added at runtime are not saved.

For each trace on an XY Plot object, samples are plotted using a horizontal X-axis representing one tag value and a vertical Y-axis representing another tag value.

To add traces to an XY Plot at runtime, you can drag one tag as the x tag and another as the y tag from the **Tag Explorer** to the chart or add from the **Properties** dialog box.

To delete traces from the chart, in the **Trace List**, right-click a trace and select **Delete Trace** or select a trace and select **Delete trace**  on the XY Plot toolbar.

## View data of traces (XY Plot)

### View live data

You can view data in real-time using live mode by clicking **Live mode**  on the XY Plot toolbar.

### View trace data

On the plot surface, right-click a trace and select **Trace Data**. In the **Trace Data** dialog box opens, you can view trace data for a specific period, and can organize it for printing.

## View trace statistics

XY Plot calculates statistics (including averages and time weighted averages) on the data being plotted on the graph. Statistics calculations are performed on the data for each time period or across all time periods. On the XY Plot toolbar, click **Statistics**  . In the **Statistics** dialog box, select a time period to view.

## Use cursors (XY Plot)

### X-axis cursors

When the x cursor is enabled, two vertical cursors that slide along the X-axis display. Each cursor can be operated individually. You can also use the X-axis cursor with the Y-axis cursor. This feature is useful when you stack the Y-axis so you can quickly see the values for a point in time for all traces at once and compare those values.

Cursors:

- Can be used while zooming in on data.
- Allow panning of data.
- Display values of data points that touch the cursor.

To user X-axis cursors, on the XY Plot toolbar, click **X-axis Cursors**  Two vertical cursors display on the XY Plot. Drag each cursor left or right on the plot surface.

### Y-axis cursors

When the Y cursor is enabled, two horizontal cursors that slide along the Y-axis display. Each cursor can be operated individually. You can also use the X-axis cursor with the Y-axis cursor.

Cursors:

- Can be used while zooming in on data.
- Allow panning of data.
- Display values of data points that touch the cursor.

To use Y-axis cursors, on the XY Plot toolbar, click **Y-axis Cursors**  Two horizontal cursors display on the XY Plot. Drag each cursor up or down on the plot surface.

## Create time periods (XY Plot)

Use time periods to analyze the same data over different time periods. When creating an XY Plot, you usually select tags that are related: pump flow, speed, temperature, and pressure over a specific period. When plant operation is not consistent, it is useful to compare the operation of a tag, such as a pump, during normal operations against its operation during a suboptimal period.

### To create a time period

1. Add one or more tags to the plot surface.
2. On the XY Plot toolbar, click **New Time Period**  The **Time** bar changes from blue to red to indicate that you can create a time period.

3. On the **Time** bar, do one of the following:
  - Click the down arrow beside the first date and time entry and select the start date and time. Select an end date and time and then click **Refresh**  between the two dates to refresh the trace.
  - Click the down arrow beside the Last 5 minutes entry and select a relative time period.

XY Plot creates a trace definition using the same tags as the first time period.

## Use shapes (XY Plot)

Shapes are a powerful analyzing tool that allows quick detection of normal or abnormal data behavior.

### Create shapes from existing traces

When creating a shape from an existing trace, XY Plot analyzes the underlying data and makes its best effort to mimic the trace data. The resulting shape will usually follow the shape of the data represented by the trace, but occasionally some data points may not be included in the shape. If this occurs, you can refine your shape.

To create a shape from an existing trace, right-click the plot surface and select **Shape > New from Trace**. A shape is added to the XY Plot.

### Create shapes

When you create a shape that is not based on an existing trace, a rectangular shape displays on the plot surface and you can manually change the shape.

#### To create a shape:

1. Right-click the plot surface and select **Shape > New Default**. A rectangular shape is added to the plot and you are placed in edit mode.
2. Use the cursor to drag the square boxes on the corners of the shape to change the shape.
3. Right-click the plot surface and select **Exit Edit** when finished.

## Zoom and pan data (XY Plot)

You can zoom in on or pan data on an XY Plot for more detailed analysis.

#### To zoom in on an XY Plot, do any of the following:

- Double-click an area of the plot surface that you want to zoom in on.
- Press CTRL-plus on the keyboard.
- At the bottom right of the plot surface, click .
- Scroll the mouse wheel up. Note that if you use the mouse wheel, the XY Plot zooms relative to the location of the cursor on the plot surface.

#### To zoom out of an XY Plot, do any of the following:

- Shift-click an area of the plot surface that you want to zoom out of.
- Press CTRL-minus on the keyboard.
- At the bottom right of the plot surface, click .
- Scroll the mouse wheel down. Note that if you use the mouse wheel, the XY Plot zooms relative to the location of the cursor on the plot surface.

**To pan data, select one of the following directions:**

- Previous data. Press and hold the left mouse button and drag right.
- More recent data. Press and hold the left mouse button and drag left.
- Up. Press and hold the left mouse button and drag up.
- Down. Press and hold the left mouse button and drag down.

## Adding logic and control

This chapter describes:

- What logic and control is.
- Creating and using macros.
- Creating and using derived tags.
- Defining and using events.
- Creating and using client keys.

### About logic and control

In FactoryTalk View Studio, the **Logic and Control** folder contains editors for setting up alternative ways of manipulating tag data and issuing commands at runtime.

You can set up:

- **Derived tags** to derive values from combined tag data.
- **Events** to trigger actions when specified events occur.
- **Macros** to issue multiple FactoryTalk View SE commands in sequence.
- **Client keys** to assign commands to keyboard keys independent of objects or displays, so that they are always available at runtime.

### Creating and using derived tags

A derived tag is a tag whose value is the result of an expression. An expression can be simply a tag name, or it can be an equation made up of tag names, mathematical and logical operations, special functions, and **If-Then-Else** logic.

Derived tag calculations are carried out at the FactoryTalk View SE Server (also called the HMI server).

A derived tags component is a file that contains the definitions of one or more derived tags. A derived tags component can be run when an HMI server starts, or it can be run using the **DerivedOn** command, after the HMI server has started.

### Parts of the Derived Tags editor

Use the **Derived Tags** editor to create derived tag components for an application.

#### To open the Derived Tags editor:

1. In FactoryTalk View Studio, in the Explorer window, open the Logic and Control folder.
2. Right-click the Derived Tags icon, and then select **New**.

You can also open the Derived Tags editor by dragging its icon into the FactoryTalk View Studio workspace, or by double-clicking an existing Derived Tags component.

### Accept and Discard buttons (Derived Tags editor)

When you enter information in the **Derived Tags** editor, the **Prev** and **Next** buttons change to **Accept** and **Discard** buttons.

Click **Accept** to save the derived tags' information. Click **Discard** to cancel changes to the derived tags file.

## Form and spreadsheet (Derived Tags editor)

In the upper part of the form, provide a name and description for the derived tag.

In the **Expression** box, create the expression that will determine the derived tag's value. You can use the buttons beneath the box to build expressions. For more information, see [Creating expressions on page 344](#).

## Checking the syntax of an expression (Expression editor)

In the Derived Tags editor, to verify that the expression you typed uses correct syntax, click **Check Syntax**. If the syntax is invalid, an error is shown in the Check Syntax box.

You can check the syntax of an expression at any time while the Derived Tags editor is open. The syntax is also checked automatically when you click **Accept** or **OK** in the editor.

## How to use derived tags

Here's an example of how a derived tag can be used. Suppose that there are five weight sensors on a conveyor belt. The tag database contains one tag for each sensor, so the weight at each point on the conveyor belt is monitored.

If the weight at any point is excessive, FactoryTalk View triggers an alarm. However, if no individual sensor detects an excessive weight, but the total of all five sensors is too high, an alarm might be triggered.

To handle this case, you could set up a derived tag to sum the weights of all five sensors. Then, if this total, which would be the value of the derived tag, is too high, FactoryTalk View can trigger an alarm.



**Tip:** Writing to a tag whose value is the result of a derived tag expression is not recommended, because a derived tag is reevaluated whenever tags in the expression change.

For example, if a derived tag named Tag3 is defined by the expression Tag1 + Tag2, you can set Tag3 to zero; however, the value of Tag3 will change again, when either Tag1 or Tag2 changes.

## Creating a derived tags component

You can use any tag to store the result of a derived tag calculation. The tag must already exist, for example, as an HMI tag (memory or device) or in a data server, before you can use it as a derived tag.

**NOTE:** Do not create derived tags that depend on the results of other derived tags.

Derived tags processing is not sequential. This means the results of an expression that includes other derived tags might not occur in the desired scan or order of evaluation. To avoid this, it is best to put all the required logic in the expression.

These are the tasks involved in setting up derived tags for an application:

1. Create the derived tags in the **Derived Tags** editor in FactoryTalk View Studio.
2. Set the maximum update rate in the **Derived Tag Setup** dialog box.
3. Save the derived tags component and give it a name.

## To create a derived tags component:

1. In the **Derived Tags** editor, specify the tag that will hold the derived value and type a description of the tag.
2. In the **Expression** box, create an expression that will determine the derived tag's value. For information about expressions, see [Creating expressions on page 344](#).
3. Click **Accept**.
4. If the derived tags component is to contain more than one derived tag, click **Next** to create another derived tag.
5. Repeat steps 1 through 4 until all derived tags for the particular component are defined.
6. From the **Setup** menu, select **Derived Tag Setup**.
7. In the **Derived Tag Setup** dialog box type a description of the component and specify a maximum update rate, and then click **OK**.
8. In the **Derived Tags** editor, click **Close**.
9. Specify the name of the derived tags component and then click **OK**.

For details about options in the **Derived Tags** editor and in the **Setup dialog** box, click **Help**.

## About the maximum update rate

The maximum update rate is the fastest rate, at which any event expression in an events component can be updated. The maximum update rate also determines the fastest rate, at which data servers can send changes in tag values.

Set a maximum update rate separately, for each event component you create.

Set the update rate as fast as, or faster than, the rate, at which the values of tags used in the expressions, change, unless it is desirable to miss changes in tag values.



**Tip:** Expressions containing derived tags are evaluated only when the value of a tag or function in the expression changes.

## Using multiple derived tag components

You can create multiple derived tag components, for example, to group derived tags that need to be evaluated at different rates.

Each HMI server in a network distributed application can run up to 20 derived tag components, with up to 1,000 tags in each component.

## Modifying existing derived tag components

You can modify derived tags during development, or when you run the application.

If you change a derived tags component at runtime, the changes will not take effect until you stop and then restart derived tags processing. For information about starting and stopping derived tags, see [Starting and stopping derived tags processing on page 425](#).

## Starting and stopping derived tags processing

There are many ways to start and stop derived tags processing. Choose the way that works best for the application.

To provide an operator with a way to start and stop derived tags processing, use FactoryTalk View commands and macros in conjunction with graphic objects. For details about the FactoryTalk View command syntax, see the FactoryTalk View Site Edition Help.

In the following methods, <component> represents the name of a derived tags file.

## Starting derived tags in network and local station applications

For network applications, it is best to start derived tags processing using a server component such as a startup macro, or using event processing.

Using clients to start and stop derived tags is not recommended, because one client might stop a derived tags component that another client is using.

Use one of the following methods for network applications:

- In the **HMI Server Properties** dialog box, click the **Components** tab, select the checkbox, **Derived tags**, and then select a derived tag component.  
The derived tag component will start the next time the HMI server runs, or when the HMI server's components are started manually.
- In the **Macros** editor, create a macro that contains the command, **DerivedOn <component>**. In the **HMI Server Properties** dialog box, click the **Components** tab, select the **Macro** checkbox, and then specify the macro. Derived tags processing will start when the HMI server runs.
- In the **Events** editor, specify the **DerivedOn <component>** command as the action for an event.
- At the command line in FactoryTalk View Studio, or the FactoryTalk View SE Administration Console, type **DerivedOn <component>** and then press **Enter**.

## Starting derived tags in local station applications only

The following methods, which use clients to start derived tags processing, are recommended for local station applications only:

- In the **Graphics** editor, create a button and then specify the **DerivedOn <component>** command as the button's press action. When the button is pressed, derived tags processing starts.
- In the **Graphics** editor, create a graphic object and then attach a touch animation with the **DerivedOn <component>** command as the object's action. When the object is touched, derived tags processing starts.
- For the Login Macro in the FactoryTalk View SE User Accounts editor, specify the **DerivedOn <component>** command, or a macro that contains the command.
- For the **Startup** command in the **Display Settings Behavior** tab, specify the **DerivedOn <component>** command, or a macro that contains the command.

## Stopping derived tags in network and local station applications

Having multiple clients stop derived tags processing is not recommended. Do this in special cases only, for example, for application maintenance.

To stop a single derived tag component, use one of these methods:

- In the **Macros** editor, create a macro that contains the command, **DerivedOff <component>**. In the **HMI Server Properties** dialog box, click the **Components** tab, select the checkbox, **On shutdown macro**, and then specify the macro.
- Derived tags processing will stop when the HMI server shuts down.
- In the **Events** editor, specify the **DerivedOff <component>** command as the action for an event.
- At the command line in FactoryTalk View Studio, or the FactoryTalk View SE Administration Console, type **DerivedOff <component>** and then press **Enter**.

## Stopping derived tags in local station applications only

The following methods, which use clients to stop derived tags processing, are recommended for local station applications only:

- In the **Graphics** editor, create a button and then specify the **DerivedOff <component>** command as the button's press action. When the button is pressed, derived tags processing stops.
- In the **Graphics** editor, create a graphic object and then attach a touch animation with the **DerivedOff <component>** command as the object's action. When the object is touched, derived tags processing stops.
- For the **Logout Macro** in the FactoryTalk View SE User Accounts editor, specify the **DerivedOff <component>** command, or a macro that contains the command.
- For the **Shutdown** command in the **Display Settings Behavior** tab, specify the **DerivedOff <component>** command, or a macro that contains the command.

## Setting up FactoryTalk View SE events

Events are expressions that trigger actions:

- Expressions are equations containing tag values, mathematical operations, if – then – else logic, and other built-in FactoryTalk View functions. For information about using expressions, see [Creating expressions on page 344](#).
- Actions are FactoryTalk View commands, symbols, or macros. For example, an action could initiate a snapshot of tag values using the **DataLogSnapshot** command, or change a tag value using the **Set** command.

---

**NOTE:** Events are detected at the FactoryTalk View SE Server (also called the HMI server); therefore, commands that run only at the FactoryTalk View SE Client are ignored when issued by an event expression. For a list of these commands, see [FactoryTalk View commands on page 442](#).

---

An events component is a file that contains the definitions of one or more events. An events component can be run when an HMI server starts, or it can be run using the **EventOn** command after the HMI server has started.

## Parts of the Events editor

Use the **Events** editor to set up events processing for an application.

### To open the Events editor:

1. In FactoryTalk View Studio, in the **Explorer** window, open the **Logic and Control** folder.
2. Right-click the **Events** icon, and then click **New**.

You can also open the **Events** editor by dragging its icon into the FactoryTalk View Studio workspace, or by double-clicking an existing Events component.

## Accept and Discard buttons (Events editor)

When you enter information in the **Events** editor, the **Prev** and **Next** buttons change to the **Accept** and **Discard** buttons. Click **Accept** to save information. Click **Discard** to cancel changes to the event component.

## Form and spreadsheet (Events editor)

In the upper part of the form, specify the action and provide a description for the event component.

In the **Expression** box, create the expression that will determine when the action is performed. You can use the buttons beneath the box to build expressions. For more information, see [Creating expressions on page 344](#).

## Checking the syntax of an expression (Events editor)

In the **Events** editor, to verify that the expression you typed uses correct syntax, click **Check Syntax**. If the syntax is invalid, an error is shown in the **Check Syntax** box.

You can check the syntax of an expression at any time while the **Events** editor is open. The syntax is also checked automatically, when you click **Accept** or **OK** in the editor.

## Creating an events component

An event component consists of two parts: an expression and an action.

When a component in the expression changes, the expression is evaluated. If the expression's true state changes from false on the previous evaluation, to true on the current evaluation, the action is triggered.

These are the tasks involved in setting up events for an application:

1. Create the events in the **Events** editor.
2. Set the maximum update rate for the events component in the **Events Setup** dialog box.
3. Save the events component and give it a name. The file extension .eds is added by the system.

### To create an events component:

1. In the **Events** editor, in the **Action** box, specify the command, macro or symbol that will run when the event expression evaluates to true.
2. In the **Expression** box, create an expression that will determine the conditions for triggering the action. For more information, see [Creating expressions on page 344](#).
3. Click **Accept**.
4. If the events component is to contain more than one event, click **Next** to create another event.
5. Repeat steps 1 through 4 until all events for the particular component are defined.
6. From the Setup menu, select **Event Setup**.

7. In the Event Setup dialog box, type a description of the component, specify a maximum update rate, and then click **OK**.
8. Click **Close**.
9. Provide a name for the events component, and then click **OK**.

For details about options in the **Events** editor and in the **Setup** dialog box, click **Help**.

## About the maximum update rate

The maximum update rate is the fastest rate, at which any event expression in an events component can be updated. The maximum update rate also determines the fastest rate, at which data servers can send changes in tag values.

Set a maximum update rate separately, for each event component you create.

Set the update rate as fast as, or faster than, the rate, at which the values of tags used in the expressions, change, unless it is desirable to miss changes in tag values.



**Tip:** Expressions containing derived tags are evaluated only when the value of a tag or function in the expression changes.

## Using multiple events components

You can create multiple events components, for example, to group events that need to be evaluated at different rates.

Each HMI server in a network distributed application can run up to 20 event components, with up to 1,000 tags in each component.

**NOTE:** Do not create events that depend on other events. Events are not processed sequentially.

## Modifying existing event components

You can modify events during development, or when you run the application.

If you change an event component at runtime, the changes will not take effect until you stop and then restart events processing.

## Starting and stopping events processing

There are many ways to start and stop events processing. Choose the way that works best for the application.

To provide an operator with a way to start and stop events processing, use FactoryTalk View commands and macros in conjunction with graphic objects. For details about the FactoryTalk View command syntax, see the FactoryTalk View Site Edition Help.

In the following methods, *<component>* represents the name of an events file.

## Ways to start events processing

To start events processing, use any of these methods:

- In the **HMI Server Properties** dialog box, click the **Components** tab, select the **Events** checkbox, and then select an event component.  
The event component will start the next time the HMI server runs, or when the HMI server's components are started manually.
- In the **Macros** editor, create a macro that contains the command, **EventOn <component>**. In the **HMI Server Properties** dialog box, click the **Components** tab, select the **Macro** checkbox, and then specify the macro. Event detection will start when the HMI server runs.
- In the **Graphics** editor, create a button object and then specify the **EventOn <component>** command as the button's press action. When the button is pressed, event detection starts.
- In the **Graphics** editor, create a graphic object and attach a touch animation to it, using **EventOn <component>** in the **Action** box. When the object is touched, event detection starts.
- At the command line in FactoryTalk View Studio, or in the FactoryTalk View SE Administration Console, type **EventOn <component>**, and then press **Enter**.
- For the Login Macro in the FactoryTalk View SE User Accounts editor, specify the **EventOn <component>** command, or a macro that contains the command.
- For the Startup command in the **Display Settings Behavior** tab, specify the **EventOn <component>** command, or a macro that contains the command.

## Ways to stop events processing

To stop a single event component, use any of these methods:

- In the **Macros** editor, create a macro that contains the command, **EventOff <component>**. In the **HMI Server Properties** dialog box, click the **Components** tab, select the checkbox, **On shutdown macro**, and then specify the macro.  
Events processing will stop when the HMI server shuts down.
- In the **Graphics** editor, create a button object and then specify the **EventOff <component>** command as the button's press action. When the button is pressed, event detection stops.
- In the **Graphics** editor, create a graphic object and attach a touch animation to it, using **EventOff <component>** in the **Action** box. When the object is touched, event detection stops.
- At the command line in FactoryTalk View Studio, or the FactoryTalk View SE Administration Console, type **EventOff <component>**, and then press **Enter**.
- For the Logout Macro in the FactoryTalk View SE User Accounts editor, specify the **EventOff <component>** command, or a macro that contains the command.
- For the **Shutdown** command in the **Display Settings Behavior** tab, specify the **EventOff <component>** command, or a macro that contains the command.

You can also stop events processing by stopping all HMI server components running on the computer. For details, see [To stop HMI server components manually on page 139](#).

## Creating and using macros

A macro is a series of FactoryTalk View SE commands stored in a macro component.

You can use a macro component anywhere a command can be used. When the component name is issued, the macro runs, carrying out all the commands in the component.

You can create macros to perform almost any action or series of actions. For example, use a macro to:

- Open a group of graphic displays, and define their initial positions.
- Define temporary key definitions.
- Close any open graphic displays.
- Set tag values.

### To create a macro:

- In FactoryTalk View Studio, open the **Logic and Control** folder, right-click the **Macros** icon, and then select **New**.

To open the **Command Wizard**, double-click anywhere in the **Macros** editor.

## Macro syntax

When typing commands in macros, follow these guidelines:

- Separate each identifier, specification, or string with a space or a tab.
  - Start each command on a new line, or separate commands on the same line with a semicolon ( ; ).
  - To insert a comment, type an exclamation mark ( ! ) at the beginning of a new line or immediately after a semicolon ( ; ). Everything between the exclamation mark and the end of the line, or the next semicolon, is treated as a comment.
  - To replace a tag name with its current value when a command is evaluated, enclose the tag name in dollar signs ( \$ ) to create a tag placeholder in the command.
- When a macro containing \$tag\$ runs, the current tag value is substituted for the tag placeholder, before commands in the macro are issued.
- To indicate percent in a macro, use two percent signs ( % % ). One percent sign indicates a parameter in the macro. For details, see [Specifying parameters in a macro on page 432](#).

---

**NOTE:** Do not use dollar signs or percent signs in comments inserted in macros. This causes errors at runtime.

---

### Example 1: A macro called Factory

A macro called Factory contains the following commands:

```
Display /Ingredients::Overview Display /Ingredients::Detail /Ingredients::Valve23 =
open
```

When the macro runs, the graphic display called *Overview* opens, then the display called *Detail* opens, and then the tag *Valve23* is set to its open state.

The displays and the tag are all in the area called *Ingredients*.

### Example 2: A macro using placeholders in commands

The first two entries in this macro are **Display** commands that use placeholders:

```
Display /$Tag1$::Process1 Display /$Tag1$::TrendDisplay$Tag2$ Valve23 = Open
```

When the macro runs, FactoryTalk View replaces the placeholders (\$Tag1\$, \$Tag2\$) with the current values of the specified tags.

For example, if Tag1(a string tag)=*Mixing*, and Tag2(an analog tag)=2, then the graphic display Process1 from the Mixing area opens, followed by the display TrendDisplay2, also from the Mixing area.

Finally, the tag Valve23 from the home area is set to its open state. In this example, the home area is also the area, in which the macro is run.

You don't need to put brackets around tag names when using placeholders in commands.

## Specifying parameters in a macro

Macros can accept up to nine parameters. To specify a parameter in a macro, type a percent ( % ) sign followed by a number.

To run the macro and parameters, specify the macro name followed by the parameters. Separate multiple parameters with spaces.

### Example: Factory macro with two parameters

Re-create the Factory macro described in Example 1, as follows:

```
Display Overview
Display %1
Valve23 = %2
```

To run the macro, type the macro name with the parameters to substitute for the placeholders %1 and %2:

```
Factory Detail Open
```

When the macro runs, it substitutes *Detail* wherever %1 occurs in the macro, and substitutes *Open* wherever %2 occurs in the macro.

The actions performed are the same as in Example 1. However, in this example, the macro runs in the home area instead of the Ingredients area.

## Typing macro names that contain spaces

If the name of a macro contains spaces, enclose the name in quotation marks when typing the macro name. If the macro contains parameters, place the parameters outside the quotation marks.

### Example: Macro name with spaces and parameters

The name of the macro in Example 3 is *Factory Display*, instead of *Factory*. To run the macro, type the following at the command line:

```
"Factory Display" Detail Open
```

## Nesting macros

You can insert a macro within another macro – this is called nesting. A macro can have up to eight nesting levels.



**Tip:** The single quotation mark ('') is a special character in FactoryTalk View and cannot be used in nested macros. Instead, use the double quotation mark ("").

### Example: Nesting macros

A macro called *Draw* contains:

```
Display Overview Display Detail
```

and a macro called *Factory* contains:

```
Draw Valve23 = Open
```

With the *Draw* macro nested in the *Factory* macro, running the *Factory* macro performs the same actions as in Example 3.

## Running macros

At runtime, an operator can run a macro anywhere they can run a FactoryTalk View command. Ensure you include a way for the operator to run commands or macros, for example, by specifying them as press actions for buttons.

For more information about running macros, see Help.

### Running a macro when an HMI server starts

If you have created an HMI server with components that must start in a particular order, create a *Startup* macro. To make this macro run when the HMI server starts, select it as the startup macro in the **Components** tab of the **HMI Server Properties** dialog box.

For details about specifying the startup macro:

- In a network distributed application, see [Selecting startup and shutdown components on page 127](#).
- In a local station application, see [Selecting startup and shutdown components on page 148](#).

## Specifying user login and logout macros

You can assign login and logout macros to any user or group of users listed in the **Runtime Security** editor, in FactoryTalk View Studio. Any macro component, containing any FactoryTalk View command, can be a login or logout macro.

For example, a login macro might contain a command to bring up a specific graphic display in an area of the plant. A logout macro might contain commands to redefine sensitive keys.

A user's login macro runs when the user signs in to the FactoryTalk View SE Client.

If the client is already running when the user signs in, the current user is logged off. If the current user is assigned a logout macro, the logout macro runs once the new user's name and password are accepted.

In a network distributed application, a login or logout macro will only run in the area where it's located.

For example, if you specify a login macro in the **Bottling** area, when the user signs in to a FactoryTalk View SE Client, the macro will run only if the **Bottling** area is the client's home area.



**Tip:** You specify the client's home area when you select components for the client configuration file. For more information, click Help in the FactoryTalk View SE Client wizard.

For more information about logging on and off at runtime, and about assigning login and logout macros to users, see [Setting up security on page 77](#).

## Creating client keys

A client key is associated with commands or macros that run when the key is pressed.

Client keys are similar to object keys and display keys, except client keys are not dependent on a particular graphic object or display. Client keys are associated with the FactoryTalk View SE Client, and are active at all times.

Use client keys to let operators perform actions without having to open a particular graphic display. For example, you might create a client key that recalls an overview menu, to let operators open the menu regardless of where they are in the application.

To create a client key, you create a client key component in the **Client Keys** editor. A single component can contain more than one client key definition.

### To open the Client Keys editor:

In FactoryTalk View Studio, in the **Explorer** window, open the **Logic and Control** folder, right-click **Client Keys**, and then select **New**.

For details about the options in the editor, click the **Help** button.

## Running client key components

To run a client key component, use one of these methods:

- Use the FactoryTalk View **Key** command in a macro or anywhere else you can use the FactoryTalk View commands. For details, see the FactoryTalk View Site Edition Help.
- Run the client key component when the FactoryTalk View SE Client runs. For details, see Help for the FactoryTalk View SE Client Wizard.



**Tip:** You cannot run more than one client key component at a time. If you try to do this, the components are not merged; instead, the second component overrides the first component.

# Managing recipes

This chapter describes:

- What recipes are
- The parts of RecipePro+ editor
- Create, modify, and delete recipes
- Refresh recipe values
- Import or export recipes

## About recipes

A recipe is a specific set of numeric and string data values (ingredients) that are associated with a specific set of tags. The function of recipes is to allow a runtime operator to write each data value to its associated tag in one operation.

Each ingredient has a pre-set data value assigned to it. The set of numeric and string tags assigned to the ingredients in the recipe is called a tag set. The ingredients, tag sets, and data values are stored together in a recipe file, all arranged in an easy-to-use spreadsheet.

Each ingredient must have a unique name. A recipe can have up to 15,000 ingredients. You can edit, import, or export recipes at design time and runtime. You can also write the recipe data values to tags, or write tag values to recipes at runtime.

The recipes can be used for manufacturing food and beverages. They can also be used for any application where you want to show, edit, or save multiple values at once. For example, recipes are used in the petrochemical and pharmaceutical industries. In the pharmaceutical industry, you could use recipes to design flexible packaging, creating recipes that specify the number of tissues to put in a box.

## About the RecipePro+ editor

The RecipePro+ editor extends the functionality of the Recipe editor with advanced recipe management. You can use the RecipePro+ editor to create, duplicate, edit and preview the recipes. By refreshing recipe values, you can view the differences between the current tag values and the preset recipe values. You can also use it to export recipes to CSV files or import recipes from CSV files.

By using the FactoryTalk View command **RecipeProEdit**, you can open the RecipePro+ editor with full functionality at runtime. In this case, you do not need a FactoryTalk View Studio license. You can also open the RecipePro+ editor with fewer functionalities by using the FactoryTalk View command **RecipeProList**.



**Tip:** The earlier operations are initiated by RecipePro+ commands that are assigned to the actions of graphic objects. Therefore, there must be a graphic object like a button on the display for each operation that you want to perform. For details about using the **Command Wizard** and about specific FactoryTalk View commands, see FactoryTalk View SE Help.

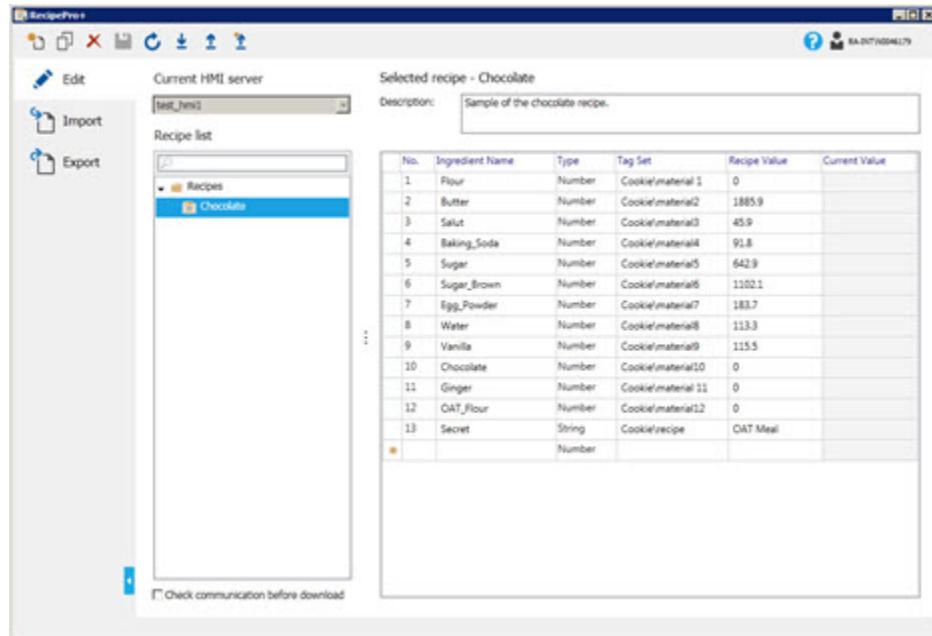
### To open the RecipePro+ editor:

In FactoryTalk View Studio, in the Explorer window, under the HMI server, double-click **RecipePro+ > RecipePro+ Editor**.



**Tip:** You can also run the **RecipeProEdit** command in the SE display client to open the editor with full functionalities. For more information about the command, see the FactoryTalk View SE Help.

The following example shows an overview of the editor:



## The parts of RecipePro+ editor

The RecipePro+ editor consists of these parts:

- **Toolbar**

Contains commonly used actions, such as creating, deleting, and refreshing recipes. When you point to an icon on the toolbar, its description is displayed below it.

- **Navigation bar**

Contains tabs to edit, import, or export recipes. You can click the left-pointing or right-pointing pointer at the bottom to collapse or expand the tabs.

- **Recipe file editor**

Provides an editable area where you can set up and configure recipes, specify importing or exporting options.



**Tip:** The **Current HMI Server** list is not available at design time. It is only available at runtime.

## Recipe edit table

The recipe file edit table shows the ingredients, tags, and data values of the selected recipe. You can edit these values, insert or delete rows in the table as needed.

The following table shows the columns of the recipe table.

Column	Description
No.	Shows the ingredient sequence number.
Ingredient Name	Specify a unique name for the ingredient. Examples might be <i>Sugar</i> , <i>Nitric acid</i> , <i>Secret sauce</i> , <i>55mm carriage bolts</i> and so on.
Type	Specify whether the data value should be interpreted as a number or a string. <ul style="list-style-type: none"> <li>• If the type is <b>Number</b>, the value, if present, must be a valid integer or floating point number.</li> <li>• If the type is <b>String</b>, the value, if present, can be any string.</li> </ul>
Tag Set	Specify the tag name associated with the data value for the ingredient. You can type the tag names or select tags from the <b>Tag Browser</b> .
Recipe Value	Specify the data value of the tag associated with the ingredient.
Current Value	Shows the current value of the tag associated with the ingredient when you click <b>Refresh</b> . After you've refreshed the current values, you can click the sort button next to <b>Current Value</b> to sort the current values that are different from the preset recipe values.

## Toolbar commands

The following table shows the commands on the toolbar of the **Edit** tab.

Commands	Description
 Create	Select this icon to create a recipe. Each recipe must have a unique name. When you create a recipe, you define the ingredients that make up the recipe, whether they are number or string type, what the data values for each ingredient will be, and what tag each value will be written to.
 Duplicate	Select this icon to duplicate an existing recipe.
 Delete	Select this icon to delete a recipe.
 Save	Select this icon to save changes for a recipe.
 Refresh	Select this icon to refresh the current and preset values of tags associated with the ingredient and show them in the recipe edit table. If the current values are different from the preset values, the current values are displayed in red.

Commands	Description
<b>Upload</b> 	<p>Select this icon to read the tag values from the location specified in the tag set of the recipe file and save those values to the recipe.</p> <p><b>Note:</b> The uploading stops if an error occurs, but the system continues to validate the recipe file and logs the error messages to FactoryTalk Diagnostics. It may take some time if a large number of errors occurs. During that time, you may have a slow response if using the RecipePro+ editor. To avoid this, use the RecipePro+ editor when the FactoryTalk Diagnostics logging completes.</p>
<b>Upload and create</b> 	<p>Select this icon to read the tag values from the location specified in the tag set of the recipe file and save those values to a new recipe. The new recipe is duplicated from the recipe selected.</p>
<b>Download</b> 	<p>Select this icon to write the ingredient values of the specified recipe file to the tags in the tag set.</p> <p>By default, only tags that are accessible will be downloaded. To check whether all tags are accessible before downloading, select <b>Check communication before download</b>. If one or more tags are not accessible, the download command will not be executed.</p>

## Create, modify, or delete recipes

Use the **Edit** tab of the RecipePro+ editor to create, modify, duplicate, and delete recipes.

### To create a recipe:

1. In the **Edit** tab, under the **Current HMI Server** list, if needed, select another HMI server where the recipe is.
2. On the toolbar command, click **New** .



**Tip:** You can also right-click **Recipes** or any recipe under the **Recipe list** and select **New**.

3. In the **New Recipe File** dialog box, enter a unique name for the recipe and click **OK**. The recipe will be displayed in the recipe list.
4. If needed, type the description for the recipe.
5. In the recipe edit table, specify the ingredient names, data types, associated tag sets, and preset recipe values.
6. When complete, click **Save**  to save the changes.

## To modify a recipe:

1. In the RecipePro+ editor, under the **Recipe list**, select the recipe you want to modify.
2. In the recipe edit table, modify the ingredient names, data types, associated tag sets, and preset recipe values.
3. When complete, click **Save**  to save the changes.

## To duplicate a recipe:

1. In the RecipePro+ editor, under the **Recipe list**, select the recipe you want to duplicate and click **Duplicate** .



**Tip:** You can also right-click the recipe and select **Duplicate**.

2. In the **Duplicate Recipe File** dialog box, enter a unique name for the recipe and click **OK**. The recipe will be displayed in the recipe list.
3. Modify the recipe if needed.

## To delete a recipe:

1. In the RecipePro+ editor, under the **Recipe list**, select the recipe you want to delete and click **Delete** .



**Tip:** You can also right-click the recipe and select **Delete**.

2. In the **Confirmation** dialog box, click **Yes**.

## Refresh recipes

If the current and preset values of tags associated with the ingredient are changed in a recipe, you can refresh these values and show them in the recipe edit table. If the current values are different from the preset values, the current values are displayed in red.

## To refresh recipes:

In the RecipePro+ editor, under the **Recipe list**, select the recipe you want to refresh and click **Refresh** .

Alternatively, you can right-click the recipe and select **Refresh**.

## Import or export recipes

Use the RecipePro+ editor to import recipes from CSV files or export recipes to CSV files.

- The imported recipe names are the same as the CSV file names. For example, if the CSV file is *Sugar.csv*, the recipe name after importing is *Sugar*.
- Each recipe will be saved to a CSV file with the same name as the recipe. For example, the recipe *Sugar* is saved to *Sugar.csv*.

## Required security permissions

Users or groups must be granted with the appropriate security permissions to perform certain actions in the RecipePro+ editor. You can set the security permissions on the **Permissions** tab. To view the permissions, in the **Security Settings** dialog box, navigate to **FactoryTalk View SE** and expand it.



**Tip:** To open the dialog box, in the FactoryTalk View Studio Explorer window, right-click the appropriate application or area and select **Security**.

Because security settings are inherited down the tree, choose the application or area at the highest possible level in the tree. For example, if you apply security settings at an application level, those settings apply to all of the areas within that application. If you apply security settings to a particular area, the settings apply to only that area and to other areas contained within that area.

To operate recipes in the RecipePro+ editor, users or groups must have at least the **RecipePro: Preview** permission. Otherwise, the recipes will not be visible in the editor.



**Tip:** The **RecipePro: Preview** permission is specifically designed for FactoryTalk ViewPoint mobile users. If you want to restrict certain users to view and access the recipes on mobile devices, you must deny this permission for those users.

The following table shows additional security permissions required for certain actions.

To do this	You need these additional security permissions
Create a recipe	RecipePro: Create
Delete a recipe	RecipePro: Delete
Download a recipe	RecipePro: Download
Duplicate a recipe	RecipePro: Create
Edit a recipe	RecipePro: Edit
Modify a tag set	RecipePro: Edit RecipePro: Modify TagSet
Import a recipe	RecipePro: Create RecipePro: Edit RecipePro: Modify TagSet
Rename a recipe	RecipePro: Edit
Upload a recipe	RecipePro: Edit RecipePro: Upload
Upload and create a recipe	RecipePro: Create RecipePro: Upload
RecipeProDownload command	RecipePro: Download
RecipeProUpload command	RecipePro: Edit RecipePro: Upload

If you have secured HMI tags, the security codes are not validated at runtime. Users or groups that are not assigned the tags' security code are able to perform the download or upload operation.

## Migrate RSView32 RecipePro to FactoryTalk View SE RecipePro+

After migrating an RSView32 project to FactoryTalk View SE, the existing recipes are also imported. To restore the previous functions in FactoryTalk View SE, edit the recipes in the RecipePro+ editor and configure the recipe display.

### Migrate RSView32 RecipePro to FactoryTalk View SE RecipePro+

1. Import an RSView32 project to FactoryTalk View SE.
2. In FactoryTalk View SE, open the RecipePro+ editor.

FactoryTalk View SE RecipePro+ has no concept of "unit" as in RSView32 RecipePro. The recipe in FactoryTalk View SE RecipePro+ is equivalent to 'unit' in RSView 32 RecipePro. After importing the project, units are automatically converted to *RecipeName\_UnitName* in RecipePro+ in FactoryTalk View SE.

3. Edit the recipe as needed, and then save the recipe file.
4. Open the display including the RecipePro Operator Console object or recipe command buttons.



**Tip:** A warning message may open, indicating that there is no equivalent to the RSView32 RecipePro Operator Console in FactoryTalk View SE. View SE has the RecipePro+ editor, which is available at design time and runtime with full functions that the RecipePro Operator Console object has. You can delete the placeholder for the RSView32 recipe control object and add a button associating the **RecipeProEdit** command to open the RecipePro+ editor at runtime.

The recipe file name format is changed:

- From **RecipeFile/Unit** to **RecipeFile\_Unit**

If the original display includes RecipePro command buttons, the command buttons will be automatically converted as below:

- **RecipeProEdit <RecipeFile>** to **RecipeProEdit**
- **RecipeProDownloadUnit <Unit>** to **RecipeProDownload <RecipeFile\_Unit>**
- **RecipeProUploadUnit <Unit>** to **RecipeProUpload <RecipeFile\_Unit>**

5. Click **Save** to save the changes.

## FactoryTalk View commands

This appendix describes:

- Where and how to use FactoryTalk View commands.
- Using absolute and relative references.
- Creating symbols.
- Using the Command Wizard.

For details about specific FactoryTalk View commands, see the FactoryTalk View Site Edition Help.

## Using commands in a FactoryTalk View SE application

In a FactoryTalk View SE application, you can use the commands in:

- Boxes that require you to enter an action.

For example, use a command as the press, release, or repeat action when you attach a touch animation to an object in a graphic display, or as the action for a button.

- Macros or symbols.
- The command line.

Commands take precedence over macros. For example, if you have a macro called *Display*, the **Display** command will run whenever you try to run the *Display* macro.

## How to use commands

When typing commands, or building them using the **Command Wizard**, keep the following guidelines in mind:

- In the descriptions of commands in the Help, parameters enclosed in angle brackets < > are required and parameters enclosed in square brackets [ ] are optional.
- Parameters do not have to be entered in the order they are listed in Help.
- Commands and parameters are not case-sensitive.
- To use several commands in sequence, start each new command on a new line, or separate commands on the same line with a semicolon ( ; ).
- Separate multiple parameters with a space.
- Enclose long file names containing spaces with double quotes when the file names are used as parameters.

For example:

```
Display PID /P"Temperature Loop 1"
```

- Enclose area and component names that contain spaces, or are ambiguous, in double quotes.
- An exclamation mark ( ! ) at the beginning of a new line or immediately after a semicolon ( ; ) indicates the start of a comment.

Everything between the exclamation mark and the end of the line, or the next semicolon, is treated as a comment.

- If a command accepts wildcard characters, this is noted in the description of the command in the Help. The wildcard characters are:

```
* matches any number of characters, including the backslash ( \ ) and period  
( . )
```

```
? matches any single character
```

## Using tag placeholders in commands

To replace a tag name with its current value when a command is evaluated, enclose the tag name in dollar signs (\$). This creates a tag placeholder in the command.

When using tag placeholders in commands, keep these points in mind:

- The maximum number of characters for floating point tag values is 17. If the value is longer than 17 digits, it is represented in scientific format.
- The maximum command length is 5000 characters. The command will be truncated if substituted tag values cause the command to exceed this length.

### Example 1: A macro using tag placeholders in commands

```
Display Screen$Tag1$  
Display $Tag3$$Tag2$  
Valve23 = Open
```

In the first two commands, the tag placeholders Tag1, Tag2, and Tag3 are string tags. When the macro runs, the tags have the values 1, 2, and 'Screen,' respectively.

Substituting the current values for the tag placeholders in commands in the macro, FactoryTalk View opens displays named Screen1 and Screen2, and then sets a tag named Valve23 to its open state.



**Tip:** If a tag placeholder is enclosed in double quotes (" ") or nested in the string value of another tag placeholder in the command, then FactoryTalk View will not substitute the tag value.

### Example 2: Ensuring that the tag value is evaluated as expected

FactoryTalk View evaluates tag placeholders before commands are run. For example, if the press action for a button object is:

```
Tag1 = 4  
Display Screen$Tag1$
```

when the button is pressed, the value of Tag1 will be substituted for the tag placeholder in the **Display** command before Tag1 is set to 4.

To ensure that the tag is set first, embed the command containing the tag placeholder in a macro that uses the tag as a parameter. Given a macro named *Display\_Detail*, this would change the button's press action to:

```
Tag1 = 4  
Display_Detail Tag1
```

Inside the macro, the tag placeholder is specified with the **Display** command, as follows:

```
Display Screen$%1$
```

Now when the button is pressed, FactoryTalk View will first set the tag to 4, and then run the **Display** command in the macro, opening a display named Screen4.

## Where commands run

Some commands run at the FactoryTalk View SE Server, some run at the FactoryTalk View SE Client, and some run wherever they were issued.

### Commands that run at the FactoryTalk View SE Server

Including:

- DataLogChangeRate, DataLogMergeToPrimary, DataLogNewFile, DataLogOff, DataLogOn, DataLogRenameFile, DataLogSnapShot, DataLogSwitchBack, DerivedOn, DerivedOff
- EventOff, EventOn
- RecipeProDownload, RecipeProUpload
- ScriptExecute

### Commands that run at the FactoryTalk View SE Client

Including:

- Abort, AE\_Acknowledge, AE\_ClearCount, AE\_Disable, AE\_Enable, AE\_Remark, AE\_Reset, AE\_Shelve, AE\_Suppress, AE\_Unshelve, AE\_Unsuppress
- Define, Display, DisplayClientClose, DisplayClientOpen, DisplayNavigationHistory, DisplayNavigationMenu, DisplayNextScreen, DisplayPreviousScreen, Download, DownloadAll
- FlushCache
- Help
- Invoke
- Key
- Language, Login, Logout
- NextPosition, NextWindow
- OpenRSLogix5000
- Password, Pin, Position, PrevPosition, PrevWindow, PrintDisplay, PullForward, PushBack
- RecipeProEdit, RecipeProList, RecipeRestore, RecipeSave
- ScreenPrint, SendKeys, SetFocus, Switchover
- Undefine, Unpin, Upload, UploadAll

### Commands that do not run in test display mode

Most commands that run at the FactoryTalk View SE Client will also run in FactoryTalk View Studio in test display mode. The following exceptions are ignored in test display mode:

- Abort
- Display, DisplayClientClose, DisplayClientOpen, DisplayNavigationHistory, DisplayNavigationMenu, DisplayNextScreen, DisplayPreviousScreen
- FlushCache
- Key
- Login, Logout
- NextWindow
- PrevWindow, PrintDisplay, PullForward, PushBack
- RecipeProEdit, RecipeProList
- SetFocus, Switchover

## Commands that run wherever they are issued

These commands run wherever they are issued. For example, if one of these commands is issued at the FactoryTalk View SE Client, it runs at the client.

- =(Equal)
- AppAbort, AppActivate, AppStart
- Beep
- DDEExecute
- If
- Pause, PlayWave
- Ramp, Remark
- Set
- Toggle

The **AppStart** command should not be used with interactive actions that use **Events**.

## Using absolute and relative references

You can use absolute or relative references to refer to application components, such as graphic displays and tags, when building FactoryTalk View commands or connecting graphic objects to process data.

---

**NOTE:** Local station applications use only relative references.

---

Absolute references work like file paths. Relative references work like file names.

**Absolute references** point directly at a specific component, by referring to the component's name and the area (or areas, in the case of nested areas) in which it is located. For example, an absolute reference to a graphic display called Detail in an area called AssemblyLineNorth is `/AssemblyLineNorth::Detail`.

An absolute reference to an application's root area does not include the application's name, even though the name is shown in the root area in FactoryTalk View Studio.

**Relative references** point at a component relative to the current server or area. For example, a relative reference to a graphic display named Detail is simply the display's name `Detail`. When a relative reference is used, FactoryTalk View assumes that the component is in the current area.

Use relative references, for example, to reuse component names in a network distributed application for a plant that has identical production lines. Such an application might contain different areas to represent each production line; however, each area would contain the same component names.

## How relative references are resolved

You only need to understand how relative references are resolved when you are creating network distributed applications. If you are creating network station or local station applications, you can skip this section.

Before FactoryTalk View can perform an action on a component that is specified using a relative reference, the relative reference must be resolved. To do this FactoryTalk View converts the relative reference to an absolute reference.

For example, if the current area is called RecipesArea, the graphic display specified using the relative reference *IngredientDisplay* is resolved to the absolute reference **/RecipesArea:IngredientDisplay** before it can be used.

For commands, relative references are resolved in different ways, depending on the type of parameters the command takes.

## Commands that take tags as parameters

Commands that take tags as parameters are resolved as follows:

- When the command runs from a graphic display, the area that contains the graphic display is used to resolve the relative reference.

For example, if the command **Display OverviewStats** runs from the graphic display called *IngredientDisplay* in the area *RecipesArea*, the area name *RecipesArea* is used to resolve the relative reference in the command.

If the command runs from the graphic display called *IngredientDisplay* in the *Mixing* area, the *Mixing* area name is used to resolve the relative reference in the command.

- When the command is run by an Events component, the *Alarm Identification* feature, or the *Startup* or *Shutdown* macros, the area that contains the HMI server is used to resolve the relative reference.

## Commands that take HMI project components as parameters

When the command runs from a graphic display, the area that contains the graphic display is used to resolve the relative reference.

Some commands are run by a part of the system that runs in the background. This background activity is managed by the HMI server.

For example, an Events component runs commands when events occur. These parts of the system resolve relative references using the area in which the HMI server is located, because the HMI server manages the background activity.

## Creating symbols

If you have long commands, or commands with parameters that are hard to remember or easy to mix up, you can rename those commands with a single word called a symbol.

Symbols can be used anywhere a command can be used: in a box that requires an action, in a macro, or at the command line.

You can define symbols in a macro or at the command line. However, symbols are mainly an operational convenience when using the system from the command line.

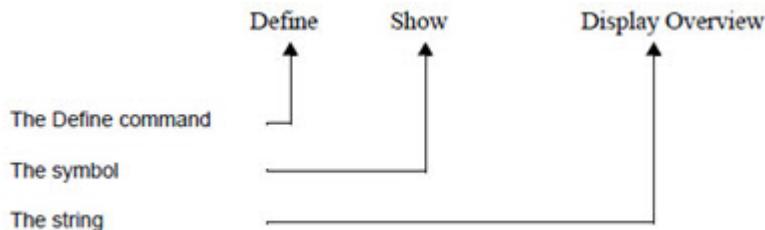
### To define a symbol:

At the command line or in a macro or box, type:

```
Define <symbol> <string>
```

- *<symbol>* is the abbreviated command, without spaces.
- *<string>* is an existing command with or without parameters. It can contain spaces and other symbols.

*Example: The Define command*



In this example, the command **Display Overview** is replaced with the word **Show**.

### To clear a symbol:

At the command line or in a macro or box, type one of the following:

This command	Does this
<code>Undefine &lt;symbol&gt;</code>  <symbol> is the name of the symbol that you want to delete.	Clears the symbol.
<code>Undefine *</code>	Clears all defined symbols.

*Example: Undefining symbols*

`Undefine Show`

Clears the symbol **Show**.

## Important guidelines

When creating symbols, keep the following in mind:

- FactoryTalk View does not check for security access on symbol names.  
Therefore, be sure to put security on the underlying FactoryTalk View command. For information about setting up security, see [Setting up security on page 77](#).
- If a symbol and a macro have the same name, the symbol runs instead of the macro.  
The order of precedence is: symbol, command, macro.

## Running and building commands

Use the command line to run commands during application development, testing, and maintenance.

Use the **Command Wizard** to build command strings.

The **Command Wizard** lists all FactoryTalk View commands and their syntax. The wizard also lists any macros that you have created.

For details about using the **Command Wizard**, and about individual FactoryTalk View commands, see Help.

### To open the Command Wizard:

- Click the **Browse** button beside a text box where a command is expected, for example, in the Press action text box for a button object, or in the Command Line.
- Double-click in a box where a command is expected or in the Macros editor.
- In the **Command Line**, **Runtime Secured Commands** editor, **Events** editor, and **Macro** editor, from the **Edit** menu, select **Commands**.

## Scripts

Use Scripts in FactoryTalk View SE to manage PowerShell script files and trigger with the ScriptExecute command. Scripts are graphic-independent and all HMI servers share script files under the same application. When executing the script, it runs on the HMI server that triggered the script. The execution supports synchronous and asynchronous waiting modes.

Scripts can read tag's values and write values to tags. You can edit the script files in any script editor and they are stored in the FactoryTalk Directory. You can also import or export multiple script files at a time.

For the entire sample script of reading and writing tags, see *FactoryTalk View Site Edition Help* **Example: PowerShell script for reading tags** and **Example: PowerShell script for writing tags**.

Supported script types include script files (.ps1), script modules (.psm1), and module manifest files (.psd1).

Difference between synchronous and asynchronous waiting modes

- **Synchronous** means that all scripts run on a shared thread. They execute in the order that they are triggered. If too many scripts are called all at once, it will take long periods of time to execute.
- **Asynchronous** means that all scripts run on its own thread. This is useful when a script takes a long time to execute, so that it does not hinder the execution of other scripts.

Because the script runs on the HMI server side, you may want to consider:

- Who can configure the script at design time.
- Who can run the ScriptExecute command that triggers the script.

To decide who can configure the script at design time, secure the resources in the explorer of FactoryTalk View Studio with common actions such as Write or List Children.

To secure the ScriptExecute command at runtime, do one of the following:

- Assign a security code to the ScriptExecute command via **Settings > Runtime Secured Commands** in FactoryTalk View Studio, so that only user groups with the same security code are allowed to run the command.
- Configure the disabled state or visibility of the button that triggers the ScriptExecute command, so that the button is only visible or enabled to user groups that are allowed to run the command.

# Creating and deploying .NET controls

This chapter describes:

- How to create Windows Forms controls.
- How to create WPF (Windows Presentation Foundation) controls.
- Deploying self-created .NET controls or third-party libraries.
- Handling complex data types of property and method parameters in .NET controls.

## Custom .NET control overview

The .NET Control object provides built-in Windows Forms and WPF controls. You can also import self-created controls or third-party libraries for more custom controls. For the custom controls to work properly in FactoryTalk View SE, you must follow specific rules when creating or deploying them.

## Required knowledge

Creating custom .NET controls requires the following knowledge:

- FactoryTalk View Studio
- Microsoft Visual Studio
- C# programming
- Microsoft .NET Framework and .NET managed code assemblies

## Environment requirements

- FactoryTalk View version 13.0 or later
- Microsoft Visual Studio 2017
- Microsoft SDK for Windows 10
- .NET Framework 4.8 developer pack (recommended)

## Creating custom Windows Forms controls

This section uses a calculator control as an example to explain how to create custom Windows Forms controls.



The calculator includes:

- Several function buttons
- An expression box
- A result box

After creating this control, you can:

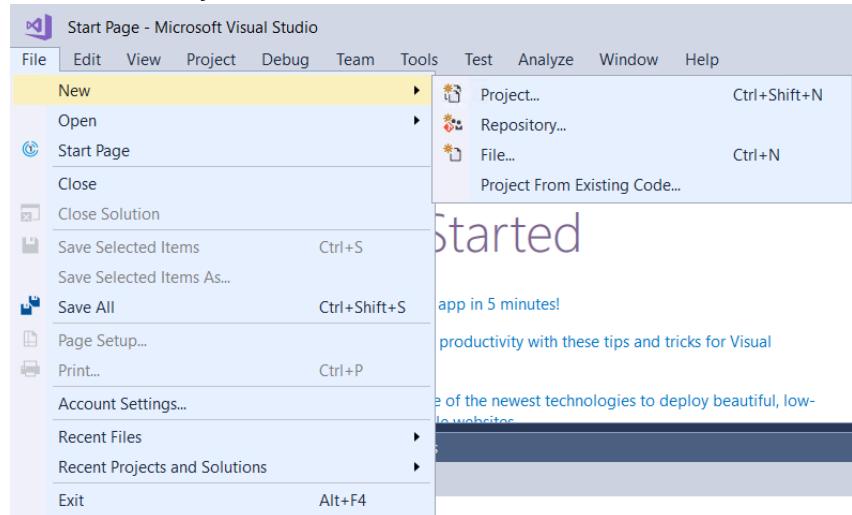
- Import it to FactoryTalk View Studio and draw it on a display.
- View and configure its properties.
- Set and get property values, invoke methods, and trigger events at runtime.
- Bind tags to its properties.

## Create a Windows Forms control in Visual Studio

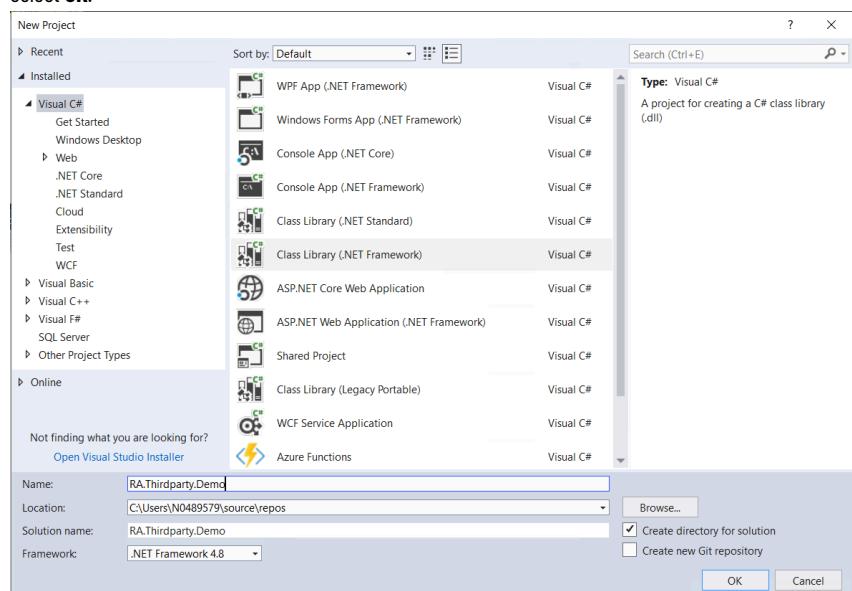
Use Visual Studio to create a custom Windows Forms control.

## To create a Windows Forms control

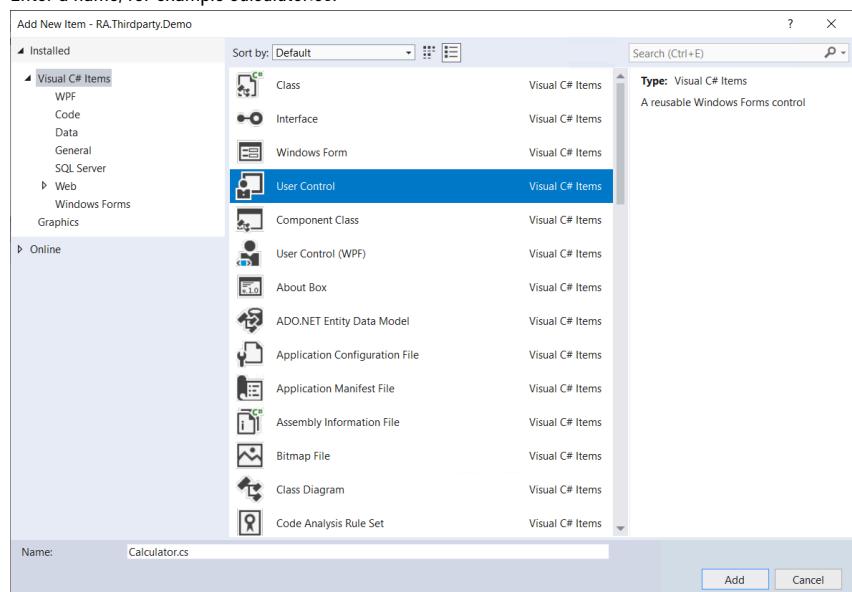
1. Create a project in Visual Studio.
  - a. Open Visual Studio.
  - b. Select **File > New > Project**.



2. In the **New Project** dialog box:
  - a. Select **Installed > Visual C# > Class Library (.NET Framework)**.
  - b. In the **Framework** list, select **.NET Framework 4.8**.
  - c. Enter a name and file path.
  - d. Select **OK**.



3. Add a user control.
  - a. Select **Project > Add user control.**
  - b. Enter a name, for example *Calculator.cs*.

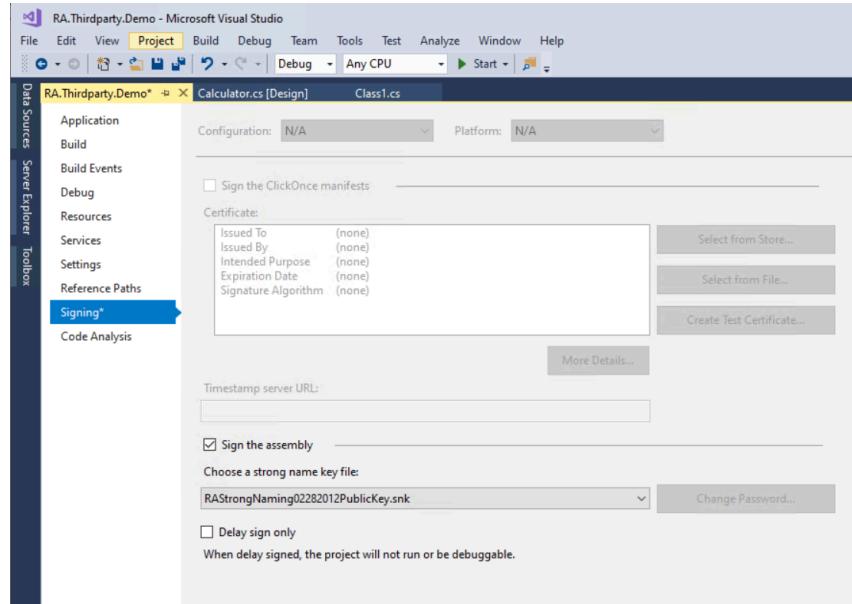


4. (optional) Configure strong name signing.

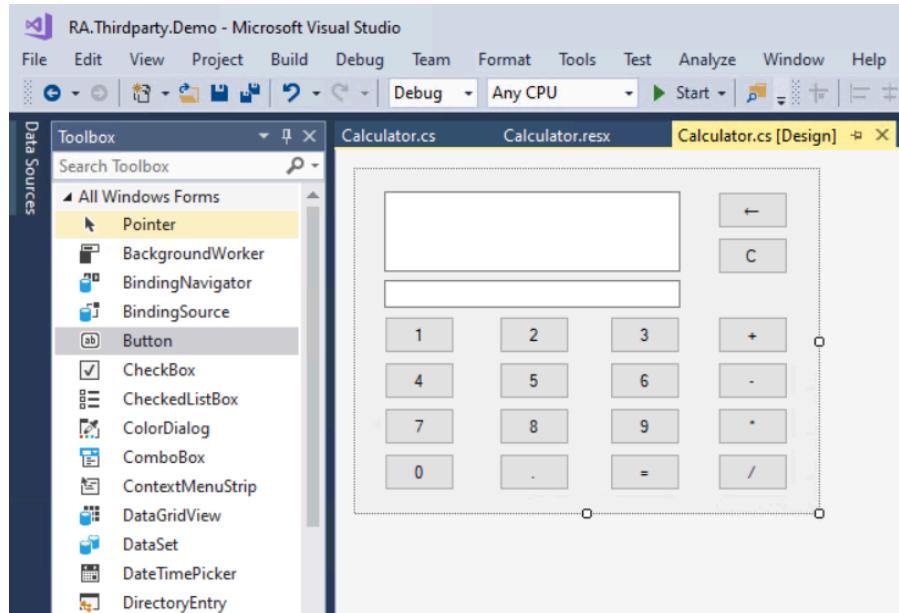


**Tip:** If you want to register the assembly to the Global Assembly Cache, a strong name signing is required.

- a. In the project's properties window, select **Signing**.
- b. Select the **Sign the assembly** checkbox.
- c. In the **Choose a strong name key file** list, select **Browse**.
- d. Choose the SNK file of your organization.



5. Select **View > Toolbox** to open the Toolbox, and then add objects for the calculator.



6. Select **View > Code** to open the code of the control.

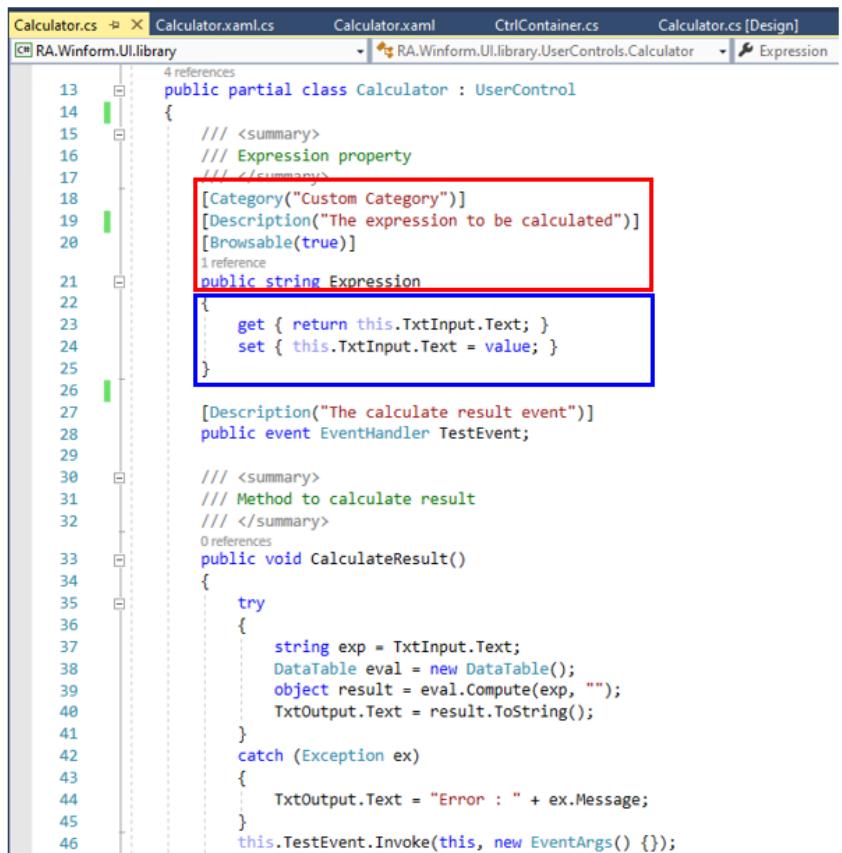
- Define a public string property **Expression**. This property will show on the Properties tab of the control's Properties dialog box in FactoryTalk View Studio.
- Define a public method **CalculateResult**. This method will show on the Methods tab of the control's Properties dialog box in FactoryTalk View Studio.

Area in red:

- Add a **Category** attribute name to classify the type of properties.
- Use the **Description** attribute to give an introduction to the property.

Area in blue:

- To get or set a property value externally in FactoryTalk View SE, you must declare the property as **public** and provide it with **getter** and **setter** methods.
- We recommend that you use common data types, such as string, integer, Boolean, and so on for better compatibility.



```

Calculator.cs  X Calculator.xaml.cs    Calculator.xaml      CtrlContainer.cs      Calculator.cs [Design]
RA.Winform.UI.library                                RA.Winform.UI.library.UserControls.Calculator      Expression

13  public partial class Calculator : UserControl
14  {
15      /// <summary>
16      /// Expression property
17      /// </summary>
18      [Category("Custom Category")]
19      [Description("The expression to be calculated")]
20      [Browsable(true)]
21      public string Expression
22      {
23          get { return this.TxtInput.Text; }
24          set { this.TxtInput.Text = value; }
25      }
26
27      [Description("The calculate result event")]
28      public event EventHandler TestEvent;
29
30      /// <summary>
31      /// Method to calculate result
32      /// </summary>
33      public void CalculateResult()
34      {
35          try
36          {
37              string exp = TxtInput.Text;
38              DataTable eval = new DataTable();
39              object result = eval.Compute(exp, "");
40              TxtOutput.Text = result.ToString();
41          }
42          catch (Exception ex)
43          {
44              TxtOutput.Text = "Error : " + ex.Message;
45          }
46          this.TestEvent.Invoke(this, new EventArgs() {});
47      }

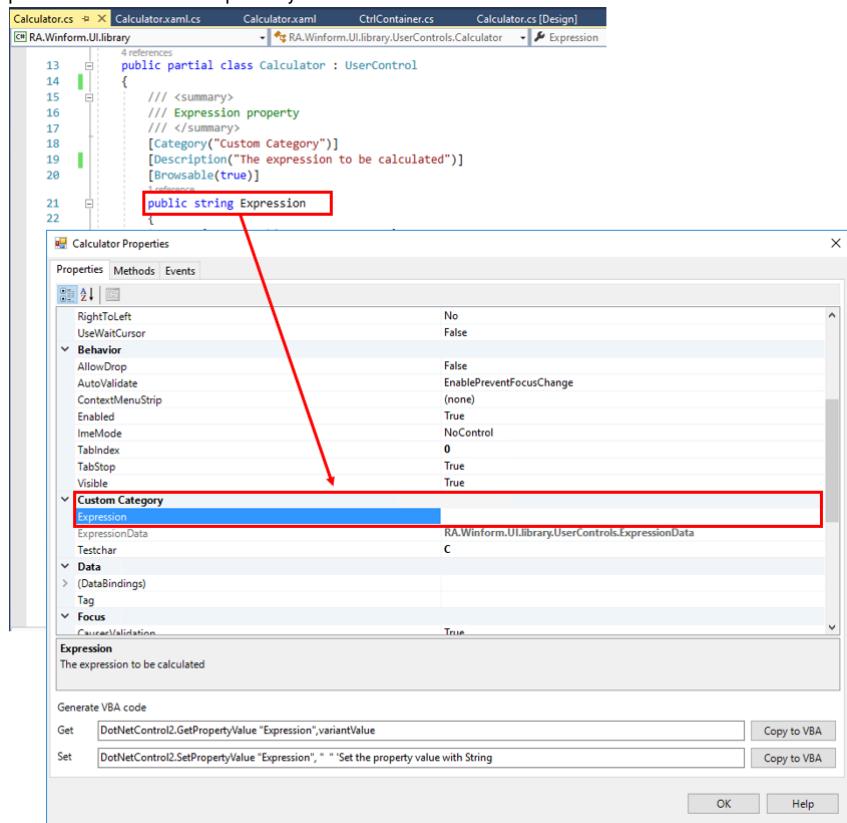
```

7. Configure the property in FactoryTalk View SE.

This picture shows how to configure the property on the Properties tab of the control's Properties dialog box in FactoryTalk View Studio.

Area in red:

- The property must be declared as **public**.
- We recommend that you use common data types, such as string, integer, Boolean, and so on as parameters for better compatibility.

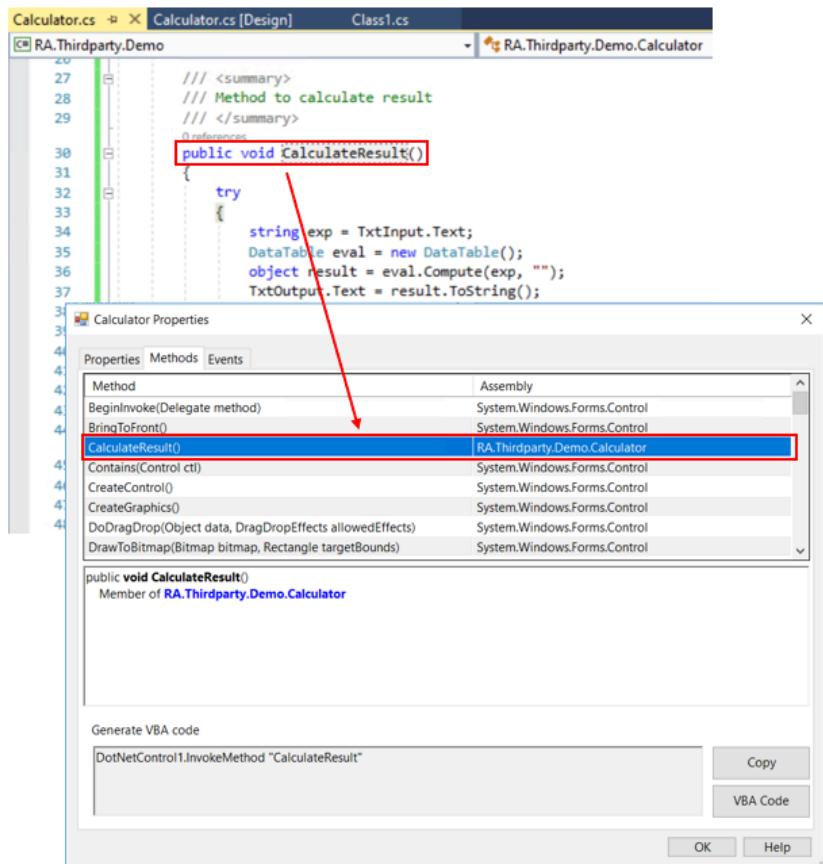


8. Configure the method in FactoryTalk View SE.

The picture shows how to configure the method on the Methods tab of the control's Properties dialog box in FactoryTalk View Studio.

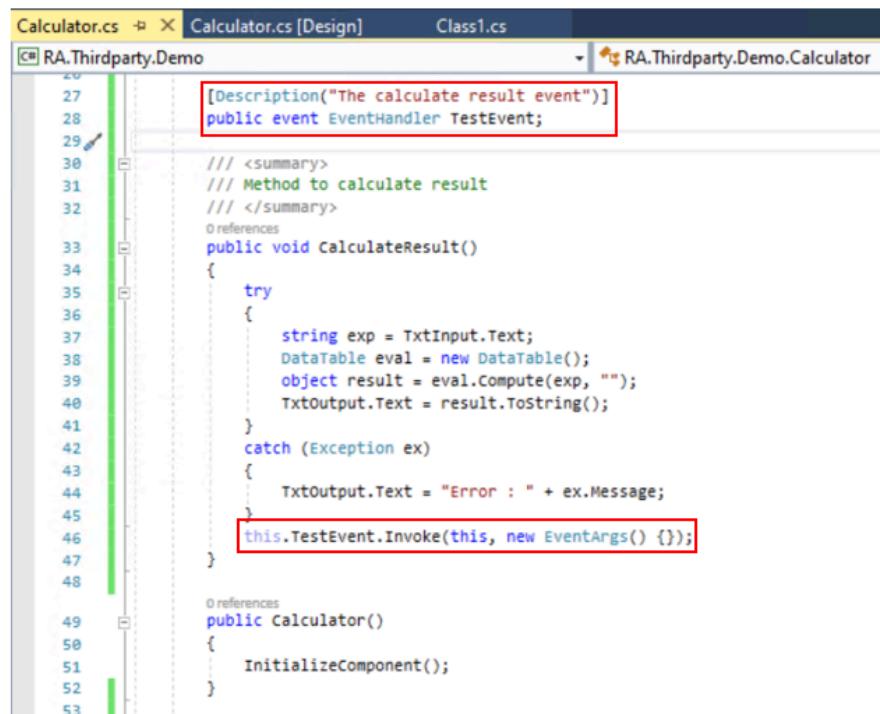
Area in red:

- The method must be declared as **public**.



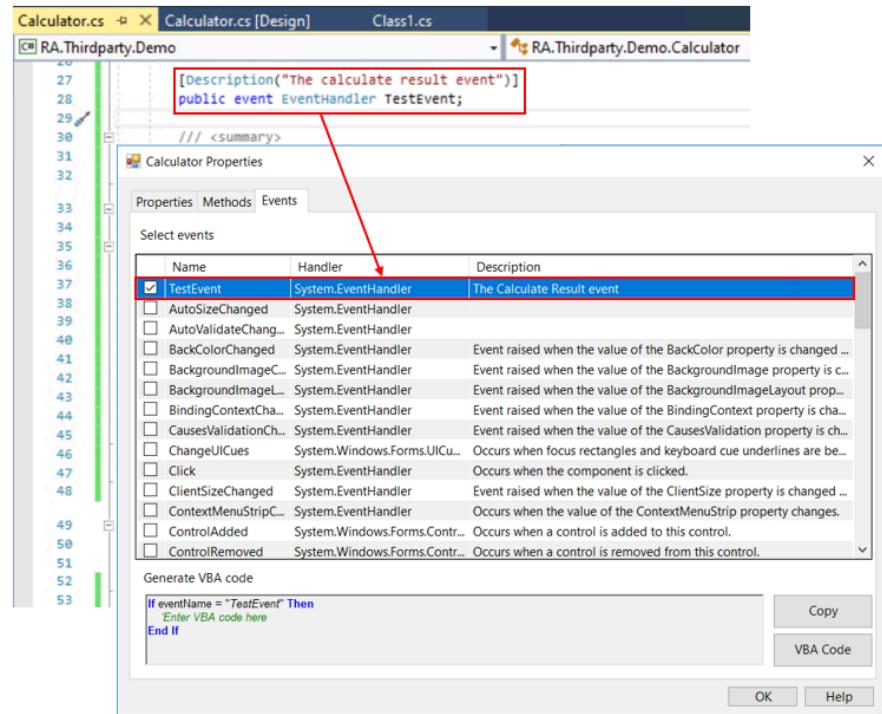
9. Select **View > Code** to open the code of the control.

Add an event named **TestEvent** in the control and invoke this event at the end of the **CalculateResult** function.

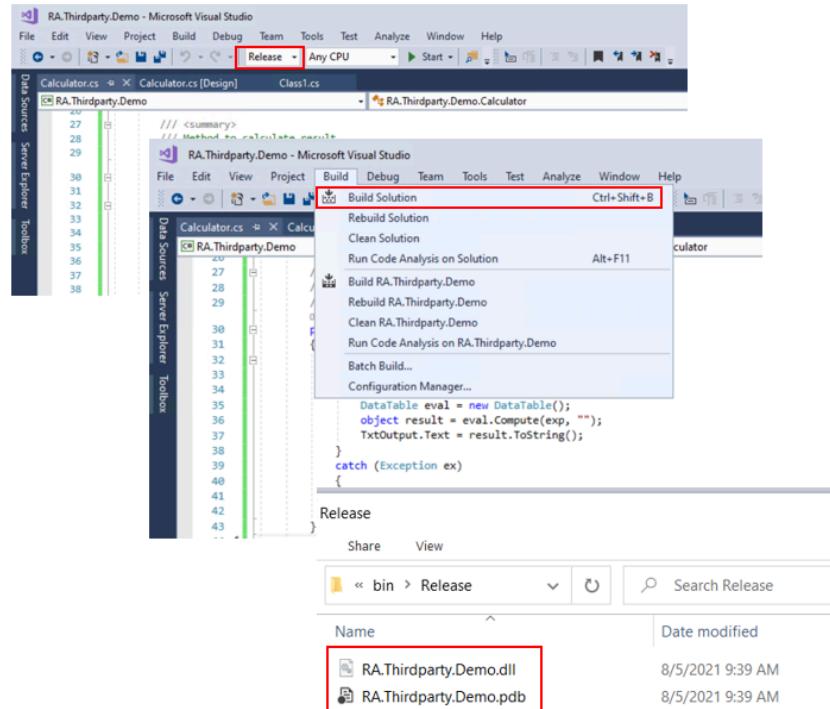


10. Subscribe to the event in FactoryTalk View SE.

The picture shows how to subscribe to the event on the Events tab of the control's Properties dialog box in FactoryTalk View Studio.



11. Create the assembly.
  - a. Change the solution configuration to **Release**.
  - b. Select **Build > Build solution**. The DLL files will be created in the Release folder.



## Creating custom WPF controls

This section uses a calculator control as an example to explain how to create custom WPF controls.



The calculator includes:

- Several function buttons
- An expression box
- A result box

After creating this control, you can:

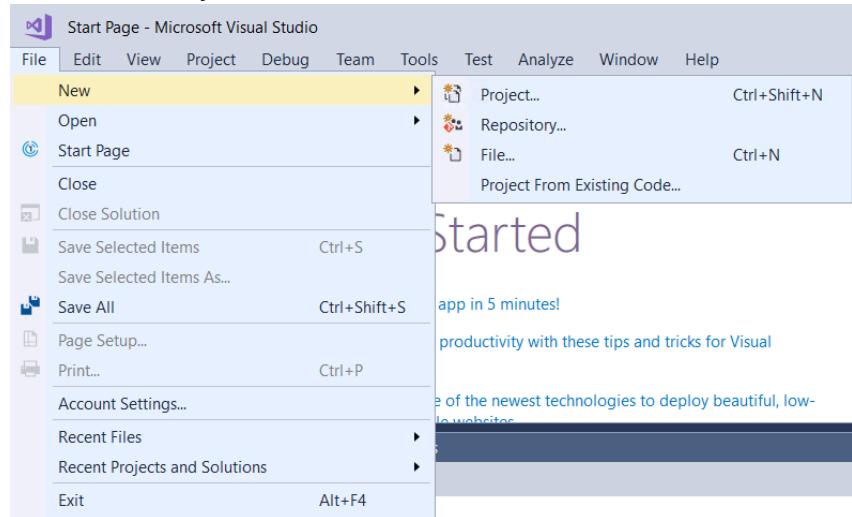
- Import it to FactoryTalk View Studio and draw it on a display.
- View and configure its properties.
- Set and get property values, invoke methods, and trigger events at runtime.
- Bind tags to its properties.

## Create a WPF control in Visual Studio

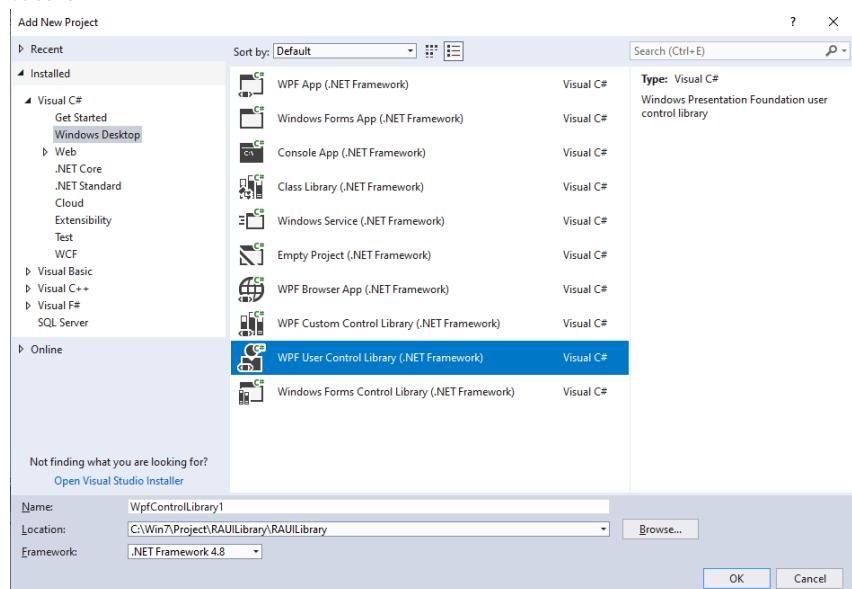
Use Visual Studio to create a custom WPF control.

## To create a WPF control

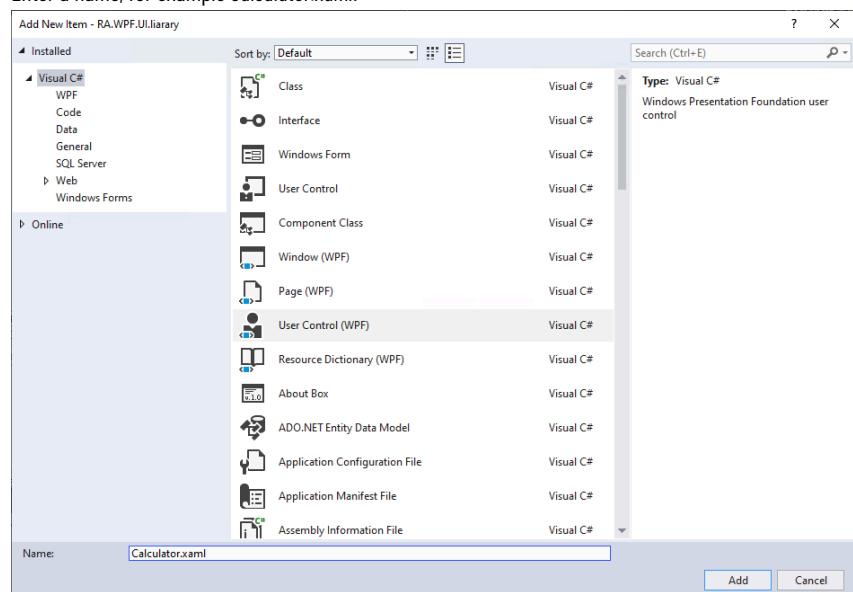
1. Create a project in Visual Studio.
  - a. Open Visual Studio.
  - b. Select **File > New > Project**.



2. In the **New Project** dialog box:
  - a. Select **Installed > Visual C# > Windows Desktop > WPF User control Library (.NET Framework)**.
  - b. In the **Framework** list, select **.NET Framework 4.8**.
  - c. Enter a name and file path.
  - d. Select **OK**.



3. Add a user control.
  - a. Select **Project > Add user control.**
  - b. Enter a name, for example *Calculator.xaml*.

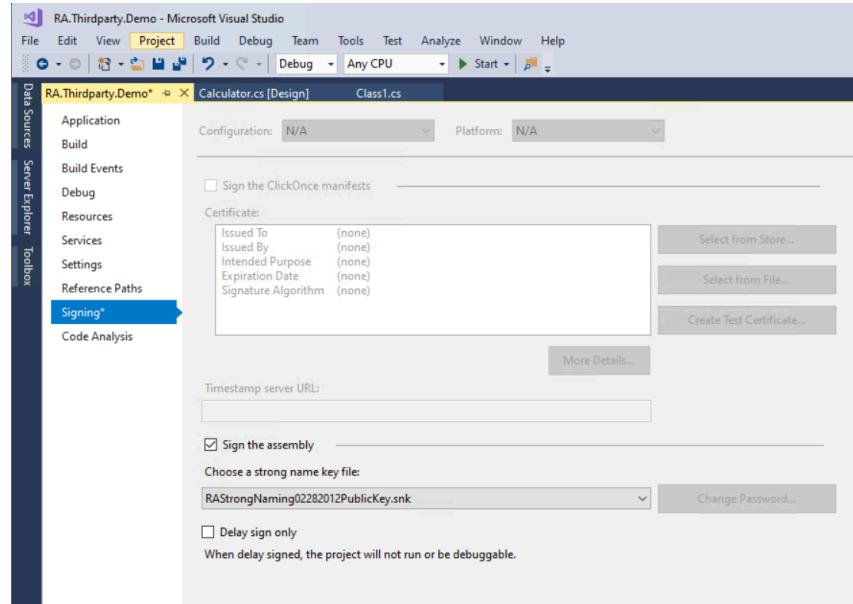


4. (optional) Configure strong name signing.

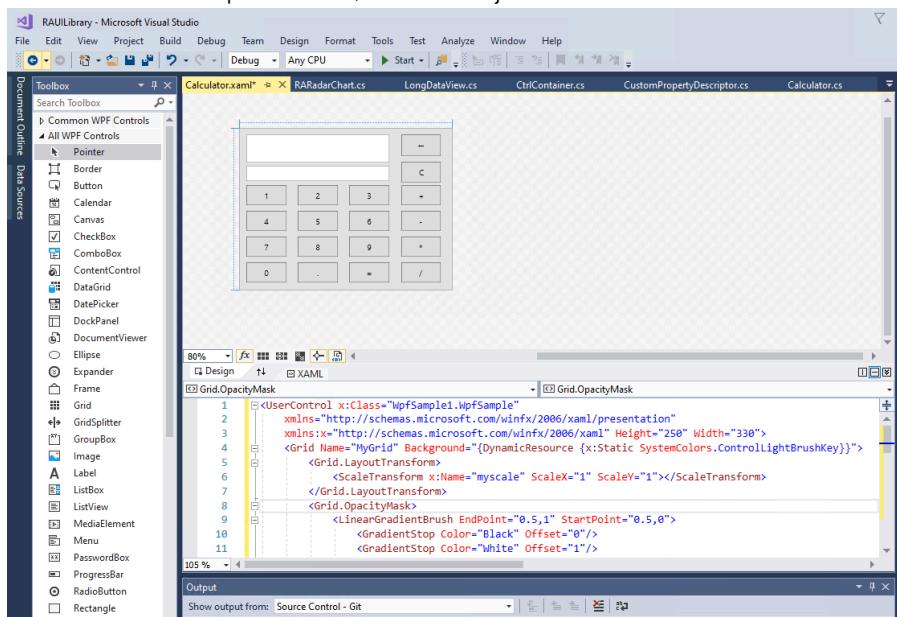


**Tip:** If you want to register the assembly to the Global Assembly Cache, a strong name signing is required.

- a. In the project's properties window, select **Signing**.
- b. Select the **Sign the assembly** checkbox.
- c. In the **Choose a strong name key file** list, select **Browse**.
- d. Choose the SNK file of your organization.



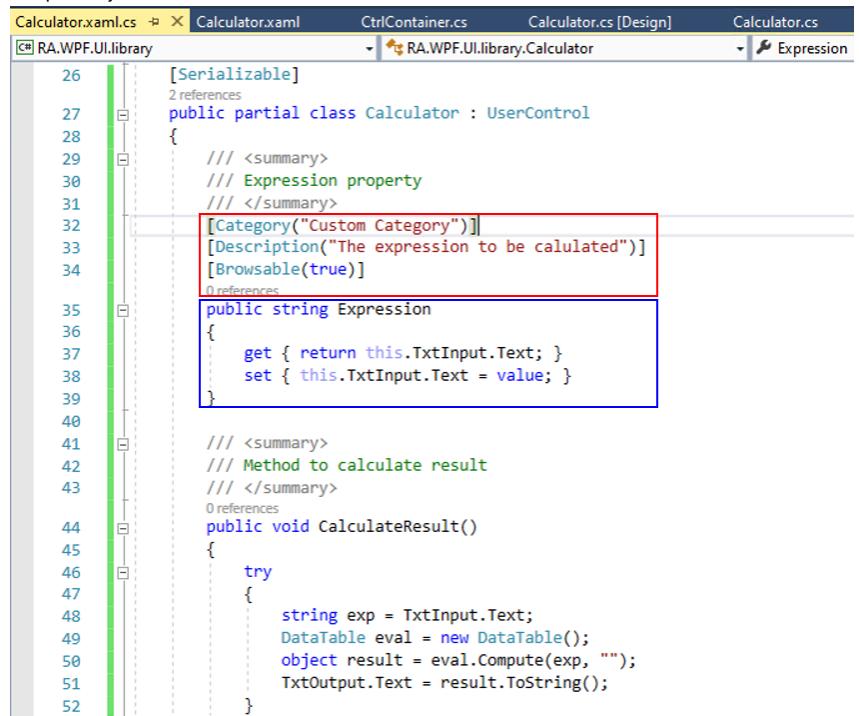
5. Select **View > Toolbox** to open the Toolbox, and then add objects for the calculator.



6. Select **View > Code** to open the code of the control.

- Define a public string property **Expression**. This property will show on the Properties tab of the control's Properties dialog box in FactoryTalk View Studio.
  - Define a public method **CalculateResult**. This method will show on the Methods tab of the control's Properties dialog box in FactoryTalk View Studio.
- Area in red:
- Add a **Category** attribute name to classify the type of properties.
  - Use the **Description** attribute to give an introduction to the property.
- Area in blue:

- To get or set a property value externally in FactoryTalk View SE, you must declare the property as **public** and provide it with **getter** and **setter** methods.
- We recommend that you use common data types, such as string, integer, Boolean, and so on for better compatibility.



```

26 [Serializable]
27 public partial class Calculator : UserControl
28 {
29     /// <summary>
30     /// Expression property
31     /// </summary>
32     [Category("Custom Category")]
33     [Description("The expression to be calculated")]
34     [Browsable(true)]
35     public string Expression
36     {
37         get { return this.TxtInput.Text; }
38         set { this.TxtInput.Text = value; }
39     }
40
41     /// <summary>
42     /// Method to calculate result
43     /// </summary>
44     public void CalculateResult()
45     {
46         try
47         {
48             string exp = TxtInput.Text;
49             DataTable eval = new DataTable();
50             object result = eval.Compute(exp, "");
51             TxtOutput.Text = result.ToString();
52         }
53     }
}

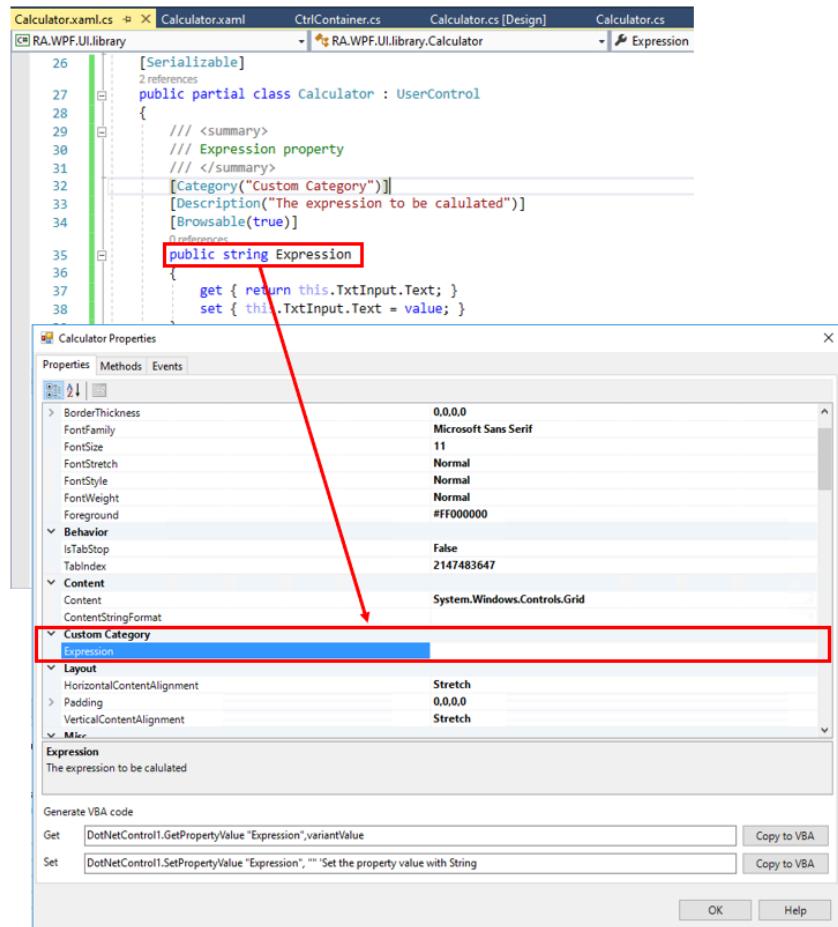
```

7. Configure the property in FactoryTalk View SE.

This picture shows how to configure the property on the Properties tab of the control's Properties dialog box in FactoryTalk View Studio.

Area in red:

- The property must be declared as **public**.
- We recommend that you use common data types, such as string, integer, Boolean, and so on as parameters for better compatibility.

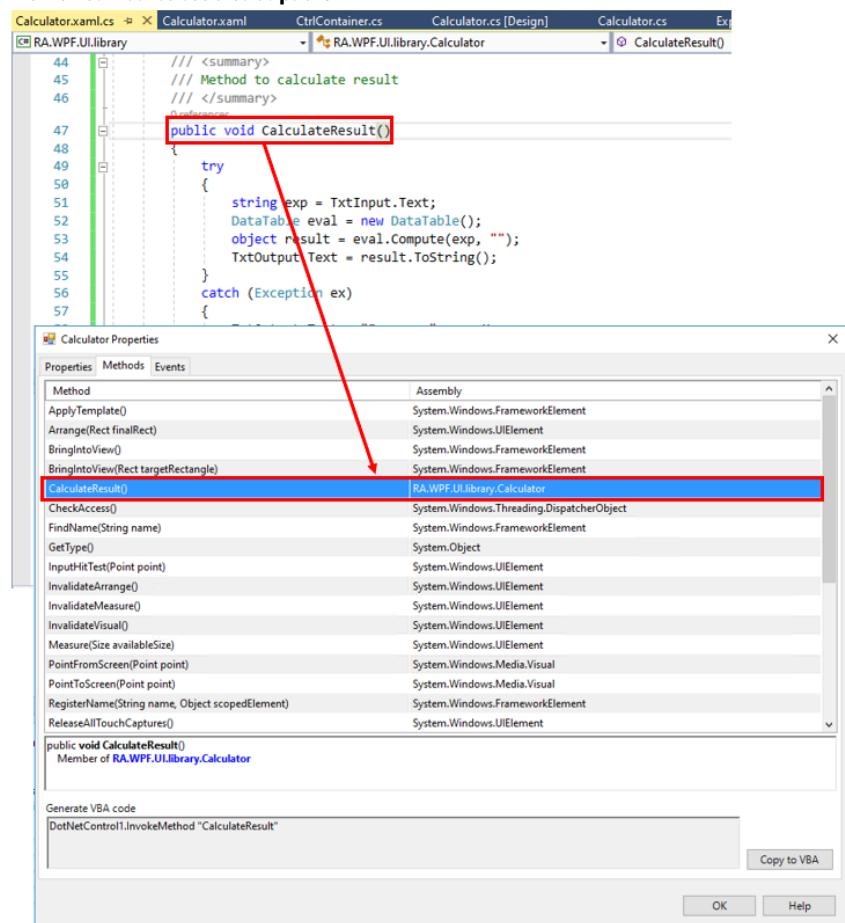


8. Configure the method in FactoryTalk View SE.

The picture shows how to configure the method on the Methods tab of the control's Properties dialog box in FactoryTalk View Studio.

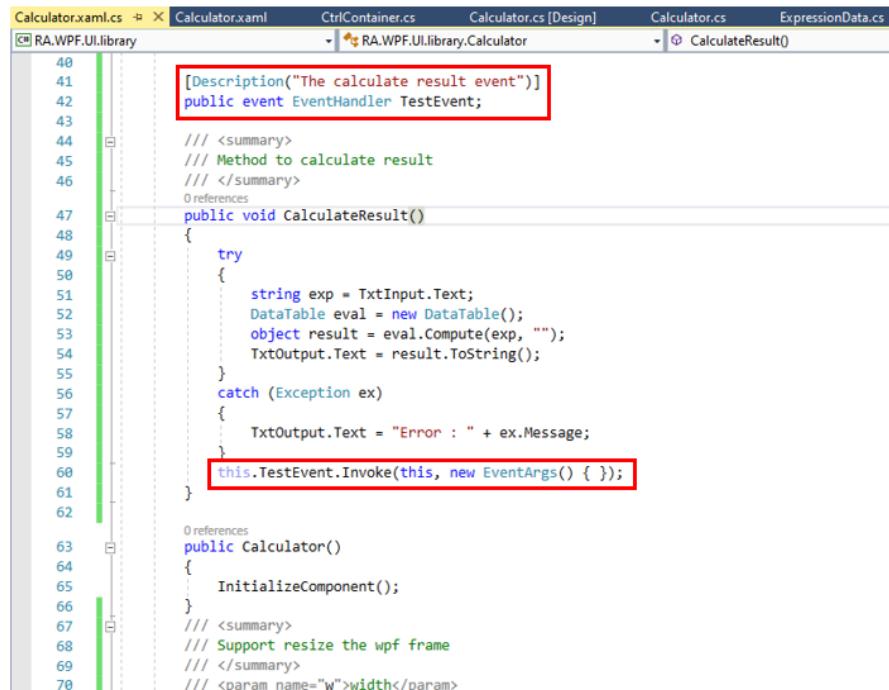
Area in red:

- The method must be declared as **public**.



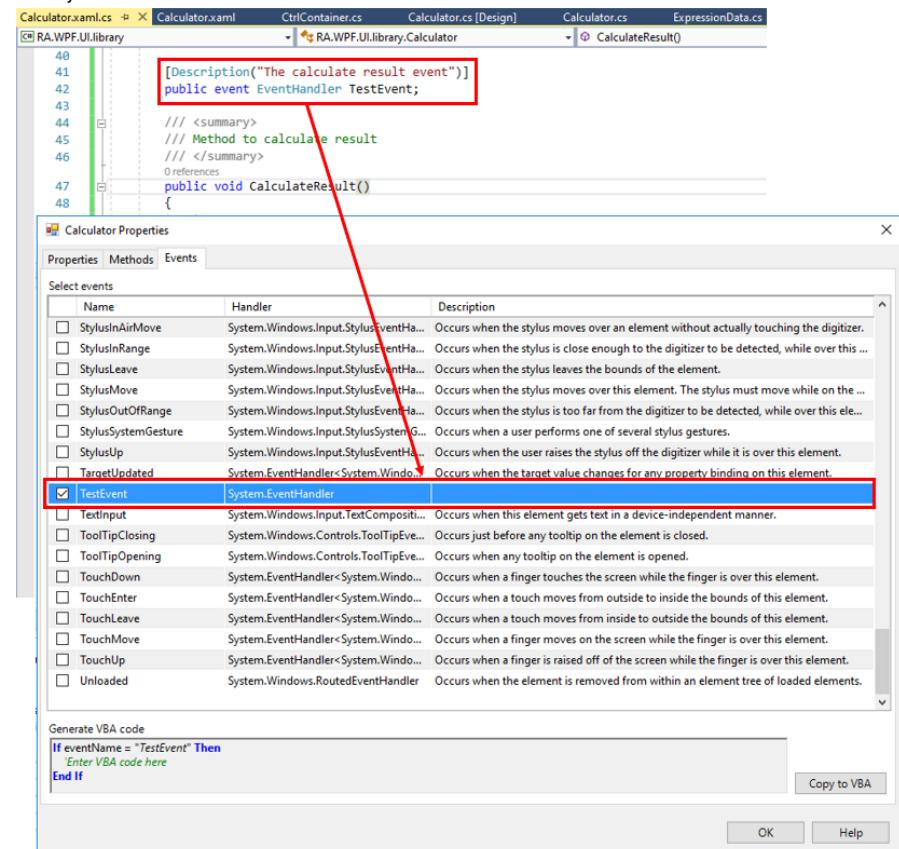
9. Select **View > Code** to open the code of the control.

Add an event named **TestEvent** in the control and invoke this event at the end of the **CalculateResult** function.

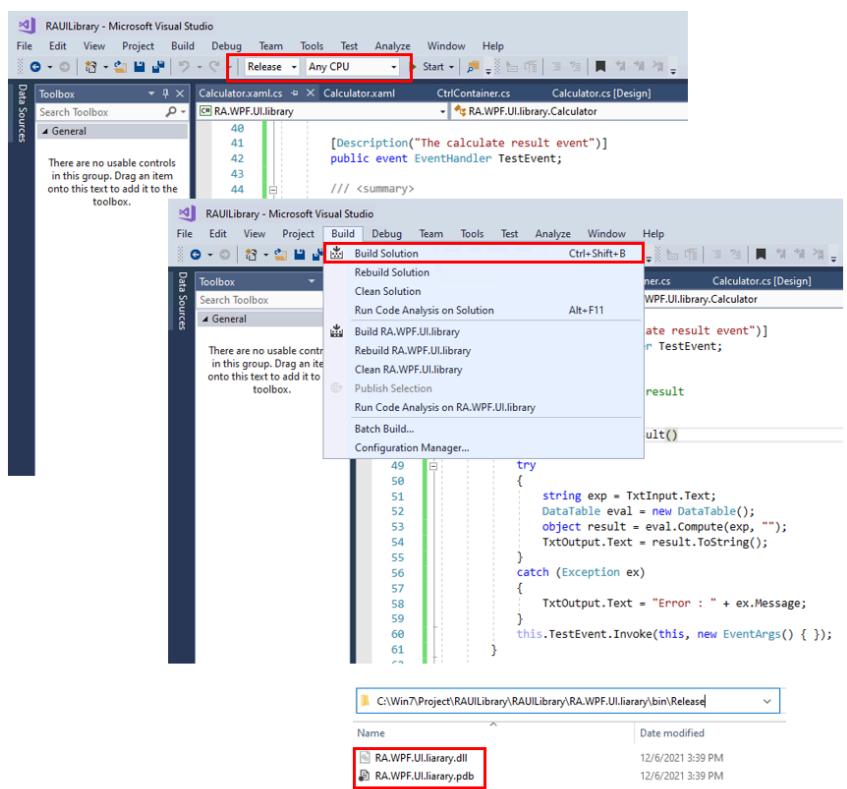


10. Subscribe to the event in FactoryTalk View SE.

The picture shows how to subscribe to the event on the Events tab of the control's Properties dialog box in FactoryTalk View Studio.



11. Create the assembly.
  - a. Change the solution configuration to **Release**.
  - b. **Select Build > Build solution.** The DLL files will be created in the Release folder.



## Importing and deploying the .NET control assembly

You can import and deploy self-created .NET controls or third-party libraries. For the custom controls to work properly, all binaries must be copied to the predefined target folder, which is C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup. Depending on the type of libraries, some libraries' binaries can be automatically copied to the target folder after importing them, while others must be manually copied to the target folder.

- A library that includes multiple binaries in the same folder  
After importing the binaries in the .NET Control dialog box in FactoryTalk View Studio, they are automatically copied to the target folder.
- A library that is installed in the Global Assembly Cache  
After importing the main binary in the .NET Control dialog box in FactoryTalk View Studio, it is automatically copied to the target folder. You don't need to manually copy any other dependent binaries to the target folder.
- A library that includes multiple binaries in a folder structure or multiple folders  
You must manually copy the binaries with the intact folder structure to the target folder C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup.

## Import self-created .NET controls or third-party libraries

If your custom control or library's binaries are in the same folder, or the library is installed in the Global Assembly Cache, you can import them in the .NET Control dialog box in FactoryTalk View Studio. After the import, the selected binaries are automatically copied to the target folder C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup.

## To import self-created .NET controls or third-party libraries

1. Open FactoryTalk View Studio.
2. Open a display and draw a .NET Control.
3. In the **.NET Control** dialog box, click **Browse**.
4. Select the DLL files of the self-created .NET controls or third-party libraries.



### Tip:

- You can select multiple files at one time.
- Network paths are not supported.

## Deploy .NET controls on multiple computers

- For custom .NET controls, copy all the binaries in C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup on the FactoryTalk View Studio computer to the same folder on all client computers.
- For third-party libraries installed in the Global Assembly Cache, install them on all client computers, and then copy all the binaries in C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup on the FactoryTalk View Studio computer to the same folder on all client computers.

## Update .NET controls

- Close FactoryTalk View Studio and display clients, and then replace the binaries with the new ones in the target folder C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup.
- If the third-party library is installed in the Global Assembly Cache, install the new version of the library, and then replace the binaries in the target folder C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup.

## Delete .NET controls

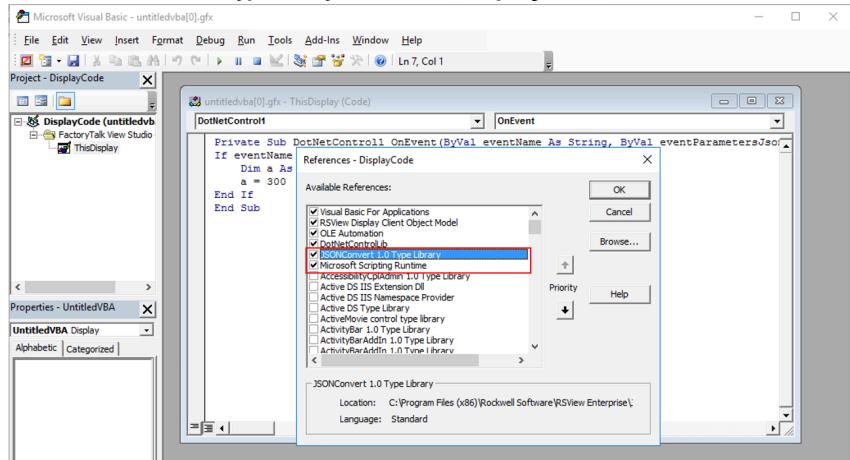
- Close FactoryTalk View Studio and display clients, and then delete the binaries in the folder C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup.
- If the third-party library is installed in the Global Assembly Cache, uninstall the library, and then delete the binaries in the folder C:\Users\Public\Documents\RSView Enterprise\DotNetControl Setup.

## JSON support in VBA

Use a built-in module called JSONConverter 1.0 Type Library in the VBA editor to support complex data types of property and method parameters in .NET control. This module will deal with those data types that the NewtonJSON library can serialize. Complex objects will be passed in and out by the JSON string format.

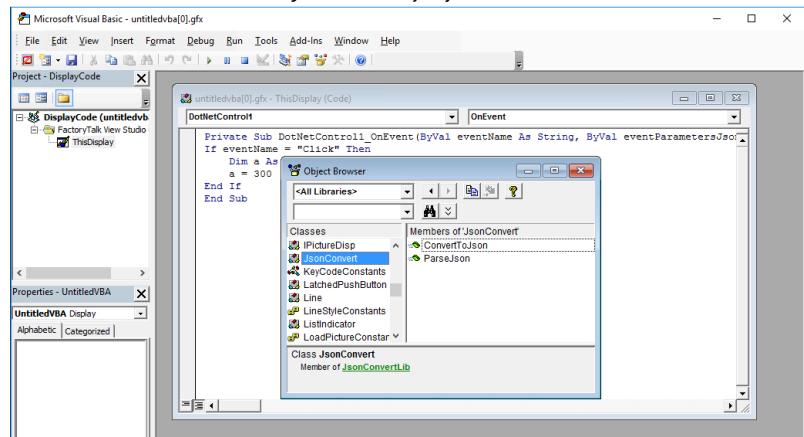
## To use JSONConverter in VBA

1. Add the reference library of JSONConverter.
  - a. In the VBA editor, select **Tools > References**.
  - b. Select **JSONConverter 1.0 Type Library** and **Microsoft Scripting Runtime**.



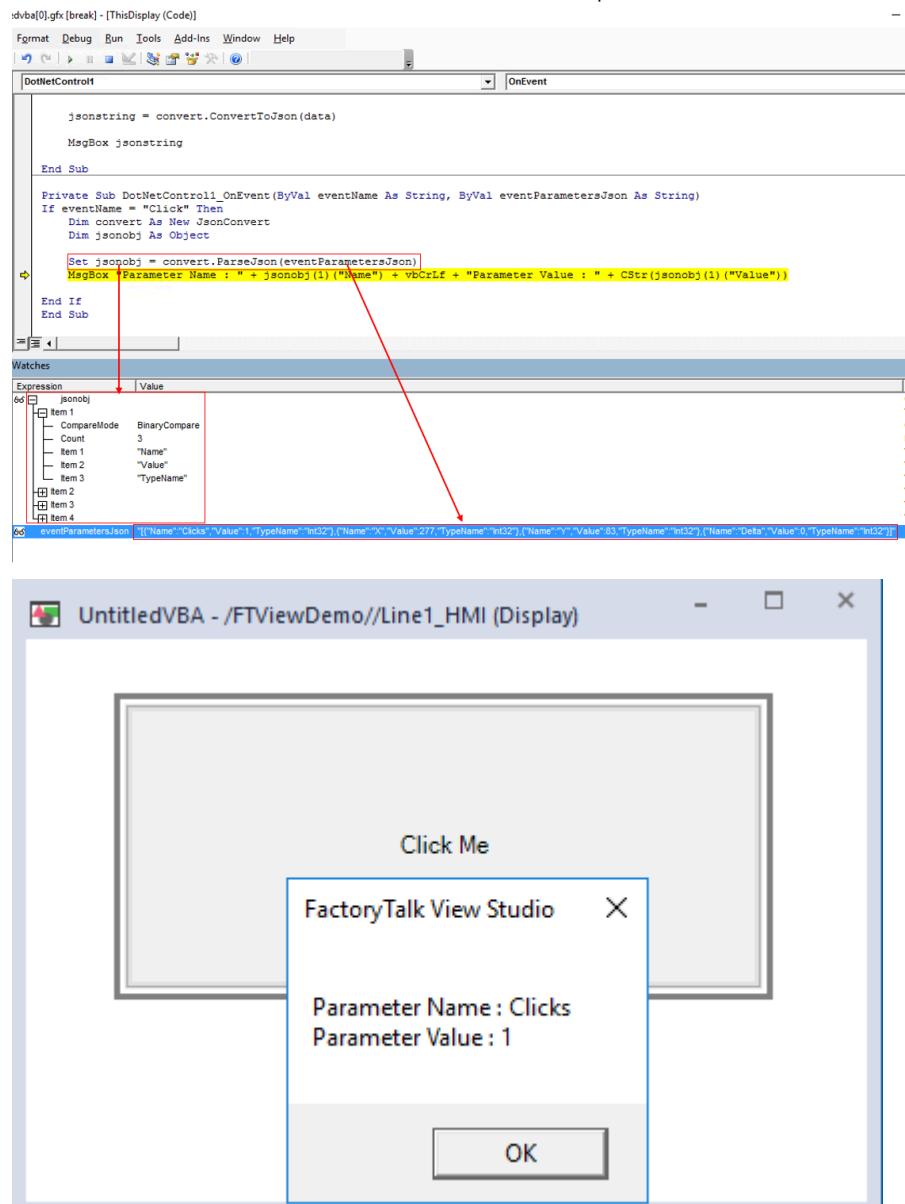
JsonConverter has two functions:

- **ConvertToJson**: Converts a dictionary object to a JSON string.
- **ParseJson**: Converts a JSON string to a dictionary object.



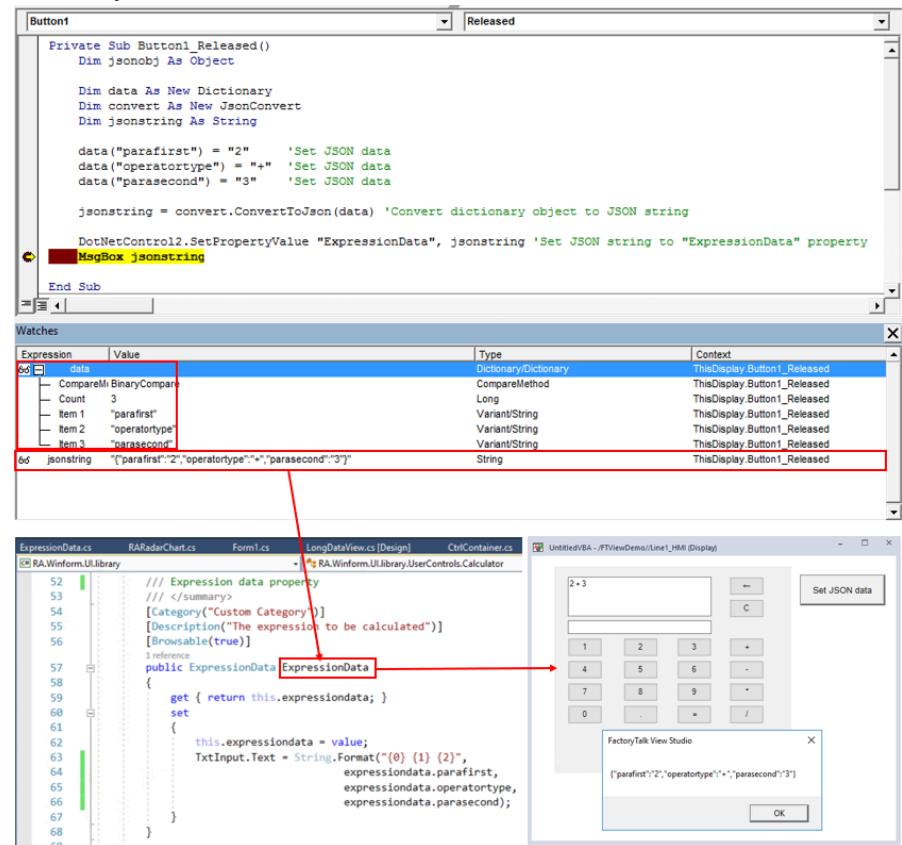
2. Parse the JSON string in VBA.

The VBA subscribes to the **Click** event, and then uses JSONConverter to parse the event information.



3. Convert a dictionary object to a JSON string.

The VBA creates a dictionary object, sets the JSON key-value pairs, and then uses JSONConverter to generate a JSON string.



## References

- Microsoft SDK

Microsoft SDK is a free tool that you can download from the Microsoft website. For more information, see <https://dotnet.microsoft.com/en-us/download/dotnet-framework/net48>.

- Global Assembly Cache

Each computer where the Common Language Runtime is installed has a machine-wide code cache called the Global Assembly Cache. The Global Assembly Cache stores assemblies specifically designated to be shared by several applications on the computer. For more information, see <https://docs.microsoft.com/en-us/dotnet/framework/app-domains/gac>.

## Setting up DDE communications for HMI tags

This appendix describes:

- What DDE communications are.
- How to set up DDE communications with HMI tags.
- Scanning for new tag values.



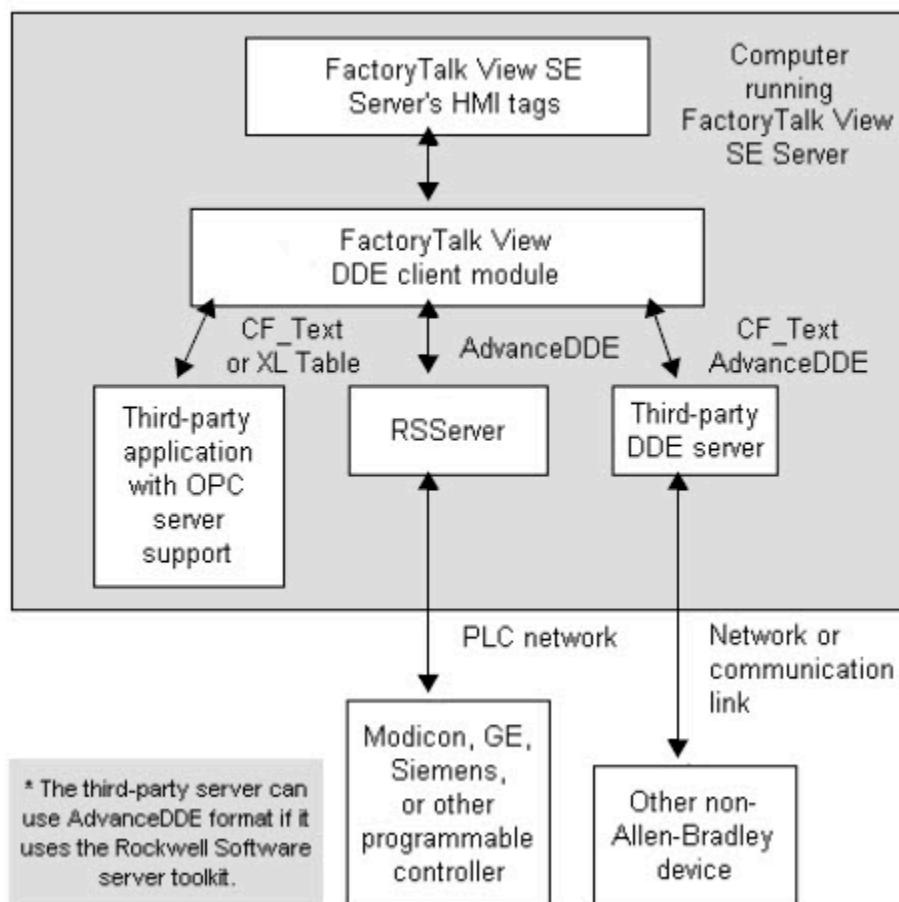
**Tip:**

This section does not apply to Microsoft Windows 7, Windows Server 2008 and higher operating systems.

### Overview of DDE communications

To support legacy applications FactoryTalk View supports DDE communications.

The following illustration shows how FactoryTalk View SE might use DDE.



DDE (Dynamic Data Exchange), used with external DDE servers such as the Rockwell Software RSServer products, or with other third-party servers, allows applications to communicate with a wide range of local and remote devices.

DDE servers provide a way to connect to an HMI server's HMI tags (the DDE client), to communication devices such as Siemens or Modicon, or to third-party DDE servers such as Microsoft Excel or Visual Basic.

## Setting up an HMI server as a DDE client

These are the tasks involved in setting up an HMI server as a DDE client:

1. Start the DDE server.
2. Start FactoryTalk View Studio, and then create or open an application (the DDE client).
3. In the **Tags** editor, create tags. Select **Device** as the data source. In the **Address** box, type the required DDE address. For information about DDE addressing syntax, see [Syntax for DDE addresses on page 473](#).



**Tip:** If the DDE server is running as an application instead of as a service, for more information, see Knowledgebase Document ID: [BF1797 - My DDE link is not working in RSView SE](#).

## Creating an HMI tag that uses DDE

This section describes how to complete the Data Source options in the **Tags** editor for tags that use DDE communications.

For details about creating HMI tags, see Help. For information about working with tags, see [Working with tags on page 159](#).

### Specifying Device as the data source

An HMI tag with Device as its data source receives its data from a source external to FactoryTalk View. This section describes how to create tags for data that comes from:

- Programmable controllers through a DDE server.
- Another Windows program through a DDE server.

#### To create a tag with Device as the data source:

1. For the data source Type, select **Device**.



2. In the **Address** box, specify the DDE item.

Type the DDE address for the tag. The DDE item name and format depend on the DDE server, and are not validated by FactoryTalk View.

## Syntax for DDE addresses

The syntax for DDE tag addresses is:

DDE : Application|Topic!Item

There are no spaces:

- Between the application name, the vertical bar, and the topic name.
- Between the topic name, the exclamation mark, and the item name.

For more information about addressing syntax, see the documentation supplied with the DDE server.

## **Scanning for new tag values**

When an HMI server containing tags is running, it must periodically update its tag values in the value table.

For applications using DDE, values are updated by the server at the rate specified in the server product. When the server detects a change, it provides the changed value to FactoryTalk View.

For details about setting the poll rate, see the server documentation.

# Using the SE Client object model and display code

This chapter describes:

- How you can use VBA with FactoryTalk View Site Edition.
- The VBA-integrated development environment.
- The objects in the FactoryTalk View SE Client object model.
- How to find information about FactoryTalk View SE Client objects.
- How to find information about VBA.

## Customizing applications using VBA with FactoryTalk View

If you regularly require data from programs such as Microsoft Excel, or Microsoft SQL Server, consider using the FactoryTalk View SE Client object model and display code with VBA to integrate these applications with FactoryTalk View SE.

You can also use the FactoryTalk View SE Client object model to customize the control system in the following ways:

- **Create custom alarm events.** You can write alarm detection algorithms using PLC logic, and then create events in FactoryTalk View SE to respond to the algorithms.
- **Validate operator input.** Use VBA logic to validate the operator's input, for example, to ensure that the value an operator enters in a numeric input object falls within 10 percent of the value of another numeric input object.
- **Create custom forms for operators.** You can use VBA to create custom forms, for example, pop-up dialog boxes that an operator can interact with at runtime.  
You can also use VBA logic to validate operator input, for example, to ensure that the value an operator enters in a numeric input object falls within the correct range of values.
- **Design intuitive graphic displays.** Use the FactoryTalk View SE Client object model to provide ActiveX controls with data, for use in graphic displays.  
For example, use list boxes or combo boxes in graphic displays to allow an operator to select options such as recipe items.
- **Manipulate the FactoryTalk View SE Client window.** Write VBA code to arrange graphic displays based on the size of the FactoryTalk View SE client window. This allows you to adapt an application dynamically to various desktop sizes and resolutions.
- **Send custom messages to the Diagnostics log.** Send specific messages to the Diagnostics List and Diagnostics log, to record operation conditions and events through VBA code.
- **Access HMI tag database.** The FactoryTalk View SE Client object model allows you to manage HMI tags and tag values through VBA code.
- **Secure the system.** The FactoryTalk View SE Client object model allows you to obtain security information about who is using the system, and to use the security information and events to control access to the system.  
For example, you can restrict a user's access to a graphic display by creating code that opens the graphic display only when the user is logged in with a particular security code.

## Summary of basic steps

There are two basic tasks involved in working with the FactoryTalk View SE Client object model and VBA:

1. Create the graphic display and the graphic objects that you want to write display code for.  
Display code is saved with a graphic display. Before you can write VBA code that manipulates graphic objects, you must create the graphic display and the graphic objects in FactoryTalk View Studio. For details about creating graphic displays, see [Creating graphic displays on page 259](#).
2. In the VBA IDE (Integrated Development Environment), using the Visual Basic programming language, write procedures in the code window for the module called ThisDisplay.  
For an overview of the VBA IDE, see [Parts of the VBA integrated development environment on page 476](#). For information about VBA Help, see [Finding information about VBA on page 478](#).

This chapter highlights aspects of VBA that are unique to FactoryTalk View Studio and describes how to open the VBA IDE from FactoryTalk View graphic displays.

This chapter does not include detailed information about using VBA – it is assumed that you are familiar with the VBA environment and Visual Basic programming language.

## About procedures

A procedure is a named block of code that runs as a unit. Examples of procedures are Visual Basic subroutines and functions. The module called ThisDisplay contains the procedures that you create to run in response to a display's events.

VBA procedures can call procedures in other user forms or modules. For information about creating Visual Basic procedures, see VBA Help.

## How VBA code runs

The VBA procedures that you create are called in response to events that occur in a graphic display at runtime.

Events that can be used in this way are defined in the Display Client object model, and include actions such as the display being opened and animated, mouse clicks, and so on.

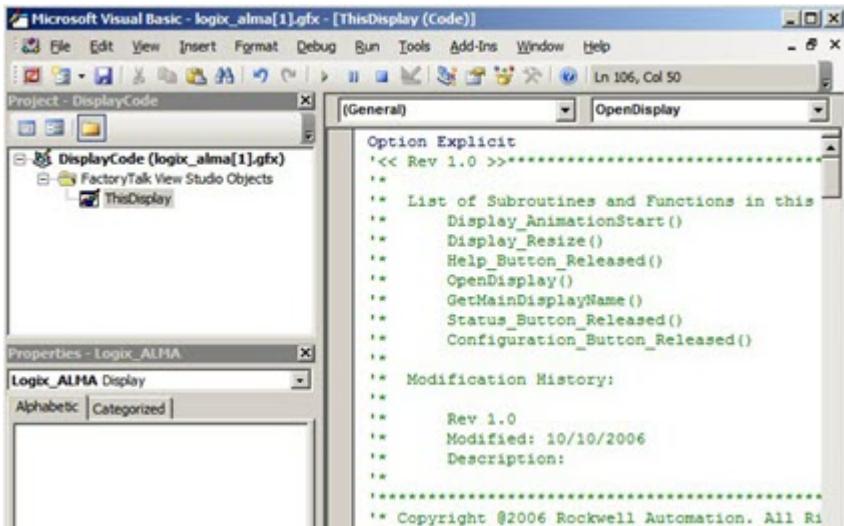
## Parts of the VBA IDE

The integrated development environment (IDE) for VBA is available from the FactoryTalk View Graphics editor.

Use the VBA IDE to create, test, and debug VBA procedures. VBA procedures can be designed to run in response to events that occur in FactoryTalk View graphic displays at runtime.

## To open the VBA IDE:

1. In FactoryTalk View Studio, in the Explorer window, open the graphic display you want to write display code for.
2. From the **Edit** menu, select **VBA Code**.



For information about opening VBA Help, see [Finding information about FactoryTalk View SE Client objects on page 478](#).

## Project Explorer

For each graphic display, a VBA project named *DisplayCode* is created. The **Project Explorer** shows the VBA project for each graphic display that is open in the **Graphics** editor.

Each project contains a FactoryTalk View Studio Objects folder. This folder contains a module called *ThisDisplay*, which represents the graphic display.

The module called *ThisDisplay* contains the code that interacts with objects on the display. It can contain any number of procedures.

You can create additional VBA modules and user forms. To use the procedures in these modules, you must call them from procedures that exist in the module called *ThisDisplay*.

## Properties window

This window lists the property settings for the code modules, class modules, and the VBA user forms and the objects on those forms.

## Procedure or code window

This window is where you write and modify VBA procedures.

In *ThisDisplay*, any FactoryTalk View SE Client objects that raise events are listed in the dropdown list at the upper left of the window. When an object is selected in the list, the object's events are listed in the dropdown list on the right.

## FactoryTalk View SE Client object model

The FactoryTalk View SE Client object model contains objects that represent the features of the SE Client. Using these objects, you can interact with an SE Client by writing code that is run when an object event occurs.

Use the Object Browser in the VBA IDE to view the FactoryTalk View SE Client objects.

### To open the Object Browser, use one of the following methods:

- Right-click anywhere in the Code window, and then select **Object Browser**.
- From the **View** menu, select **Object Browser**.
- Click anywhere in the code window and then click F2.

## Finding information about FactoryTalk View SE Client objects

The FactoryTalk View SE Help contains detailed information about the objects, properties, methods, and events in the FactoryTalk View SE Client object model.

### To open context-sensitive Help

In the **Object Browser**, click the item, and then press **F1**, or click the **?** button on the toolbar.

Alternatively, in a line of code in the code window, you can highlight the name of a FactoryTalk View object, a property, or a method, and then press **F1**.

### To open the object model Help:

1. In FactoryTalk View Studio or the FactoryTalk View SE Administration Console, click **Help**, and then click **Contents**.
2. In the Contents list, double-click **Creating Graphic DisplaysWorking with VBA and FactoryTalk Object Models**.
3. In the list of topics, do one of the following:
  - For a list of overview topics, double-click **Creating VBA Display Code**.
  - For details about the objects, double-click **Using the FactoryTalk View SE Client Object Model**.

Help topics for VBA code and the FactoryTalk View SE Client object model open in a separate window, on top of the FactoryTalk View Help window.

To return to the FactoryTalk View Help window, close the FactoryTalk View SE Client object model Help window.

## Finding information about VBA

If you're new to Visual Basic, you might want to look at these Microsoft publications:

- *Visual Basic Getting Started*
- *Visual Basic Programmer's Guide*

### To open VBA Help:

From the **Help** menu in the VBA IDE, select **Microsoft Visual Basic Help**.

Alternatively, in any area of the IDE, press **F1**.

## Importing and exporting XML files

This appendix describes:

- What XML is.
- Exporting FactoryTalk View graphics data to XML files.
- Modifying exported XML files.
- Importing graphic display XML files.
- The XML file structure for graphic displays.

### About XML

XML is the Extensible Markup Language used to create documents with structured information.

XML has a standardized format and structure. You can modify the elements and attributes of a graphic display by changing them, or by adding new ones, in the XML file.

For more, general information about XML, see the World Wide Web Consortium's webpage about XML at <http://www.w3.org/XML>.

For specific information about importing and exporting XML files in FactoryTalk View SE, see the FactoryTalk View Site Edition Help.

### Exporting graphics data to an XML file

The quickest way to create XML files for FactoryTalk View graphic displays is to export the data from FactoryTalk View.

After exporting, you can open the XML file in Notepad, modify it, and then import the modified file back into FactoryTalk View.

#### To export graphic display information to an XML file:

1. In FactoryTalk View Studio, in the Explorer window, right-click the **Displays** editor.
2. Click **Import and Export** to open the **Graphics Import Export Wizard**.
3. In the **Operation Type** dialog box, select **Export graphic information from displays**, and then click **Next**.
4. Follow the instructions in the wizard. For details, click **Help**.

FactoryTalk View creates XML files for the selected graphic displays, in the location you specify. In the same location.



**Tip:** FactoryTalk View also creates a file called BatchImport\_Application name.xml. Use this file to import several displays at once. To import a different set of displays than you exported, modify the list of display names in the BatchImport\_Application name.xml file.

### Modifying exported XML files

When you modify an XML graphic display file, you can change objects that already exist in the display or you can add additional objects.

When you import the file back into FactoryTalk View, the revised file overwrites the previous XML file, so the changes you have made are visible when you open the display in FactoryTalk View.

When you import the file, you specify whether you are importing changes or new objects. You can't do both actions in a single import operation.

It is recommended that you use Notepad to edit XML files.



**Tip:** If you include attributes for an object whose name does not match one of those in the graphic display, those attributes are not imported. Attributes for all other objects in the file whose names do match the ones in the graphic display are imported.

## Saving XML files in Notepad

An XML file begins with an XML declaration, containing the XML version and the encoding information. For example,

```
<?xml version="1.0" encoding="UTF-8" ?>  
  
<?xml version="1.0" encoding="UTF-16" ?>
```

If you create an XML document in Notepad, the file can be saved in UTF-8 or UTF-16 file format. If no declaration appears, the UTF-8 format is assumed. If you save the file in Notepad with Encoding=Unicode, be sure the declaration at the top of the XML file specifies encoding="UTF-16".



**Tip:** Notepad's Unicode file type corresponds to the UTF-16 file format.

For files containing strings in English or other Latin-based languages, UTF-8 is recommended to reduce the size of the XML file. For other languages, such as Chinese, Japanese, or Korean, UTF-16 is recommended.

You must specifically save as the desired encoding in Notepad. The default is to save to ANSI, and that format will not import properly.

## Testing XML files

An XML file must be well-formed to be imported. To find out whether your XML file is well-formed, test it.

### To test an XML file:

Open the XML file in Internet Explorer.

If you can see the XML code, your file is well-formed. If the XML code is not well-formed, Internet Explorer displays an error message.

## Importing XML files

You can import graphic display information in an XML file that has been created using an external editor, or you can import an XML file that you originally exported from FactoryTalk View and then modified.

**NOTE:** When you import modified graphic displays, the existing displays are overwritten. To save a copy of existing displays, select **Yes** when the **Graphics Import Export Wizard** shows the prompt: *"Do you want to back up the displays that will be modified by the import?"*

Alternatively, you can create backup copies by exporting the existing displays, before importing the modified ones.

## Error log file

If errors occur during importing, the errors are logged to a text file. The file opens automatically when importing is finished. The last paragraph of the file lists the location of the file.

## Importing graphic display XML files

You can import a single graphic display XML file at a time, or import multiple graphic displays. You can also choose whether to import new objects or update existing objects.

To import multiple graphic displays, specify the names of the graphic displays in the file *BatchImport\_Application name.xml*. FactoryTalk View creates this file when you export multiple graphic displays.

### To import graphic display information from an XML file:

1. In FactoryTalk View Studio, in the Explorer window, right-click the **Displays** editor.
2. Click **Import and Export** to open the **Graphics Import Export Wizard**.
3. In the **Operation Type** dialog box, select **Import graphic information into displays**, and then click **Next**.
4. Follow the instructions in the wizard.

For details about using the **Graphics Import Export Wizard**, see Help.

## Graphic display XML file structure

A graphic display XML file is a FactoryTalk View XML document that describes the objects and settings for a graphic display. The root element of the XML document is called **gfx**. It represents the graphic display.

An XML document can contain only one root element. All other elements must be contained within the beginning and end markers of the root element.

In an XML document, the start of an element is marked by **<element name>**, and the end is marked by **</element name>**.

If an element contains no sub-elements, the end can be marked by **/>**. For example, **<trigger id="T1" type="value" ack-all-value="0" />**.

The syntax for specifying an attribute for an element is **attribute="value"**. The attribute value must be enclosed in single or double quotes. You can specify multiple attributes for an element. For example, the caption element contains 13 possible attributes.

Elements for group objects begin with **<group name>**, and end with **</group name>**.

The **<group>** element contains all the elements and attributes for each object in the group.

For more information about graphic object elements and their attributes, see Help.

## FactoryTalk View documentation

In addition to *FactoryTalk View Site Edition User's Guide*, the FactoryTalk View documentation set includes:

- **FactoryTalk View Installation Guide** contains information about installing and deploying FactoryTalk View SE applications.
- **Release Notes** contains system requirements, system features, anomalies, functional changes, and application notes for when you begin installing or working with FactoryTalk View.
- **Help** contains procedures and comprehensive reference information that is available from the **Help** menu in FactoryTalk View Studio, and from editors and dialog boxes used to develop FactoryTalk View SE applications.

### Other documents

Document	Description
FactoryTalk Alarms and Events System Configuration Guide, publication <b>FTAE-RM-001</b>	Describes how to install, configure, and use FactoryTalk Alarms and Events services as part of a FactoryTalk-enabled automation system.
FactoryTalk Security System Configuration Guide, publication <b>FTSEC-QS001</b>	Describes how to configure the FactoryTalk Security, create accounts, and set permissions.

## Troubleshoot KEPServer

If you are using KEPServer Enterprise as an OPC server with FactoryTalk View version 9.00.00 or later and using an HMI tag to connect to a third-party device, you will not be able to access the tag value. The cause is that the security enhancements in FactoryTalk View version 9.00.00 or later affect the communications with third-party devices. To access the tag value, the additional Distributed Component Object Model (DCOM) configuration is required for KEPServer Enterprise on the computers.

If you are using a direct reference tag through the KEPServer Enterprise OPC server, you can access the tag value successfully.

If you try to add tags from the local KEPServer Enterprise OPC server in DataLogPro or data log, the Tag Browser cannot browse any tags. In such cases, you should also configure DCOM.

## Set DCOM configuration for KEPServer Enterprise

In the following steps, KEPServer Enterprise 5.21 is used as an example. The steps also apply to earlier versions of KEPServer Enterprise.

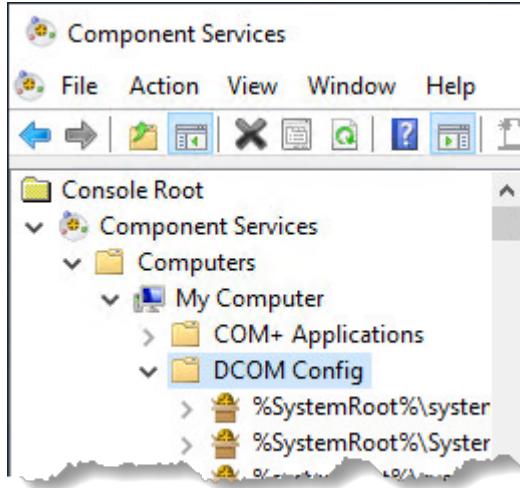
### To set DCOM configuration for KEPServer Enterprise

1. Open Component Services.



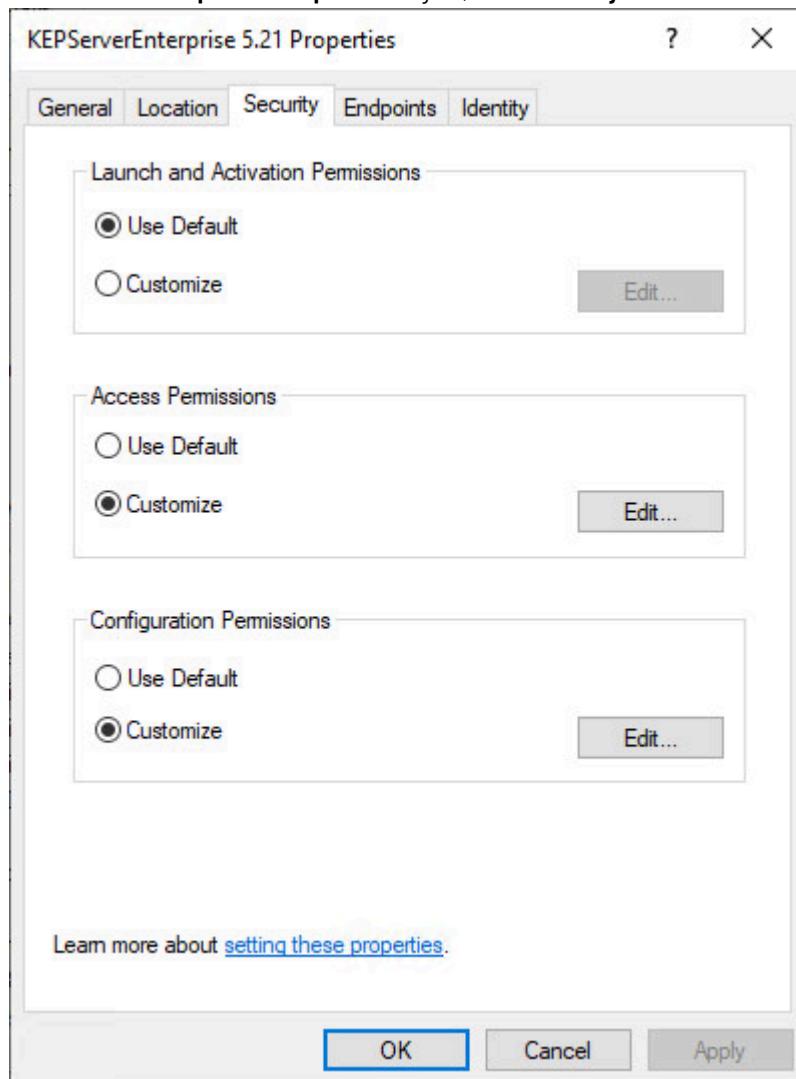
**Tip:** You can open Component Services from **Control Panel > Administrative Tools** or use the search box on the Windows **Start** menu or the taskbar. For more information about how to open Component Services, see Windows Help.

2. From the console tree, select **Component Services > Computers > My Computer > DCOM Config**.

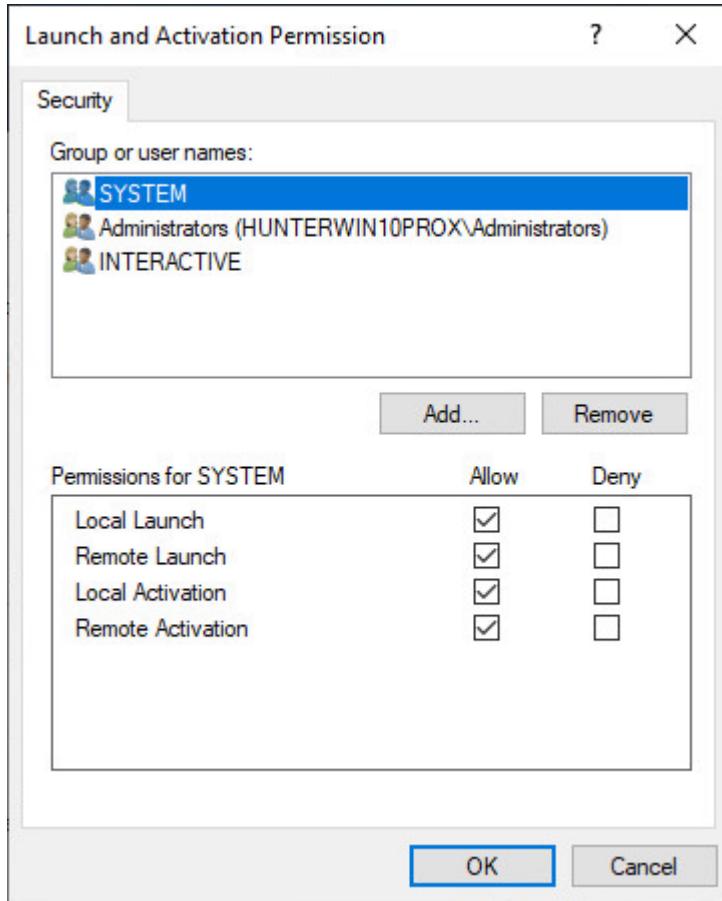


3. Under the **DCOM Config** node, right-click **KEPServiceEnterprise 5.21** and select **Properties**.

4. In the **KEPServerEnterprise 5.21 Properties** dialog box, click the **Security** tab.

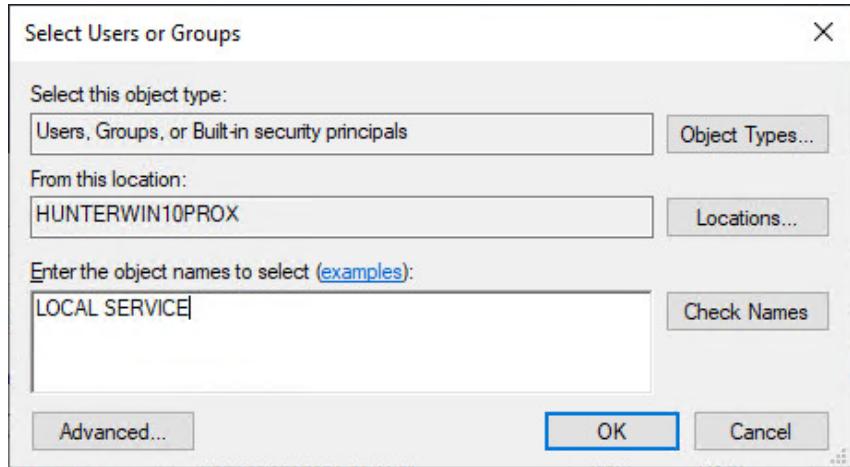


5. In the **Launch and Activation Permissions** area, select **Customize** and click **Edit**. The **Launch and Activation Permission** dialog box opens.



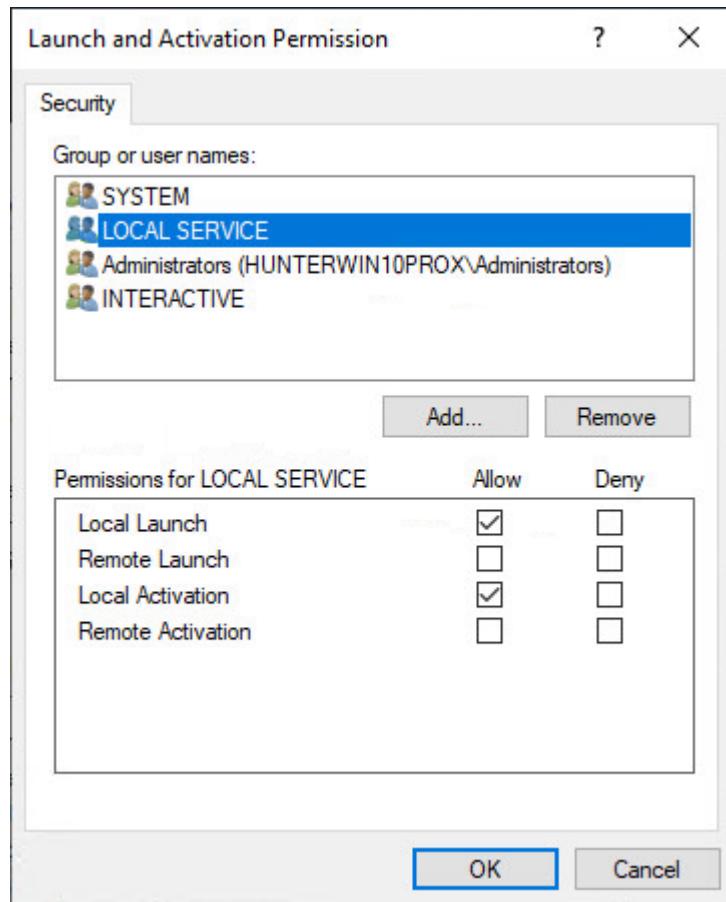
6. If **LOCAL SERVICE** is not in the **Group or user names** box, add **LOCAL SERVICE**.

- Click **Add**.
- In the **Select Users or Groups** dialog box, enter **LOCAL SERVICE**.



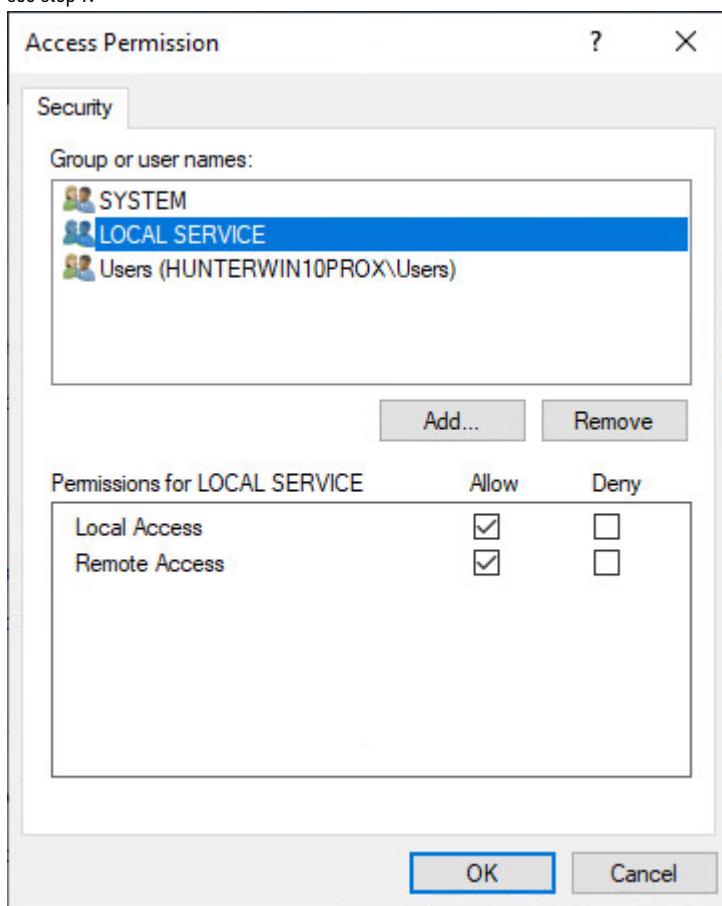
- Click **OK**.
- LOCAL SERVICE shows in the **Group or user names** list.

7. Allow Local Launch permission and Local Activation permission for LOCAL SERVICE.
  - a. In the **Group or user names** box, select **LOCAL SERVICE**.
  - b. In the **Permissions for LOCAL SERVICE** box, select the **Allow** checkboxes for **Local Launch** and **Local Activation**.



- c. Click **OK**.
8. On the **Security** tab, in the **Access Permissions** area, select **Customize** and click **Edit**. The **Access Permission** dialog box opens.
9. If **LOCAL SERVICE** is not in the **Group or user names** box, add **LOCAL SERVICE**. For detailed instruction, see step 6.

10. Allow **Local Access** permission and **Remote Access** permission for LOCAL SERVICE. For detailed instruction, see step 7.



11. Click **OK** to close the **KEPServerEnterprise 5.21 Properties** dialog box.  
12. Restart the computer to apply the changes.

## Legal Notices

Rockwell Automation publishes legal notices, such as privacy policies, license agreements, trademark disclosures, and other terms and conditions on the [Legal Notices](#) page of the Rockwell Automation website.

### Software and Cloud Services Agreement

Review and accept the Rockwell Automation Software and Cloud Services Agreement [here](#).

### Open Source Software Licenses

The software included in this product contains copyrighted software that is licensed under one or more open source licenses.

You can view a full list of all open source software used in this product and their corresponding licenses by opening the oss\_license.txt file located in your product's OPENSOURCE folder on your hard drive. This file is divided into these sections:

- Components
  - Includes the name of the open source component, its version number, and the type of license.
- Copyright Text
  - Includes the name of the open source component, its version number, and the copyright declaration.
- Licenses
  - Includes the name of the license, the list of open source components citing the license, and the terms of the license.

The default location of this file is:

C:\Program Files (x86)\Common Files\Rockwell\Help\<product name>\Release Notes\OPENSOURCE\oss\_licenses.txt.

You may obtain Corresponding Source code for open source packages included in this product from their respective project web site(s). Alternatively, you may obtain complete Corresponding Source code by contacting Rockwell Automation via the **Contact** form on the Rockwell Automation website: <http://www.rockwellautomation.com/global/about-us/contact/contact.page>. Please include "Open Source" as part of the request text.

### Commercial Software Licenses

This software also includes these commercially licensed software components:

Component	Copyright
PDFControls.NET 2.0.114	Copyright 2006-2021 TallComponents BV.
Symbol Factory 3.01.94	Copyright 1996-2023 Software Toolbox, Inc.

### Other Third-Party Free Software Licenses

This software also includes these free software components:

Component	Copyright
SQL Server LocalDB 2022 U13	Copyright Microsoft. All rights reserved.

<b>Component</b>	<b>Copyright</b>
FarPoint Spread 3.0	Copyright @1991-1999 FarPoint Technologies Incorporated
Microsoft Visual C++ 2015-2022 Redistributable	Copyright (c) Microsoft Corporation. All rights reserved.
Microsoft .NET Framework 4.8	Copyright Microsoft. All rights reserved.
VBA 7.1.11	Copyright @2018 Microsoft Corporation. All rights reserved.

# Rockwell Automation Support

Use these resources to access support information.

Technical Support Center	Find help with how-to videos, FAQs, chat, user forums, and product notification updates.	<a href="http://rok.auto/support">rok.auto/support</a>
Knowledgebase	Access Knowledgebase articles.	<a href="http://rok.auto/knowledgebase">rok.auto/knowledgebase</a>
Local Technical Support Phone Numbers	Locate the telephone number for your country.	<a href="http://rok.auto/phonesupport">rok.auto/phonesupport</a>
Literature Library	Find installation instructions, manuals, brochures, and technical data publications.	<a href="http://rok.auto/literature">rok.auto/literature</a>
Product Compatibility and Download Center (PCDC)	Get help determining how products interact, check features and capabilities, and find associated firmware.	<a href="http://rok.auto/pcdc">rok.auto/pcdc</a>

## Documentation feedback

Your comments help us serve your documentation needs better. If you have any suggestions on how to improve our content, complete the form at [rok.auto/docfeedback](http://rok.auto/docfeedback).

## Waste Electrical and Electronic Equipment (WEEE)



At the end of life, this equipment should be collected separately from any unsorted municipal waste.

Rockwell Automation maintains current product environmental information on its website at [rok.auto/pec](http://rok.auto/pec).

Rockwell Otomasyon Ticaret A.Ş. Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400 EEE Yönetmeliğine Uygundur

Connect with us.

[rockwellautomation.com](http://rockwellautomation.com) expanding **human possibility**™

AMERICAS: Rockwell Automation, 1201 South Second Street, Milwaukee, WI 53204-2496 USA, Tel: (1) 414.382.2000, Fax: (1) 414.382.4444

EUROPE/MIDDLE EAST/AFRICA: Rockwell Automation NV, Pegasus Park, De Kleetlaan 12a, 1831 Diegem, Belgium, Tel: (32) 2 663 0600, Fax: (32) 2 663 0640

ASIA PACIFIC: Rockwell Automation, Level 14, Core F, Cyberport 3, 100 Cyberport Road, Hong Kong, Tel: (852) 2887 4788, Fax: (852) 2508 1846