Performance and Visibility

Allen-Bradley

FactoryTalk® View Site Edition User's Guide

User's Guide

Allen-Bradley • Rockwell Software

Rockwell Automation

Table of contents

Chapter 1

Getting started with FactoryTalk View SE

Chapter 2

Exploring FactoryTalk View Studio

Chapter 3

Planning an application

Chapter 4

Setting up the FactoryTalk Directory

Chapter 5

Setting up security

Chapter 6

Working with network distributed applications

Chapter 7

Working with network station applications

Chapter 8

Working with local station applications

Chapter 9

Setting up communications

Chapter 10

Working with tags

Chapter 11

Setting up HMI tag alarms

Chapter 12

Setting up FactoryTalk alarms

Chapter 13

Setting up language switching

Chapter 14

Setting up FactoryTalk system availability

Chapter 15

Logging system activity

Chapter 16

Creating graphic displays

Chapter 17

Creating graphic objects

Chapter 18

Animating graphic objects

Chapter 19

Setting up navigation

Chapter 20

Creating expressions

Chapter 21

Creating embedded variables

Chapter 22

Setting up data logging

Chapter 23

Setting up trends

Chapter 24

Adding logic and control

Chapter 25

FactoryTalk View commands

Chapter 26

Setting up DDE communications for HMI tags

Chapter 27

Using the SE Client object model and display code

Chapter 28

Importing and exporting XML files

Legal Notices

Index

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.
- FactoryTalk View tools and utilities.
- How to set up the software you need.
- Running FactoryTalk View SE without activation.
- Exploring the InstantFizz application.
- How to start creating a new network distributed application.

Welcome to FactoryTalk View Site Edition

FactoryTalk View Site Edition 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.

A member of the FactoryTalk View family of products, FactoryTalk View Site Edition (also called FactoryTalk View SE) provides all the tools you need to create powerful, dependable process monitoring and supervisory control applications.

In FactoryTalk View Studio, you can create FactoryTalk View SE 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.

About FactoryTalk systems

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

FactoryTalk Services Platform

The 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 services, 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 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 77.
- **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 85.
- **FactoryTalk Live Data** manages connections between FactoryTalk products and data servers. For information about data communications in FactoryTalk View SE applications, see Setting up communications on page 177.
- **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 383.
- **FactoryTalk Administration Console** is an optional, stand-alone tool for developing, managing, and securing multiple 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.

For more information about FactoryTalk Alarms and Events and FactoryTalk Activation, see the list of FactoryTalk View SE software below.

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.

To open the FactoryTalk Help:

Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **FactoryTalk Help**.

You can also open the FactoryTalk **Help** by clicking Help in dialog boxes used to set up FactoryTalk components and services.

FactoryTalk View Site Edition software

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

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

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

For an overview of FactoryTalk Services Platform, see FactoryTalk Services Platform on page 26.

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

FactoryTalk View Studio contains editors for creating complete applications, and includes client and server software for testing the applications you create. Use the editors to create applications that are as simple or as complex as you need.

For information about FactoryTalk View Studio features, see Exploring FactoryTalk View Studio on page 41.

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 85.

Note: FactoryTalk View Studio is also the configuration software for developing FactoryTalk View Machine Edition (ME) applications. For information about FactoryTalk View ME, see the *FactoryTalk View Machine Edition 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 run time.

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

The 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 run-time clients to receive FactoryTalk device-based and tag-based alarm information, you need to set up application servers to support FactoryTalk Alarms and Events services. For details, see Setting up FactoryTalk alarms on page 261.

FactoryTalk Activation provides 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 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 run time.
- 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 run time. In a network distributed application, multiple clients can run in different languages simultaneously.
- Modify HMI tag and alarm properties at run time, and have changes take effect at connected clients, without restarting the clients.

  In a redundant network distributed application, changes made at the primary HMI server can be replicated to the secondary server.
- Create a complete alarm monitoring system that includes FactoryTalk Alarms and Events and HMI tag alarms.

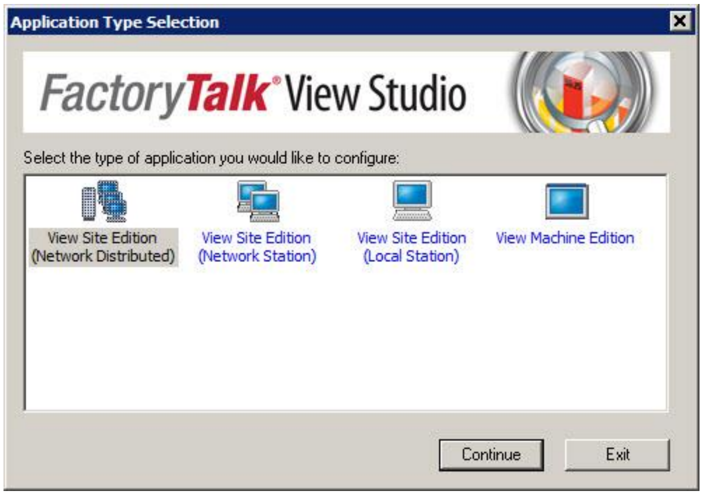
  Using FactoryTalk Alarms and Events services, FactoryTalk View SE applications can subscribe to and display device-based alarms, programmed directly into Logix5000 controllers.
- Customize alarm summaries and banners to provide specific alarm data, rather than displaying alarms for the entire system.
- 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 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 and VBA 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. The available applications are shown below.


Application Type Selection

Select the type of application you would like to configure:

FactoryTalk View Studio
View Site Edition (Network Distributed)
View Site Edition (Network Station)
View Site Edition (Local Station)
View Machine Edition
Continue
Exit

- 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 located 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 located 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 that uses more than 10 computers, you must use a network domain controller. For network distributed applications consisting of 10 computers or fewer, you can use a Windows workgroup.

Note: Do not run FactoryTalk Directory, or any other application software, on the same computer as a Windows domain controller.

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 RSLinx Enterprise

RSLinx Enterprise is a FactoryTalk Live Data server that can run on multiple platforms, from PanelView Plus dedicated terminals to desktop computers.

For communications with Allen-Bradley local and remote devices—particularly with Logix5000 controllers—RSLinx Enterprise is the recommended data communications software for FactoryTalk View applications.

When to use RSLinx Classic

RSLinx Enterprise is the preferred choice when using Logix controllers, but there are some cases where Linx Classic must be used.

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.

With RSLinx Classic, you can also create alias topic shortcuts and perform online tasks such as uploading and downloading RSLogix 5000 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 supports the OPC-DA(Open Platform Communications Data Access) 2.05a specification.

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 and RSLinx 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.

With a Site Edition network distributed application in demo mode, you can:

- Create or load up to five HMI servers locally in FactoryTalk View Studio.
- Create or load up to five graphic displays per HMI server.
- Run a local FactoryTalk View SE Client for up to two hours. Remote clients cannot connect to an application in demo mode.
- Import only the first graphic in each category from Symbol Factory. Symbol Factory is a graphics library with over 5,000 quality graphic images.

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.

There are two types of FactoryTalk Directory:

- **Local Directory** manages applications that are confined to a single computer, for example, FactoryTalk View SE local station applications.
- **Network Directory** manages applications that can consist of multiple clients and servers on separate computers connected over a network, for example, FactoryTalk View SE network distributed and network station applications.

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 77.

Exploring the InstantFizz application

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

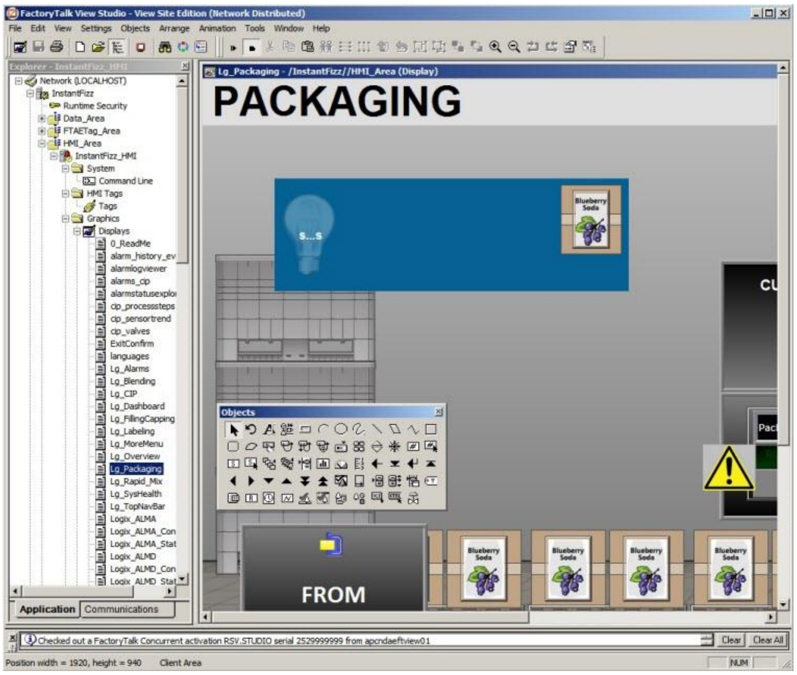
To explore the FactoryTalk View SE development and run-time environments, open the InstantFizz application in FactoryTalk View Studio and launch the SE client.

Tip: Before running the client, you need to download the control logix project **IF2_DEMO.ACD** located at **\Documents and Settings\All Users\Documents\RSView Enterprise\SE\HMI Projects\InstantFizz_HMI** to a programmable controller or a SoftLogix controller.

To open the InstantFizz application

1. Start FactoryTalk View Studio, click **View Site Edition (Network Distributed)**, and then click **Continue**.

2. In the **Existing** tab, click **InstantFizz**, select an application language, and then click **Open**.



Checked out a FactoryTalk Concurrent activation RSV.STUDIO serial 2529999999 from apcndaeftview01

In the illustration, the graphic display named Packaging and the undocked Objects toolbar are open in the Graphics editor.

For information about working with FactoryTalk View Studio and the editors, see Exploring FactoryTalk View Studio on page 41.

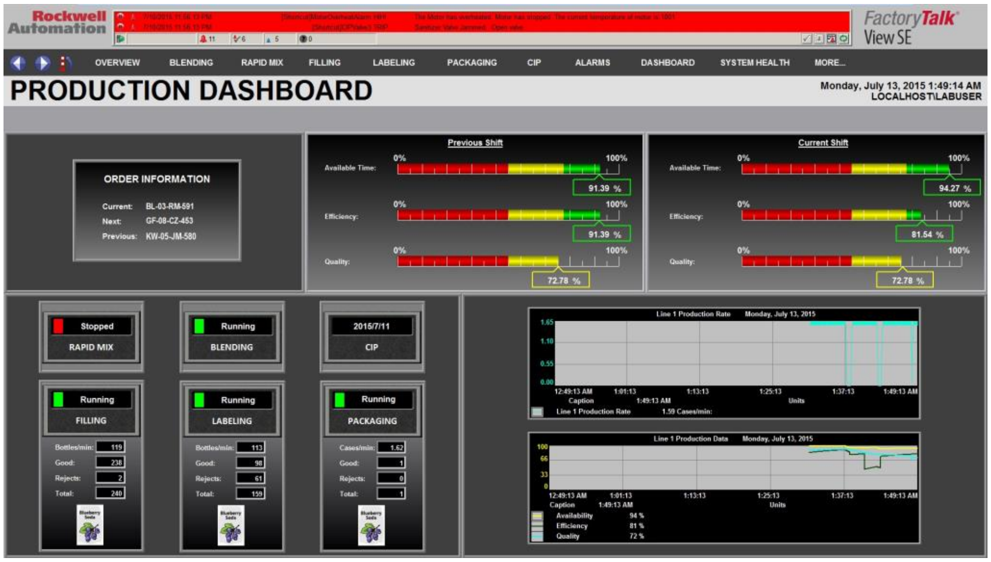
To test a display in FactoryTalk View Studio

1. In FactoryTalk View Studio, in the Explorer window, right-click the display you want to test, and then click **Open**.

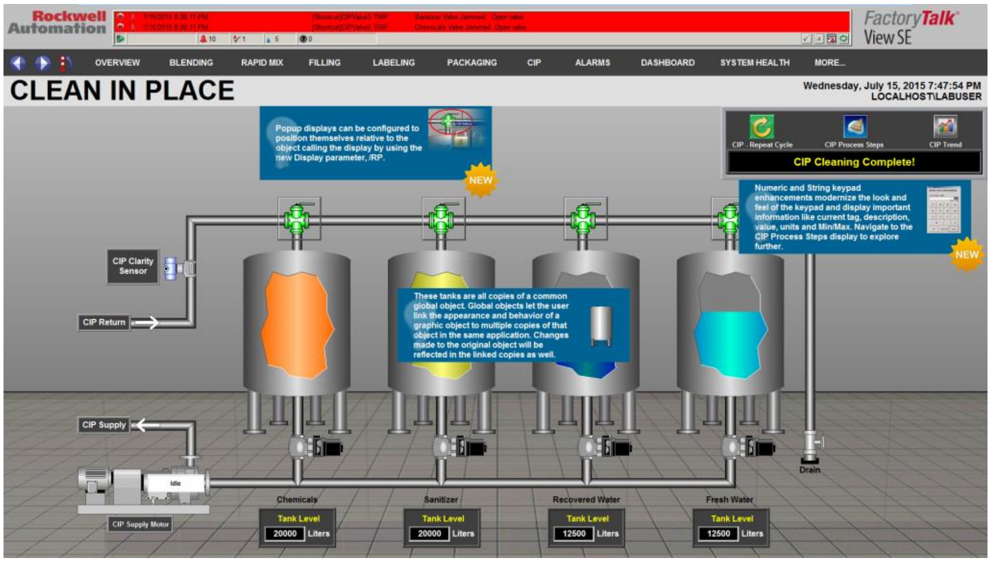
2. On the View menu, click **Test Display**. To stop testing, on the View menu, click **Edit Display**.

To run the InstantFizz client

Open **C:\Users\Public\Documents\RSView Enterprise\SE\Client** and 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**.


Popup displays can be configured to position themselves relative to the object calling the display by using the new Display parameter, /RP.

Numeric and String keypad enhancements modernize the look and feel of the keypad and display important information like current tag, description, value, units and Min/Max. Navigate to the CIP Process Steps display to explore further.

These tanks are all copies of a common global object. Global objects let the user link the appearance and behavior of a graphic object to multiple copies of that object in the same application. Changes made to the original object will be reflected in the linked copies as well.

CIP Cleaning Complete!

Creating a new FactoryTalk View SE application

You can also create a simple, new application to exercise some of the development and run-time 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.

The HMI server stores project components (for example, graphic displays) and serves these components to clients. The server also contains a database of tags, and performs alarm detection, and historical data management (logging).

To create a new network distributed application

1. Select **Start > All Programs > FactoryTalk View Studio**.
2. In the Application Type Selection dialog box, click **Site Edition (Network Distributed)**, and then click **Continue**.
3. In the New/Open Site Edition (Network Distributed) Application dialog box, click the **New** tab.
4. Type a name and description for the application, select an application language, and then click **Create**.

   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 323.

After creating the application, you can add an HMI server to the root of the application, or to another area in the application. For this example, add an area to contain the new HMI server.

To add an area to the application

1. In FactoryTalk View Studio, in the Explorer window, right-click the application icon, and then click **New Area**.
2. Type a name and description for the area, and then click **OK**.

To add an HMI server

1. In FactoryTalk View Studio, in the Explorer window, right-click the area you just created, click Add New Server, and then click HMI Server.
2. In the Add HMI Server Wizard, in the Select Operation window, click Create a new HMI server, and then click Next.

   For information about other options in the Select Operation window, see Adding an HMI server on page 156.

3. Type a name and description for the HMI server, and then click **Finish**.

For details about options in the Add HMI Server Wizard, click **Help**.

Tip: If it's set to display when you create a new HMI server, the Add Controller Instruction Faceplates dialog box opens. If you don't want to add faceplates, click **Cancel** to close the dialog box, without affecting HMI server creation. For more information about adding faceplates, see Adding controller instruction faceplates to an application on page 422.

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 51. For instructions about using the editors, click **Help** in the editor dialog boxes.

For this example, use the Graphics editor to create a graphic display, and then add a graphic object to that display.

To create a new display

- In FactoryTalk View Studio, in the Explorer window, open the Graphics folder, right-click the **Displays** icon, and then click **New**.

  An untitled display opens in the workspace to the right of the Explorer window.

To add a graphic object to the display

1. In the Graphics editor, on the Objects menu, click **Drawing**, and then click **Rectangle**.

   You can also click the button on the Objects toolbar that represents the object you want to add.

   For information about using the Graphics editor, see Creating graphic displays on page 393 and Creating graphic objects on page 449.

2. Position the pointer where you want to draw the rectangle, click and hold the left mouse button, and then drag the mouse down and to the right.

3. Release the mouse button to place the rectangle on the display.

To save the new display

1. On the **File** menu, click **Close**.
2. Click **Yes** to save changes and close the display.
3. In the **Save** dialog box, type a name for the display, and then click **OK**.

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 run time.

To create a FactoryTalk View SE Client configuration file

1. Select **Start > All Programs > Rockwell Software > FactoryTalk View > FactoryTalk View Site Edition Client**.
2. In the FactoryTalk View SE Client Wizard, click **New**, and then follow the instructions in the wizard.

For details about options in the FactoryTalk View SE Client Wizard, click **Help**.

To run the FactoryTalk View SE Client

1. In the Completion Options window of the FactoryTalk View SE Client Wizard, select **Save configuration and open FactoryTalk View SE Client now**.

2. Click **Finish**.

   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 configuration software for developing and testing FactoryTalk View Site Edition applications.

FactoryTalk View Studio contains editors for creating complete applications, and contains software for testing the applications you create. Use the editors to create applications that are as simple or as complex as you need.

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 85.

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 run FactoryTalk View Studio, select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **FactoryTalk View Studio**.

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 permissions you might require.

For an overview of FactoryTalk Security services, see Setting up security on page 85. For details, see FactoryTalk Security Help.

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. Start FactoryTalk View Studio, select the application type, and then click **Continue**.
2. In the **New/Open Application** dialog box, click the **Existing** tab.
3. Click the name of the application you want to open, select an application language, and then 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 323.

Resolving access problems to the HMI Projects folder

If you have problems accessing the HMI projects folder, you might have to change your permissions. To open an application in FactoryTalk View Studio, users need Windows read and write access to the folder that contains the HMI projects.

A Windows user in the Power Users category might not have write access.

To set up write access for Power Users:

1. In Windows Explorer, right-click the HMI Projects folder, and then select **Properties**.

   Tip: The default path for the HMI Projects folder is:

   - \Users\Public\Public Documents\RSView Enterprise\SE\HMI projects
   - (Windows XP) \Documents and Settings\All Users\Shared Documents\RSView Enterprise\SE\HMI projects

2. In the **Security** tab, select **Power Users** from the list of group or user names.

3. In the Permissions list under the list of names, beside Full Control, select the **Allow** check box.

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.

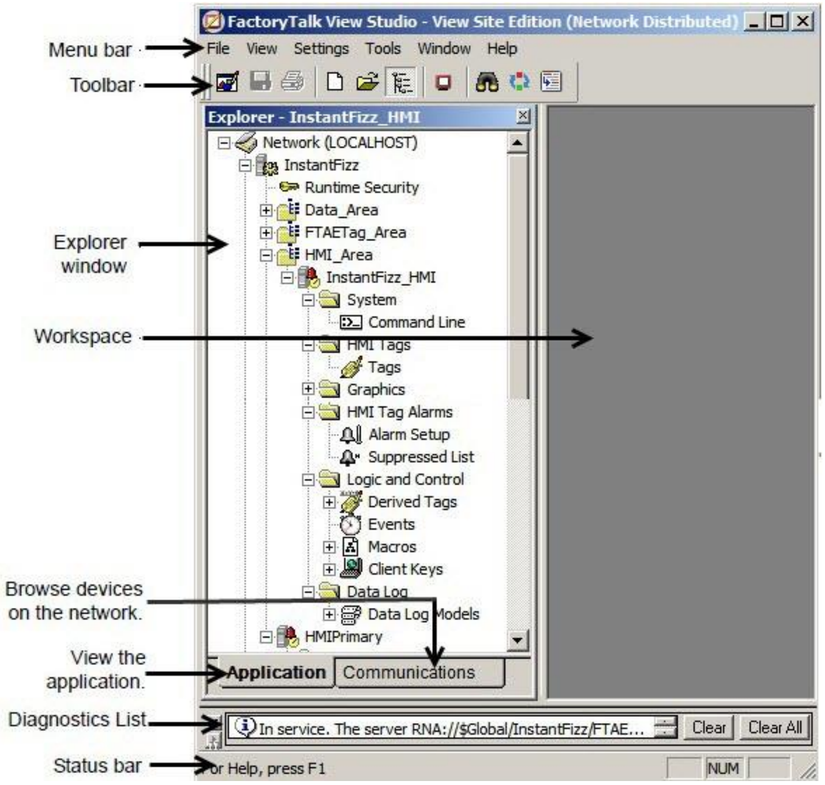
To open the Samples application:

1. Start FactoryTalk View Studio, click **View Site Edition (Network Distributed)**, and then click **Continue**.

2. In the **Existing** tab, click **InstantFizz**, select an application language, and then click **Open**.

Parts of the FactoryTalk View Studio main window

When you create or open an application, its contents are shown in the FactoryTalk View Studio main window.



In service. The server RNA://$Global/InstantFizz/FTAE...

For Help, press F1

Menu bar

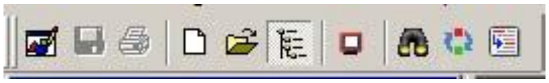
The menu bar contains the menu items for the active window. Each editor has its own set of menus.

Toolbars

The toolbars contain buttons that provide quick access to commonly used menu items. When you point to a toolbar button, the name of the button is shown in a tooltip.

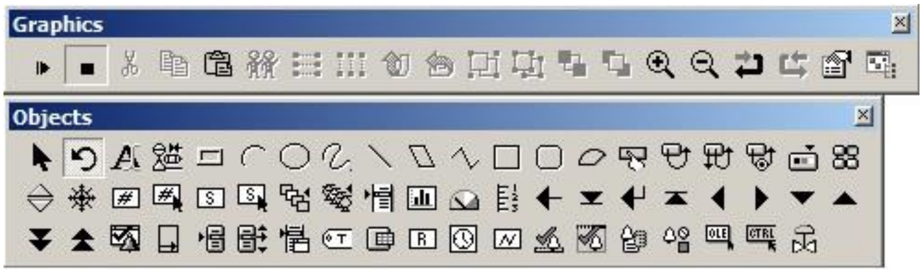
Tip: Find, Replace, and Cross Reference toolbar buttons have been added with FactoryTalk View 7.0.

All FactoryTalk View editors use the Standard toolbar, shown here in its undocked form:



The Graphics editor has additional toolbars that are shown when you open a graphic display, a global object display, or a library.

The next illustration is of the Graphics and Objects toolbars, in their undocked forms:



Explorer

The Explorer is the main tool for working in FactoryTalk View Studio. It provides access to the editors you use to set up the application and create its components. For more information about the Explorer, see Working in the Explorer window on page 48.

Workspace

The workspace is 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. For details, see Working in the Explorer window on page 48.

Application tab

The Application tab contains the Explorer window.

Communications tab

The Communications tab shows the devices on the network available to the computer hosting RSLinx Enterprise. Use this tab for setting up network types and browsing devices on the configured networks.

For more information, see Setting up communications on page 177. For details, see RSLinx Enterprise Help.

Diagnostics List

The Diagnostics List shows information about system activities. It’s located above the status bar at the bottom of the FactoryTalk View Studio main window.

You can hide, move, resize, and clear messages from the Diagnostics List.

Use the Diagnostics Setup tool to select the types of messages that are shown in the Diagnostics List. For details, see Message routing on page 385, or click **Help** in the Diagnostics Setup tool.

Diagnostic messages are preceded by a blue, yellow, or red icon. Blue indicates information, yellow indicates a warning, and red indicates an error.

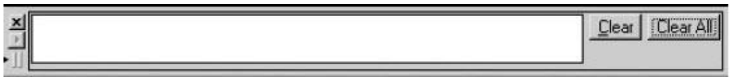
To show or hide the Diagnostics List

- In FactoryTalk View Studio, from the View menu, select **Diagnostics List**.

When Diagnostics List has a check mark beside it, the list is visible.

To move the Diagnostics List

1. Click and hold the grab bars at the bottom left of the Diagnostics List.



Tip: If you can’t see the grab bars, drag the top edge of the Diagnostics List, to make it larger.

2. Drag the list to its new location. To prevent the Diagnostics List from docking automatically, press and hold the **Ctrl** key as you drag.

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

Removing messages from the Diagnostics List

Use one of these methods:

- Click **Clear**, to remove the most recent message (at the top of the list), or to remove the selected message.
- Click **Clear All**, to remove all the messages in the list.

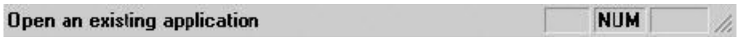
Removing a message from the Diagnostics List does not delete the message from the Diagnostics log.

Status bar

The information shown in the status bar depends on what you are doing in FactoryTalk View Studio and where the pointer is. The status bar can provide information about:

- The active window or selected tool.

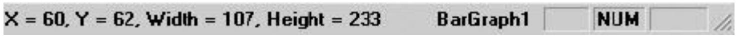
  For example, if you position the pointer over the Open button on the Standard toolbar, the status bar shows the following message:

  Open an existing application

- The selected graphic object.

  For example, if you select an object in the Graphics editor, the status bar shows the object’s position, size, and name:

  X = 60, Y = 62, Width = 107, Height = 233 BarGraph1

Workbook mode

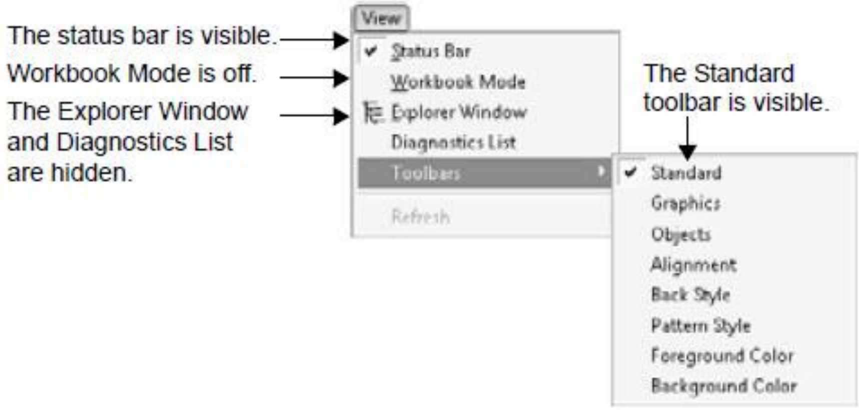
Workbook mode provides a different way to switch between open windows in the FactoryTalk View Studio workspace.

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

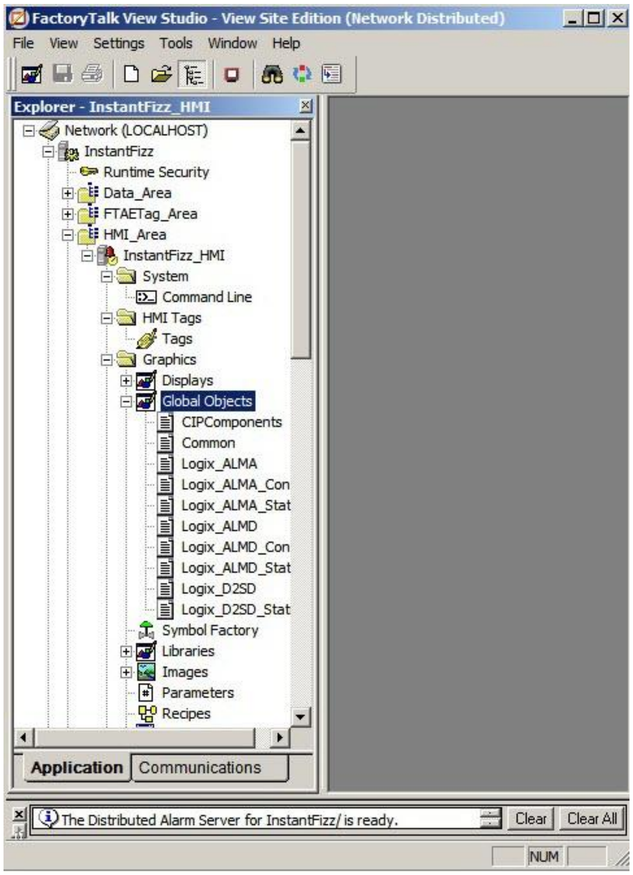
To show or hide elements of the main window (except the menu bar), select items on the View menu: if there is a check mark beside an item, the item is visible; if there is no check mark, the item is hidden.



Working in the Explorer window

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

The following illustrations shows what a network distributed application looks like in the Explorer window.



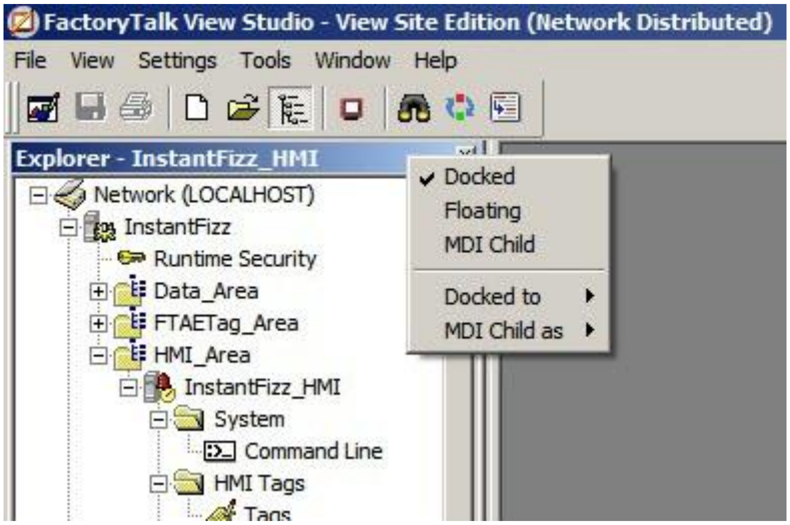
The Distributed Alarm Server for InstantFizz/ is ready.

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 in the middle of the main window. You can then drag it to a new position.

- Right-click the Explorer’s title bar, and then click a new docking location: Floating, or MDI (Multiple Document Interface) Child.



If you click MDI Child, the Explorer becomes a window that you can move, minimize, maximize, or restore, within the main window. You cannot move the Explorer outside the main window.

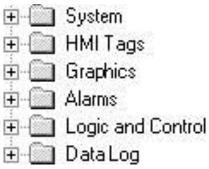
As long as it is undocked, you can resize the Explorer. To do this, click an edge or corner of the window, and then drag until it’s the size you want.

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.



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 on 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.

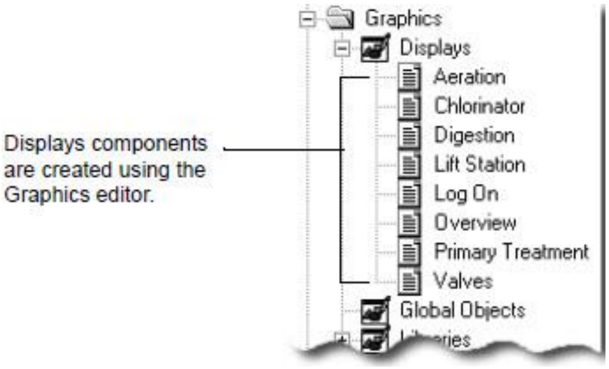
Following is a list of FactoryTalk View SE editors and their functions. If you can open an editor from the FactoryTalk View Studio menu bar, the name of the menu is noted.

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. **TIP**: The settings you specify in this editor apply **only** to the computer you are using.	Diagnostics Setup (Tools menu)
For HMI tag alarms only, specify what, where, when, and how alarm activity is to be logged. **TIP**: The settings you specify in this editor apply **only** to the computer you are using, and only if the computer contains an HMI server.	HMI Tag Alarm Log Setup (Tools menu)
Add languages to an application, set up a default language, and export and import text strings.	Languages (Tools menu)
Open a command line to run FactoryTalk View commands and macros.	Command Line
Create HMI tags and set up HMI tag alarms.	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 run time.	Parameters
Create recipe files to specify values for input and display objects in graphic displays.	Recipes

Create local message files containing trigger values, and the corresponding messages.	Local Messages
For HMI tag alarms only, set up general features such as alarm severities and user messages.	Alarm Setup
For HMI tag alarms only, display a list of tags that have alarm notification suppressed.	Suppressed List
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
Re-map 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 located in folders under the application directory.



Displays 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 new 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

To add a component, using drag and drop:

1. In Windows Explorer, locate the file to add to the application.
2. Drag the file to the Explorer window in FactoryTalk View Studio.

   The component is added to the Explorer automatically, under the editor used to modify it.

To add a component, using the Add Component option:

1. In FactoryTalk View Studio, in the Explorer window, right-click the editor used to create the component you want to add, and then click **Add Component into Application**.

   For example, to add a graphic display, right-click the Displays icon, and then click **Add Component into Application**.

2. Find and select the component you want to add, and then click **Open**.

   Selected components are added to the Explorer, under the editor's icon.

   Tip: 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.

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

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 filename, you need to keep in mind 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 115 characters in the following path and file name, as the limit is 260, leaving 145 additional characters, including spaces, allowable:

```
C:\Documents and Settings\All Users\Documents\RSView
Enterprise\SE\HMI Projects\InstantFizz_SE\Gfx\cip overview.gfx
```

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

```
C:\Documents and Settings\All Users\Documents\RSView
Enterprise\SE\HMI Projects\InstantFizz_SE\Mcr
```

That is 108 total characters just for the path, 260 - 108 = 142. The filename can be up to 142 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, right-click the component you want to delete, and then select **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 after renaming the physical file

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 Application**.

   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, that require similar information. Knowing about these features saves time.

Gaining quick access to common operations

Shortcut menus provide quick access to actions 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.

Tag names are used, for example, to specify connections for input objects in FactoryTalk View graphic displays. 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 run time.
- Click the **Browse** or **Tags** button (whichever is available) to open the Tag Browser, where you can select or create a tag.

For more information about tags, see Working with tags on page 189.

Selecting and building commands

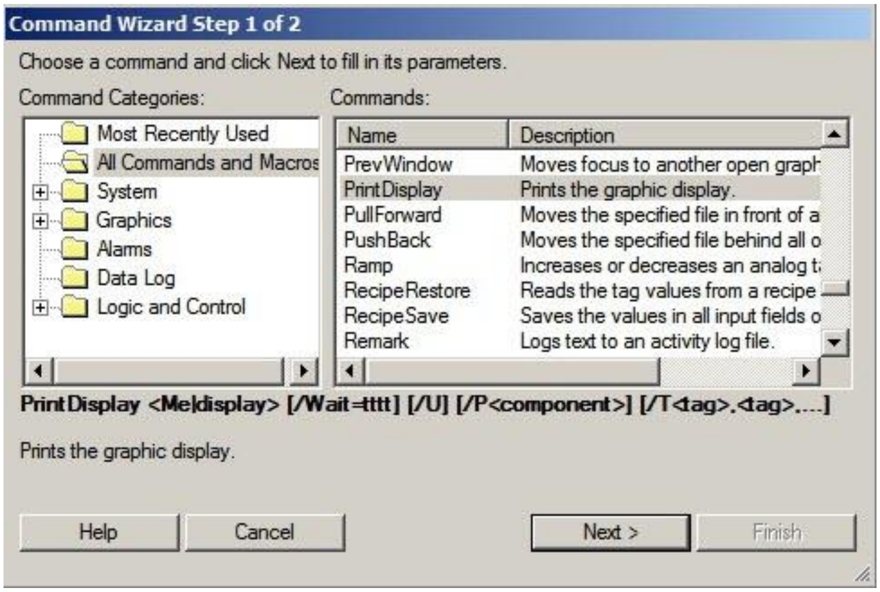
You can use FactoryTalk View commands to interact with and control application components. Most commands accept parameters for added precision and control.

You can set up keys and graphic objects to run commands at run time. 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.

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.
- From the **Edit** menu, select **Commands**.



Command Wizard Step 1 of 2

Choose a command and click Next to fill in its parameters.

Command Categories:

Commands:

Name	Description
PrevWindow	Moves focus to another open graph
PrintDisplay	Prints the graphic display.
PullForward	Moves the specified file in front of a
PushBack	Moves the specified file behind all o
Ramp	Increases or decreases an analog t
RecipeRestore	Reads the tag values from a recipe
RecipeSave	Saves the values in all input fields o
Remark	Logs text to an activity log file.

PrintDisplay <Me|display> [/Wait=tttt] [/U] [/P<component>] [/T<tag>,<tag>,...]

Prints the graphic display.

For information about specific FactoryTalk View commands, click **Help**.

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 555.

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 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 run time

To print graphic displays at run time, 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, even if parts are covered by other displays. If you want to print an image of whatever shows on the monitor, use the ScreenPrint command.

For more information about the PrintDisplay command, 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 run-time 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 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 FactoryTalk View SE network distributed application.

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

Choosing a Windows domain or workgroup

For network applications consisting of more than 10 computers, the use of a domain controller is strongly recommended, but not required. The following domains are supported:

- Windows Server 2012 (Standard and Datacenter editions)
- Windows Server 2012 R2 (Standard and Datacenter editions)
- Windows Server 2008 with Service Pack 2 (Standard Edition)
- Windows Server 2008 R2 with Service Pack 1 (Standard Edition)

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

Windows workgroups

For network distributed applications consisting of 10 computers or fewer, 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 85.

Tip: For workgroup applications running in Windows XP, you must turn off simple file sharing on each computer in the workgroup. For more information, see *FactoryTalk View Site Edition Installation Guide*.

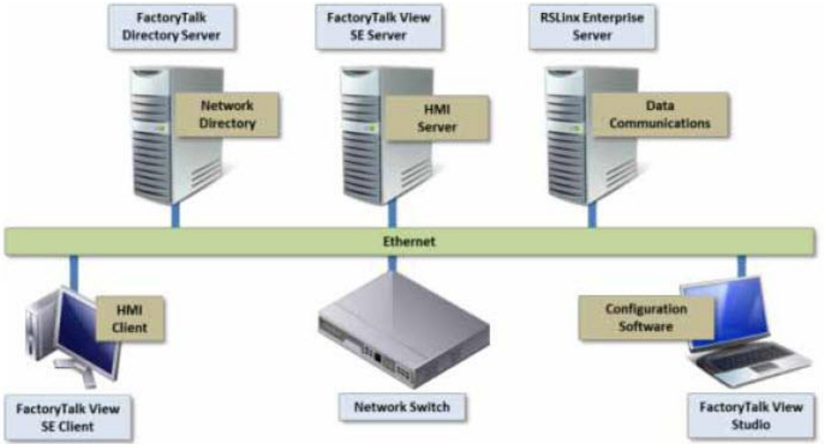
Determining which computers you'll need

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 FactoryTalk View SE system requirements and limits related to running network distributed application components, see the *FactoryTalk View Site Edition Installation Guide*.

Tip: If you plan to deploy a network distributed application that uses more than two servers (or two pairs of redundant servers) and 20 clients, or if you would like architectural assistance, contact your local Rockwell Automation Sales office for architectural assistance . For information about setting up redundant FactoryTalk View SE Servers, see Setting up FactoryTalk system availability on page 351.

Following is a sample architecture of a distributed system. It includes a FactoryTalk Directory Server, FactoryTalk View SE Server and Client, RSLinx Enterprise 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**

  The 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 manages network distributed applications. All the computers in a given network application must point at the same FactoryTalk Network Directory.

For more information, see Setting up the FactoryTalk Directory on page 77 and Working with network distributed applications on page 125.

- **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 alarm detection and 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 351.

- **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 data servers that comply with the OPC-DA 2.05a specification.

  For information about data servers in a FactoryTalk View SE application, see Setting up communications on page 177 and Working with network distributed applications on page 125.

  Tip: If you use an OPC data server instead of RSLinx Enterprise, 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 41.

  Tip: FactoryTalk View Studio is also the tool for developing FactoryTalk View Machine Edition applications. For information about developing machine-level applications, see the *FactoryTalk View Machine Edition User's Guide*.

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

The FactoryTalk View SE Client software provides a run-time 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 sub-set of editors available in the Administration Console to make minor changes to an application without installing FactoryTalk View Studio.

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 177.

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 need to 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 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 set up a traditional alarm system, using HMI tags. You can also use FactoryTalk Alarms and Events services, to centralize the distribution of alarm information from various devices in your application, to run-time clients and logs.

For more information about alarm monitoring and control, see Setting up HMI tag alarms on page 217, and Setting up FactoryTalk alarms on page 261.

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

There are advantages to using FactoryTalk Alarms and Events services for alarm monitoring and control, when compared with using traditional, HMI tag alarms.

For example, FactoryTalk Alarms and Events provides a single, integrated set of alarm information, giving applications a system-wide view and record of alarm activity, enabled by alarm servers that support FactoryTalk services.

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

Device-based alarms are set up by programming alarm detection instructions directly into Logix5000 controllers that support built-in alarms.

Tip: For a list of Logix5000 controllers that support FactoryTalk Alarms and Events services, see the FactoryTalk Alarms and Events Help.

If you use device-based alarms, you only have to program 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 alarms by way of Rockwell Automation Device Servers (RSLinx Enterprise) that are set up to support FactoryTalk Alarms and Events. For details, see Working with Rockwell Automation Device Servers on page 279.

Choosing tag-based alarms

Tag-based alarms are set up by specifying alarm conditions for tags in programmable controllers (PLC-5 or SLC 500), or third-party devices communicating through OPC data servers, that do not have built-in alarm detection.

If you want to take advantage of integrated FactoryTalk Alarms and Events services, use tag-based alarms as an alternative to using HMI tag alarms.

FactoryTalk View SE Clients receive tag-based alarm information, by way of FactoryTalk 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 281, and Setting up FactoryTalk tag-based alarms on page 282.

Traditional HMI tag alarms

In a FactoryTalk View SE application, you can also set up alarms for tags in a FactoryTalk View SE Server’s tag database. These alarms are called HMI tag alarms.

If your application uses HMI tags for other purposes, and you want to monitor these tags for alarms, use a traditional HMI alarm system.

FactoryTalk View SE Clients receive HMI tag alarm information, by way of the HMI servers where the HMI tags reside.

HMI tag alarm information is not distributed by FactoryTalk Alarms and Events services. This means that you cannot monitor and interact with HMI tag alarms, using FactoryTalk alarm displays and logs.

For more information about HMI tag alarm monitoring, see Setting up HMI tag alarms on page 217.

Deciding when to use HMI tags

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

- Triggering an alarm when the value of an HMI tag crosses a specified threshold.
- 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 re-use 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 189.

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
- HMI tag alarm requirements 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 2000.

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 log on 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 85.

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 these 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 (RSLinx Enterprise), and OPC data servers (RSLinx Classic, and other OPC 2.05a Data Access servers).
- **Support for online changes** to HMI tag and alarm properties. For information about this feature, see Modifying HMI tag and alarm properties at run time on page 373.
- **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 366.

- **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 380.

For more information, see Setting up FactoryTalk system availability on page 351.

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 351.

In addition, you can set up FactoryTalk View SE to send HMI tag alarm, 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:

- HMI tag alarm logging, see Setting up HMI tag alarms on page 217.
- data logging, see Setting up data logging on page 585.
- activity logging, see Logging system activity on page 383.

Setting up the run-time 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.
- 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 393 and Creating graphic objects on page 449.

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 543.

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.

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.

Tip: Use global objects to make templates easier to distribute and maintain. For information about global objects, see Creating graphic displays on page 393.

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 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 603.

Planning run-time language switching

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

- How operators will switch between languages at run time.

  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 323.

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 393.
- 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 RSLogix 5000, 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 633.

- 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.

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:

Adding 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.

Validating 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.

Creating custom operator forms

Use the FactoryTalk View SE Client object model to populate VBA form objects with data, for use in graphic displays. For example, provide list boxes or combo boxes for operators to select recipe items from a display.

Manipulating the FactoryTalk View SE Client window

Write VBA code to arrange graphic displays based on the size of the FactoryTalk View SE Client window, so that the run-time application can adapt dynamically to various desktop sizes and resolutions.

Sending custom messages to the Diagnostics log

Send specific messages to the Diagnostics List and Diagnostics log, to notify the operator of problems with VBA code.

Securing the system

Use the FactoryTalk View SE object model to obtain security information about who is using the system and to control access to the system.

For example, you can restrict a user’s access to a graphic display on a secured computer by creating code that displays the graphic display on a workstation with a specific computer name, only when the user is logged in with a particular user name.

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

The FactoryTalk Directory centralizes access to 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**, or the Local Directory, manages local station applications.

  All local application components, except for OPC data servers, must be located on the same computer.

- **FactoryTalk Network Directory**, or the Network Directory, manages network distributed and network station applications.

  Network distributed applications can consist of multiple clients and servers, distributed across several computers connected over a network.

  One Network Directory manages all of the FactoryTalk products that participate in a single network distributed or network station application.

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*.

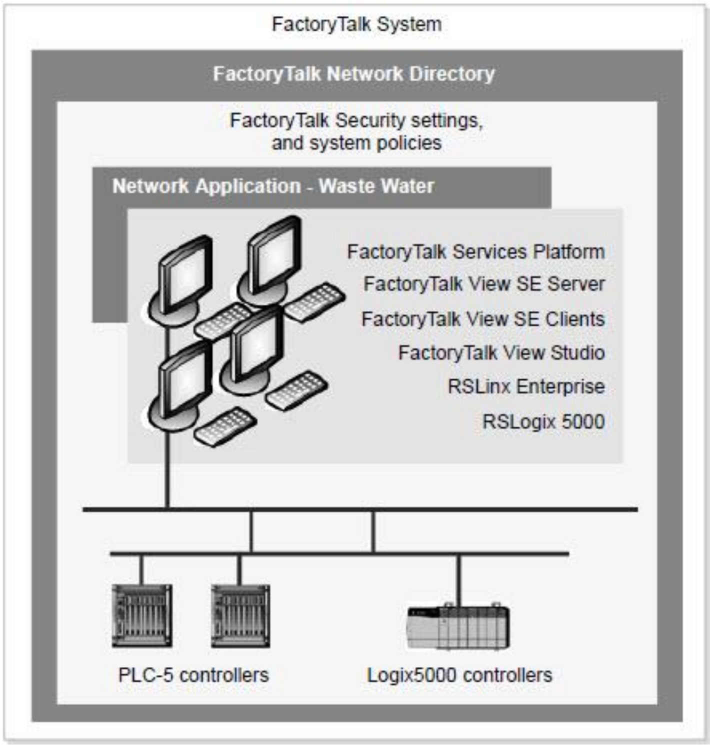
FactoryTalk Directory in a networked system

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.

For an overview of FactoryTalk services, see About FactoryTalk systems on page 127.

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


FactoryTalk System
FactoryTalk Network Directory
FactoryTalk Security settings, and system policies
Network Application - Waste Water
FactoryTalk Services Platform
FactoryTalk View SE Server
FactoryTalk View SE Clients
FactoryTalk View Studio
RSLinx Enterprise
RSLogix 5000
PLC-5 controllers
Logix5000 controllers

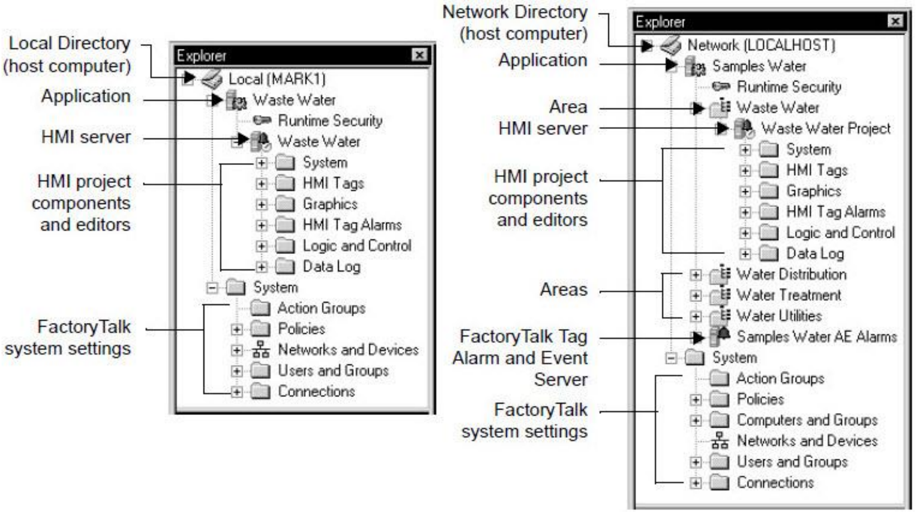
In the illustration, computers hosting FactoryTalk View SE Clients and FactoryTalk View Studio software use the Network Directory to find computers hosting FactoryTalk View SE Servers.

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 RSLinx Enterprise, to gain access to data on network devices (for example, PLC-5 and Logix5000 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 the these illustrations:



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 located on the same computer as FactoryTalk View Studio.
- The host computer name, if the Network Directory is located 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 RSLinx Enterprise data server, or one FactoryTalk Tag Alarm and Event Server, which also must be 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 167.

Running network applications

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.

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 could develop one application while another is operating in a production environment. Alternatively, you could run multiple applications simultaneously, with each application controlling a different facility within a corporate network.

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.

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.

  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.

Specifying the location of the FactoryTalk Directory server

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 run FactoryTalk Directory, or any other application software, on the same computer as a Windows domain controller.

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*.

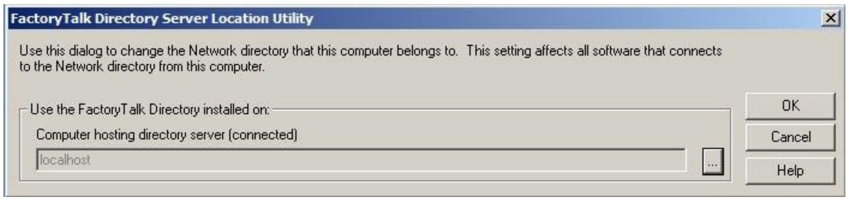
To specify localhost on the Network Directory server computer:

1. On the computer hosting the Network Directory server, select **Start > All Programs > Rockwell Software > FactoryTalk Tools > Specify FactoryTalk Directory Location**.

2. You are prompted to log on. Type your name and password, and then click **OK**.

   To use the FactoryTalk Directory Server Location Utility, you must log on 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**.



FactoryTalk Directory Server Location Utility

Use this dialog to change the Network directory that this computer belongs to. This setting affects all software that connects to the Network directory from this computer.

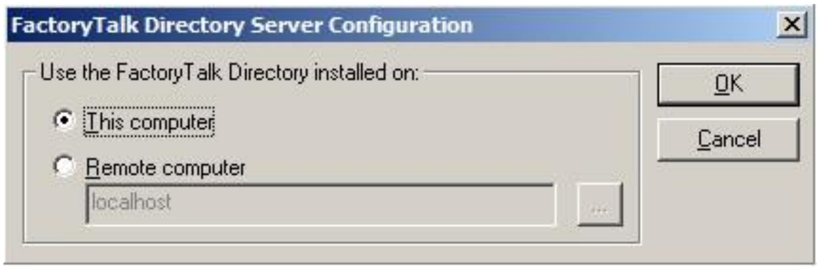
Use the FactoryTalk Directory installed on:

Computer hosting directory server (connected)

localhost

   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**.



FactoryTalk Directory Server Configuration

Use the FactoryTalk Directory installed on:

● This computer

○ Remote computer

localhost

5. Click **OK**.

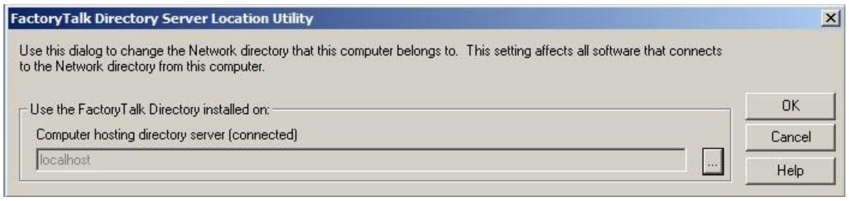
To specify the directory location on application computers:

1. Select **Start > All Programs > Rockwell Software > FactoryTalk Tools > Specify FactoryTalk Directory Location**.

2. You are prompted to log on to FactoryTalk. Type your name and password, and then click **OK**.

Tip: To use the FactoryTalk Directory Server Location Utility, you must log on 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, click the Browse button.



FactoryTalk Directory Server Location Utility

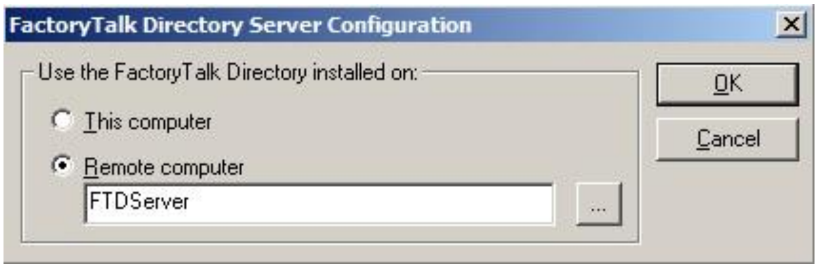
Use this dialog to change the Network directory that this computer belongs to. This setting affects all software that connects to the Network directory from this computer.

Use the FactoryTalk Directory installed on:

Computer hosting directory server (connected)

localhost

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



FactoryTalk Directory Server Configuration

Use the FactoryTalk Directory installed on:

○ This computer

● Remote computer

FTDServer

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 log on 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 new security accounts, or change system security policies.

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

Monitoring the Network Directory status

In the FactoryTalk Directory Server Location Utility, you can view the current status of the active Network Directory server:

- **(Connected)** means all FactoryTalk products and components participating in a FactoryTalk system located 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 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.

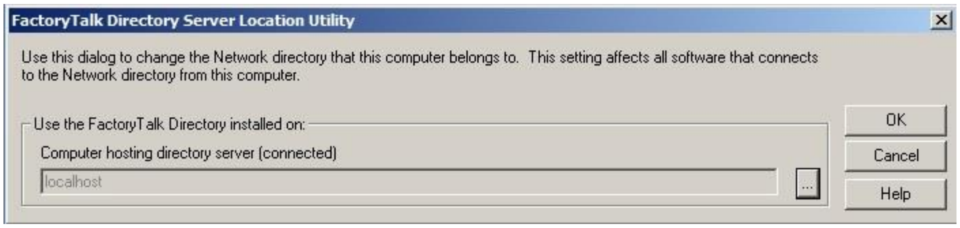
To check the status of the Network Directory server:

1. Select **Start > All Programs > Rockwell Software > FactoryTalk Tools > Specify FactoryTalk Directory Location**.

2. You are prompted to log on to FactoryTalk. Type your name and password, and then click **OK**.

   Tip: To use the FactoryTalk Directory Server Location Utility, you must log on 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, look for the status of the Network Directory server beside the label, Computer hosting directory server:



FactoryTalk Directory Server Location Utility

Use this dialog to change the Network directory that this computer belongs to. This setting affects all software that connects to the Network directory from this computer.

Use the FactoryTalk Directory installed on:

Computer hosting directory server (connected)

localhost

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 run-time security for HMI project components.
- Other ways to control run-time 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 Site Edition, 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 logs on 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 112.

About the FactoryTalk Directory

The FactoryTalk Directory software works like a telephone directory, or electronic address book, allowing parts of an application to find each other on a computer, or across a network. There are two types of FactoryTalk Directory:

- **FactoryTalk Local Directory** (also called the Local Directory) manages FactoryTalk View SE local station applications. All local station application components, except for data servers, must be located on the same computer.
- **FactoryTalk Network Directory** (also called the Network Directory) manages FactoryTalk View SE network distributed and network station applications. Network applications can involve multiple clients and servers on several computers, connected over a network. Network station applications can contain one client and HMI server, and must be located on a single computer.

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 77.

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 the FactoryTalk Security Help.

To open the FactoryTalk Security System Configuration Guide:

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

To open the FactoryTalk Security Help:

1. Select **Start > All Programs > Rockwell Software > FactoryTalk Tools > FactoryTalk Help**.

2. In the **Contents** tab, open the book, Setting up FactoryTalk Security.

You can also gain access to the FactoryTalk Security Help, by clicking **Help** in dialog boxes used to set up security for FactoryTalk resources.

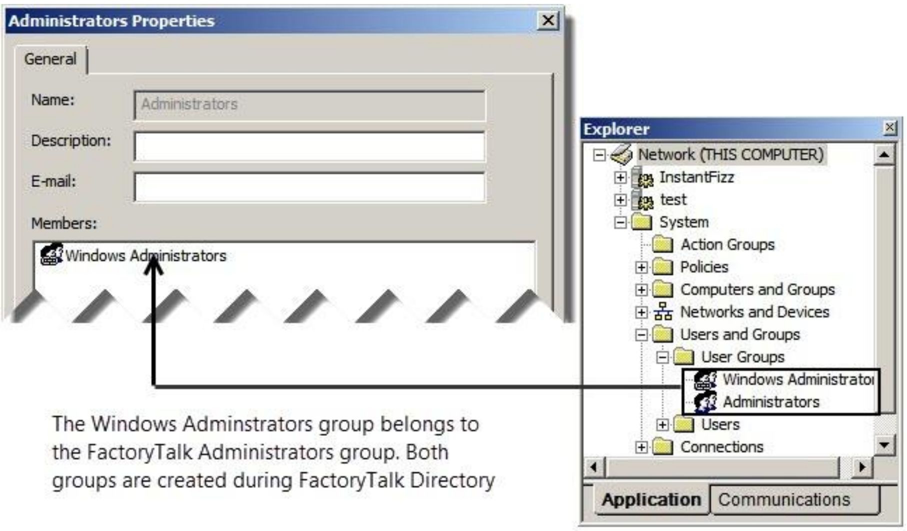
Gaining initial access to a FactoryTalk system

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 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 Adminstrators 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 run-time 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 FactoryTalk system resources for selected users and computers, you need to 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.

For more information, see Deciding how to secure a FactoryTalk View SE application on page 93.

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 99 and About the All Users account on page 88 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 log on, 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 log on and off to change users, or to gain access to *secured* parts of the FactoryTalk system.

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

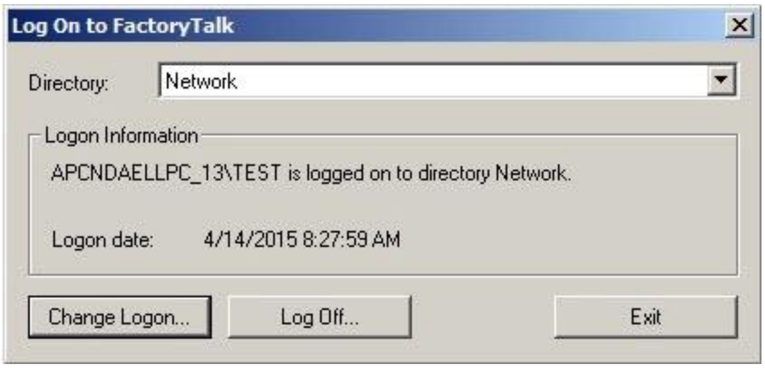
Logging on to the FactoryTalk Directory

To log on 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 > All Programs > Rockwell Software > FactoryTalk Tools > Log On to FactoryTalk**.



Log On to FactoryTalk

Directory: Network

Logon Information

APCNDAELLPC_13\TEST is logged on to directory Network.

Logon date: 4/14/2015 8:27:59 AM

Tip: In the illustration, the current Network Directory user is named **administrator**. If there was no current user, **Not logged on to directory Network** would be shown, the **Logon date** would be blank, and there would only be a **Log On** button in the Log On to FactoryTalk tool.

For more information about logging on to the FactoryTalk Directory, see the FactoryTalk Security 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 log on 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 log on 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 log on as a user with administrative privileges. For details, see Specifying the location of the FactoryTalk Directory server on page 81.

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

Logging on to FactoryTalk View Studio

To log off 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 log on 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 89.

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 run-time 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.

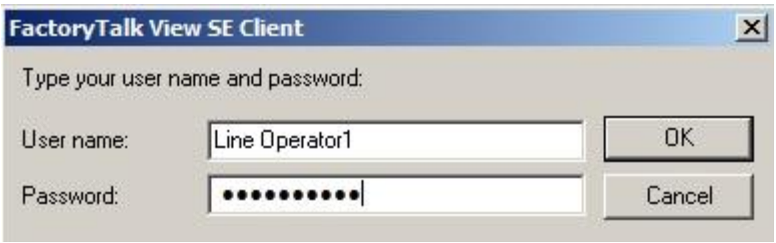
In a secured FactoryTalk system, you must remove the All Users account, add users to the Runtime Security list, and then give the users the security permissions needed to run an application. For more information, see Removing All Users from the Runtime Security list on page 99, Setting up accounts in the Runtime Security editor on page 96, and Performing secured tasks in FactoryTalk View SE on page 120, 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 log on the current FactoryTalk Directory user.

If there is no user logged on to the FactoryTalk Directory, the client will attempt to log on 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 log on. To open the FactoryTalk View SE Client Login dialog box (shown in the following illustration), click **Retry**.



FactoryTalk View SE Client

Type your user name and password:

User name: Line Operator1

Password: ●●●●●●●●●●●

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

Changing the current user while the client is running

To allow users to log on to a FactoryTalk View SE Client at run time, 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 SE Client 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 89.

To log on a different user

- In the FactoryTalk View SE Client Login dialog box, type the user name and password of the user that wants to log on, 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 log in using the syntax **domain\user name.**

Logging off the current user at run time

To allow the current FactoryTalk View SE Client user to log off from the client at run time, 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 currently running displays close.
2. The current user's logout macro runs, and the user is logged off.

   For information about user macros, see Specifying login and logout macros on page 100.

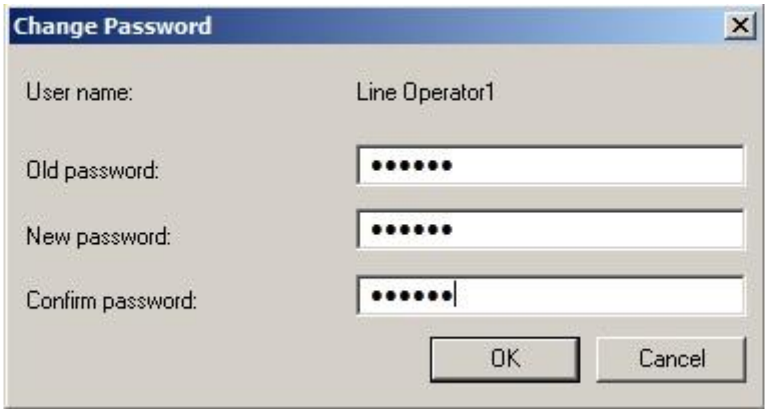
3. If the FactoryTalk View SE Client is using an activation key, the key is released and made available for other clients to use.
4. The FactoryTalk View SE Client Login dialog box opens, to let another user log on.

   Tip: You can set up the FactoryTalk View SE Client to log off 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 run time

To allow the current FactoryTalk View SE Client client user to change their password at run time, 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.



User name: Line Operator1

Old password: ••••••

New password: ••••••

Confirm password: ••••••

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 the FactoryTalk Security 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 run time; 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 109.

- 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 log on 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 run time.

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

Securing FactoryTalk View SE applications at run time

FactoryTalk View SE manages run-time 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 run-time security for a FactoryTalk View SE application:

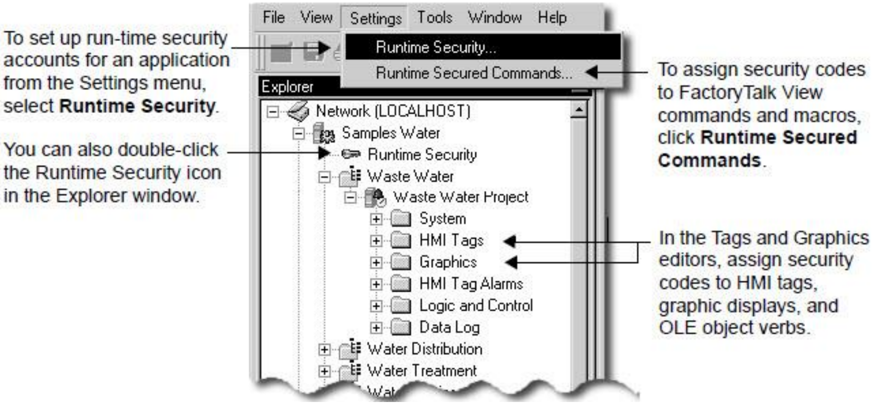
- In the Runtime Security editor:

- Add the FactoryTalk user and group accounts you want to secure.
- Assign FactoryTalk View security codes (A-P) to the accounts 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 120.

- 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 following illustration.



To set up run-time security accounts for an application from the Settings menu, select **Runtime Security**.

You can also double-click the Runtime Security icon in the Explorer window.

To assign security codes to FactoryTalk View commands and macros, click **Runtime Secured Commands**.

In the Tags and Graphics editors, assign security codes to HMI tags, graphic displays, and OLE object verbs.

For details, see the following:

- Setting up FactoryTalk accounts in FactoryTalk View SE on page 96.
- Setting up run-time security for HMI project components on page 101.
- Other ways to control run-time access to an application on page 105.

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 106
- Setting up user and computer accounts on page 110
- Setting up system-wide policies on page 112
- Setting up security for FactoryTalk system resources on page 114
- Understanding inherited permissions on page 118
- Performing secured tasks in FactoryTalk View SE on page 120

For comprehensive information and setup instructions, see the FactoryTalk Security Help.

Setting up FactoryTalk accounts in FactoryTalk View SE

To secure a FactoryTalk user's access to FactoryTalk View SE application components at run time, you must set up run-time 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 100.

After setting up Runtime Security accounts, you can set up the HMI project components you want to secure. For more information, see Setting up run-time security for HMI project components on page 101.

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.

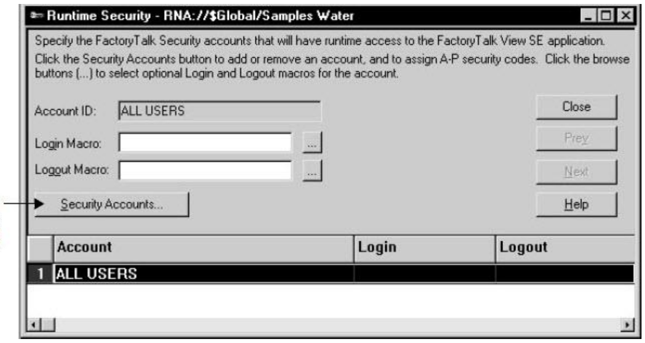
For information about creating FactoryTalk accounts, see Setting up user and computer account on page 110.

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.



Runtime Security - RNA://$Global/Samples Water

Specify the FactoryTalk Security accounts that will have runtime access to the FactoryTalk View SE application.
Click the Security Accounts button to add or remove an account, and to assign A-P security codes. Click the browse buttons (...) to select optional Login and Logout macros for the account.

Account ID: ALL USERS

Login Macro:

Logout Macro:

	Account	Login	Logout
1	ALL USERS		

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

To add users and assign security codes to them:

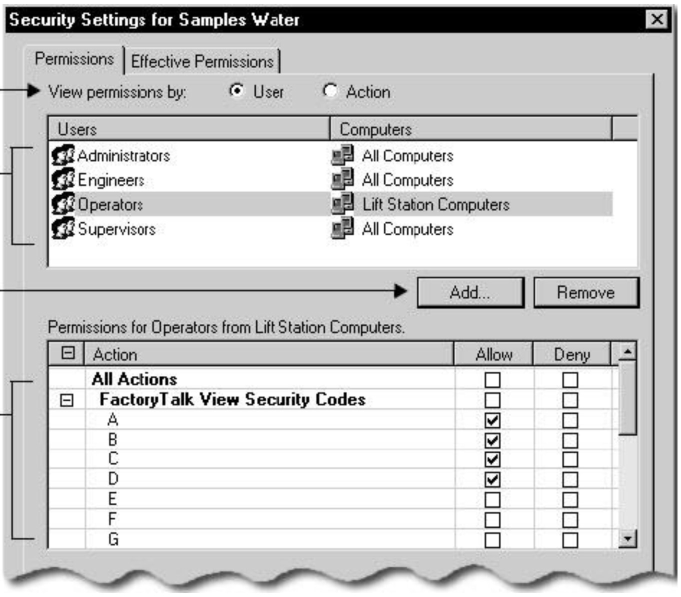
1. In the Runtime Security editor, click **Security Accounts**.

Set up security codes by user or action.

List of user groups with levels of access to securable HMI project components.

To add FactoryTalk users or groups, click Add.

To assign security codes to the selected user or group, select the Allow check boxes.



Security Settings for Samples Water

View permissions by: ● User ○ Action

Users	Computers
Administrators	All Computers
Engineers	All Computers
Operators	Lift Station Computers
Supervisors	All Computers

Permissions for Operators from Lift Station Computers.

Action	Allow	Deny
All Actions	□	□
⊟ FactoryTalk View Security Codes	□	□
A	☑	□
B	☑	□
C	☑	□
D	☑	□
E	□	□
F	□	□
G	□	□

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 check box 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 check box beside All Actions, or the check box beside the FactoryTalk View Security Codes heading.

6. Select the Deny check box 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 check box. 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 118.

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

To add users and assign security codes to them

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

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 groups 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 run-time 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 run-time 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 run-time 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 88.

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**.

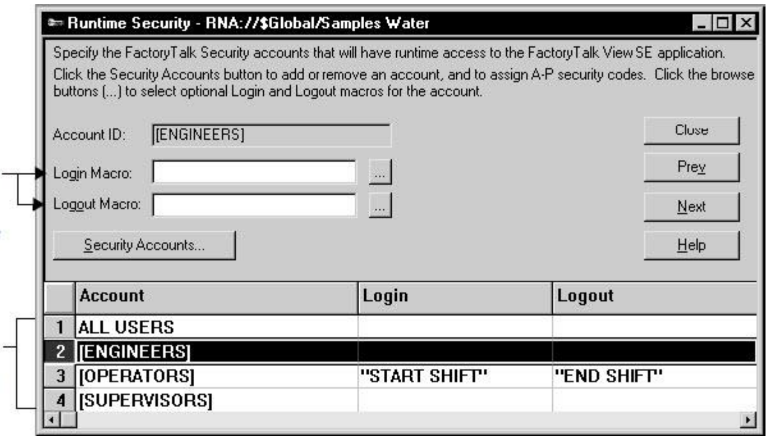
Specifying login and logout macros

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) log on to or off from the application at run time.

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 distribution application, a login or logout macro will only run in the area where it's located. For example, if you specify a login macro located in the Bottling area, when the user logs on to the 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.



Specify the FactoryTalk Security accounts that will have runtime access to the FactoryTalk View SE application. Click the Security Accounts button to add or remove an account, and to assign A-P security codes. Click the browse buttons (...) to select optional Login and Logout macros for the account.

Account ID: [ENGINEERS]

Login Macro:

Logout Macro:

	Account	Login	Logout
1	ALL USERS		
2	[ENGINEERS]		
3	[OPERATORS]	"START SHIFT"	"END SHIFT"
4	[SUPERVISORS]		

Type a macro name, or click the Browse button to find and select a macro for the selected account.

In the accounts list, square brackets [ ] indicate a user group.

For more information about logging on and off at run time, see Logging users on to and off from FactoryTalk View SE on page 88.

For information about creating macros, see Adding logic and control on page 633.

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 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 run-time security for HMI project components

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

You set up security for these HMI project components in FactoryTalk View Studio, in their respective editors.

Assigning security codes to commands and macros

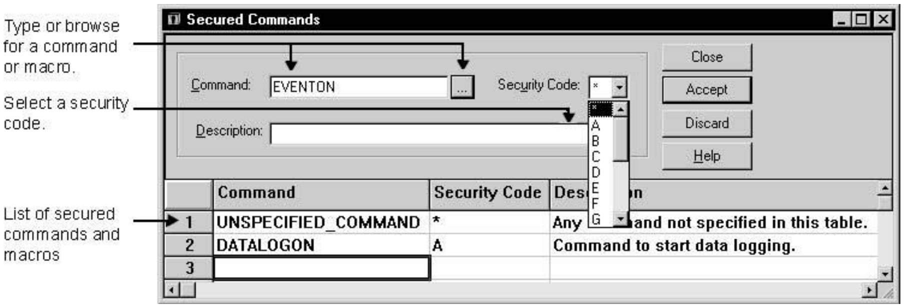
In the **Runtime Secured Commands** editor, you can create a list of the commands and macros to be secured, and then assign a security code to each.

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 run time.

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

To assign security to commands and macros:

1. In FactoryTalk View Studio, from the **Settings** menu, select **Runtime Secured Commands.**



Type or browse for a command or macro.

Select a security code.

List of secured commands and macros

Command: EVENTON

Security Code: *

Description:

	Command	Security Code	Description
1	UNSPECIFIED_COMMAND	*	Any and not specified in this table.
2	DATALOGON	A	Command to start data logging.
3			

2. In the **Command** text box, enter the command or macro you want to secure. To find and select a command or macro, click **Browse**.

3. Provide an optional description, select a security code, and then click **Accept**.

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.

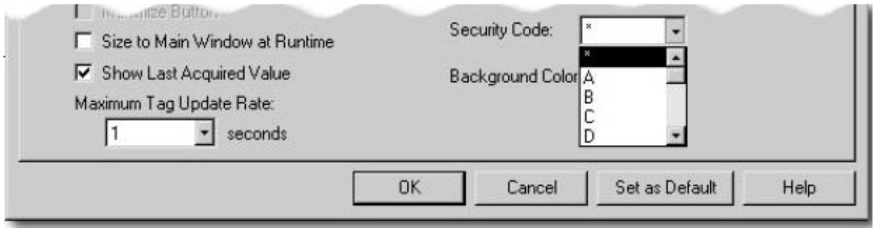
Assigning security codes to graphic displays

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 run time. 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 list, select a security code other than the asterisk (*), and then click **OK**.



☐ Size to Main Window at Runtime
☑ Show Last Acquired Value
Maximum Tag Update Rate: 1 seconds
Security Code: *

Assigning security codes to OLE objects

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

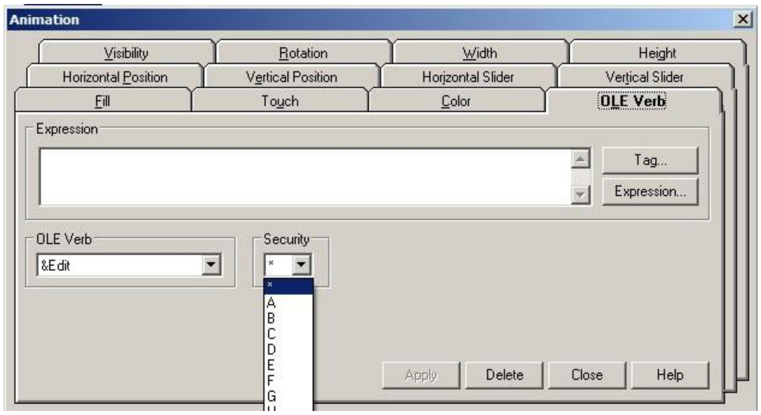
Only users or groups assigned the OLE verb’s security code will be able to activate or modify the OLE object at run time. Users must have at least one security code assigned to activate OLE objects assigned an asterisk (*).

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**.



OLE Verb: &Edit

Security: *

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.

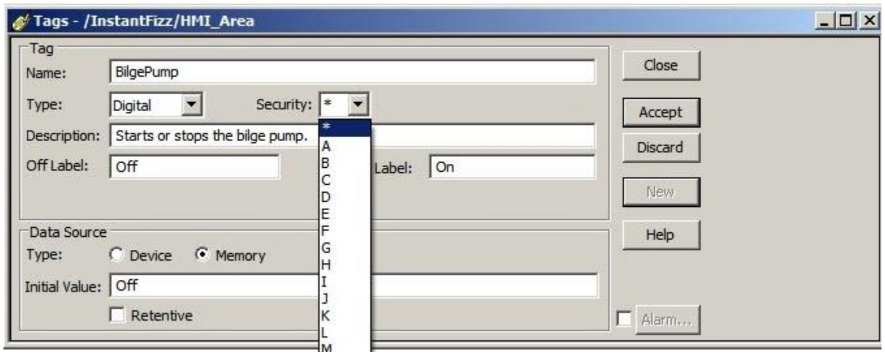
Only users or groups assigned the tag's security code, will be able to change the tag's value at run time. Users must have at least one security code assigned, to change the value of tags assigned an asterisk (*).

Tip: You cannot assign a security codes 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.



Tags - /InstantFizz/HMI_Area

Tag

Name: BilgePump

Type: Digital

Security: *

Description: Starts or stops the bilge pump.

Off Label: Off

Label: On

Data Source

Type: ○ Device ● Memory

Initial Value: Off

□ Retentive

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

4. Click **Accept**.

Other ways to control run-time access to an application

To further restrict access to a FactoryTalk View SE application at run time, 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 run-time environment

To lock operators into the run-time 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 check boxes of Title Bar, Minimize Button, and Maximize Button. For details, see Creating graphic displays on page 393.

- **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 check boxes 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 check box, 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 > All Programs > Rockwell Software > FactoryTalk View > Tools > DeskLock**. For details about using DeskLock, click **Help** within the tool.

Use the signature button to control user actions

To further secure commands, graphic objects, and tags, provide a signature button in a FactoryTalk View SE graphic display.

With the signature button, you can control operator actions at run time, 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 449.

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 and a network distributed application on the same computer.

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.

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 need to 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 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 run-time 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 87.

For information about setting up security permissions, see Setting up system-wide policies on page 112. For details about adding users to a group, see the FactoryTalk Security 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 Security 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 logs on to the Windows network after restarting, so it can run control programs automatically.

For more information, see FactoryTalk Security 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 located 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 log on and off 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 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.

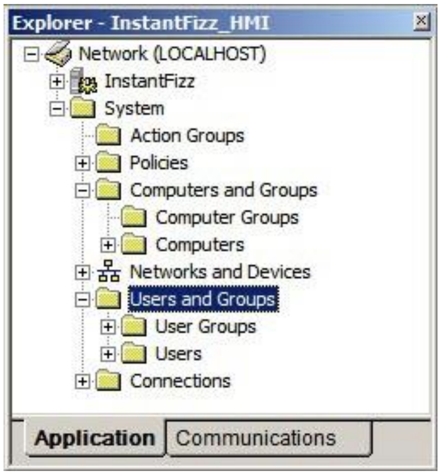
Later, if you want to change an individual user's permissions, you can move the user's account to a different group. You can also change permissions for several individual users at once, by modifying the group that the users belong to.

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 the FactoryTalk Security Help.

Setting up user and computer accounts

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

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



To create a new 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 new 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 new Windows user or group, you must do so in Windows. For details, see Windows help.

Creating computer accounts for network applications

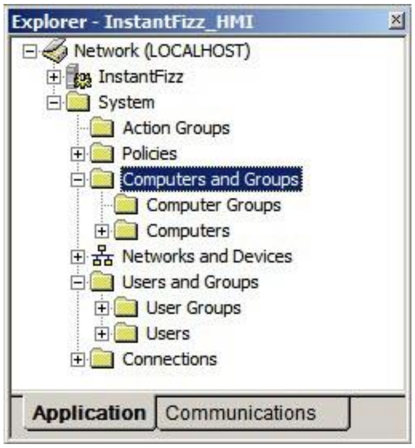
For FactoryTalk View SE network applications, in addition to specifying which users have access to system resources, you can specify the computers where they have access.

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 new 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 new computer group:

- In the Computers and Groups folder, right-click the Computer Groups folder, and then select **New Computer Group**.

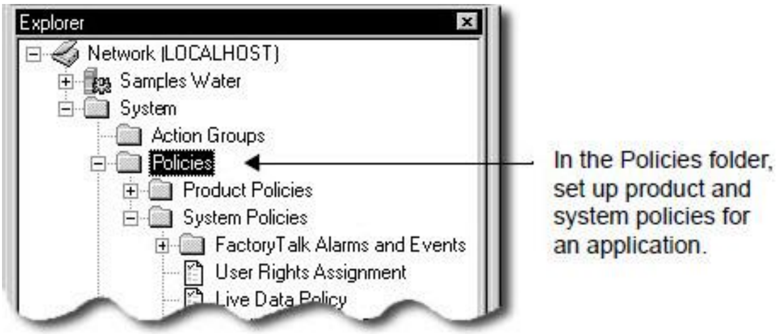
  For details about options in the New Computer Group dialog box, click **Help**.

Setting up system-wide policies

In FactoryTalk View Studio you can set up product and system policies that determine 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 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.



In the Policies folder, set up product and system policies for an application.

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 configuring and using secure web sites under Internet Information Services. For details, click **Help** in the FactoryTalk View SE Secure Web Site Setup tool.

For more information about product policies, see the FactoryTalk Security Help.

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.
- **User rights assignment** settings determine which users can backup 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 changes to take effect, all computers on the network must be shut down and restarted.

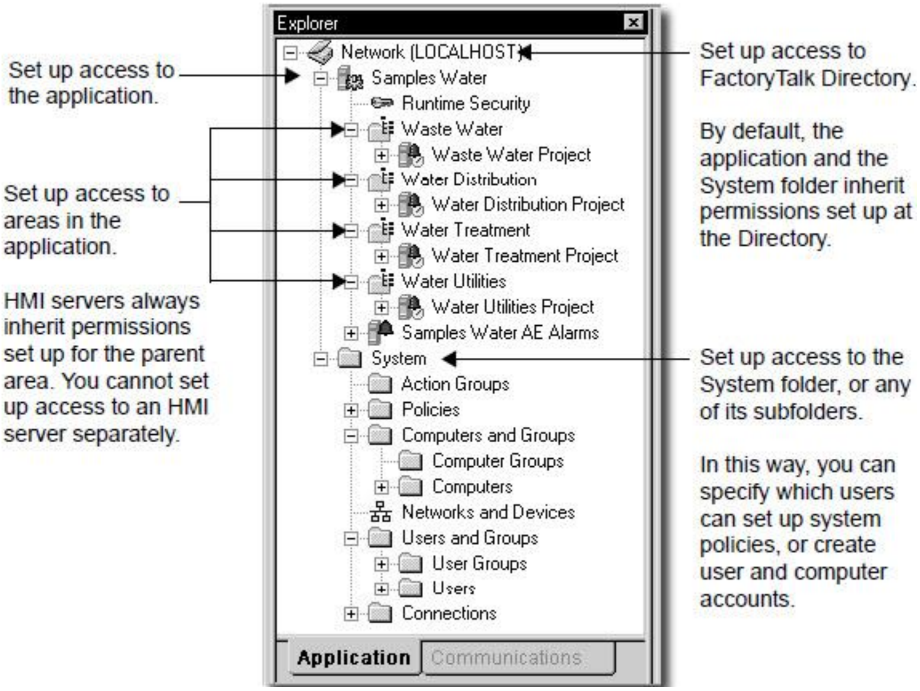
- **Audit policy** settings determine what security information is recorded while the system is in use. This includes whether FactoryTalk Diagnostics logs an audit message when a user attempts an action and is allowed or denied access.
- **Security policy** settings determine general characteristics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

For details about setting up system policies, see the FactoryTalk Security 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:



Set up access to the application.

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.

Set up access to FactoryTalk Directory.

By default, the application and the System folder inherit permissions set up at the Directory.

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

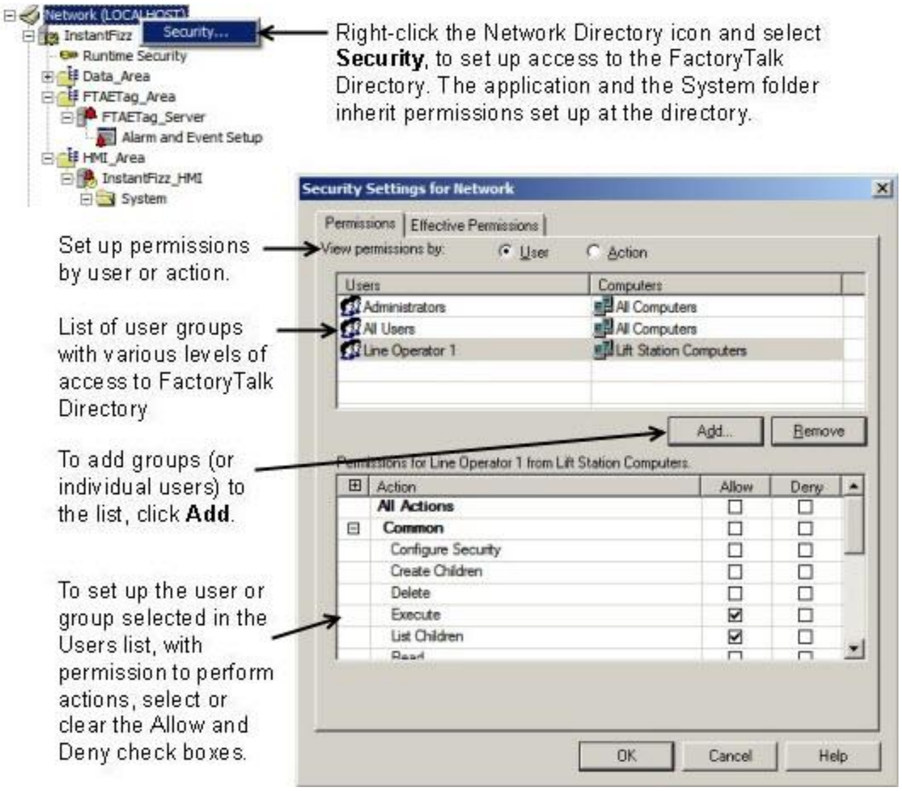
Use the Security Settings dialog box to add the users or groups of users that are to have access to the resource, and then allow or deny permission to perform the actions that define levels of access.

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**.



Users	Computers
Administrators	All Computers
All Users	All Computers
Line Operator 1	Lift Station Computers

Action	Allow	Deny
All Actions	□	□
Common	□	□
Configure Security	□	□
Create Children	□	□
Delete	□	□
Execute	☑	□
List Children	☑	□
Read	□	□

Right-click the Network Directory icon and select **Security**, to set up access to the FactoryTalk Directory. The application and the System folder inherit permissions set up at the directory.

Set up permissions by user or action.

List of user groups with various levels of access to FactoryTalk Directory

To add groups (or individual users) to the list, click **Add**.

To set up the user or group selected in the Users list, with permission to perform actions, select or clear the Allow and Deny check boxes.

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 user 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 check box beside the action.

   You can also select the Allow check box 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 check box beside the action.

5. Repeat steps 2 to 4 for each user or group of users 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 the FactoryTalk Security 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 120.

About explicit and implicit permission

Selecting the Deny check box 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 check box for an action, to implicitly deny permission to perform the action.

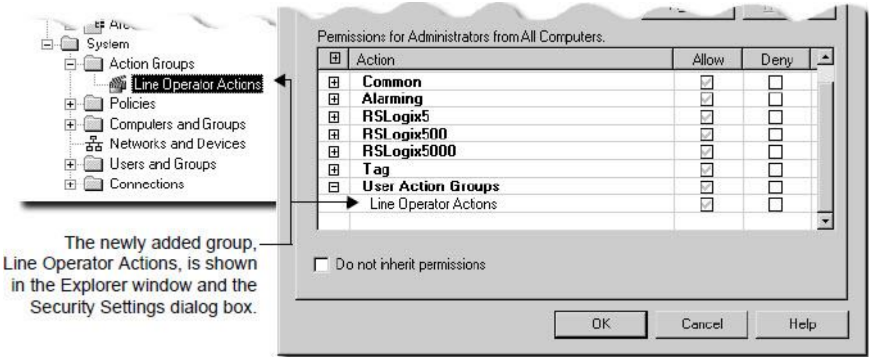
In this case, if a user belongs to one group that is allowed to delete application, and belongs to another group that is implicitly denied that permission, then the user will be allowed to delete applications.

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 user 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.



Permissions for Administrators from All Computers.

⊞	Action	Allow	Deny
⊞	**Common**	☑	□
⊞	**Alarming**	☑	□
⊞	**RSLogix5**	☑	□
⊞	**RSLogix500**	☑	□
⊞	**RSLogix5000**	☑	□
⊞	**Tag**	☑	□
⊟	**User Action Groups**	☑	□
	Line Operator Actions	☑	□

□ Do not inherit permissions

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 116.

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 run time.

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 run-time security for HMI project components on page 101.

Tip: To secure individual HMI tags at run time, 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.

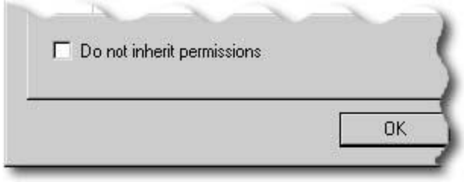
For more information about how inheritance works, see FactoryTalk Security Help.

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 check box, Do not inherit permissions.



☐ 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.

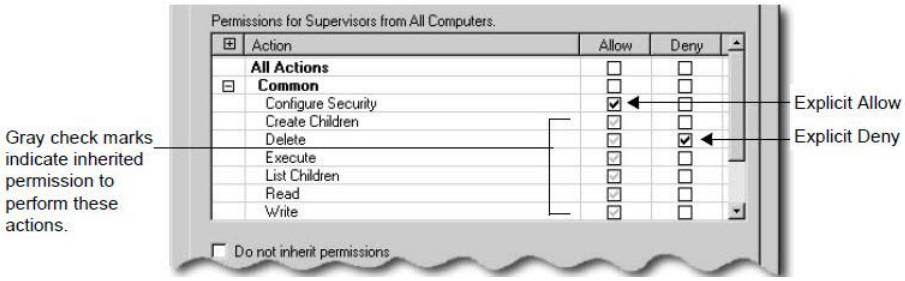
If you choose to copy inherited permissions, the change is applied to all listed users that have any inherited permissions. If you choose to remove all inherited permissions, all listed users that have only inherited permissions are removed from the Security Settings dialog box.

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 on page 1-38, 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.



Permissions for Supervisors from All Computers.

⊞	Action	Allow	Deny
	All Actions	□	□
⊟	**Common**	□	□
	Configure Security	☑	□
	Create Children	☑	□
	Delete	☑	☑
	Execute	☑	□
	List Children	☑	□
	Read	☑	□
	Write	☑	□

□ Do not inherit permissions

Gray check marks indicate inherited permission to perform these actions.

Explicit Allow

Explicit Deny

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 check box beside the Configure Security action.

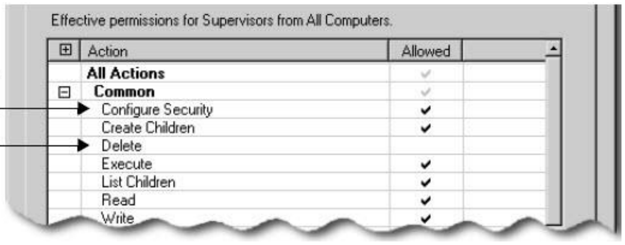
  Explicitly allowing the Configure Security action on the application means that Supervisors can set up security for the application, add

Runtime Security accounts and assign security codes to those users, and secure HMI project components.

- Select the Deny check box 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	✓
Delete	
Execute	✓
List Children	✓
Read	✓
Write	✓

Check mark means Configure Security is allowed.

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 116.
- How inherited permissions work, see Understanding inherited permissions on page 118.

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 368.
- Enabling or disabling, suppressing or unsuppressing, or acknowledging FactoryTalk alarms, see Setting up FactoryTalk alarms on page 261.

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

To do this	In this component or tool	You need these security permissions
Open an existing network distributed, network station, or local station application and view its contents.	FactoryTalk View Studio FactoryTalk View SE Administration Console	Common Read Common List Children
Run an existing network distributed, network station, or local station application and write to tags.	FactoryTalk View SE Client	Common Read Common List Children Tag Write Value At least one security code (A to P) set up in the Runtime Security editor. For details, see Setting up FactoryTalk accounts in FactoryTalk View SE on page 96.
The Common actions Read and List Children allow basic access to a network or local system. To increase access, add security permission, as shown in the remainder of this table.		
Open an existing application and modify its properties.	FactoryTalk View Studio FactoryTalk View SE Administration Console	Common Read Common List Children Common Write
Create an application and modify its properties. Add new areas, HMI servers, data servers, or Tag Alarm and Event Servers to an application.	FactoryTalk View Studio	Common Read Common List Children Common Create Children Common Write
Modify the properties of existing HMI servers, data servers, or Tag Alarm and Event Servers.	FactoryTalk View Studio	Common Read Common List Children Common Write
Create or modify HMI project components, for example, graphic displays or derived tags files. Add HMI project components into an application.	FactoryTalk View Studio	Common Read Common List Children Common Create Children Common Write Common Delete
Delete areas, HMI servers, data servers, or Tag Alarm and Event Servers from an application.	FactoryTalk View Studio	Common Read Common List Children Common Write Common Delete
Secure access to application resources, for example, the areas in an application.	FactoryTalk View Studio	Common Configure Security
Create and administer FactoryTalk user and computer accounts.	FactoryTalk View Studio	Common Configure Security Common Create Children Common Write Common Delete

Add user accounts to FactoryTalk View (in the Runtime Security editor). Secure FactoryTalk View commands and macros (in the Runtime Secured Commands editor). **IMPORTANT**: to perform these tasks, the necessary permissions must be set up at the application level.	FactoryTalk View Studio	Common Configure Security Common Write
Delete a network distributed, network station, or local station application.	Application Manager	Common Read Common List Children Common Delete
Rename a network distributed, network station, or local station application.	Application Manager	Common Read Common List Children Common Write
Copy a local station application.	Application Manager	Common Read Common List Children Common Write
To back up and restore applications, in addition to having the following permissions, users must be allowed to back up and restore FactoryTalk Directory contents:		
Back up a network distributed application.	FactoryTalk View Studio	Common Read Common List Children Common Write
Restore a network distributed application.	FactoryTalk Administration Console	Common Read Common List Children Common Write
Back up and restore a local station application	Application Manager	Common Read Common List Children Common Write

Example – Using the Common actions to set up security for user groups in a FactoryTalk View SE network distributed application

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.

For information about overriding inherited permissions, see Understanding inherited permissions on page 118.

To set up this level of access	For this group	Set up these permissions at the Network Directory	And then override inherited permissions
Full access. This includes the ability to: • Create applications • Add areas and servers • Create HMI project components • Set up permissions for all resources the FactoryTalk Directory manages • Create new user accounts • Add Runtime Security accounts • Secure HMI project components For a complete list of tasks users with full access can perform, see the previous table in this example.	Administrators	Allow Common actions: • Configure Security • Create Children • Delete • List Children • Read • Write Allow Tag action: • Write Value	No changes. Retain inherited permissions at all lower-level resources.
Same access as Administrators, *except members of this group cannot*: • Set up security for the Users and Groups sub folder of the System folder. • Create users at the FactoryTalk Directory. However, members of this group *can* add Runtime Security accounts, assign security codes to them, and use the Runtime Secured Commands editor.	Engineers	Allow Common actions: • Same as Administrators Allow Tag action: • Write value	Explicitly deny these Common actions on the Users and Groups folder (in the System folder): • Configure Security • Delete • Create Children Retain all other inherited permissions.
Run-time access (see Operators group, next), *plus, members of this group can*: • Modify existing applications. • Modify HMI server properties. • Create HMI project components.	Supervisors	Allow Common actions: • Read • List Children Tag action: • Write Value	Explicitly allow these Common actions on the application: • Delete • Write • Create Children Retain all other inherited permissions.
Run-time access. *Members of this group can only*: • Load existing applications. However, members of this group cannot modify HMI server properties, nor view HMI project components. HMI servers show in the Explorer window as locked. • Run applications in the FactoryTalk View SE Client. • Write to tags at run time. **IMPORTANT**: to restrict access to individual HMI tags, in the Runtime Security editor, assign security codes to this group.	Operators	Allow Common actions: • Read • List Children Allow Tag action: • Write Value	No changes. Retain all inherited permissions at lower-level resources.

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 run-time security for individual HMI tags. For details, see Assigning security codes to HMI tags on page 104.

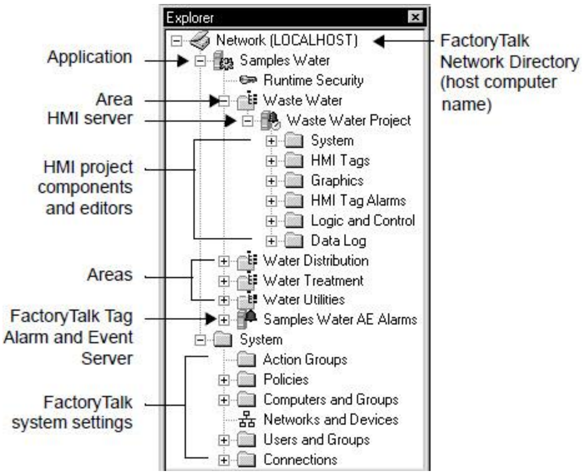
Working with network distributed applications

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:



Network station applications are described in Working with network station applications on page 149. Local station applications are described in Working with local station applications on page 167.

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, providing clients with access to information in programmable controllers, devices, and other data servers that comply with the OPC-DA 2.05a specification, with or without the use of HMI tags.

  In a network distributed application, you can use multiple RSLinx Enterprise and OPC data servers (including RSLinx Classic), running on different computers. You can also set up a redundant pair of host computers for each data server in the application.

  RSLinx Enterprise servers can be set up to subscribe to alarms detected in devices such as ControlLogix controllers. For more information, see Setting up FactoryTalk alarms on page 261.

  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 RSLinx and Logix5000.

- A list of users, plus the security codes that allow or deny these users permission to access secured HMI project components at run time.

- 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 261.

The software programs behind a network distributed application—the FactoryTalk Network Directory, the HMI servers, the HMI clients, and the data servers—can be located 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 77.

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 26.

An automation and control system that uses FactoryTalk services, and integrates FactoryTalk products and components, is known as a FactoryTalk system.

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.

To open the FactoryTalk Help:

Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **FactoryTalk Help**.

You can also open the FactoryTalk **Help** by clicking Help in dialog boxes used to set up FactoryTalk components and services.

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

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

For example, to access graphic displays in a network distributed application, HMI clients use FactoryTalk Directory to find out which computers on the network are hosting the HMI servers that provide the displays.

FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed and network station applications. All of the computers participating in a particular network distributed application must share a common Network Directory, located on a network server.

For more information about FactoryTalk Network Directory, see Setting up the FactoryTalk Directory on page 77.

Note: Do not run FactoryTalk Directory, or any other application software, on the same computer as a Windows domain controller.

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 85. For details, see the FactoryTalk Security 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 permissions you might require.

For an overview of FactoryTalk Security services, see Setting up security on page 85. For details, see FactoryTalk Security Help.

HMI servers

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 HMI tag alarms, 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 sub-area 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 135.
- Setting up redundant HMI servers, see Setting up HMI server redundancy on page 142.
- Specifying which components will run when an HMI server starts, see Selecting startup and shutdown components on page 142.

HMI projects

HMI projects contain graphic displays, data log models, HMI tags, HMI tag alarms, and other services. An HMI project is created when you add a new 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 141.

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

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 new production line to the application, you could create a new 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 sub-areas, and one or more data servers. Each area or sub-area can contain only one HMI server.

An area or sub-area 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

In a network distributed application, 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.

- **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
  ```

  Tip: 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.

  Use absolute references to ensure that a specific component in a specific location is used, regardless of where it is referenced from.

- **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 located in the current area.

  Use relative references, for example, to re-use 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, use 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 a control system remain available to connected clients, FactoryTalk View SE provides features such as server status monitoring and support for online changes.

For information about these and other availability features, see Setting up FactoryTalk system availability on page 351.

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 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 362.

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 run time in up to 40 different languages, set up language switching for the application.

For more information, see Setting up language switching on page 323.

Creating network distributed applications

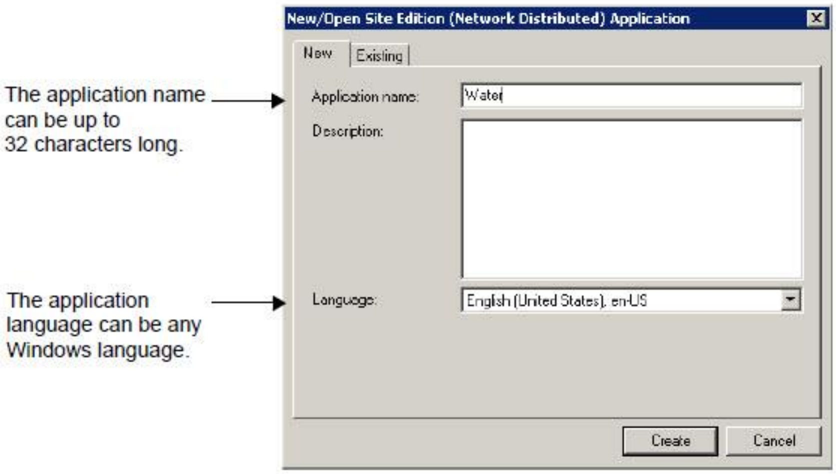
To create a network distributed application, in FactoryTalk View Studio, first you create the application, and then you add elements such as areas, HMI servers, data servers, and FactoryTalk Tag Alarm and Event Servers.

To create a network distributed application:

1. Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **FactoryTalk View Studio**.

2. In the Application Type Selection dialog box, select **View Site Edition (Network Distributed)**, and then click **Continue**.

3. In the New/Open Site Edition (Network Distributed) Application dialog box, click the **New** tab.



New/Open Site Edition (Network Distributed) Application

Application name: Water

Description:

Language: English (United States), en-US

The application name can be up to 32 characters long.

The application language can be any Windows language.

4. Type a name and description for the application, select an application language, and then click **Create**.

   In the Explorer window, the application icon and name are shown beneath the Network Directory icon.

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 you want to delete, and then click **Delete**.

When you delete an area, HMI servers and data servers located in the area are not deleted from disk.

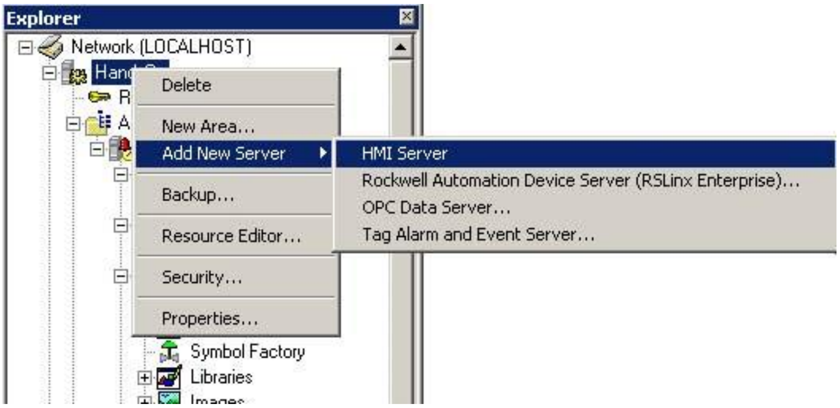
Adding an HMI server

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.
   - **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.

Tip: The Add Controller Instruction Faceplates dialog box opens, if it's set to show when you create a new HMI server. If you do not want to add faceplates, click **Cancel** to close the dialog box, without affecting server creation. For more information about adding faceplates, see Adding controller instruction faceplates to an application on page 422.

After adding the HMI server, you can set up its properties. For details, see Setting up HMI server properties on page 159.

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 RSLinx Enterprise data server in the root area or individual area.

To add an RSLinx Enterprise 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 (RSLinx Enterprise)**.

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 RSLinx Enterprise 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 Data Server**.

2. Set up properties, such as the name and location of the server, and whether to provide redundancy using a secondary server.

   For details, see the RSLinx Classic (or other OPC data server) product documentation.

For additional information about setting up RSLinx and OPC data servers, see Setting up communications on page 177.

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 261.

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 goes out of service, set up redundant HMI servers and data servers.

Tip: If you plan to deploy a network distributed application that uses more than two servers (or two pairs of redundant servers) and 20 clients, or if you would like architectural assistance, contact your local Rockwell Automation Sales office for architectural assistance . For information about setting up redundant FactoryTalk View SE Servers, see Setting up FactoryTalk system availability on page 351.

You must set up redundancy separately for each server in an application. For more information, see Setting up FactoryTalk system availability on page 351.

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 359.

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 re-use 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 disk.
2. Create an HMI analog or digital alarm tag.

   To indicate when disk space drops below preset levels above 500 MB, set up alarm conditions for an HMI analog tag. To indicate only when disk space drops below 500 MB, set up alarm conditions for an HMI digital tag.

For information about setting up events, see Adding logic and control on page 633.

Setting up HMI server properties

After adding an HMI server to a network distributed application, use the Properties dialog box to:

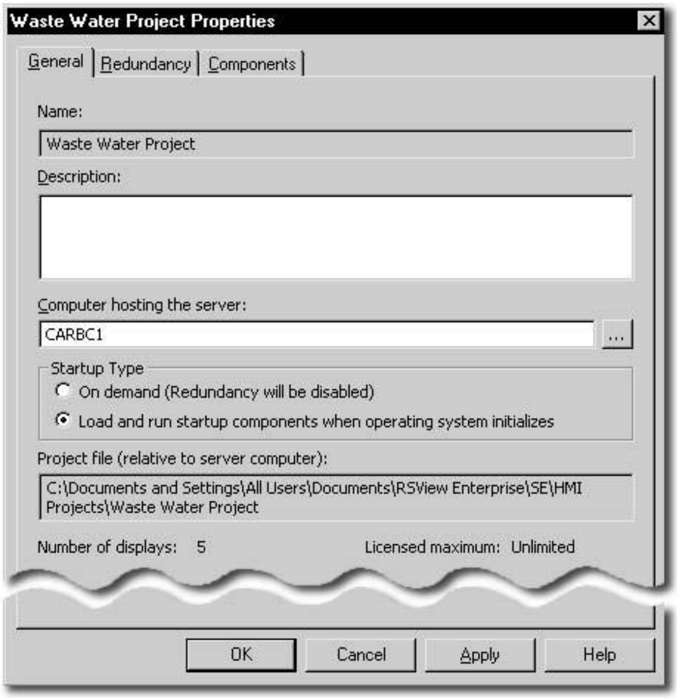
- Select a startup type for the HMI server.
- Set up redundancy for the HMI server.

- 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**.



Name: Waste Water Project

Description:

Computer hosting the server: CARBC1

Startup Type
○ On demand (Redundancy will be disabled)
● Load and run startup components when operating system initializes

Project file (relative to server computer):
C:\Documents and Settings\All Users\Documents\RSView Enterprise\SE\HMI Projects\Waste Water Project

Number of displays: 5 Licensed maximum: Unlimited

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 run-time environment, the HMI server should start when the operating system starts to ensure that no HMI tag alarms or logged data are 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 log on 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 174.

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 the check box **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 160.

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 setting up redundant HMI servers, see Setting up FactoryTalk system availability on page 351.

Replicating changes to the secondary HMI server

In a non-redundant FactoryTalk View SE network distributed application, you can modify tag properties and alarm conditions (after issuing AlarmAcceptEdits) at run time, and have those changes take effect at connected clients, without restarting the clients.

In an application containing redundant HMI servers, you can replicate HMI application component changes made at the primary server to the secondary server. For details, see Replicate changes to the secondary HMI server on page 366.

Selecting startup and shutdown components

In the Components tab of the HMI Server Properties dialog box, select the check box 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 633.

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

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. In FactoryTalk View Studio, open the network distributed application.
2. In the Explorer window, right-click the HMI server, and then click **Properties**.
3. 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. In FactoryTalk View Studio, open the network distributed 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 also will 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

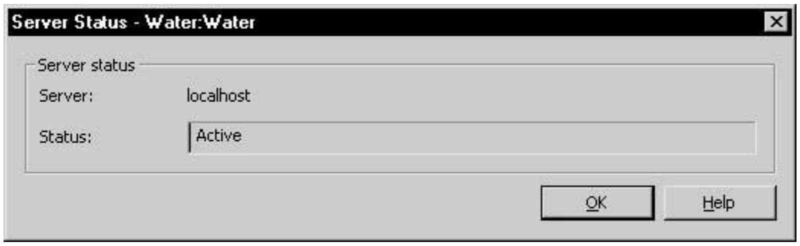
In FactoryTalk View Studio, in the Server Status dialog box, you can view the operational status of an HMI server to determine whether the server is ready to provide service, and to troubleshoot problems.

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 353.

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**.


Server Status - Water:Water

Server status

Server: localhost

Status: Active

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:
   - \Users\Public\Public Documents\RSView Enterprise\SE\HMI projects
   - (Windows XP) \Documents and Settings\All Users\Shared 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 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 run the Application Manager:

Select **Start > All Programs > Rockwell Software > FactoryTalk View > Tools > Application Manager**.

For details about options in the Application Manager, click **Help**.

To rename a network distributed application:

1. In the Application Manager, click **Site Edition (Network Distributed)**, and then click **Next**.
2. Click **Rename application**, and then click **Next**.
3. Follow the instructions in the Application Manager to complete the rename operation.

To delete a network distributed application:

1. In the Application Manager, click **Site Edition (Network Distributed)**, and then click **Next**.
2. Click **Delete application**, and then click **Next**.

3. Follow the instructions in the Application Manager to complete the delete operation.

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 145.
- Removing a data server and deleting its cache files, see Deleting an OPC data server on page 188.

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*.

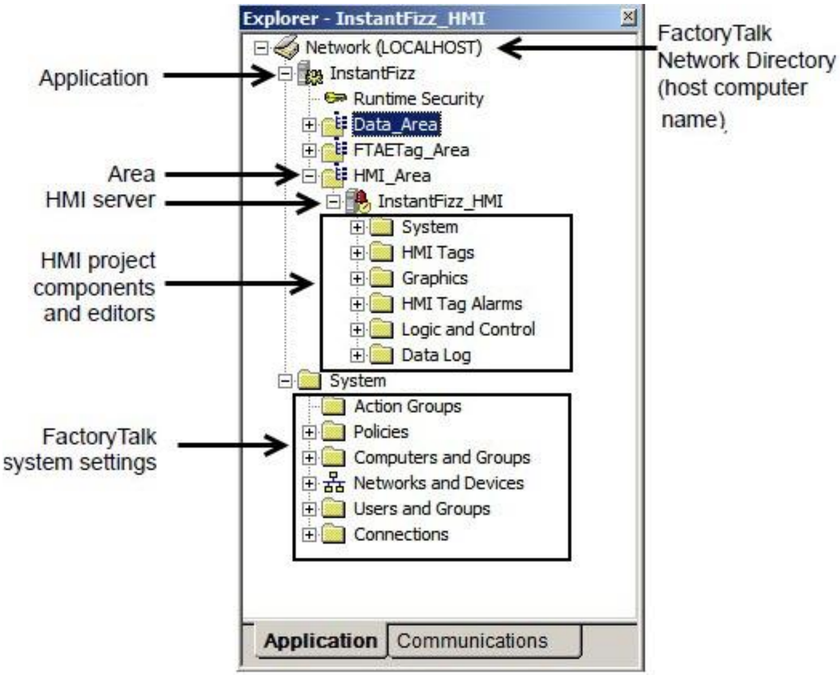
Working with network station applications

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:



Network distributed applications are described in Working with network distributed applications on page 125. Local station applications are described in Working with local station applications on page 167.

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, providing clients with access to information in programmable controllers, devices, and other data servers that comply with the OPC-DA 2.05a specification, with or without the use of HMI tags.

  In a network station application, you can use multiple RSLinx Enterprise and OPC data servers (including RSLinx Classic), running on different computers. You can also set up a redundant pair of host computers for each data server in the application.

  RSLinx Enterprise servers can be set up to subscribe to device-based alarms detected in Logix5000 controllers. For more information, see Setting up FactoryTalk alarms on page 261.

  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 RSLinx and RSLogix.

- A list of users, plus the security codes that allow or deny these users permission to access secured HMI project components at run time.
- 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 261.

The HMI server and the HMI client are the software programs behind a network station application that must be located on the same computer. The FactoryTalk Directory Server, the RSLinx Enterprise data server, and the OPC data servers can be located 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 26.

An automation and control system that uses FactoryTalk services, and integrates FactoryTalk products and components, is known as a FactoryTalk system.

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.

To open the FactoryTalk Help:

Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **FactoryTalk Help**.

You can also open the FactoryTalk **Help** by clicking Help in dialog boxes used to set up FactoryTalk components and services.

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

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

For example, to access graphic displays in a network distributed application, HMI clients use FactoryTalk Directory to find out which computers on the network are hosting the HMI servers that provide the displays.

FactoryTalk Network Directory (also called the Network Directory) manages FactoryTalk View SE network distributed and network station applications. All of the computers participating in a particular network distributed application must share a common Network Directory, located on a network server.

For more information about FactoryTalk Network Directory, see Setting up the FactoryTalk Directory on page 77.

Note: Do not run FactoryTalk Directory, or any other application software, on the same computer as a Windows domain controller.

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 85. For details, see the FactoryTalk Security 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 permissions you might require.

For an overview of FactoryTalk Security services, see Setting up security on page 85. For details, see FactoryTalk Security Help.

HMI servers

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 HMI tag alarms, 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 156.

HMI projects

HMI projects contain graphic displays, data log models, HMI tags, HMI tag alarms, 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

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 RSLinx Enterprise 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

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 run time, 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 a control system remain available to connected clients, FactoryTalk View SE provides features such as server status monitoring and support for online changes.

For information about these and other availability features, see Setting up FactoryTalk system availability on page 351.

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 run time in up to 40 different languages, set up language switching for the application.

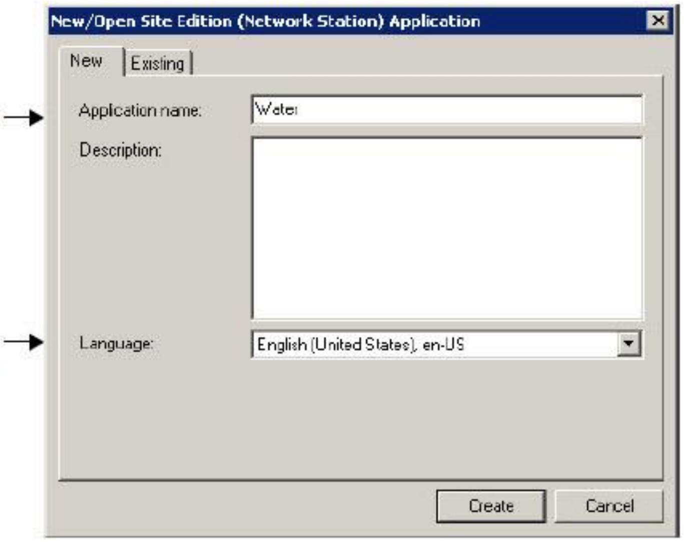
For more information, see Setting up language switching on page 323.

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 > All Programs > Rockwell Software > FactoryTalk View > 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 **New** tab.



New/Open Site Edition (Network Station) Application

Application name: Water

Description:

Language: English (United States), en-US

The application name can be up to 32 characters long.

The application language can be any Windows language.

4. Type a name and description for the application, select an application language, and then click **Create**.

In the Explorer window, the application icon and name are shown beneath the Network Directory icon.

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 you want to delete, and then click **Delete**.

When you delete an area, HMI servers and data servers located in the area are not deleted from disk.

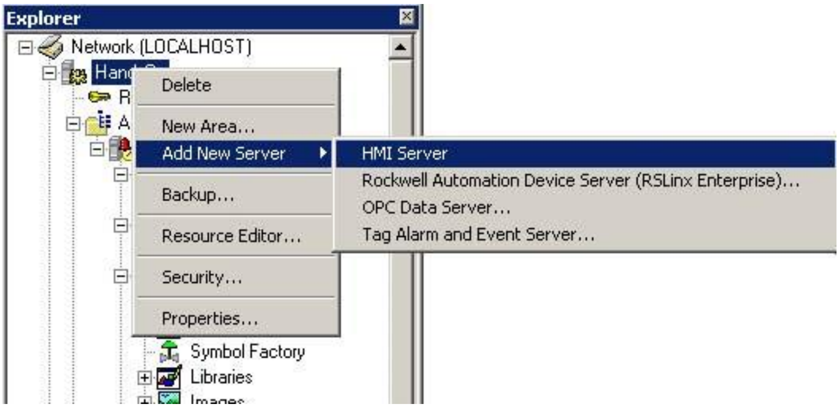
Adding an HMI server

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.

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 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.
   - **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.

   Tip: The Add Controller Instruction Faceplates dialog box opens, if it's set to show when you create a new HMI server. If you do not want to add faceplates, click **Cancel** to close the dialog box, without affecting server creation. For more information about adding faceplates, see Adding controller instruction faceplates to an application on page 422.

After adding the HMI server, you can set up its properties. For details, see Setting up HMI server properties on page 159.

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 RSLinx Enterprise data server in the root area or individual area.

To add an RSLinx Enterprise 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 (RSLinx Enterprise)**.

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 RSLinx Enterprise 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 Data Server**.

2. Set up properties, such as the name and location of the server, and whether to provide redundancy using a secondary server.

   For details, see the RSLinx Classic (or other OPC data server) product documentation.

For additional information about setting up RSLinx and OPC data servers, see Setting up communications on page 177.

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 261.

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 re-use 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 disk.
2. Create an HMI analog or digital alarm tag.

   To indicate when disk space drops below preset levels above 500 MB, set up alarm conditions for an HMI analog tag. To indicate only when disk space drops below 500 MB, set up alarm conditions for an HMI digital tag.

For information about setting up events, see Adding logic and control on page 633.

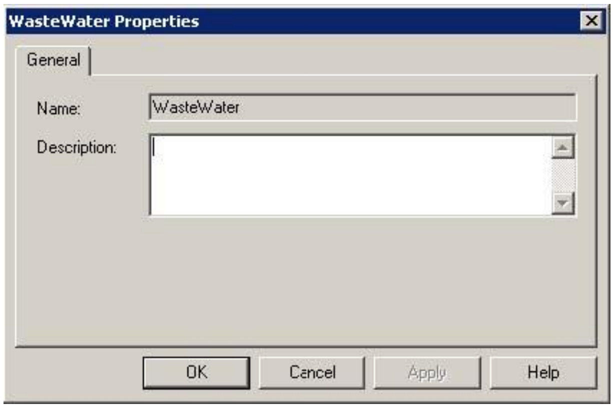
Setting up HMI server properties

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**.



Name: WasteWater

Description:

For details about options in the HMI Server Properties dialog box, click **Help**.

Starting and stopping HMI server components manually

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 also will run.

Monitoring the status of an HMI server

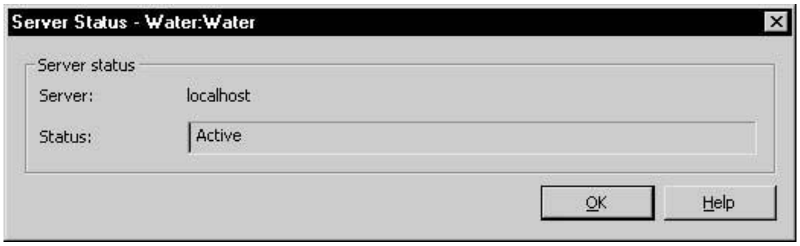
In FactoryTalk View Studio, in the Server Status dialog box, you can view the operational status of an HMI server to determine whether the server is ready to provide service, and to troubleshoot problems.

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 353.

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**.



Server Status - Water:Water

Server status

Server: localhost

Status: Active

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:
   - \Users\Public\Public Documents\RSView Enterprise\SE\HMI projects
   - (Windows XP) \Documents and Settings\All Users\Shared 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 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 run the Application Manager:

Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **Tools** > **Application Manager**.

For details about options in the Application Manager, click **Help**.

To rename a network station application:

1. In the Application Manager, select **Site Edition (Network Station)**, and then click **Next**.
2. Select **Rename application**, and then click **Next**.
3. Follow the instructions in the Application Manager to complete the rename operation.

To delete a network station application:

1. In the Application Manager, select **Site Edition (Network Station)**, and then click **Next**.
2. Select **Delete application**, and then click **Next**.
3. Follow the instructions in the Application Manager to complete the delete operation.

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 145.
- Removing a data server and deleting its cache files, see Deleting an OPC data server on page 188.

Backing up and restoring network station applications

You also use the Application Manager tool to copy, back up, and restore a network station application. For details, see information about deploying local applications, in the *FactoryTalk View Site Edition Installation Guide*.

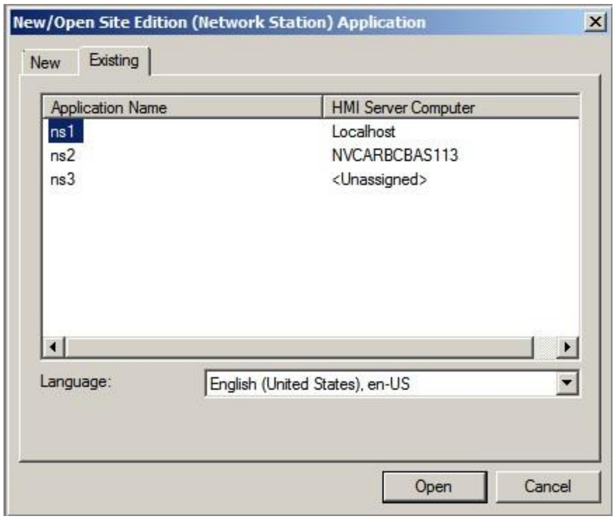
Determing where the application is located

You can browse 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** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **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.



Application Name	HMI Server Computer
ns1	Localhost
ns2	NVCARBCBAS113
ns3	<Unassigned>

Language: English (United States), en-US

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 located on the current computer.
- computer name - the HMI server is located on the computer named (for example, NVCARBCBAS113).
- <Unassigned> - no HMI server has been added to the application.

To rehost an HMI server:

The applications restored from a .bak file need to be rehosted. For more information about restoring a .bak file, refer to the FactoryTalk Help.

1. Use the HMI Server Backup and Restore utility to restore the HMI server. For more information, please refer to the HMI Server Backup and Restore utility help.
2. Select **Start > All Programs > Rockwell Software > FactoryTalk View > FactoryTalk View Studio**.
3. In the Application Type Selection dialog box, select **View Site Edition (Network Station)**, and then click **Continue**.
4. In the New/Open Site Edition (Network Station) Application dialog box, click the **Existing** tab.
5. Select the application containing the HMI server that you want to rehost, and click **Open**.
6. Right-click the HMI server, and select **Properties.**

7. In the **Computer hosting the server** field, type another computer name or click the browse button to search for and select one.

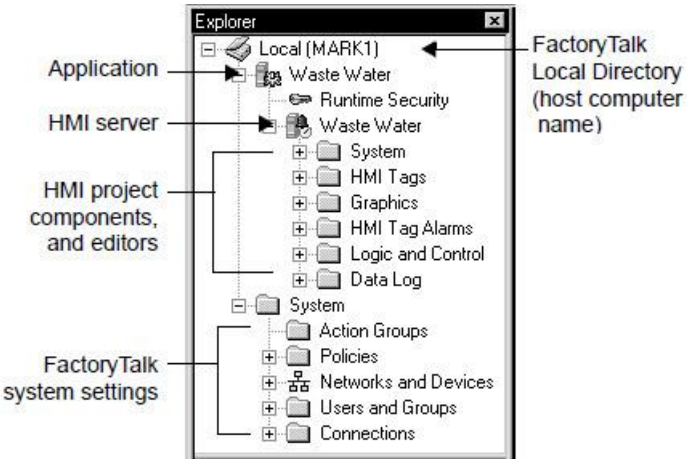
Working with local station applications

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:



Network distributed applications are described in Working with network distributed applications on page 125. 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, providing clients with access to information in programmable controllers, devices, and other data servers that comply with the OPC-DA 2.05a specification, with or without the use of HMI tags.

  Local station applications support a single RSLinx Enterprise data server, which must run on the same computer as the application. Only OPC data servers (including RSLinx Classic) can be located on other computers.

  RSLinx Enterprise servers can be set up to subscribe to device-based alarms detected in Logix5000 controllers. For more information, see Setting up FactoryTalk alarms on page 261.

  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 RSLinx and Logix5000.
- A list of users, plus the security codes that allow or deny these users permission to access secured HMI project components at run time.
- 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 261.

The software programs behind a local station application—the FactoryTalk Local Directory, the HMI server, the HMI client, and the RSLinx Enterprise data server—must be located 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 26.

An automation and control system that uses FactoryTalk services, and integrates FactoryTalk products and components, is known as a FactoryTalk system.

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.

To open the FactoryTalk Help:

Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **FactoryTalk Help**.

You can also open the FactoryTalk **Help** by clicking Help in dialog boxes used to set up FactoryTalk components and services.

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

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

For example, to access tag values in a data server, HMI clients use FactoryTalk Directory to find out which computer is hosting the data server.

FactoryTalk Local Directory (also called the Local Directory) manages 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 77.

Note: Do not run FactoryTalk Directory, or any other application software, on the same computer as a Windows domain controller.

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 85. For details, see the FactoryTalk Security 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 permissions you might require.

For an overview of FactoryTalk Security services, see Setting up security on page 85. For details, see FactoryTalk Security Help.

HMI servers

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 HMI tag alarms, 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

HMI projects contain graphic displays, data log models, HMI tags, HMI tag alarms, 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

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 run time, use

```
Display Detail
```

as the button’s press action.

Language switching

To make user-defined text strings in an application available at run time in up to 40 different languages, set up language switching for the application.

For more information, see Setting up language switching on page 323.

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 an RSLinx Enterprise server, a FactoryTalk Tag Alarm and Event Server, and OPC data servers**.**

To create a local station application:

1. Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **FactoryTalk View Studio**.

2. In the Application Type Selection dialog box, select Site Edition (Local Station), and then click **Continue**.

3. In the New/Open Site Edition (local station) Application dialog box, click the **New** tab.

4. Type a name and description for the application, select an application language, and then click **Create**.

   Tip: The Add Controller Instruction Faceplates dialog box opens, if it's set to display when you create a new application. If you don't want to add faceplates, click Cancel to close the dialog box, without affecting application creation. For more information about adding faceplates, see Adding controller instruction faceplates to an application on page 422.

   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, 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 RSLinx Enterprise 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 an RSLinx Enterprise server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root, select **Add New Server > Rockwell Automation Device Server (RSLinx Enterprise)**.
2. Set up server properties, such as the name of the server, and whether the server will support alarms and events.

   Tip: The location is set to *localhost* automatically, and cannot be changed. Local station applications support a single RSLinx Enterprise data server, which must run on the same computer as the application.

   For details, see the RSLinx Enterprise product documentation.

To add an OPC data server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root, select **Add New Server > OPC Data Server**.
2. Set up properties, such as the name and location of the server, and whether to provide redundancy using a secondary server.

   For details, see the RSLinx Classic (or other OPC data server) product documentation.

For additional information about setting up RSLinx and OPC data servers, see Setting up communications on page 177.

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 261.

Setting up HMI server properties

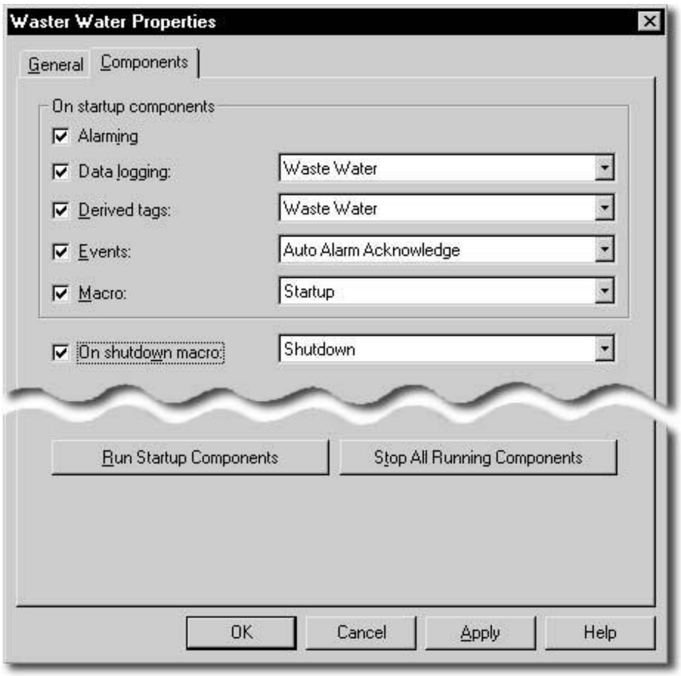
After creating a local station application, use the HMI Server Properties dialog box to:

- 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**.



Waster Water Properties

On startup components

☑ Alarming

☑ Data logging: Waste Water

☑ Derived tags: Waste Water

☑ Events: Auto Alarm Acknowledge

☑ Macro: Startup

☑ On shutdown macro: Shutdown

For details about the options in the HMI Server Properties dialog box, click **Help**.

Selecting startup and shutdown components

In the Components tab of the HMI Server Properties dialog box, select the check box 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

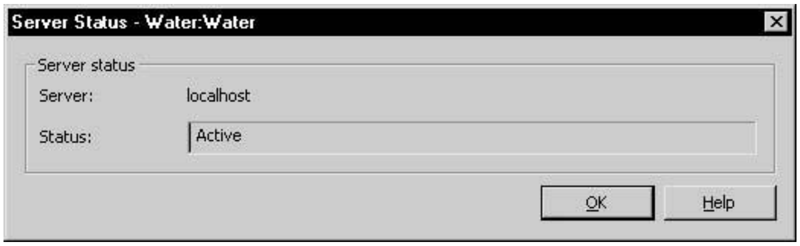
In FactoryTalk View Studio, in the Server Status dialog box, you can view the operational status of an HMI server to determine whether the server is ready to provide service, and to troubleshoot problems.

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 353.

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**.



Server Status - Water:Water

Server status

Server: localhost

Status: Active

For details about options in the Server Status dialog box, click **Help**.

Renaming, deleting, and copying local station applications

Use the Application Manager tool to rename, delete, or copy a local station application.

To run the Application Manager:

Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **Tools** > **Application Manager**.

For details about options in the Application Manager, click **Help**.

To rename a local station application:

1. In the Application Manager, select **Site Edition (Local Station)**, and then click **Next**.

2. Select **Rename application**, and then click **Next**.

3. Follow the instructions in the Application Manager to complete the rename operation.

To delete a local station application:

1. In the Application Manager, select **Site Edition (Local Station)**, and then click **Next**.

2. Select **Delete application**, and then click **Next**.

3. Follow the instructions in the Application Manager to complete the delete operation.

For local station applications, you can choose whether to delete the HMI server project associated with the application. For details, click **Help** within the Application Manager.

For details about removing a data server and deleting its cache files, see Deleting an OPC data server on page 188.

To copy a local station application:

1. In the Application Manager, select **Site Edition (Local Station)**, and then click **Next**.

2. Select **Copy application**, and then click **Next**.

3. Select the application to copy, and then click **Next**.

4. Provide a name for the copy, and then click **Finish**.

About backing up and restoring local station applications

You also use the Application Manager tool to copy, back up, and restore a local application. For details, see information about deploying local applications, in the *FactoryTalk View Site Edition Installation Guide*.

Setting up communications

This chapter describes:

- Data servers and FactoryTalk Live Data.
- Basic steps for setting up communications.
- Adding RSLinx Enterprise data servers to an application.
- Setting up communications in RSLinx Enterprise.
- 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** RSLinx Enterprise is the recommended data server for all Rockwell Automation devices and networks.

  RSLinx Enterprise does not require activation or licensing and can be installed as often as needed in any application.

  RSLinx Enterprise also provides service for device-based alarms and events. For more information, see Setting up FactoryTalk alarms on page 261.
- **OPC Data Servers** are any data server that supports the OPC-DataAccess Standard v2.05a.

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 Access 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 in the following situations:

- CIP to non-CIP routing is required to reach the device.
- Unsolicited messages are used to send data from controllers.
- Alias topics (multiple paths to the same device) are required.

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 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 26.

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 - RSLinx Enterprise, 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 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 RSLinx Enterprise and OPC data servers, located on different computers on the network.

Local applications can only use one RSLinx Enterprise server, which must be located on the same computer as the application.

Tip: OPC data servers can be located on different computers; however, it is not recommended.

Communicating with multiple controllers

RSLinx Enterprise is designed to serve multiple controllers across multiple networks simultaneously. RSLinx Enterprise recovers from program downloads, disconnects, power-offs, and online changes without affecting service to other controllers.

RSLinx Classic can also serve multiple controllers across multiple networks simultaneously; however, is not designed to manage abnormal conditions. When controllers or devices are powered off, RSLinx Classic will wait for a time-out before servicing other devices. RSLinx Classic also does not automatically manage online changes and may require re-starting after program downloads.

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 (RSLinx Enterprise)
   - Other OPC-DA 2.05a-compliant server (RSLinx Classic or KEPServer Enterpriser)

   Tip: RSLinx Enterprise is the recommended server for FactoryTalk View applications. For more information about choosing a data server type, see About data servers on page 177.

2. Install and set up the data servers.

- RSLinx Enterprise is installed automatically with FactoryTalk View SE. Add a Rockwell Automation Device Server (RSLinx Enterprise) to the FactoryTalk View SE application and set up the server's properties. For more information, see Adding RSLinx Enterprise data servers to an application on page 181.
- 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 185.

For information about installing RSLinx Enterprise or RSLinx Classic, see the FactoryTalk View Site Edition Installation Guide.

3. Once the application can communicate with 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 189.

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 189 and Creating graphic objects on page 449.

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 permissions you might require.

For an overview of FactoryTalk Security services, see Setting up security on page 85. For details, see FactoryTalk Security Help.

Adding RSLinx Enterprise data servers to an application

A FactoryTalk View SE network application can contain more than one RSLinx data server, and you can create multiple data servers in a single area.

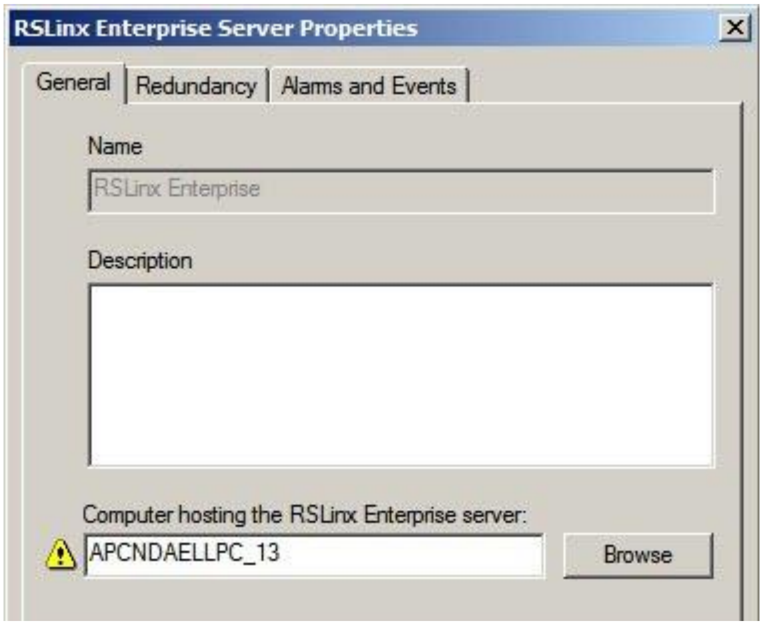
A local application can contain only one RSLinx Enterprise data server, which must be located on the same computer as the application.

To add an RSLinx Enterprise 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 (RSLinx Enterprise)**.
2. In the RSLinx Enterprise 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 general properties

In the General tab of the RSLinx Enterprise server's 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.


RSLinx Enterprise Server Properties

Name: RSLinx Enterprise

Description:

Computer hosting the RSLinx Enterprise server: APCNDAELLPC_13

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 RSLinx Enterprise server

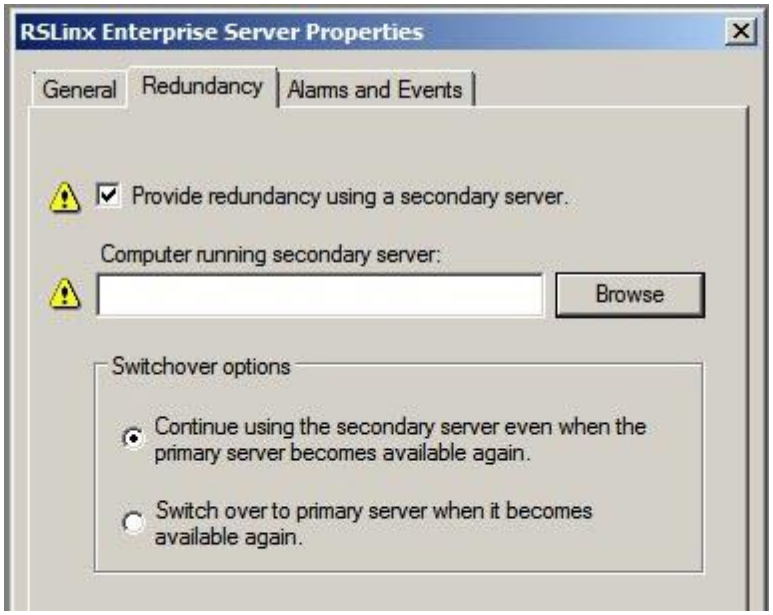
Type the name of the computer where the RSLinx Enterprise data server is running. To find and select a computer, click the **Browse** button.

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 RSLinx Enterprise data server, which must be located on the same computer as the application.

Setting up RSLinx Enterprise data server redundancy

To minimize disruptions to clients in the event that the RSLinx Enterprise 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 RSLinx Enterprise server's 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.



☑ Provide redundancy using a secondary server.

Computer running secondary server:

Switchover options

● Continue using the secondary server even when the primary server becomes available again.

○ Switch over to primary server when it becomes available again.

Tip: In a local application, there is no Redundancy tab in the RSLinx Enterprise Server Properties dialog box. Local applications do not support data server redundancy.

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.

Provide redundancy using a secondary server

Select or clear this check box, 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**.

Setting up support for FactoryTalk Alarms and Events

To enable an RSLinx Enterprise server to use FactoryTalk Alarms and Events services to receive and send alarms detected in Logix5000 controllers, in the Alarms and Events tab in RSLinx Enterprise Server Properties dialog box, select the check box, **Enable alarm and event support**.

For more information, see Setting up FactoryTalk alarms on page 261.

Deleting an RSLinx Enterprise data server

If an RSLinx Enterprise data server is no longer required, you can delete it from the application.

To delete an RSLinx Enterprise data server:

In FactoryTalk View Studio, in the Explorer window, right-click the RSLinx Enterprise data server you want to delete, and then click **Delete**.

Setting up communications for RSLinx Enterprise

Use the RSLinx 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:

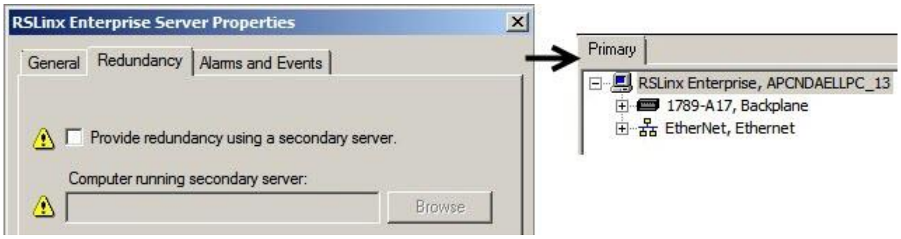
1. In FactoryTalk View Studio, in the Explorer window, click the + symbol beside the RSLinx Enterprise data server icon.

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 an RSLinx Enterprise data server, the Communication Setup editor will contain only a Primary tab.



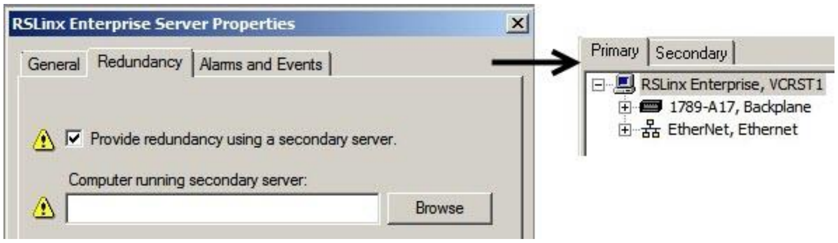
RSLinx Enterprise Server Properties

General | Redundancy | Alarms and Events

□ Provide redundancy using a secondary server.

Computer running secondary server:

In a network application, if you set up both a primary and a secondary RSLinx Enterprise data server, the Communication Setup editor has a Primary and a Secondary tab.



RSLinx Enterprise Server Properties

General | Redundancy | Alarms and Events

☑ Provide redundancy using a secondary server.

Computer running secondary server:

You can use these tabs to point RSLinx Enterprise 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 RSLinx Classic or KEPServer Enterprise 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 RSLinx Enterprise data servers, see Using multiple data servers in an application on page 178.

To add an OPC data server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the application root or an area name, select **Add New Server > OPC Data Server**.
2. In the OPC Data Server Properties dialog box, select options and provide information in each tab.
3. When you are finished, click **OK**.

Setting up general properties

In 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 theOPC server. To browse for the programmatic ID, click the **Browse** button.

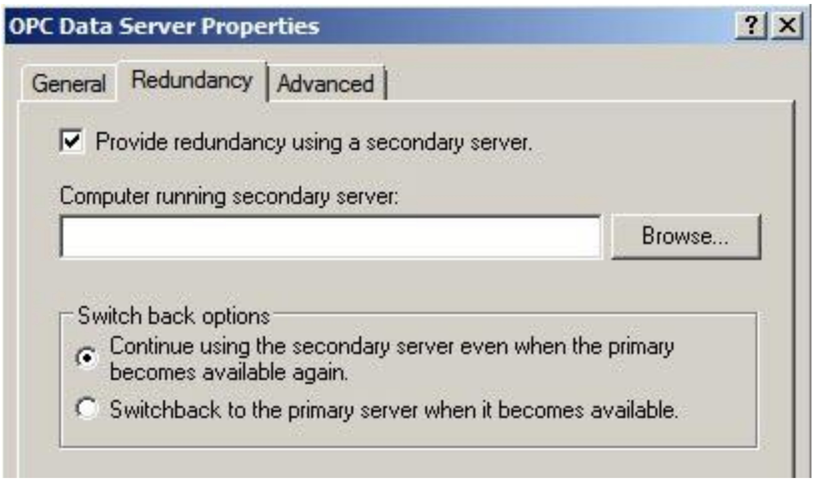
To use RSLinx Classic as the OPC server, click the **Browse** button, and then click **RSLinx Remote OPC Server**.

Tip: In a network application, always select **RSLinx Remote OPC Server**, instead of RSLinx OPC Server.

Setting up OPC data server redundancy

To minimize disruptions to clients in the event that the OPC 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 OPC server's 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.



☑ Provide redundancy using a secondary server.

Computer running secondary server:

Switch back options

● Continue using the secondary server even when the primary becomes available again.

○ Switchback to the primary server when it becomes available.

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 check box, 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.

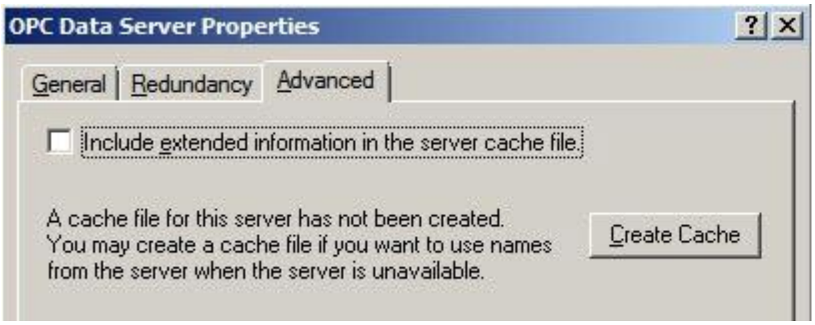
Switch back options

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**.

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.

A cache file for this server has not been created.
You may create a cache file if you want to use names from the server when the server is unavailable.

Include extended information in the server cache file

Select this check box 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 check box, **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 their attributes.
- 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.
- Defining alarm conditions for HMI tags.

About data server tags and HMI tags

FactoryTalk View Site Edition conforms to the OPC-DA 2.05a specification 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 two types of tags:

- **Data server tags** (also called direct reference tags, or data items in the OPC-DA specification) provide direct access to the basic attributes of all OPC-DA-compliant tags, through data servers you add to an application.
- **HMI tags** provide additional properties for run-time 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 sections in this chapter starting on Working with HMI tags on page 204.

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 (Logix5000 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 extended capabilities of HMI tags

To take advantage of extended capabilities, such as run-time 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 run time.

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 104.

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 474.

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 205.

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, by changing the physical addresses that the HMI tag names are mapped to.

HMI tags can also have descriptive names, which you cannot create in some controllers or OPC servers.

Monitoring alarms using HMI tags

HMI tags also provide an alternative method for triggering an alarm when a tag has a certain value. You can set up alarm conditions for HMI analog and digital tags.

For more information, see Setting up HMI tag alarms on page 217.

Tip: In FactoryTalk View SE, you can also use FactoryTalk Alarms and Events services to monitor and control alarms. For information, see Chapter 12, Setting up FactoryTalk alarms

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 Logix5000 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-DA servers that have their own tag database, see the documentation provided with the OPC-DA server.
   - In a Logix5000 controller, see the programming software documentation for the controller.
2. In FactoryTalk View Studio, in the Explorer window, if it doesn't 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 run time, 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 209.

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 run time, 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 run time.

To do this, attach Fill animation to the object using a tag that is updated by a network device that monitors the vat level. At run time, 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 run time, 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 196.

- Create the tag, if it does not exist.

  Create new data server tags in the device or OPC server. For example, in a Logix5000 device, create the tag using RSLogix 5000 programming software.

Create new HMI tags in the Tags editor, in FactoryTalk View Studio. For details, see Creating, modifying, and deleting HMI tags on page 209.

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 run time.

For details about setting up data logging, see Setting up data logging on page 585.

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 set up with alarms. Of these, 10,000 can be HMI analog tags.

Tip: RSLinx Enterprise 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 131.

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 located 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 located in the current area.

Use relative references, for example, to re-use 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.

The home area

In network distributed applications, the home area is the area, in which an application component (for example, a tag or graphic display) is located.

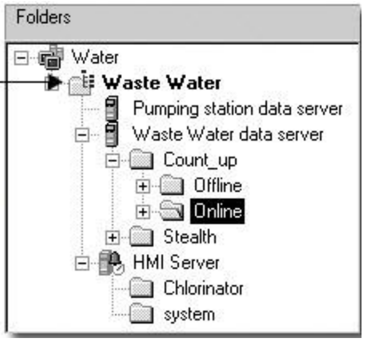
When you refer to an application component without specifying the area, FactoryTalk View SE uses the home area to locate the component.

For example, if you create a graphic display in the Cooling area, and add a relative reference to a tag (for example, FanStart), FactoryTalk View SE looks for the tag in the home area of the graphic display (Cooling).

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.

Finding the home area in the Tag Browser

In the Tag Browser, the home area in a network distributed application is shown in bold type, as shown in the following illustration.



The home area in this application is called Waste Water.

The home area contains two data servers, and an HMI server.

The Tag Browser composes references to tags automatically, using correct syntax.

If the tags you select are in the home area, the Tag Browser automatically uses relative references. If the tags are not in the home area, the Tag Browser uses absolute references.

Parts of the Tag Browser

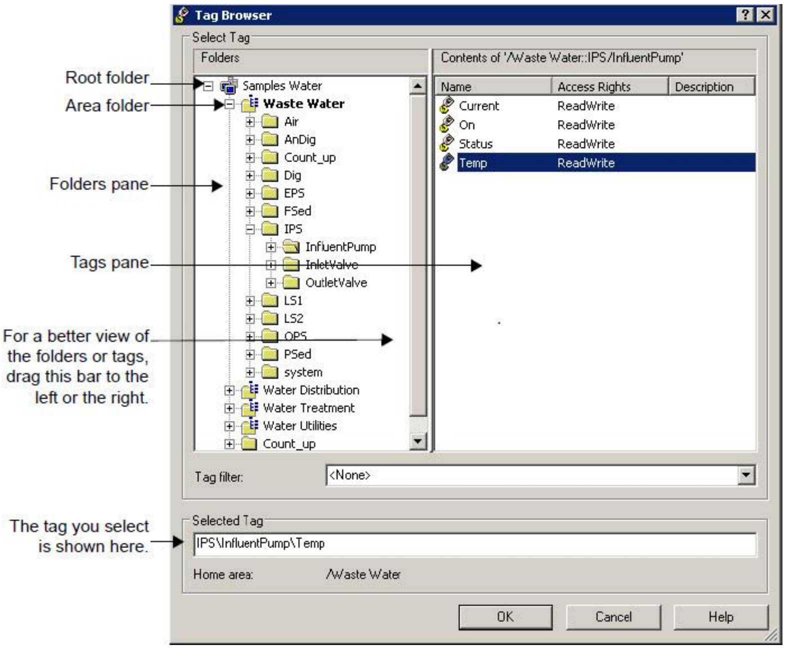
To use a tag in a FactoryTalk View SE application, you can type the tag's name and path, or you can use the Tag Browser to search for and select the tag.

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.



Contents of '/Waste Water::IPS/InfluentPump'

Name	Access Rights	Description
Current	ReadWrite	
On	ReadWrite	
Status	ReadWrite	
Temp	ReadWrite	

Tag filter: <None>

Selected Tag: IPS\InfluentPump\Temp

Home area: /Waste Water

Root folder

Area folder

Folders pane

Tags pane

For a better view of the folders or tags, drag this bar to the left or the right.

The tag you select is shown here.

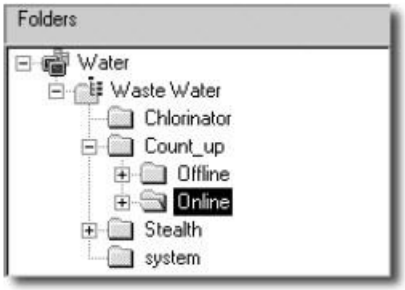
Viewing tags in folders

In the Tag Browser, the Folders display on the left shows the application root folder, plus all the folders that contain tags, in the application’s HMI servers and data servers. In a network distributed application, there is also a folder representing each area in the application.

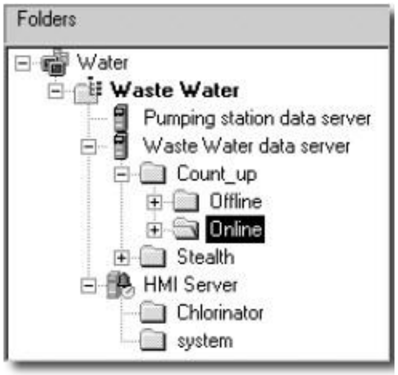
To view the tags in a folder, select the folder. In the previous illustration, the InfluentPump folder is open, and contains the tags shown on the right.

Showing server names

By default, the Tag Browser shows folders, but not the HMI servers or data servers the folders belong to. To show server names, right-click a blank area of the Folders pane, and then select **Show Server Names**.



The folders list with server names hidden.



The same folders list with server names shown.

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.

For more information about the home area, see The home area on page 195.

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 and all the components that refer to those tags.

You can also use the tag browser.

Finding a tag or text string

1. From the tool bar, 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

1. From the tool bar, 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.

To find 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 editor.

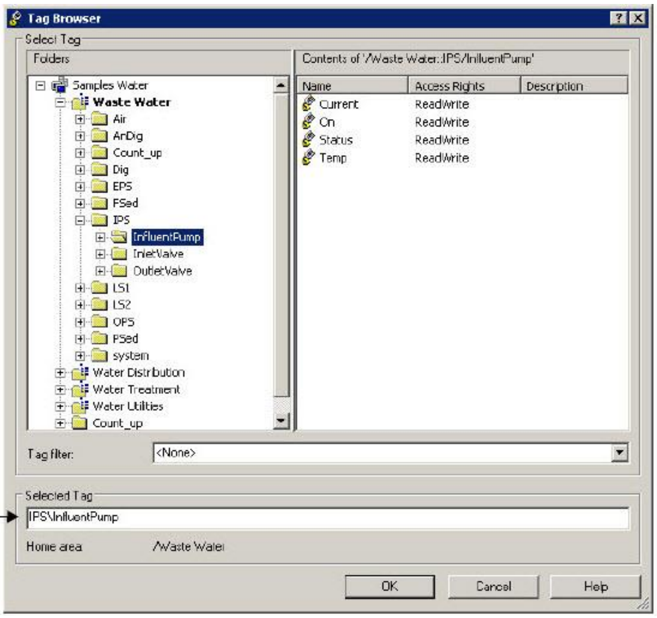
The tag or tags you select can be from a data server or an HMI server. The currently selected tag is shown under Selected Tag, in the lower part of the Tag Browser. If multiple servers use the same tag name, the server name is shown with the tag name

Tip: If you are able to select multiple tags, their names are shown under the label Selected Tags.

Selecting folders instead of individual tags

You can also select folders in the Tag Browser, for example, to select structure tags for faceplate objects.

When you select a folder, you select all the member elements in the folder. Instead of a tag name, the folder name is shown under Selected Tag, as shown in the next illustration



Contents of '/Waste Water::IPS/InfluentPump'

Name	Access Rights	Description
Current	ReadWrite	
On	ReadWrite	
Status	ReadWrite	
Temp	ReadWrite	

Tag filter: <None>

Selected Tag: IPS\InfluentPump

Home area /Waste Water

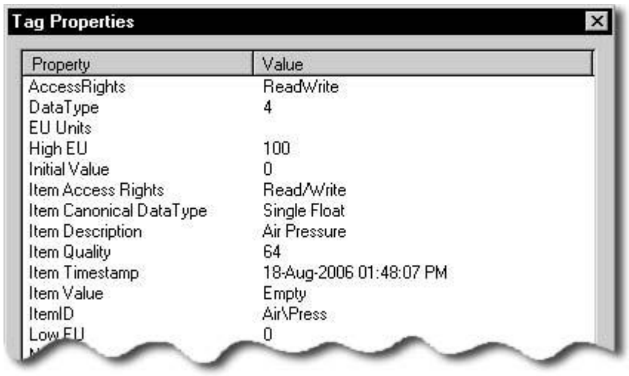
The folder you select is shown under Selected Tag.

Showing a tag’s properties

To show the properties of a tag in the Tag Browser, right-click the tag, and then click **Properties**. The Tag Properties dialog box shows information about the tag.

The properties are a snapshot, and do not update in real time. You cannot show the properties of multiple tags at the same time.

Tip: By default, the Tag Browser does not show tag descriptions. To show or hide tag descriptions, right-click a blank area of the right pane, and then select **Show Description**.

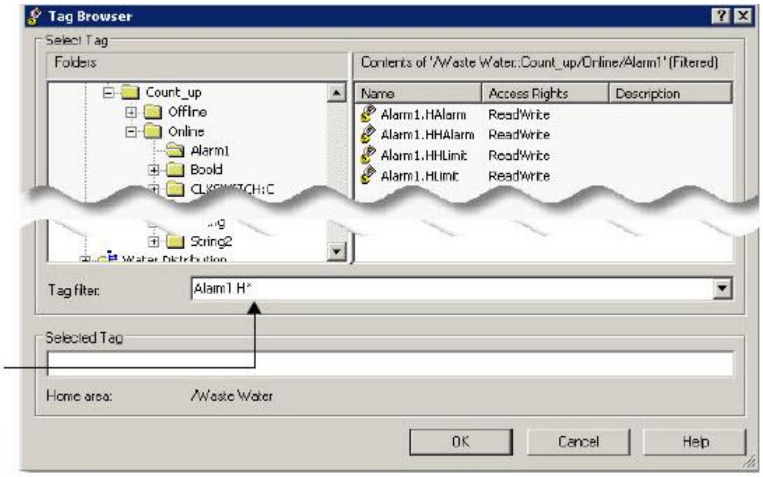


Tag Properties

Property	Value
AccessRights	ReadWrite
DataType	4
EU Units	
High EU	100
Initial Value	0
Item Access Rights	Read/Write
Item Canonical DataType	Single Float
Item Description	Air Pressure
Item Quality	64
Item Timestamp	18-Aug-2006 01:48:07 PM
Item Value	Empty
ItemID	Air\Press
Low EU	0

Filtering tags

To show only tags whose names match a pattern, type the pattern in the Tag Filter list box, and then press Enter.



Tag Browser

Select Tag

Contents of '/Waste Water::Count_up/Online/Alarm1' (Filtered)

Name	Access Rights	Description
Alarm1.HAlarm	ReadWrite	
Alarm1.HHAlarm	ReadWrite	
Alarm1.HHLimit	ReadWrite	
Alarm1.HLimit	ReadWrite	

Tag filter: Alarm1 H*

Selected Tag

Home area: /Waste Water

Type the tag filter here.

To clear the filter, click **<None>** in the list.

The Tag Filter list contains the last 10 filters you applied. You can create filters using wildcard characters:

This wildcard character	Does this
?	Matches any single character.
*	Matches any number of characters, including the backslash (\ ) character.

To remove a tag filter:

- In the Tag Browser, in the Tag Filter list, click <**None**>.

Creating modifying, and importing HMI tags

In the Tag Browser, you can also create new folders for HMI tags, create and modify the HMI tags, or import tags from a PLC or SLC database.

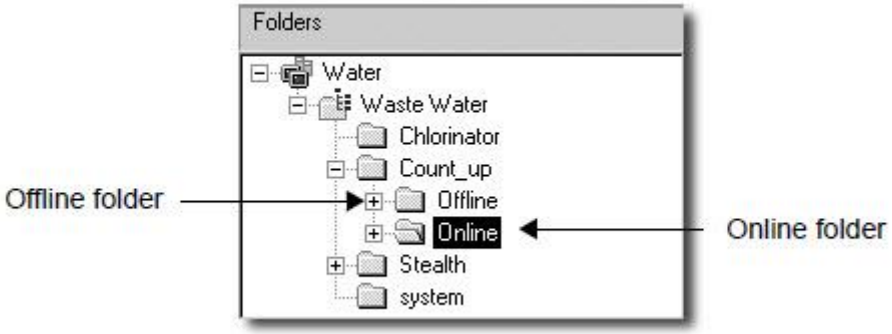
To perform these actions, right-click in the Tag Browser's right pane, and select:

- **New HMI Tag** to create a new HMI tag
- **Edit HMI Tag** to modify an existing HMI tag
- **Import PLC Tags** to import a tag from another database

For information about creating HMI tags and modifying their properties, see Creating, modifying, and deleting HMI tags on page 209.

Browsing for offline tags

In the Tag Browser, tags that are available when the application is connected to a device are located in the Online folder. Tags that are available from an offline file, such as a PLC program file, are located in the Offline folder.



Browsing for offline tags from RSLinx Enterprise

For each RSLinx Enterprise 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 Logix5000 program stored on disk. If the shortcut in RSLinx Enterprise 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 RSLinx Enterprise, see the RSLinx Enterprise Help.

To browse an offline file in the Communication Setup dialog box:

1. In FactoryTalk View Studio, in the Explorer window, double-click the RSLinx Enterprise icon.

2. Double-click Communication Setup to open the RSLinx Enterprise Communication Setup dialog box.

3. Click the **Browse** button beside Offline Tag File.

Browsing from RSLinx Classic

For each RSLinx Classic topic in an application, an Offline and an Online folder is shown in the Tag Browser.

Use the Offline folder to browse for tags in a PLC or Logix5000 program stored on disk. If the RSLinx topic can gain access to symbols, you can also browse for the symbols in the Offline folder.

For details about adding symbols to a topic in RSLinx, see the RSLinx Classic Help.

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. (In RSLinx, the offline tags are located in a separate folder.)

For details about setting up and managing an OPC data server's cache files, see Setting up advanced properties on page 187.

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 209.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.
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 bar code scanner that uses an alphanumeric product code.
System	Information generated while the system is running, including HMI tag alarm information, system time and date, and so on. 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 RSLinx, 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 check box 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.

Organizing 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 run time.

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 633.

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 2000.

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. From the View menu, select **Tag Statistics**.

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.

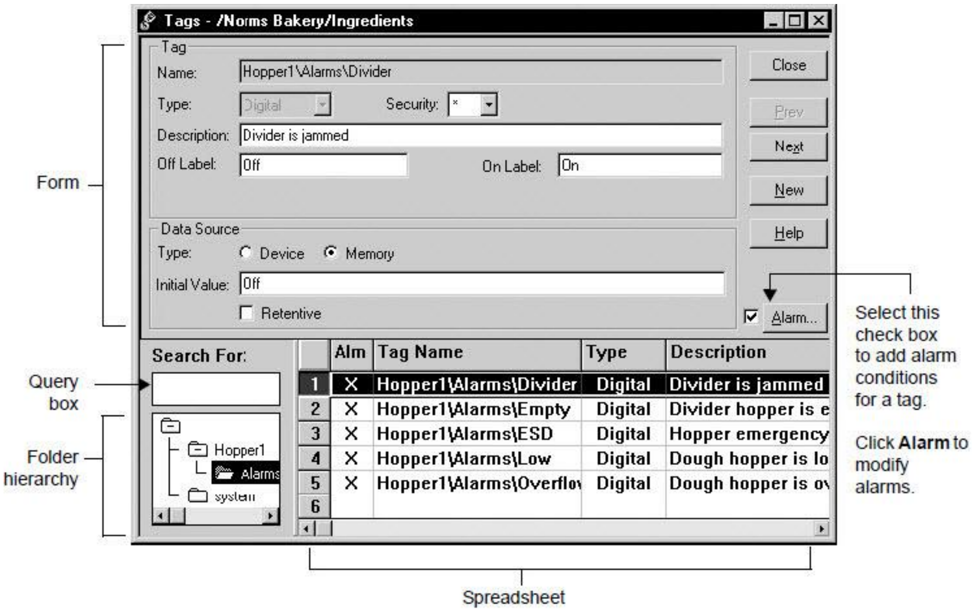
While the Tags editor is open, use items on the View menu and the toolbar, to control the editor's appearance.

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.



Tag

Name: Hopper1\Alarms\Divider

Type: Digital Security: *

Description: Divider is jammed

Off Label: Off On Label: On

Data Source

Type: ○ Device ● Memory

Initial Value: Off

□ Retentive

☑ Alarm...

Search For:

	Alm	Tag Name	Type	Description
1	X	Hopper1\Alarms\Divider	Digital	Divider is jammed
2	X	Hopper1\Alarms\Empty	Digital	Divider hopper is e
3	X	Hopper1\Alarms\ESD	Digital	Hopper emergency
4	X	Hopper1\Alarms\Low	Digital	Dough hopper is lo
5	X	Hopper1\Alarms\Overflo\	Digital	Dough hopper is o\
6				

Select this check box to add alarm conditions for a tag.

Click **Alarm** to modify alarms.

Accept and Discard buttons

When you enter information in the Tags editor, the **Prev** and **Next** buttons change to **Accept** and **Discard** buttons.

Click **Accept** to save tag and alarm information. Click **Discard** to cancel changes to a tag's properties.

Form and spreadsheet

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).

To define alarm conditions for an analog or a digital tag, select the Alarm check box. To modify alarms once they have been defined, click the Alarm button.

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. Select the Alarm box if the tag is to have an alarm set up for it.

   For more information, see Defining alarm conditions for HMI tags on page 215.

8. 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 a data source for an HMI tag

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 device as its data source receives its data from a source external to FactoryTalk View. The data can come from:

- Allen-Bradley or SoftLogix 5 programmable controllers through RSLinx.
- 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 177.

For information about DDE communications, see Setting up DDE communications for HMI tags on page 659.

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 OPC server, the access path is the name of a device shortcut or DDE/OPC topic in RSLinx.

- 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 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 633.

   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** check box.

   To let the memory tag revert to its initial value the next time the HMI server starts, clear the **Retentive** check box.

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.
- Create tags from the Tag Browser.
- 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 213.

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.

Creating tags in other FactoryTalk View editors

Any editor that uses tags has access to the tag database.

To create tags in the Tag Browser:

1. To open the Tag Browser, click **Tags** or the **Browse** button, whichever is available.
2. In the Tag Browser, open the folder in which you want to create the tag, right-click a blank area of the Contents pane, and then select **New HMI Tag**.

   For details about creating tags in the Tag Browser, click **Help** in the browser.

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** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **Tools** > **Tag Import and Export Wizard.**

For details about options in the Tag Import and Export Wizard, click **Help**.

Importing tags from a PLC database

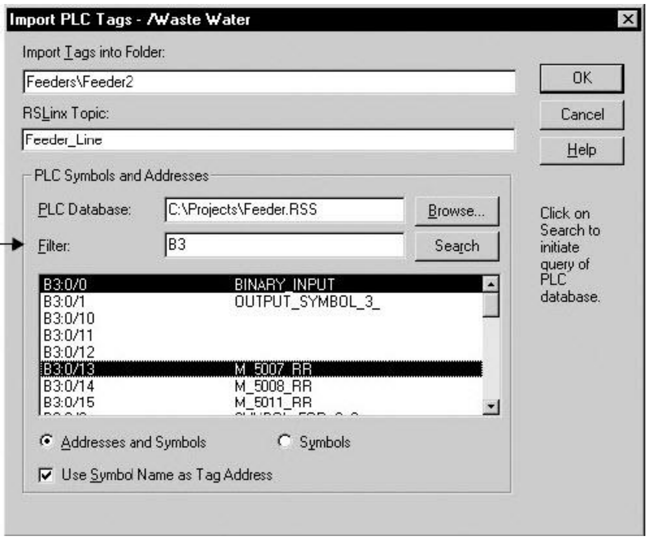
To import tags selectively from a PLC database into the FactoryTalk View HMI tag database, use the Import PLC Tags dialog box.

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, use one of these methods:

- In the Tags editor, click the **DB Browser** button on the toolbar, or from the **Edit** menu, select **Other Databases**.
- In the Tag Browser, right-click a blank area of the Contents pane, and then select **Import PLC Tags**.



Import PLC Tags - /Waste Water

Import Tags into Folder: Feeders\Feeder2

RSLinx Topic: Feeder_Line

PLC Database: C:\Projects\Feeder.RSS

Filter: B3

B3:0/0	BINARY_INPUT
B3:0/1	OUTPUT_SYMBOL_3_
B3:0/10	
B3:0/11	
B3:0/12	
B3:0/13	M_5007_RR
B3:0/14	M_5008_RR
B3:0/15	M_5011_RR

● Addresses and Symbols ○ Symbols

☑ Use Symbol Name as Tag Address

Use the filter to search for a particular tag or set of tags.

You cannot use wildcards, or filter beyond a slash. For example, a filter containing B3/1 will not show B3/10, B3/11, and so on.

Click on Search to initiate query of PLC database.

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.

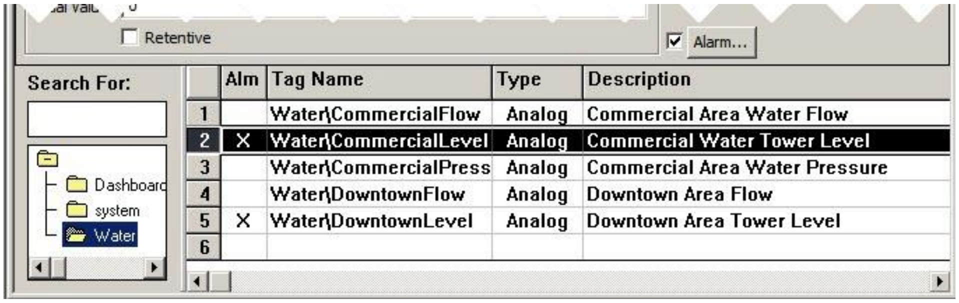
For details about options in the Import PLC Tags dialog box, click **Help**.

Defining alarm conditions for HMI tags

You can define alarms for HMI analog and digital tags.

At run time, FactoryTalk View scans the tag values in the tag database and compares them to the limits set for these tags. If a tag value crosses a limit, an alarm is triggered.

When an HMI tag is set up with an alarm, an **X** is shown in the Alm column of the Tags editor’s spreadsheet. When the tag is selected, the **Alarm** button in the editor’s form becomes available.



	Alm	Tag Name	Type	Description
1		Water\CommercialFlow	Analog	Commercial Area Water Flow
2	X	Water\CommercialLevel	Analog	Commercial Water Tower Level
3		Water\CommercialPress	Analog	Commercial Area Water Pressure
4		Water\DowntownFlow	Analog	Downtown Area Flow
5	X	Water\DowntownLevel	Analog	Downtown Area Tower Level
6				

For more information about setting up an alarm system using HMI tags, see Setting up HMI tag alarms on page 217.

Modifying HMI tag and alarm properties at run time

You can modify HMI tags and their alarm definitions in a FactoryTalk View SE network distributed application, and have the changes take effect at connected run-time clients without restarting the clients.

For a redundant network distributed application, you can also replicate changes made at the primary server to the secondary server.

For more information, see Modifying HMI tag and alarm properties at run time on page 373.

Setting up HMI tag alarms

This chapter describes:

- What HMI tag alarms are.
- Key HMI tag alarms concepts.
- The steps involved in setting up HMI tag alarms.
- Setting up general alarm behavior.
- Setting up alarm conditions for HMI tags.
- What FactoryTalk View SE alarm events are.
- Setting up HMI tag alarm logging.
- Viewing HMI tag alarm log files.
- Suppressing alarm logging.
- Creating an HMI tag alarm summary.
- Using tag placeholders to specify alarms.
- Running commands or custom programs in response to alarms.
- Starting and stopping HMI tag alarm monitoring.

About HMI tag 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 Site Edition, you can specify alarm conditions for tags you want to monitor in an HMI server’s tag database.

Alarms set up for HMI analog and digital tags are detected at the HMI server.

When an alarm is detected, notification is sent to connected FactoryTalk View SE clients, where operators can view and acknowledge the alarms in graphic displays.

HMI tag alarm features

Using HMI tag alarms, you can:

- Monitor any analog or digital HMI tag for alarms, up to a maximum of 40,000 tags per HMI server (10,000 of these can be analog tags).
- Show the most recent 2,000 alarm transactions in an HMI tag alarm summary.
- Define up to eight different severity levels to distinguish alarms visually.
- Use system default messages or custom messages to describe HMI tag alarms.
- Log alarm messages to a file, to a printer, or to both.
- Suppress HMI tag alarms for maintenance and tuning purposes.
- Associate a command or macro with an alarm to provide custom handling of the alarm.
- Share alarm information with other FactoryTalk View SE components. For example, you can use alarm HMI tag alarm functions in expressions in a graphic display.
- Use the Acknowledge bit to synchronize alarm acknowledgment on redundant HMI server pairs.
- Use the **AlarmEvent** command to respond to alarm detection algorithms you create for notification, logging, printing, and for display in HMI tag alarm summaries.

About FactoryTalk Alarms and Events

An HMI tag alarm system only detects alarms set up for tags in an HMI server's tag database. HMI tag alarm detection does not include FactoryTalk alarms.

To centralize alarm monitoring and control at the FactoryTalk Directory, set up these types of FactoryTalk alarms:

- **Device-based alarms**, set up by programming alarm detection instructions directly into Logix5000 controllers.

  FactoryTalk View SE Clients receive device-based alarms by way of Rockwell Automation Device Servers (RSLinx Enterprise) that you add to a FactoryTalk View SE application.
- **Tag-based alarms**, set up to specify alarm conditions for tags in older programmable controllers (PLC-5 or SLC 500), in third-party devices

communicating through OPC data servers, or in an HMI server’s tag database (HMI tags).

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

For more information, see Setting up FactoryTalk alarms on page 261.

Key HMI tag alarms concepts

This section presents some of the common terms and concepts used to describe HMI tag alarm services, components, and data.

Alarm thresholds for analog tags

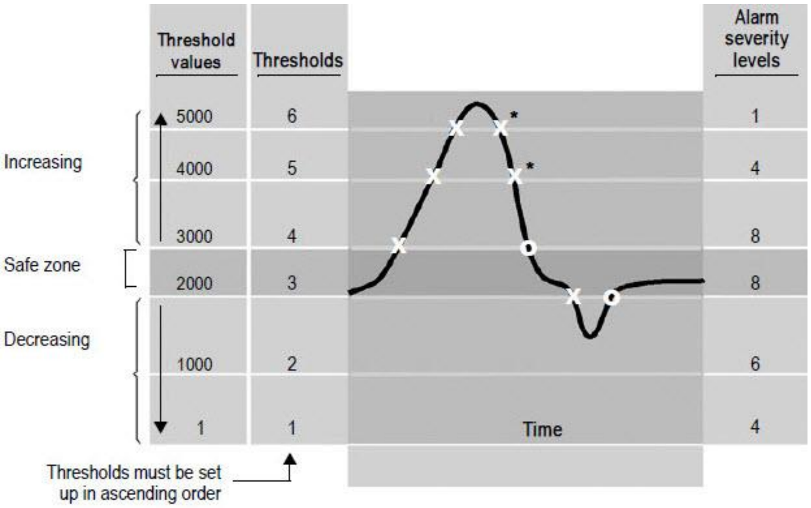
An HMI analog tag can trigger a number of alarms when its value crosses various levels, or thresholds.

You can assign up to eight alarm thresholds, each with a different level of severity to indicate the alarm’s importance.

The lowest threshold is 1 and the highest is 8. You do not have to use all eight thresholds for a tag, but the ones you do use must be set up in ascending order. For example, you can assign thresholds 1, 2, and 8, as long as you assign them in that order.

Thresholds can be increasing (monitoring for a rising value that crosses the threshold) or decreasing (monitoring for a falling value that crosses the threshold). The tag in the following example has both increasing and decreasing thresholds.

Example: Alarm thresholds



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 tag goes into alarm, and an O shows when the tag goes out of alarm.

If the motor speed	It triggers an alarm of this severity
exceeds 3000 rpm	8
exceeds 4000 rpm	4
exceeds 5000 rpm	1
falls below 5000 rpm*	4
falls below 4000 rpm*	8
falls below 2000 rpm	8
falls below 1000 rpm	6
falls below 1 rpm	4

With the given threshold settings, the motor must run between 2,000 rpm and 3,000 rpm, or an alarm is triggered.

If the check box, **Generate alarms when approaching normal operating range**, is selected in the Alarm Setup editor, alarms will occur when the motor speed falls below 5,000 rpm and 4,000 rpm, and when it rises above 1 rpm and 1,000 rpm.

Variable thresholds

Threshold values can be constant or variable. The previous example uses constant thresholds.

To define a variable threshold, specify a tag name when setting up alarm thresholds for an analog tag. As the value of the specified tag changes, the threshold changes.

Tip: Variable thresholds use more system resources than constant thresholds, due to the continuous scanning of threshold values, and the processing necessary to detect alarm faults.

Alarm faults

A variable threshold must not become higher than the threshold above it or lower than the threshold below it. If this happens, an alarm fault is generated for the tag being monitored.

To correct an alarm fault, change the variable threshold so it does not overlap either of its neighbors. This can be complex when the neighboring thresholds are also variable, because these boundaries are determined dynamically at run time.

When an alarm fault is generated, the following events occur:

- The tag's alarm status stays as it was before the fault was generated.
- An alarm fault is reported to all alarm reporting devices you have set up, such as the alarm log file and the printer.
- The alarm fault status bit for the tag is set in the value table. This notifies other applications that an alarm fault has been generated.
- *Alarm Fault* is posted in the Alarm Type column in the alarm summary.

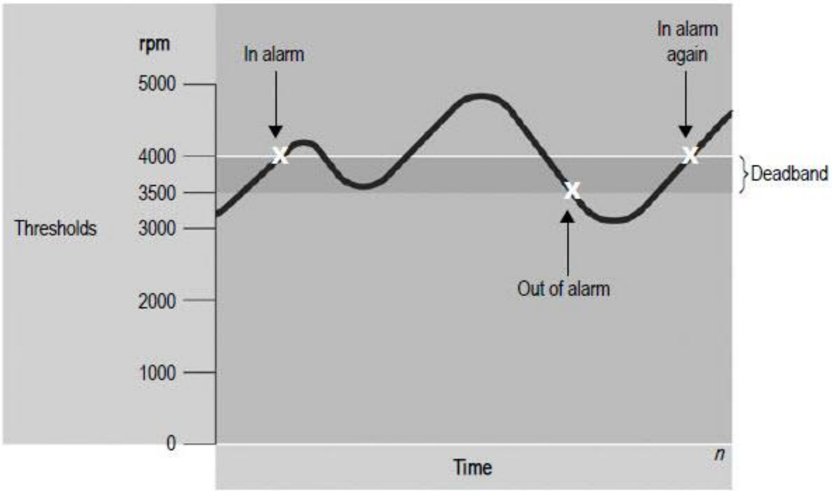
When the faulty thresholds return to their normal operating range, the alarm fault condition is cleared, the out-of-alarm-fault status is generated and logged, and alarm monitoring for the tag returns to normal.

Deadband

With some measured values, such as line pressure, tag values can fluctuate rapidly above and below a critical threshold. Where such conditions exist, you can create a deadband as a buffer to prevent the fluctuations from triggering unnecessary alarms.

If the threshold is increasing (monitoring rising values), the deadband range lies below the threshold. If the threshold is decreasing (monitoring falling values), the deadband lies above it.

The following illustration shows an increasing threshold of 4,000 rpm with a deadband value of 500 rpm. The rpm has to fall to 3,500 and then rise above 4,000 before it will trigger the alarm again.



A deadband range can be absolute, as in the previous illustration, or it can be a percentage of the minimum or maximum range for a tag. For example, if the deadband in the illustration were two percent, its range would be two percent of 5,000 rpm, or 100 rpm.

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 thresholds.

Note: Use the deadband carefully in safety-critical applications. In the preceding illustration, the variable has to fluctuate by more than 500 rpm before an alarm is triggered again.

Alarm states for digital tags

An HMI digital tag is either on or off. Instead of thresholds, digital tags have alarm states.

To trigger an alarm	Create this type of digital alarm
When a tag has a value of 1.	On
When a tag has a value of 0.	Off
When a tag value changes from 0 to 1 or from 1 to 0	Any Change*
Only when a tag value changes from 0 to 1.	Changes to On*
Only when a tag value changes from 1 to 0.	Changes to Off*

Tip: In the table, the asterisk (*) indicates change-of-state digital alarm types. These are considered out of alarm immediately after the change of state.

Alarm severity

HMI tag alarms can range in severity from 1 (most severe) to 8 (least severe), to indicate different levels of importance.

For example, a level-4 alarm might warn that a vat is half full of liquid, while a level-1 alarm indicates that the vat is about to overflow. Both alarms monitor the same tag but have different severity levels.

When you set up alarm severity, you specify what the severity levels mean and what actions they will trigger. Severity determines the order in which alarms are shown in an alarm banner.

Alarm messages

Alarm messages report information about alarms. There are three types of messages you can set up for HMI tag alarms:

- **In Alarm** messages are generated when tags go into alarm.

- **Out of Alarm** messages are generated when tags goes out of alarm.
- **Acknowledged** messages are generated when operators acknowledge alarms.

For each message type, you can use the default message or create a custom message, and route the messages to a log file or a printer, or to both destinations.

Alarm log file

The alarm log file records HMI tag alarm incidents. You can view the contents of an alarm log file in the HMI Tag Alarm Log Viewer.

Use the HMI Tag Alarm Log Setup tool to set up how often log files will be created and deleted. You can also use the tool to set up automatic logging to a central ODBC-compliant database.

If you export alarm log data to an external ODBC database, you can view records in the database using third-party, ODBC-compliant applications, such as Microsoft Excel or Microsoft Visual Basic.

Tip: To manually export the alarm log file to an ODBC database at run time, use the **AlarmLogSendToODBC** command. For details, see the FactoryTalk View Site Edition Help.

HMI tag alarm displays

HMI tag alarm information is recorded in alarm log files and on the HMI server computer where the alarms were generated.

To show records in the alarm log file, use the HMI Tag Alarm Log Viewer.

To show alarm information recorded in the HMI server computer’s memory, use graphic objects, such as the HMI tag alarm summary, in FactoryTalk View SE graphic displays.

Tip: HMI tag alarm summaries and alarm log viewers can only display HMI tag alarm information. To display FactoryTalk alarm information, you must set up the FactoryTalk objects installed with FactoryTalk View SE. For more information, see Setting up FactoryTalk alarms on page 261.

Alarm log viewer

The HMI Tag Alarm Log Viewer shows the contents of the alarm log file. The contents of the file depend on how severities are defined in the Alarm Setup editor. For details, see Setting up alarm severities on page 233.

By default, the log file will generate a record when:

- A tag goes into alarm.
- A tag goes out of alarm.
- An alarm is acknowledged.
- A tag set up with an alarm is suppressed.
- An alarm fault is generated.

For information about setting up the alarm log viewer, see Viewing HMI tag alarm log files on page 245.

Tip: Alarm logging uses system resources. To avoid slowing down the system, ensure that only essential alarm information is being logged.

Alarm summary

The HMI tag alarm summary is a graphic object, created in the FactoryTalk View SE Graphics editor, that shows HMI tag alarm information recorded in the HMI server computer’s memory.

A single HMI tag alarm summary can show alarms from multiple HMI servers located on different computers. Each HMI server maintains a list of up to 2,000 alarms.

As new alarms occur, they are shown at the top of the list in the alarm summary. When the list is full and a new alarm is generated, the alarm at the bottom of the list (the oldest alarm) is acknowledged automatically by the system, and then dropped from the list.

In the alarm log file, alarms acknowledged automatically by the system are identified by SysAk (system-acknowledged) in the TransType (transaction type) field.

Alarms are also dropped from the alarm summary when they are out of alarm, and acknowledged manually. When alarms are dropped, they are removed from memory. When the **AlarmOff** command is run, the entire contents of the alarm summary are erased.

Tip: AlarmOff is just one of the FactoryTalk View commands used with HMI tag alarms. For details about AlarmOff and other commands, see the FactoryTalk View Site Edition Help.

When setting up an HMI tag alarm summary, you can specify which alarm information to show, and how the information will appear in the alarm summary.

For example, you can specify a different color for each alarm incident: low-severity alarms could be set up as blue, medium-severity as yellow, and high-severity as red. This would make it possible for an operator to identify the severity of alarms at a glance.

For information about creating alarm summaries, see Creating an HMI tag alarm summary on page 249.

Graphic displays and objects

FactoryTalk View SE makes HMI tag alarm status information available to graphic displays through a set of system tags.

You can use these system tags with numeric and string display objects, to show specific information about HMI tag alarms in graphic displays. For details, see "Alarm system tags" next.

You can also attach visibility and color animation to affect the appearance of objects that display alarm status information.

The HMI Tag Alarm Information graphic library contains alarm-related graphic objects, for example, an alarm banner, that you can drag into a graphic display. Use the graphic objects in the library as they are, or modify them to suit the application's needs.

Alarm system tags

System tags that provide HMI tag alarm status information are created and updated on the HMI server. You can use these system tags wherever you can specify a tag name in FactoryTalk View SE.

To display this alarm status information	Use this system tag	Of type
The most recent, most severe alarm. If an alarm of an equal or higher severity occurs, it replaces the first alarm, whether or not the previous alarm has been acknowledged.	system\AlarmBanner	String
The date of the most recent, most severe alarm.	system\AlarmMostRecentDate	String
The threshold label of the tag of the most recent, most severe alarm.	system\AlarmMostRecentLabel	String
The severity of the most recent, most severe alarm.	system\AlarmMostRecentSeverity	Analog
The description of the tag of the most recent, most severe alarm.	system\AlarmMostRecentTagDesc	String
The name of the tag of the most recent, most severe alarm.	system\AlarmMostRecentTagname	String

The time of the most recent, most severe alarm.	system\AlarmMostRecentTime	String
The units of the most recent, most severe alarm.	system\AlarmMostRecentUnits	String
The number of unacknowledged and suppressed alarms.	system\AlarmStatus	String
The number of alarm transactions in an unfiltered alarm summary.	system\AlarmSummaryItems	Analog
The number of unacknowledged alarms in an unfiltered alarm summary.	system\AlarmSummaryItemsUnacked	Analog
The number of tags with alarms suppressed.	system\AlarmSuppressedCount	Analog

Alarm acknowledgment

An operator can acknowledge HMI tag alarms shown in an alarm summary or some other HMI tag alarm display. Acknowledging an alarm does not correct the condition causing the alarm, but indicates that the operator is aware of the alarm.

A *tag*, not an alarm, is acknowledged: a single tag might cause several alarms.

For example, a tag representing temperature might trigger Warm, Hot, and Overheat alarms by the time it is acknowledged. The tag could also go in and out of alarm several times before being acknowledged.

One acknowledgment is all that is required for all previous and current alarms for a tag, so alarm log files often show fewer acknowledgments than alarms.

Acknowledging alarms

To acknowledge HMI tag alarms, you can use any of these methods:

- Click **Acknowledge** or **Acknowledge All** in the HMI tag alarm summary.
- Use the FactoryTalk View command **Acknowledge** by itself, or with parameters to name a tag or group of tags.
- Use the FactoryTalk View command **AcknowledgeAll**. For details, see the FactoryTalk View Site Edition Help.

Unacknowledged alarms remain outstanding until the HMI tag alarm summary is full, alarm monitoring is turned off, or the system is shut down.

Alarm suppression

You can suppress alarm monitoring for single or multiple HMI tags. This is useful for testing or for performing repairs or maintenance on a piece of equipment.

To suppress alarm monitoring, use the FactoryTalk View command SuppressOn. To view a list of the tags not being monitored, use the Suppressed List. You can also turn monitoring back on from this list.

For details about using the **SuppressOn** and **SuppressOff** commands, see the FactoryTalk View Site Edition Help.

Alarm functions in expressions

When an HMI tag goes into alarm, or a FactoryTalk View SE alarm event occurs, information about the alarm is recorded along with the value of the tag or alarm event.

Using expressions, you can retrieve this information on a tag-by-tag basis.

Checking for HMI tags in alarm

If the system detects that an HMI tag is in alarm, it sets an internal alarm bit, and then resets the bit when the tag is out of alarm.

The following expression checks whether an HMI tag is in alarm:

```
ALM_IN_ALARM (tag )
```

where *tag* is the name of the HMI tag you want to check for alarms.

When a tag is in alarm, the expression result is 1. When a tag is out of alarm, the expression result is 0.

One way to use this expression is to animate the visibility of a graphic object in a display. When the tag goes into alarm, the ALM_IN_ALARM expression is set to 1, making the object visible. This is an effective way to draw the operator’s attention to the alarm.

Checking for acknowledged HMI tag alarms

The following expression checks whether an HMI tag alarm has ever been acknowledged:

```
ALM_ACK (tag )
```

If an alarm has been acknowledged, the expression ALM_ACK returns 1. If a tag goes out of alarm without being acknowledged, the expression returns 0.

Tip: When alarm monitoring starts, the ALM_ACK expression returns 1 by default, for tags that have never been in alarm. To reverse the default behavior, create the registry key Alarm Initially Acked on the computer running the HMI server. Then, change the key's string value to False.

To create or change the key, use the Windows Regedit32 utility. The key is located at:

```
HKEY_LOCAL_MACHINE\Software\Rockwell Software\RSView
Enterprise\Alarm Server
```

For details about using the Regedit32 utility, click **Help** in the Registry editor window.

The following expression checks whether several alarms, of a specified pattern, have ever been acknowledged:

```
ALM_ALLACKED (tag *)
```

If all tags that match the pattern have been acknowledged, the expression ALM_ALLACKED (tag *) returns 1. If *any* of the tags have not been acknowledged, the expression returns 0.

Other functions are also used with expressions to monitor alarms. For a complete list of *al*arm functions, see Tag functions on page 567.

Tip: If a graphic display requires more than 20 HMI tag alarm functions, it is recommended that you use the alarm functions in derived tags. For details, see the FactoryTalk View Site Edition Help.

Using wildcard characters in alarm expressions

For built-in alarm functions, you can use wildcard characters in the expression 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 checks whether any of a group of tags is in alarm:

```
ALM_IN_ALARM (alarm *)
```

where *alarm* * represents all tags whose names begin with *alarm*.

If one or more of these tags are in alarm, the expression result is 1. If all of the tags are out of alarm, the expression result is 0.

Tip: If the wildcard pattern in an alarm expression matches a large number of tags, it could take more time to resolve the expression.

Acknowledge bit

An acknowledge bit is a digital tag that can be used to:

- acknowledge an HMI tag alarm.
- monitor the acknowledged state of an HMI tag alarm.

When you set up an acknowledge bit for a tag, the HMI server monitors the value of the acknowledge bit and acknowledges the alarm automatically, when the value of the acknowledge bit changes from 0 to 1.

In the alarm log file, this transaction is identified by RmAck (Remote Acknowledge) in the TransType (transaction type) field.

When an operator acknowledges an alarm, the acknowledge bit is set to 1. If the Auto Reset feature is enabled, the HMI server resets the acknowledge bit to 0 when the tag goes back into alarm.

The acknowledge bit must be set up as a digital tag in the HMI tag database. For details about setting up an acknowledge bit, see the FactoryTalk View Site Edition Help.

Tip: The HMI server monitors the value of each acknowledge bit tag. Using many acknowledge bits can slow down the system considerably.

Handshake bit

A handshake bit is a digital tag that can be used to monitor the status of an HMI tag alarm.

When you set up a handshake bit for a tag, the HMI server sets the handshake bit to 1 when the tag goes into alarm. If the Auto Reset feature is enabled, the HMI server sets the handshake bit to 0 when the tag goes out of alarm.

The handshake bit must be set up as a digital tag in the HMI tag database. For details about setting up a handshake bit, see the FactoryTalk View Site Edition Help.

Switching handshaking on

Handshaking is turned off by default. To switch handshaking on, you can:

- use the */H* parameter with the **AlarmOn** command.
- use the **HandshakeOn** command.

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

Summary of steps for setting up HMI tag alarms

Following are the tasks involved in setting up HMI tag alarms in a FactoryTalk View SE application. Each task is described in detail, in subsequent sections of this chapter:

1. In the Alarm Setup editor, set up the attributes of all alarms on an HMI server:
   - In the Setup tab, specify the maximum update rate, behavior whether HMI server redundancy is set up, and whether to generate alarms for analog tags when their values approach normal range.
   - In the Severities tab, specify the logging destination and type of notification for the alarm severity levels (1 through 8) and for alarm events.
   - In the User Msgs tab, create user default messages to use in place of system default message, when alarms occur.
2. Use the HMI Tag Alarm Log Setup tool to set up the alarm log file. You can specify where the file is stored, and when (or whether) log files will be created and deleted.
3. In the Tags editor, in the Analog or Digital Alarm editor, specify alarm conditions for each HMI tag you want to monitor. You can define alarms for analog and digital tags, but not for string tags.
4. In the Graphics editor, set up alarm information displays. In graphic displays, you can:

- Set up alarm summaries.
- Embed numeric or string displays that use alarm functions or alarm system tags.
- Provide ways to start and stop alarm monitoring.

Note: The FactoryTalk View **AlarmOn** and **AlarmOff** commands control alarm monitoring on the HMI server. How you choose to start and stop alarm monitoring is particularly important for network distributed applications. For details, see Starting and stopping HMI tag alarm monitoring on page 258.

Setting up general HMI tag alarm behavior

Use the Alarm Setup editor to specify the general features of all HMI tag alarms for a given HMI server. In the:

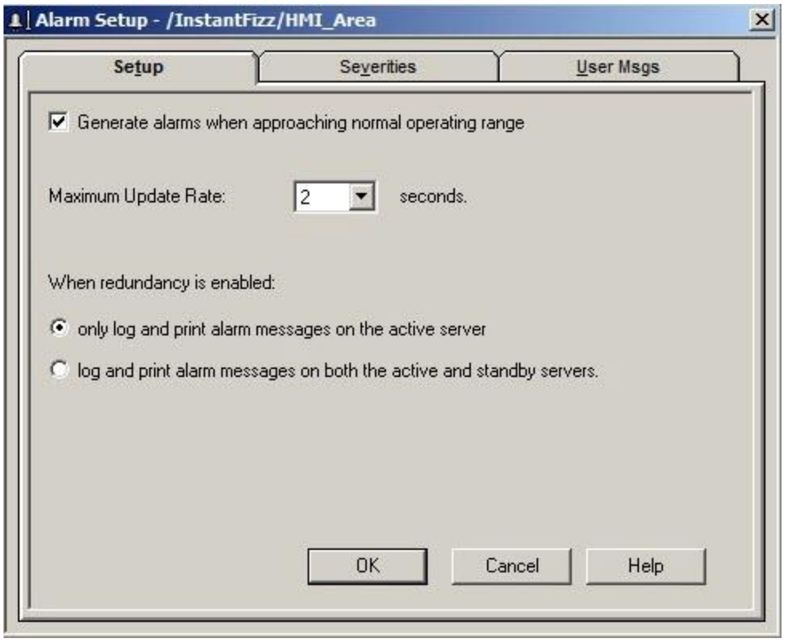
- **Setup** tab, specify general behavior for alarm monitoring and annunciation.
- **Severities** tab, set up logging and alarm behavior for individual severities.
- **User Msgs** tab, create custom alarm messages to use in place of the default system messages.

For details about options in Alarm Setup editor tabs, click **Help**.

To open the Alarm Setup editor:

1. In FactoryTalk View Studio, in the Explorer window, open the HMI Tag Alarms folder.
2. Right-click the Alarm Setup icon, and then select **Open**.

You can also double-click the Alarm Setup icon, or drag the icon into the FactoryTalk View Studio workspace.



☑ Generate alarms when approaching normal operating range

Maximum Update Rate: 2 seconds.

When redundancy is enabled:

● only log and print alarm messages on the active server

○ log and print alarm messages on both the active and standby servers.

Setting up alarm monitoring

In the Setup tab of the Alarm Setup editor, you can specify:

- How the system behaves when alarms cross back over trigger thresholds.
- How often the system checks for new alarms.
- Where alarms are logged and printed when redundant HMI servers are in use.

Alarm trigger thresholds

An HMI tag alarm is triggered when the value of an analog tag crosses a specified alarm threshold.

If the value of the tag fluctuates above and below its alarm threshold, alarms are triggered each time the tag value rises above or falls below the threshold value. You can specify whether to use this feature. For an example of how it works, see Alarm thresholds for analog tags on page 219.

Specifying how often the system checks for alarms

The system does not check for alarms more frequently than the maximum update rate specified in the Alarm Setup editor.

Match the maximum update rate to the rate, at which you expect tag values to change. For example, if you are monitoring temperatures that fluctuate slowly, check for alarms less frequently than if you are monitoring manufacturing processes that change rapidly.

Setting up alarm monitoring on redundant servers

If you have set up redundant HMI servers, HMI tag alarms can be logged and printed on the active HMI server only, or on both the Active and the Standby HMI server.

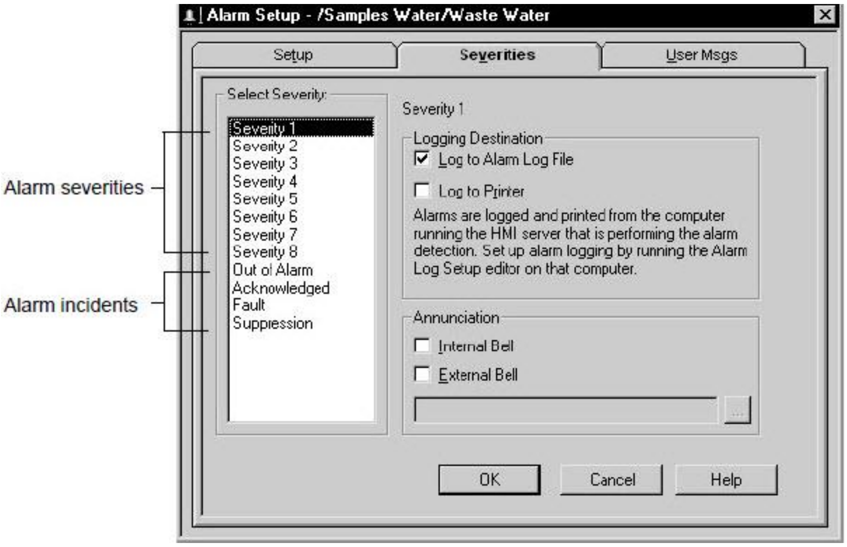
For more information, see Setting up FactoryTalk system availability on page 351.

Note: For HMI tag alarm states to be synchronized properly, the clocks on the primary and secondary HMI servers must be kept synchronized to a time server. If the clocks are not synchronized, when a fail-over occurs, multiple alarms or inconsistent information might display in an alarm summary.

Setting up alarm severities

In the Severities tab of the Alarm Setup editor, you can set up logging destinations and annunciation options for each HMI tag alarm severity and alarm incident.

For each severity or incident, you can log to the alarm log file, to a printer, or to both. You can also specify how notification is handled, for alarms of particular severities.



Severity 1

Logging Destination

☑ Log to Alarm Log File

□ Log to Printer

Alarms are logged and printed from the computer running the HMI server that is performing the alarm detection. Set up alarm logging by running the Alarm Log Setup editor on that computer.

Annunciation

□ Internal Bell

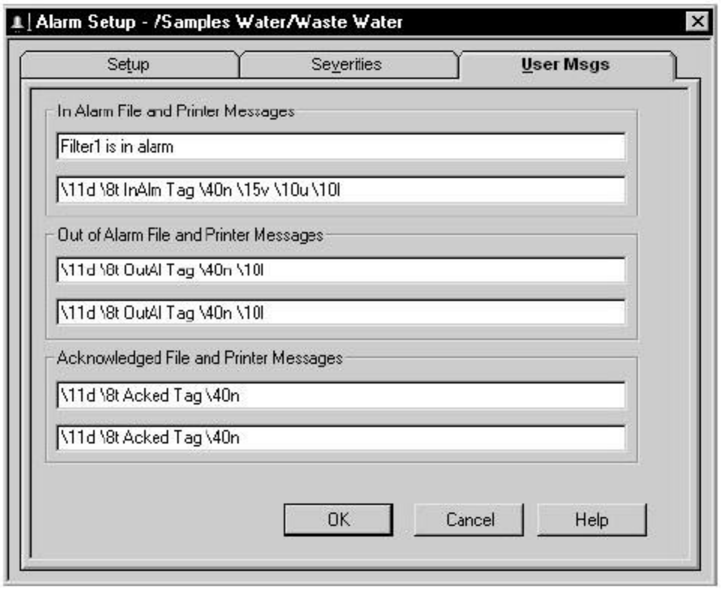
□ External Bell

Alarm or incident	What is logged (to file, printer, or both)
Severity 1 through 8	In Alarm messages.
Out of alarm	Out of Alarm messages (for any severity that has been set up to log In Alarm messages).
Acknowledged	Acknowledged messages (for any severity that has been set up to log In Alarm messages).
Fault	All Fault messages.
Suppression	All Fault messages.

Tip: When logging to a printer, you can use only continuous-feed printers, such as dot-matrix printers. Page printers, such as laser printers, are not supported.

Setting up alarm messages

Alarm messages report information about HMI tag alarms. When setting up alarm messages, you can use the system defaults, or you can define the content of each message.



Alarm Setup - /Samples Water/Waste Water

In Alarm File and Printer Messages

Filter1 is in alarm

\11d \8t InAlm Tag \40n \15v \10u \10l

Out of Alarm File and Printer Messages

\11d \8t OutAl Tag \40n \10l

\11d \8t OutAl Tag \40n \10l

Acknowledged File and Printer Messages

\11d \8t Acked Tag \40n

\11d \8t Acked Tag \40n

To create a user-default message, type the message in the appropriate box.

You can create user messages for logging to a file or a printer.

In the User Msgs tab of the Alarm Setup editor, you can define the content of:

- **In Alarm** messages, which are logged when an analog tag’s alarm threshold has been crossed, or when a digital tag has changed state.
- **Out of Alarm** messages, which are logged when a tag is no longer in alarm.

- **Acknowledged** messages, which are logged when an alarm has been acknowledged.

For details about options in the User Msgs tab, click **Help**.

Tip: Alarm messages appear in HMI tag alarm log files, not in HMI tag alarm summaries.

Defining the content of the message

To define the content of a message, use any combination of system default messages, user default messages, and custom messages.

System default messages are a series of placeholders. At run time, these placeholders are replaced with information about the alarm that has been triggered.

This placeholder	Is replaced with
\C	Current value of the tag.
\D	Date when the message is sent.
\L	Alarm label specified in the Alarm Label field of the Analog Alarm and Digital Alarm editors.
\N	Name of the tag in alarm.
\S	Description of the tag in alarm.
\T	Time when the message is sent.
\U	Units specified in the Units field, in the Tags editor.
\V	Threshold value that was crossed.

Tip: \C, \L, \U, and \V do not contain any information when used in alarm Acknowledged messages.

The system default messages are used automatically, for all alarms and alarm events. For alarm events, they are used if you don’t specify a log message string using the */L* parameter with the **AlarmEvent** command. For details, see the FactoryTalk View Site Edition Help.

Example: Message placeholders

An out-of-alarm message set up like this:

\11D \8T Tag \10N is out of alarm.

is shown like this at run time:

```
01/22/06     11:45:02 Tag PUMP3 is out of alarm.
01/22/06     11:47:23 Tag PUMP4 is out of alarm.
01/22/06     11:48:14 Tag PUMP5 is out of alarm.
```

The width of a column in a message is the number of spaces specified between the backslash and the placeholder character. In this example, the date column has 11 spaces (\11D), the time has 8 spaces (\8T), and the tag name has 10 spaces (\10N).

User default messages are messages you create to replace the system default messages. You can use both words and placeholders to define user defaults.

Custom messages apply on a per-tag basis. You set them up in the Tags editor, in the Alarm Messages tab in either of the Analog or Digital Alarm editors.

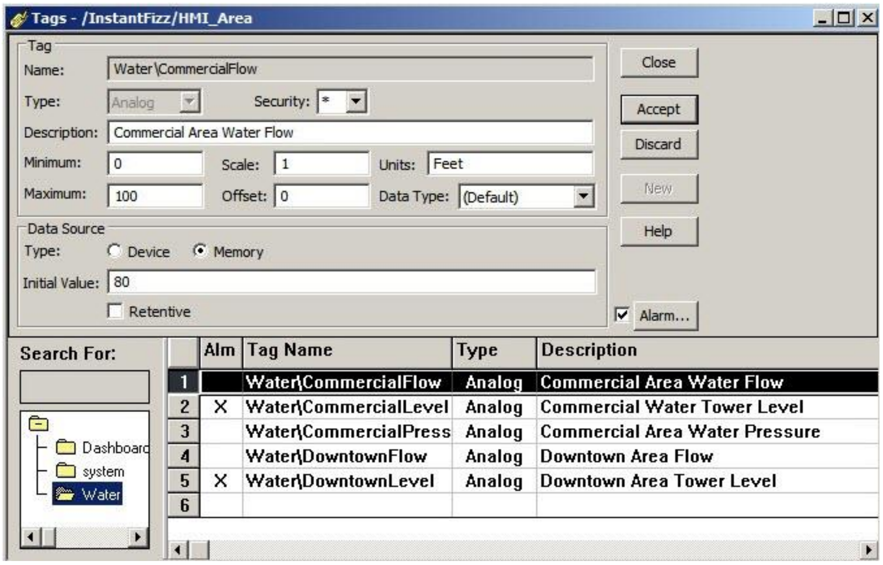
To create a custom message, type any message you want. You can use both words and placeholders to define custom messages.

Setting up alarm conditions for HMI tags

In the Tags editor, for each analog and digital tag you want to monitor for alarms, use the Analog Alarm and Digital Alarm editors to specify conditions that will trigger an alarm.

To open the Tags editor:

- In FactoryTalk View Studio, in the Explorer window, open the HMI Tags folder, right-click the Tags icon, and then click **Open**.



Tags - /InstantFizz/HMI_Area

Tag
Name: Water\CommercialFlow
Type: Analog
Security: *
Description: Commercial Area Water Flow
Minimum: 0
Scale: 1
Units: Feet
Maximum: 100
Offset: 0
Data Type: (Default)

Data Source
Type: ○ Device ● Memory
Initial Value: 80
□ Retentive

☑ Alarm...

	Alm	Tag Name	Type	Description
1		Water\CommercialFlow	Analog	Commercial Area Water Flow
2	X	Water\CommercialLevel	Analog	Commercial Water Tower Level
3		Water\CommercialPress	Analog	Commercial Area Water Pressure
4		Water\DowntownFlow	Analog	Downtown Area Flow
5	X	Water\DowntownLevel	Analog	Downtown Area Tower Level
6				

You can set up alarms for a tag when you create the tag, or you can set up alarms later.

To add alarm conditions to an analog or digital tag:

1. In the spreadsheet section of the Tags editor, select the analog or digital tag you want to set up alarm conditions for.

2. In the form section of the Tags editor, select the check box beside the Alarm button.

To modify alarm conditions already set up for a tag:

1. In the spreadsheet section of the Tags editor, select the analog or digital tag with alarm conditions, that you want to modify.

2. In the form section of the Tags editor, click **Alarm**.

Setting up alarms for HMI analog tags

These are the tasks involved in setting up alarms for HMI analog tags:

1. Set up the alarm threshold, and then choose the message to show when the threshold is crossed.

2. Set up the messages that are sent when a tag goes out of alarm, and when an operator acknowledges an alarm.

3. Set up advanced features, such as deadband and handshake.

For details about options in the Analog Alarm editor, click **Help**.

Setting up alarms for HMI digital tags

These are the tasks involved in setting up alarms for HMI digital tags:

1. Set up alarm states.

2. Set up the messages that are sent when a tag goes out of alarm, and when an operator acknowledges an alarm.

3. Set up advanced features, such as handshake.

For details about options in the Digital Alarm editor, click **Help**.

Modifying HMI tag and alarm properties at run time

In a FactoryTalk View SE network distributed application, you can modify HMI tag and alarm properties at run time, and have the changes take effect at connected clients, without restarting the clients.

HMI tag properties can be modified in FactoryTalk View Studio, in the Tags editor, in the HMI Tag dialog box (opened from the Tag Browser), or in the Tag Import & Export Wizard.

HMI tag alarm properties can be modified in the Analog or Digital Alarms editors, opened from the Tags editor. The **AlarmAcceptEdits** command must be run for alarm changes to take effect.

In FactoryTalk View Studio, for a redundant network distributed application, you can also replicate changes made at the primary server, to the secondary server.

For information about:

- Modifying HMI tag properties, see Modifying HMI tag properties on page 374.
- Modifying HMI tag alarm properties, see Modifying HMI tag alarm properties on page 376.
- Replicating changes from the primary to the secondary server, see Replicate changes to the secondary HMI server on page 366.

About FactoryTalk View SE alarm events

You can customize and extend HMI tag alarm monitoring, by writing alarm-detection algorithms using PLC ladder logic, custom programs, or other appropriate tools.

You can then add alarm events to the HMI tag alarm subsystem, to respond to the alarm-detection algorithms for annunciation, logging, printing, and display in alarm summaries.

By using a tag name for an alarm event, you can customize the alarm features of the tag. For example, you can use alarm events to provide a tag with more than eight thresholds. You can also use alarm events to specify an alarm's time stamp.

Since alarms are scanned in the background, alarms that occur rapidly might be shown out of sequence in FactoryTalk View SE, because they might all be scanned at the same time, and therefore produce the same time stamp.

If the sequence, in which alarms occur is important, you might want to record accurate time stamps for HMI tag alarms by buffering the alarms in the PLC, and then using alarm events to record the alarms with accurate time stamps in FactoryTalk View SE.

To create a FactoryTalk View SE alarm event, use the **AlarmEvent** command. For details, see the FactoryTalk View Site Edition Help.

How event-based alarms work

FactoryTalk View SE alarm events let you create alarms without using HMI tags. Event-based alarms work just like HMI tag alarms. You can:

- Show and filter event-based alarms in HMI tag alarm summaries.
- Acknowledge event-based alarms individually, or with wildcard characters, using the **Acknowledge** command.
- Use event-based alarms with alarm system tags.
- Log event-based alarms to disk or printer.

You can also use FactoryTalk View SE alarm events with HMI tag alarm functions in expressions.

In FactoryTalk View SE, or in alarm-detection algorithms you create for the application, you can stamp event-based alarms with the current time, or with a specific date and time.

Note: Alarm events are not processed until the **AlarmOn** command is run, and alarm events are no longer processed after the **AlarmOff** command is run.

How event-based alarms differ from HMI tag alarms

FactoryTalk View SE alarm events differ from HMI tag alarms in the following ways:

- You cannot specify thresholds for analog alarm events. All analog alarm events have a value of zero.
- You cannot specify alarm labels for event-based alarms. That is, you cannot use the IntoAlarm and OutOfAlarm labels for HMI digital tag alarms, or the threshold labels for HMI analog tag alarms.
- You cannot suppress event-based alarms.
- Alarm events have no acknowledge and handshake bits.
- You cannot use the Identify feature with event-based alarms, to run a command, macro, or custom program.
- Event-based alarms are not retained after the **AlarmOff** command is run, or after the HMI server shuts down.

Naming alarm events

You must provide a name for each alarm event you create. The event name can be a tag in the HMI server's tag database, as long as the tag has no alarm conditions set up for it.

Alarm event names can be up to 255 characters long and can contain the following characters:

- A to Z
- 0 to 9
- underscore (_) and dash (–)

When an alarm event name starts with a number or contains a dash, enclose the name in brackets { } when you use it in an expression, for example, {N33-0}.

Also use brackets when using wildcard characters to represent multiple alarm events in an expression, for example, {alarm*}.

Alarm event names preserve upper and lower case for readability but are not case sensitive. For example, the alarm event name HopperOverflow is the same as hopperoverflow.

How event-based alarms are logged

The alarm log file shows event-based alarms in the order, in which the alarm transactions were logged. If you specify a time stamp for alarm events, the alarm log could show the transactions out of order.

About alarm event types

Use the **AlarmEvent** command to create into-alarm and out-of-alarm events. Multiple into-alarm events can be processed for the same event name before an out-of-alarm event is received.

Use the InAndOutOfAlarm event type for change-of-state alarms. An out-of-alarm event is ignored if no into-alarm events preceded it.

Setting up HMI tag alarm logging

Use the HMI Tag Alarm Log Setup tool to specify:

- Where to store alarm log files.
- When alarm log files are created or deleted.
- Optionally, whether to log alarm data to a central, ODBC-compliant database.

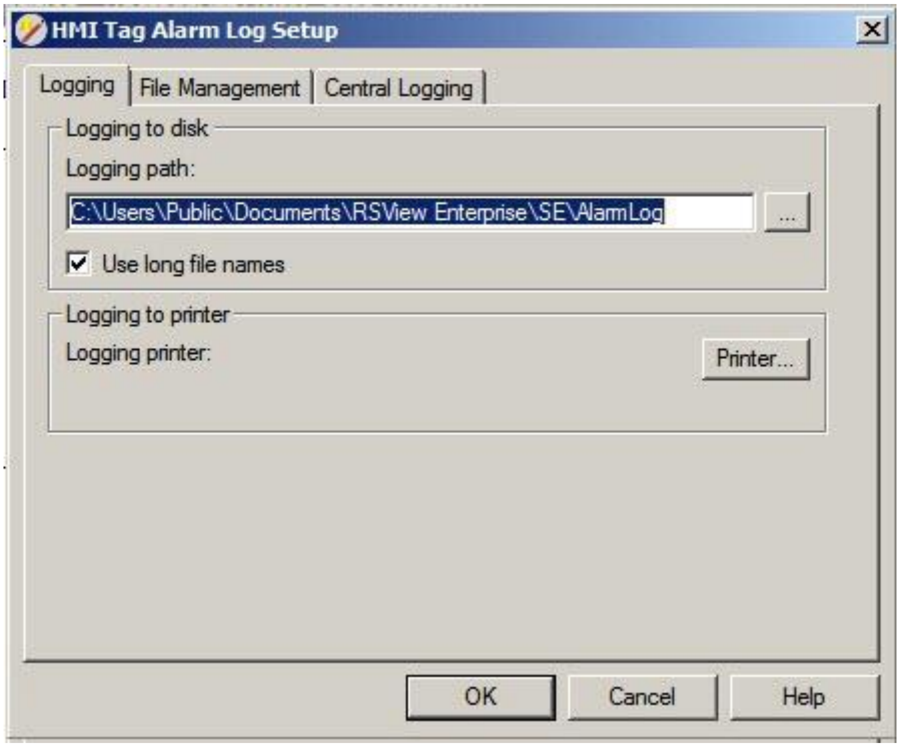
Note: Alarm logging must be set up separately on each computer running an HMI server that is monitoring tags for alarms; otherwise, the settings will have no effect.

If two HMI servers are running on a computer, the alarm log will contain data for both HMI servers. If the hard disk space is used up on the computer, alarm logging will stop.

To open the HMI Tag Alarm Log Setup tool:

Use one of these methods:

- In FactoryTalk View Studio or the FactoryTalk View SE Administration Console, from the Tools menu, select **HMI Tag Alarm Log Setup**.
- Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **Tools** > **SE HMI Tag Alarm Log Setup**.



Logging path: C:\Users\Public\Documents\RSView Enterprise\SE\AlarmLog

☑ Use long file names

For more information about HMI tag alarm log files, see:

- Adding remarks to the alarm log file at run time on page 247.
- Viewing HMI tag alarm log files on page 245.

Specifying where log files are stored or printed

In the Logging tab of the HMI Tag Alarm Log Setup tool, you can specify where alarm log files are stored on the computer. The default location is:

- ..\Users\Public\Public Documents\RSView Enterprise\SE\AlarmLog
- (Windows XP) ..\Documents and Settings\All Users\Shared Documents\RSView Enterprise\SE\AlarmLog

If alarm information is being logged to a printer, you can also specify where the information will be printed. For details about options in the Logging tab, click **Help**.

Specifying when log files are created and deleted

In the File Management tab of the HMI Tag Alarm Log Setup tool, you can specify when alarm log files are created and deleted.

You can set up the system to create new log files periodically, or you can prevent new files from being created.

In each 24-hour period, up to 26 new files can be created. If you attempt to create a 27th file, FactoryTalk View continues logging data to the 26th file. At midnight, the sequence starts again with the first new file for the new day.

You can set up file management to delete log files after a specified period, or once a specified number of files have been created. Files scheduled to be deleted are deleted whenever a new file is created.

For example, if you set up logging to start a new file daily, and to delete the oldest files every third day, the system will save alarm log files for the current day and for the three previous days. The oldest file is deleted whenever a new file is created.

To never delete files, clear the check boxes under Delete oldest files.

For details about the options in the File Management tab, click **Help**.

Note: If file management is set up to delete the oldest files when a new file is started, and you are exporting data to an ODBC database, make sure you export the data before the oldest file is deleted, or the oldest data will be lost.

Setting up periodic logging to a central ODBC database

In the Central Logging tab of the HMI Tag Alarm Log Setup tool, you can set up logging to a central, ODBC-compliant database.

Central logging works by exporting the contents of the alarm log file periodically to an ODBC-compliant database. FactoryTalk View supports the following ODBC-compliant databases: Sybase SQL Server, Oracle, and Microsoft SQL Server.

For details about options in the Central Logging tab, click **Help**.

Setting up security to log alarms to a remote computer:

If you are logging HMI tag alarms to an ODBC database that is not located on the HMI server's computer, you must change certain DCOM security settings. For details, see the FactoryTalk View Site Edition Help.

Exporting alarm log files manually to ODBC format

You can also use the FactoryTalk View command AlarmLogSendToODBC, to export alarm log file contents manually to an ODBC-compliant database.

If the specified database table is not ODBC compliant, the export will fail. If the table does not exist, FactoryTalk View will try to create one.

When you export data to an ODBC table, FactoryTalk View keeps track of the data in a control file called Alarm.exp, which is located in the same path that stores the alarm log .dat files. Whenever you export data, only the newest data is exported.

However, if the control file is deleted, all the alarm log data in the .dat files is exported when the **AlarmLogSendToODBC** command is run.

Example: Exporting alarm log files to ODBC format on demand

To export the contents of the alarm log files to an ODBC-compliant database on demand, create a graphic display containing a button that uses the **AlarmLogSendToODBC** command as its press action.

When the operator presses the button at run time, the contents of the alarm log file will be exported to the ODBC database.

The **AlarmLogSendToODBC** command exports only the records added to the alarm log files since the last export.

ODBC database schemas for HMI alarm logging

FactoryTalk View generates HMI tag alarm and data log information that can be stored in ODBC-compliant databases.

HMI tag alarm data stored in ODBC format uses the following table. For information about how alarm log files are named, see Alarm log file names on page 246.

This column	Contains	SQL data type	Length
TimeStmp	The time and date data was logged, in UTC (Coordinated Universal Time) format. **TIP**: If the ODBC database does not support SQL_TIMESTAMP, the data might be truncated. If you are using the Oracle ODBC driver, for example, to avoid truncating date and time data, use SQL_DATE instead of SQL_TIMESTAMP.	SQL_TIMESTAMP	Driver dependent
MilliTime	Millisecond time when data was logged.	SQL_INTEGER	3
TransType	The type of alarm that was generated. The transaction type is one of: SysAk (System Acknowledged) InAlm (In Alarm) OutAlm (Out of Alarm) InFlt (In Fault) OutFt (Out of Fault) Ack (Acknowledged) RmAck (Remote Acknowledged) SupOn (Suppress On) SupOf (Suppress Off) Remar (Remark)	SQL_VARCHAR, *or* SQL_CHAR	6
AlarmType	The number FactoryTalk View assigns to the transaction type: 0 for SysAk (System Acknowledged) 1 for InAlm (In Alarm) 2 for OutAlm (Out of Alarm) 3 for InFlt (In Fault) 4 for OutFt (Out of Fault) 5 for Ack (Acknowledged) 6 for RmAck (Remote Acknowledged) 7 for SupOn (Suppress On) 8 for SupOf (Suppress Off) 9 for Remar (Remark)	SQL_SMALLINT, *or* SQL_INTEGER	1
SrcArea	The area, in which the alarm occurred.	SQL_VARCHAR, *or* SQL_CHAR	80
TagName	The name of the tag or alarm event that caused the alarm.	SQL_VARCHAR, *or* SQL_CHAR	255
TagValue	The value of the tag at the time the alarm occurred.	SQL_DOUBLE, *or* SQL_INTEGER, *or* SQL_SMALLINT	20 (10 decimal places)
TagType	The type of the tag in alarm: 2 for Analog, 1 for Digital.	SQL_VARCHAR, *or* SQL_CHAR	1
ThreshVal	The threshold value.	SQL_DOUBLE, *or* SQL_FLOAT	20 (10 decimal places)
ThreshNum	The threshold number.	SQL_SMALLINT, *or* SQL_INTEGER	1
ThreshLabl	The threshold label.	SQL_VARCHAR, *or* SQL_CHAR	21

Severity	The alarm severity.	SQL_SMALLINT, *or* SQL_INTEGER	1
UserID	The name of the user (including domain name, if there is one) that initiated the alarm transaction. If the transaction was initiated by an HMI server, the user is NT Authority\System.	SQL_VARCHAR, *or* SQL_CHAR	38
Dscrptn	The message that is associated with the transaction (alarm) type, the log message for the alarm event, or the alarm log remark.	SQL_VARCHAR, *or* SQL_CHAR	132
SrcComputr	The name of the computer that initiated the command or action that caused the alarm.	SQL_VARCHAR, *or* SQL_CHAR	15
UserName	The name of the user (including domain name, if there is one) that initiated the alarm transaction. If the transaction was initiated by an HMI server, the user is NT Authority\System.	SQL_VARCHAR, *or* SQL_CHAR	255

Viewing HMI tag alarm log files

The HMI Tag Alarm Log Viewer shows the contents of the alarm log file. The contents of the file depend on how severities are defined in the Alarm Setup editor. For details, see Setting up alarm severities on page 233.

You can show alarm log files stored on the local computer or on a remote computer, and you can select the number of log files to be shown in the alarm log viewer. Each alarm log file is shown in a separate tab in the HMI Tag Alarm Log Viewer.

To open the HMI Tag Alarm Log Viewer:

Use one of these methods:

- In FactoryTalk View Studio or the FactoryTalk View SE Administration Console, from the Tools menu, select **HMI Tag Alarm Log Viewer**.
- Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **Tools** > **SE HMI Tag Alarm Log Viewer.**

To open the alarm log viewer Help:

- Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk View** > **Tools** > **SE HMI Tag Alarm Log Viewer Help**.

To show the alarm log viewer at run time:

1. In a graphic display, create a button that an operator can use to open the HMI Tag Alarm Log Viewer.
2. For the button's press action, type the following command:

```
AppStart "C:\Program Files\Rockwell Software\RSView
Enterprise\ AlarmLogViewer.exe"
```

You must include the quotation marks, because there are spaces in the parameter.

Tip: At run time, due to operating system rules, the alarm log 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. You can also use the programmatic solution documented in Answer ID 9041 in the Rockwell Automation Knowledgebase.

Alarm log file names

HMI tag alarm information is stored in file sets, in proprietary format.

When an alarm log file is created, it is named automatically. The name is based on the date the file was created and the type of data it contains.

The format for the name is YYYYMMDD*nz*.dat, where:

- YYYY are the four digits of the year.
- MM is the month.
- DD is the day.
- *n* is the sequence letter (*A*, *B*, *C*, and so on). This letter indicates the sequence, in which the files were created. You can have up to 26 files (*A* to *Z*) per day. At midnight, the sequence starts at *A* again.
- *z* is the file type: *L* (uppercase L) is for alarm.

If you are using short file names, or if the path where the log files are stored does not support long file names, the format for the name is YYMMDD*nz*.dat, where YY are the last two digits of the year.

Example: HMI tag alarm log file name

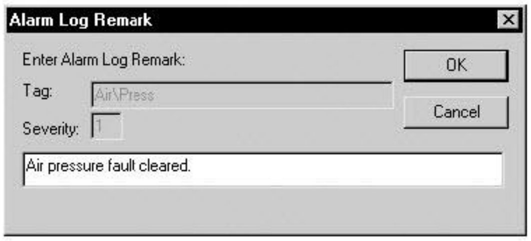
The log file named 20061015BL.dat was created in the year 2006, month 10, and day 15. The *B* indicates that this is the second file created that day. The *L* (uppercase L) indicates that this is an alarm log file.

Adding remarks to the alarm log file at run time

Use the **AlarmLogRemark** command to add information to the HMI tag alarm log file at run time.

Parameters for the command let you add a text comment, the alarm severity, and the tag name to the alarm log file. Other parameters let you prompt the operator for the text part of the remark, and log the remark to a printer, in addition to the alarm log file.

You cannot show alarm log remarks in HMI tag alarm summaries.



Tag: Air\Press

Severity: 1

Air pressure fault cleared.

To prompt the operator to type a remark for the alarm log file, use the */P* parameter with the **AlarmLogRemark** command.

At run time, only one Alarm Log Remark dialog box opens at a time, and the operator must respond to the dialog box before the next one is opened. For details about using AlarmLogRemark, see the FactoryTalk View Site Edition Help.

Tip: Alarm logging must be turned on before you can use the **AlarmLogRemark** command. To start alarm logging, run the **AlarmLogOn** command. For details, see the FactoryTalk View Site Edition Help.

Suppressing alarm logging

You can suppress alarm monitoring for specified HMI tags. This is useful for testing or performing repairs or maintenance on equipment.

To suppress alarm monitoring, use the **SuppressOn** command. To suppress all alarms for the specified tags, run the **SuppressOn** command before the **AlarmOn** command. For details about using these commands, see the FactoryTalk View Site Edition Help.

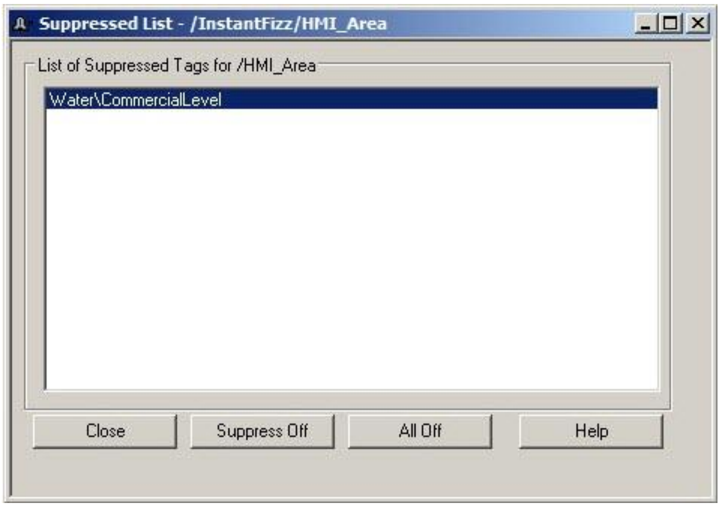
Use the Suppressed List to see which tags are not being monitored for alarms, and to turn alarm monitoring back on.

To open the Suppressed List:

1. In FactoryTalk View Studio or the FactoryTalk View SE Administration Console, in the Explorer window, open the HMI Tag Alarms folder.

2. Right-click the Suppressed List icon, and then select **Open**.

You can also double-click the Suppressed List icon, or drag the icon into the FactoryTalk View Studio workspace.


Suppressed List - /InstantFizz/HMI_Area
List of Suppressed Tags for /HMI_Area
Close
Suppress Off
All Off
Help

Water\CommercialLevel

For more information about the Suppressed List, click **Help**.

Tip: You cannot open the Suppressed List at run time, from a FactoryTalk View SE Client.

Suppressing alarm printing

To suppress alarm printing for all HMI tags, use the **AlarmPrintOff** command. To reset alarm printing, use the **AlarmPrintOn** command.

Using AlarmOn and AlarmPrintOff in the correct order

When the **AlarmOn** command is run, and HMI tag alarm monitoring is started, if an alarm log file and a printer are set up, alarms are logged to destinations. To stop logging to the printer, you can run the **AlarmPrintOff** command.

The order, in which AlarmOn and AlarmPrintOff run affects what happens at run time:

- If AlarmOn runs first, alarms are logged to the alarm log file and the printer until AlarmPrintOff is run.
- If AlarmPrintOff runs first, followed by AlarmOn, alarms are logged to the alarm log file, but not to the printer. To restart logging to the printer, you would have to run AlarmPrintOn.

The **AlarmPrintOff** and **AlarmPrintOn** commands are not retained across FactoryTalk View sessions. If you run AlarmPrintOff before AlarmOn in the current session, alarms will not print to the printer.

However, if you shut down FactoryTalk View SE, restart it, and run the **AlarmOn** command, alarms will be logged to both the alarm log file and the printer, until you run AlarmPrintOff.

Creating an HMI tag alarm summary

The HMI tag alarm summary is a table that shows the active alarms recorded in an HMI server (or servers). At run time, operators can use alarm summaries in FactoryTalk View graphic displays to view and interact with alarms.

You can create a new alarm summary object in a display, or use the pre-built alarm summary available in the HMI Tag Alarm Information graphic library.

To create a new HMI tag alarm summary object:

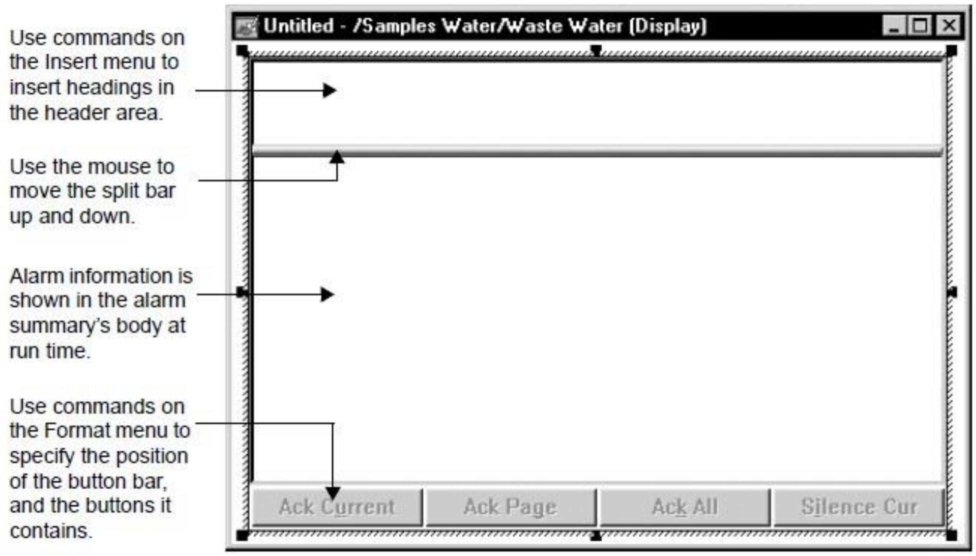
1. In FactoryTalk View Studio, in the Explorer window, right-click the Displays icon, and then select **Open**.
2. Select **Objects > Advanced > HMI Tag Alarm Summary.**
3. Drag the mouse diagonally on the display, to describe the approximate size of the alarm summary, and then release the mouse button.

   The object can be part of the display, or it can fill the entire graphic display.

For details about creating an HMI tag alarm summary, see the FactoryTalk View Site Edition Help.

The parts of an HMI tag alarm summary

A new HMI tag alarm summary object in a graphic display looks like this:



Use commands on the Insert menu to insert headings in the header area.

Use the mouse to move the split bar up and down.

Alarm information is shown in the alarm summary's body at run time.

Use commands on the Format menu to specify the position of the button bar, and the buttons it contains.

When setting up an alarm summary, you determine how many columns the table will have, how wide the columns will be, and what information the columns will contain.

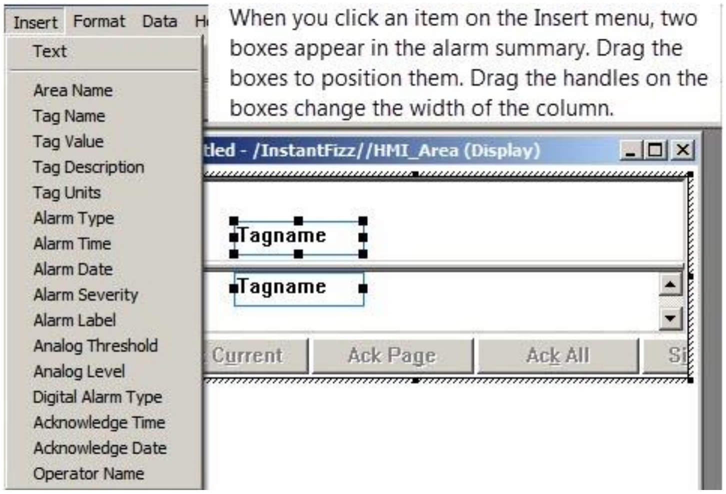
You can also specify the HMI tags and areas (in a network distributed application) that will provide information to the alarm summary.

Inserting headings

Use headings from the Insert menu (shown in the next illustration) to define what information is shown in the HMI tag alarm summary at run time.

To add a heading, click an item on the menu, and then click on the header area in the alarm summary. You can add any number of headings, and place them in any order.

Tip: Insert the Operator Name heading to include an operator’s name. When a tag goes into alarm, the Operator Name column will show the name of the Windows system account. When a user acknowledges an alarm, the column will show the name of that user.



When you click an item on the Insert menu, two boxes appear in the alarm summary. Drag the boxes to position them. Drag the handles on the boxes change the width of the column.

Tip: In local applications, the Insert menu does not contain an Area Name heading.

Choosing fonts

The header and body fonts in an HMI tag alarm summary can be different. For example, you might choose a larger font for the header and a smaller font for the body.

To change a font, from the **Format** menu, select **BodyFont** to show all fonts available on the computer.

To accommodate a larger font in the header area, increase the size of the area by dragging the split bar down.

Choosing colors and blink styles

To specify the color or blink style of an item in the HMI tag alarm summary, on the **Format** menu, select **Colors**. In the Colors dialog box, you can specify colors for:

- Severity 1 through 8 messages.
- Out of Alarm and Fault State messages.
- Header, body, and button areas.

You can also set up the blink style for:

- Severity 1 through 8 messages.
- Out of Alarm and Fault State messages.

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

Selecting buttons

In the Buttons dialog box, you can specify where in the HMI tag alarm summary the button bar will be positioned, and which buttons it will contain.

For descriptions of the individual buttons, see the FactoryTalk View Site Edition Help.

To select the buttons for an HMI tag alarm summary

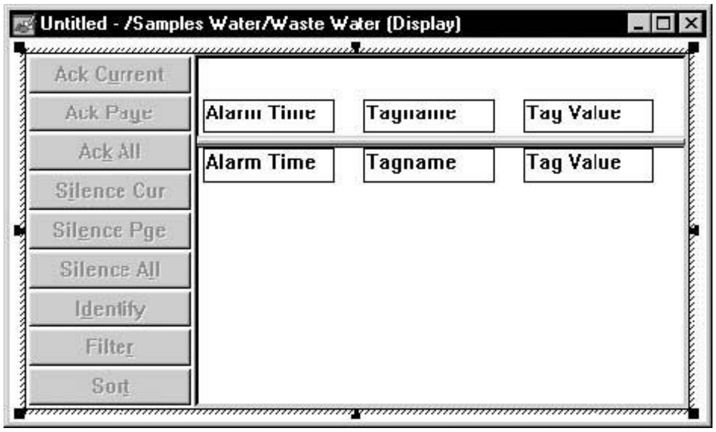
1. In the alarm summary editor, on the Format menu, click **Buttons**.
2. In the Buttons dialog box, select the check box beside each button that is to be in the alarm summary. To leave a button out, clear the check box.

   Tip: You can provide Filter and Sort buttons for the operator to use at run time. You can also set up sorting and filters for the alarm summary at design time. The result of the design-time filter or sort operation is retained each time the alarm summary is activated at run time. For more information, see Choosing the data to show on page 253.

To position the button bar

1. In the alarm summary editor, from the **Format** menu, select **Button Bar Position**.
2. Select a location.

In the following illustration, the button bar is on the left:



To change the button text

1. Double-click the button you want to change.

2. Type new text for the button.

Choosing the data to show

You can set up sorting and filters for an HMI tag alarm summary, to determine how and what alarm information is shown at run time. For network applications, you can also show full tag names in the alarm summary.

Showing area names with tag names

To show tag names with the names of the areas they belong to, from the Data menu, select **Display full tag name**.

This option is not available for local applications.

Sorting alarm information

Use Sort to specify the order, in which alarm information is shown. You can sort filtered or unfiltered data. For details about options in the Sort dialog box, click **Help**.

You cannot sort by area name in local applications.

Filtering alarm information

Use Filter to select information, that is, to specify information that you *do* want show in the alarm summary. Anything you don’t specify in the filter will not show in the alarm summary. For details about options in the Filter dialog box, click **Help**.

You cannot filter by area name in local applications.

Examples: Using area and tag names in filters

To collect all alarm information from the area called Powertrain, type:

```
/Powertrain::*
```

To collect alarm information from the current area for the tag name Coolant_East, type:

```
Coolant_East
```

To collect alarm information from the current area for all tags whose names begin with Coolant, type:

```
Coolant*
```

To collect all alarm information for the current area, type:

```
*
```

To collect alarms from specific areas for tag names that match a pattern, you can type:

```
/Powertrain::Coolant*
/Press::FlowValve*
```

To collect alarms from more than one HMI server, you can type:

```
/Powertrain::*
/Press::*
```

You can also use the Areas and Tags buttons to browse for area or tag names, instead of typing them.

Sorting and filtering at run time

By default, alarm information in an HMI tag alarm summary is sorted first by date and time, then by severity, then by area name (network distributed applications only), and then by tag name.

This means that alarms are presented in chronological order: if two or more alarms have the same time and date, they are presented in order of severity; if any alarms have the same time and date and the same severity, they are presented by tag name.

Add Filter and Sort buttons to the button bar, to let the operator change the type or order of information shown in the alarm summary. Changes made to filter and sort criteria at run time are not saved.

Using tag placeholders to specify alarm information sources

You can use tag placeholders to specify areas, subareas, or folder names, for tags you want to show in an HMI tag alarm summary.

A tag placeholder is the cross-hatch character (#) followed by a number from 1 to 500. A tag placeholder can also contain wildcard characters and folder names.

At run time, 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 about using tag placeholders in graphic displays, see Using placeholders to specify tag values on page 428.

Examples: Using tag placeholders to specify areas and tags

In these examples, the tag placeholder **#1** is used in various ways.

The tag placeholder can represent the name of an area, from which you want alarm information, for all tags the area contains (indicated by the wildcard):

```
/#1::*
```

The tag placeholder can represent the name of a subarea:

```
/Area1/#1/Area3::*
```

The tag placeholder can represent the name of a tag in the specified area:

```
Area1::#1
```

The tag placeholder can represent the name of a a folder in the specified area, from which you want alarm information for all tags the area contains (indicated by the wildcard):

```
Area1::Folder/#1/*
```

Running commands or custom programs for selected alarms

In an HMI tag alarm summary, use the **Execute** and **Identify** buttons to run a command, macro, or custom program, in response to an alarm.

Using the Execute button to run commands or programs

Use the **Execute** button to specify a command, macro or custom program for the operator to run with reference to the highlighted alarm.

You can also use parameters with the **Execute** command, to pass data derived from the highlighted alarm to the command or custom program.

For example, you might specify Remark as the **Execute** command, to transfer information about the highlighted alarm to the FactoryTalk Diagnostics log.

To append the tag value and the severity of the alarm, you would select Severity and Value from the list of parameters that can be used with the **Execute** command.

Tip: Severity and value are shown only for alarms of type IntoAlarm. For alarm types OutOfAlarm and IntoFault, the severity is always 0.

Appending Execute command parameters

Parameters are appended to the **Execute** command in the same order they are listed in the Execute Command dialog box.

You can substitute parameters into any position within a command by passing the parameters to a macro. For details about using parameters with macros, see Specifying parameters in a macro on page 646.

Execute command parameters are separated by a space, unless you select the check box, **Separate Parameters with commas**.

Tip: A space is not added automatically to the end of the command text. If you are using parameters, to insert a space between the end of a command, macro, or custom program and the first parameter, be sure to type the space at the end of the command text.

Adding area names to parameters

To include the area name with parameters for the **Execute** command, in the **Execute Command** dialog box, select the check box, **Insert Area name**.

At run time, the area of the alarm highlighted in the alarm summary will be inserted between the first and second word in the **Execute Command Text** field.

Since you cannot change the position of the area name, when setting up the **Execute** command, ensure that the syntax shown at the bottom of the dialog box is correct before you click **OK**.

Alternatively, you can use a macro to substitute the parameters into any position within the command.

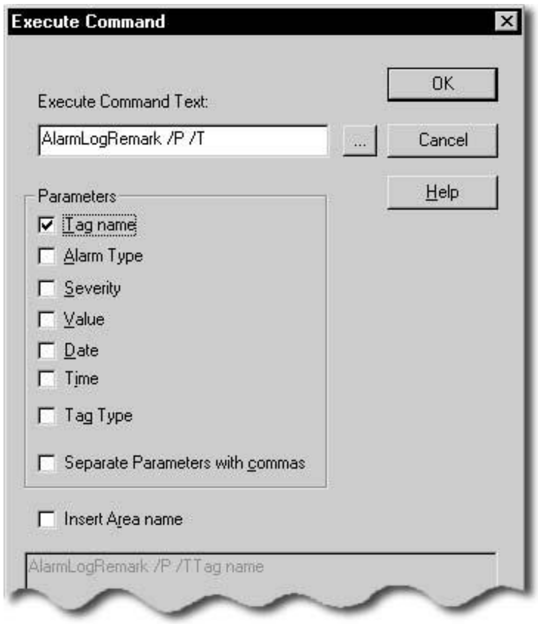
Tip: If you add the **Tag name** parameter to the **Execute** command, the area name is always included, even if the area is not shown in the alarm summary.

Using Execute to run a custom program

If you set up the Execute button to run a custom program, and you copy the alarm summary from one application to another, you must ensure that the custom program is available in the new application, otherwise the Execute button will not run the program.

Example: Using AlarmLogRemark with the Execute button

To prompt the operator to add a remark to the alarm log file, in the **Execute Command** dialog box, type **AlarmLogRemark /P /T** in the **Execute Command Text** box, and then select the **Tag name** parameter, as shown in the following illustration.



Execute Command Text: AlarmLogRemark /P /T

Parameters

☑ Tag name
□ Alarm Type
□ Severity
□ Value
□ Date
□ Time
□ Tag Type
□ Separate Parameters with commas

□ Insert Area name

AlarmLogRemark /P /TTag name

At run time:

- The */P* parameter for the **AlarmLogRemark** command shows a prompt for the operator, to enter a remark.
- The */T* parameter for the **AlarmLogRemark** command logs a string in the **Tagname** column of the alarm log file.
- The **Tag name** parameter for the **Execute** command records the name of the tag highlighted in the alarm summary, in the Tagname column of the alarm log file.

You cannot change the order, in which parameters are passed to the command line for the alarm summary Execute button.

To use the alarm summary Execute button with the **AlarmLogRemark** command, and have the tag name added correctly to the /T parameter, ensure that the /T parameter is positioned last on the command line.

For details about using the **AlarmLogRemark** command, see Adding remarks to the alarm log file at run time on page 247, or see the FactoryTalk View Site Edition Help.

Using the Identify button to run commands or programs

Use the **Identify** button to specify a command, macro or custom program that the operator can run when an HMI tag is in alarm, to provide additional information about the alarm.

For example, use the **Identify** button to run the **Display** command, to open a display that contains instructions about how to handle a motor that is running too fast.

For information about setting up the Identify button, see the FactoryTalk View Site Edition Help.

Starting and stopping HMI tag alarm monitoring

There are many ways to start and stop HMI tag alarm monitoring. Choose the way that works best for the application.

To provide the operator with a way start and stop HMI tag alarming, use FactoryTalk View commands and macros in conjunction with graphic objects. For details about FactoryTalk View command syntax, see the FactoryTalk View Site Edition Help.

Ways to start HMI tag alarm monitoring

To start alarm monitoring, use any of these methods:

- In the HMI Server Properties dialog box, in the Components tab, select the Alarming check box. Alarm monitoring will start the next time the HMI server runs, or when the HMI server's components are started manually.
- In the Macro editor, use the **AlarmOn** command in a macro. Specify the macro in the HMI Server Properties dialog box, in the Components tab. Alarm monitoring will start when the HMI server runs.
- In the Graphics editor, create a button object with AlarmOn as the press action. Alarm monitoring will start when the button is pressed.
- In the Graphics editor, attach touch animation to a graphic object, with AlarmOn as the action. Alarm monitoring will start when the object is touched.

- In the Events editor, type the **AlarmOn** command in the **Action** box for an event.
- At the command line in FactoryTalk View Studio (or the FactoryTalk View SE Administration Console), type AlarmOn and then press **Enter**.

  Tip: If the HMI server performing alarm monitoring is not in the home area, in the preceding examples, you must include the area name in the **AlarmOn** command. For details about using the **AlarmOn** command, see the FactoryTalk View Site Edition Help.

Ways to stop HMI tag alarm monitoring

To stop alarm monitoring only, use any of these methods:

- In the Graphics editor, create a button object with AlarmOff as the press action. Alarm monitoring will stop when the button is pressed.
- In the Graphics editor, attach Touch animation to a Graphic object, with AlarmOff as the action. Alarm monitoring will stop when the object is touched.
- In the Events editor, type the **AlarmOff** command in the **Action** box for an event.
- At the command line in FactoryTalk View Studio (or the FactoryTalk View SE Administration Console), type AlarmOff and then press **Enter**.

  Tip: If the HMI server performing alarm monitoring is not in the home area, in the preceding examples, you must include the area name in the **AlarmOff** command. For details about using the **AlarmOff** command, see the FactoryTalk View Site Edition Help.

You can also stop alarm monitoring by stopping all HMI server components running on the computer. For details, see Starting and stopping HMI server components manually on page 160.

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 alarm and event history logging.
- Setting up alarm and event displays.
- Using an alarm and event summary to monitor and respond to alarms.
- 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 Site Edition, you can set up a traditional alarm system, using HMI tags.

You can also use FactoryTalk Alarms and Events services, to centralize the distribution of device-based and tag-based alarm data to run-time clients, through FactoryTalk alarm servers that you add to a FactoryTalk View SE application.

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

There are advantages in using FactoryTalk Alarms and Events services for alarm monitoring and control, when compared with using traditional, HMI tag alarms.

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 and tag-based 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.

Choosing FactoryTalk device-based alarms

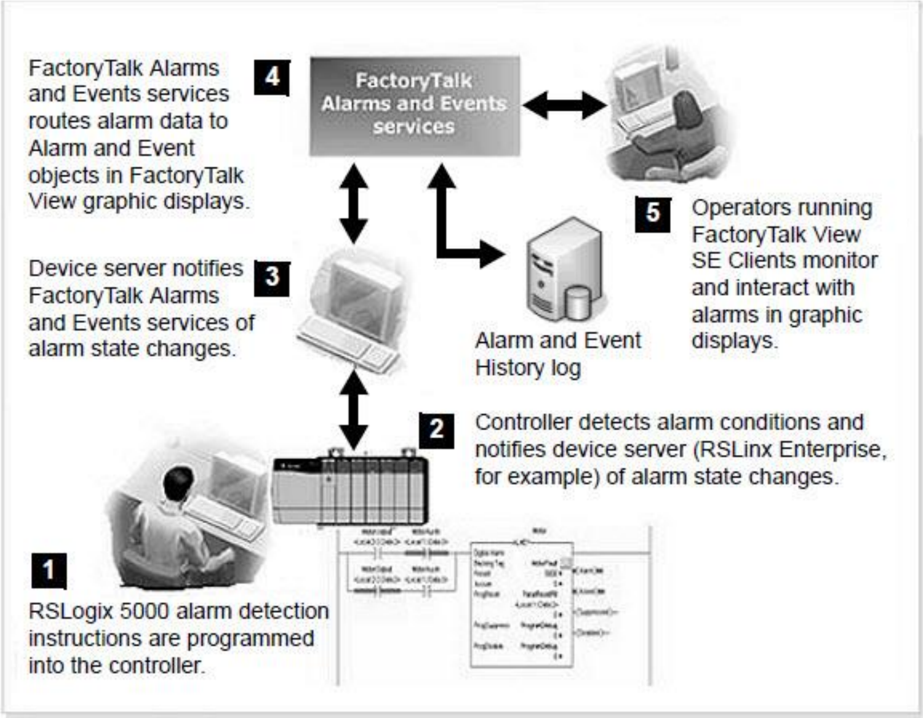
FactoryTalk device-based alarms are set up by programming alarm instructions directly into Logix5000 controllers that support built-in alarm detection.

Following are some reasons why you might choose device-based alarms, when compared with using HMI tag alarms.

Using device-based alarms	Using HMI tag alarms
Alarm detection instructions are programmed once, in the controller.	Alarm detection is programmed in the controller, and in FactoryTalk View SE. To set up HMI tag alarms, first you must create the HMI tags, which are mapped to tags in the controller. Then, in a separate step, you must add alarm conditions to the HMI tags you want to monitor. This duplication of programming effort can lead to errors.
Alarms are processed once, in the controller. Data polling is eliminated. This reduces network overhead and speeds up alarm detection.	Alarms are processed in the controller, and in FactoryTalk View SE. To monitor alarms, the HMI software polls tags in the controller. If alarms are detected, the HMI software notifies clients.
Alarm states are managed, processed, and preserved in the controller.	Alarm states are stored on the HMI server computer. If the computer is disconnected from the network, or if the HMI server fails or is restarted, alarm states might be lost.

Time stamps are applied in the controller. This makes time stamps more accurate. Also, the same time stamp is delivered to multiple servers and clients. **Tip:** If multiple controllers are used in an application, to ensure accurate time stamps, synchronize the controller clocks.	Time stamps are applied on the HMI server computer. To ensure that time stamps are accurate and consistent, clocks at server and client computers must always be synchronized.

The following illustration shows how device-based alarm information flows, from a Logix5000 controller, to operators running FactoryTalk View SE clients.


FactoryTalk Alarms and Events services
Alarm and Event History log

1 RSLogix 5000 alarm detection instructions are programmed into the controller.

2 Controller detects alarm conditions and notifies device server (RSLinx Enterprise, for example) of alarm state changes.

3 Device server notifies FactoryTalk Alarms and Events services of alarm state changes.

4 FactoryTalk Alarms and Events services routes alarm data to Alarm and Event objects in FactoryTalk View graphic displays.

5 Operators running FactoryTalk View SE Clients monitor and interact with alarms in graphic displays.

FactoryTalk View SE Clients receive device-based alarm data by way of Rockwell Automation Device Servers (RSLinx Enterprise) that you add to a FactoryTalk View SE application.

For more information, see Working with Rockwell Automation Device Servers on page 279.

Tip: RSLogix 5000 version 16 or later, is required to program alarm instructions into a Logix5000 controller. For a complete list of controllers in the Logix5000 family, that support FactoryTalk Alarms and Events services, see the FactoryTalk Alarms and Events Help.

Choosing FactoryTalk 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 Logix5000 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 more information, see Working with Tag Alarm and Event Servers on page 281.

About traditional HMI tag alarms

HMI tag alarms are set up by specifying alarm conditions for HMI tags in a FactoryTalk View SE Server's tag database.

If your application uses HMI tags for other purposes, and you want to monitor these tags for alarms, use a traditional HMI tag alarm system.

FactoryTalk View SE Clients receive HMI tag alarm information by way of the FactoryTalk View SE Servers (also called HMI servers) that contain the HMI tags.

HMI tag alarm data is not managed by FactoryTalk Alarms and Events services. To monitor and respond to HMI tag alarms, you must use the HMI tag alarm displays and logs available in FactoryTalk View SE.

For details, see Setting up HMI tag alarms. on page 217

Finding more information

This section provides an overview of 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 (RSLinx Enterprise), 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 run-time 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 the FactoryTalk View Site Edition Help.

Tip: This manual does not contain information about using RSLogix 5000 to program alarm instructions into Logix5000 controllers. For details, see the RSLogix 5000 documentation, or see the FactoryTalk Alarms and Events System Configuration Guide.

Opening the FactoryTalk Alarms and Events Help

For comprehensive information about FactoryTalk Alarms and Events, including detailed setup instructions, see the FactoryTalk Alarms and Events Help.

To open the Help, use one of these methods:

- Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **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.

Opening the 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 two types of alarm servers:

- **Rockwell Automation Device Servers (RSLinx Enterprise)** provide access to alarms detected in Logix5000 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 Logix5000 controllers that *do* support device-based alarms, if you prefer not to set up built-in alarm detection.

For information about setting up alarm servers in a FactoryTalk View SE application, see:

- Working with Rockwell Automation Device Servers on page 279.
- Working with Tag Alarm and Event Servers on page 281.

Level alarms

A level alarm compares an analog value against predefined limits (also called thresholds) and triggers an alarm when a limit is exceeded. For information about setting up tag-based level alarms, see Creating tag-based level alarms on page 284.

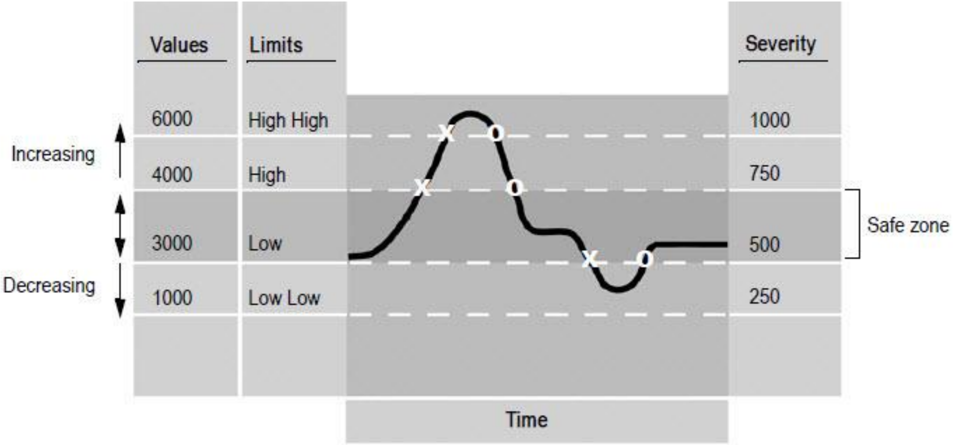
Level alarm limits

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 run time.

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.

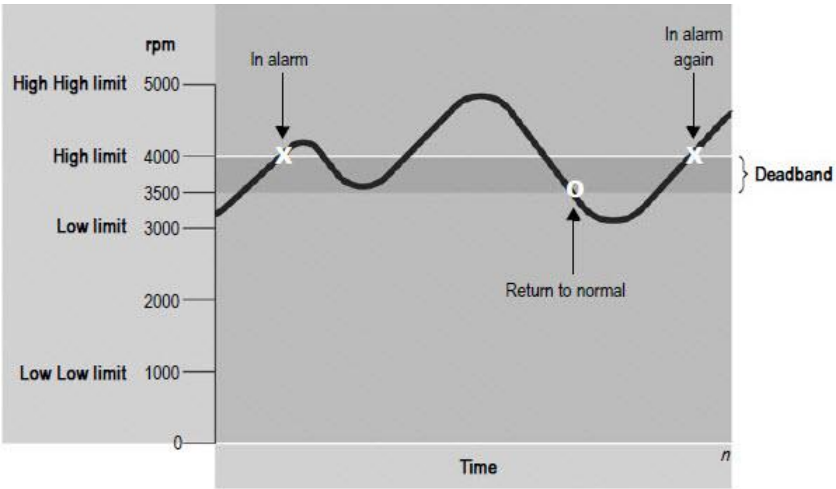
Deadband

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 re-triggering 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 the 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.

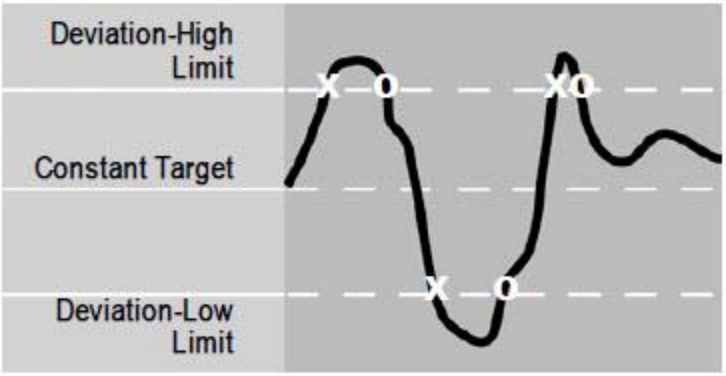
Deviation alarms

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.



For information about setting up tag-based deviation alarms, see Creating tag-based deviation alarms on page 285.

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.

Tip: If you want to set up digital change-of-state alarms, use HMI digital tags. For more information, see Setting up HMI tag alarms on page 217.

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.

For information about setting up tag-based digital alarms, see Creating tag-based digital alarms on page 283.

FactoryTalk alarm and event displays

To monitor and respond to FactoryTalk device-based and tag-based alarms at run time, use the following FactoryTalk Alarm and Event objects, embedded in FactoryTalk View SE graphic displays.

Tip: FactoryTalk Alarm and Event objects do not show HMI tag alarms. To monitor and respond to HMI tag alarms, you must use the HMI tag alarm objects available in FactoryTalk View SE. For details, see Setting up HMI tag alarms on page 217.

Alarm and event summary

The Alarm and Event Summary shows alarm information from Rockwell Automation Device Servers (RSLinx Enterprise) and Tag Alarm and Event Servers that support FactoryTalk Alarms and Events.

Use the alarm and event summary to view, acknowledge, suppress, and disable the alarms on display. The alarm and event summary also provides a detail 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 293.

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 296.

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, and 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 306.

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 310.

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 277.

If priority values are defined at an alarm server, the server-defined priorities take precedence over the system-wide priorities.

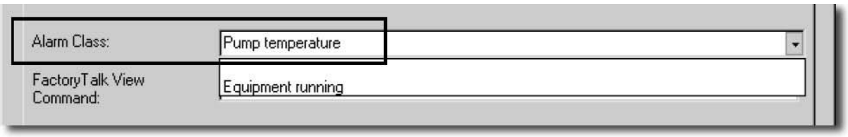
You can set up server-defined priorities in the Properties dialog box for the Rockwell Automation Device Server or Tag Alarm and Event Server. For details, see the FactoryTalk Alarms and Events Help.

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.



Alarm Class: Pump temperature

FactoryTalk View Command: Equipment running

Each time you create a new 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 the alarm condition has this state
🔔	In Alarm and Unacknowledged
🔔✓	In Alarm and Acknowledged
🔔	Normal and Unacknowledged

For more information, see Using an alarm summary to monitor and respond to alarms on page 299.

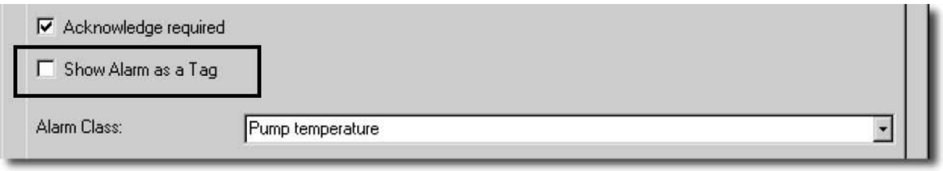
Alarm tags

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 run time, 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 check box, Show Alarm as a Tag (shown in the next illustration).



☑ Acknowledge required

☐ Show Alarm as a Tag

Alarm Class: Pump temperature

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 312.

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 287.

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 288.

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 audit messages logged 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 diagnostic messages in the Diagnostics Viewer.

- **Historical alarm and event messages** record all the alarm activity that occurs at an alarm server at run time.

  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 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 291.

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 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 information, see Setting up system-wide alarm and event policies on page 277.

2. If the application is to use device-based alarms, in RSLogix 5000, program alarm instructions into Logix5000 controllers that support built-in alarm detection.

   For information, see the RSLogix 5000 documentation.

3. To let run-time clients receive device-based alarm data, add one or more Rockwell Automation Device Servers (RSLinx Enterprise) to the application, and set up the servers to support FactoryTalk Alarms and Events.

   For information, see Working with Rockwell Automation Device Servers on page 279.

4. If the application is to use tag-based alarms, add one or more Tag Alarm and Event Servers.

   For information, see Working with Tag Alarm and Event Servers on page 281.

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 information, see Setting up FactoryTalk tag-based alarms on page 282.

6. 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 run time.

   For information, see Setting up alarm and event displays on page 292.

7. Set up historical alarm and event logging.

   For information, see Setting up alarm and event history logging on page 291.

8. Set up FactoryTalk View SE Clients to run the graphic displays.

   For 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.

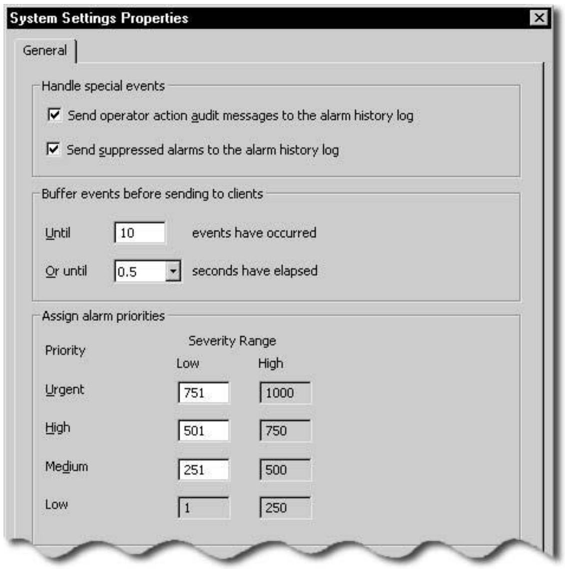
To modify system-wide alarm settings

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 (shown in the following illustration), modify special events, event buffering, and alarm priority settings, as needed.

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.


System Settings Properties

Handle special events

☑ Send operator action audit messages to the alarm history log

☑ Send suppressed alarms to the alarm history log

Buffer events before sending to clients

Until 10 events have occurred

Or until 0.5 seconds have elapsed

Assign alarm priorities

Priority	Severity Range Low	Severity Range High
Urgent	751	1000
High	501	750
Medium	251	500
Low	1	250

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

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 permissions you might require.

For an overview of FactoryTalk Security services, see Setting up security on page 85. For details, see FactoryTalk Security Help.

Working with Rockwell Automation Device Servers

To enable FactoryTalk View SE Clients to receive and display FactoryTalk alarms detected in Logix5000 controllers, do the following:

1. Add one or more Rockwell Automation Device Servers (RSLinx Enterprise) to a FactoryTalk View SE network distributed application.

   In a local station application, you can only add one RSLinx Enterprise 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 RSLinx Enterprise, to the Logix5000 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 RSLogix 5000, to program alarm instructions into Logix5000 controllers, nor does it describe how to set up redundancy for these controllers. For details, see the product documentation.

To add an RSLinx Enterprise 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 (RSLinx Enterprise)**.

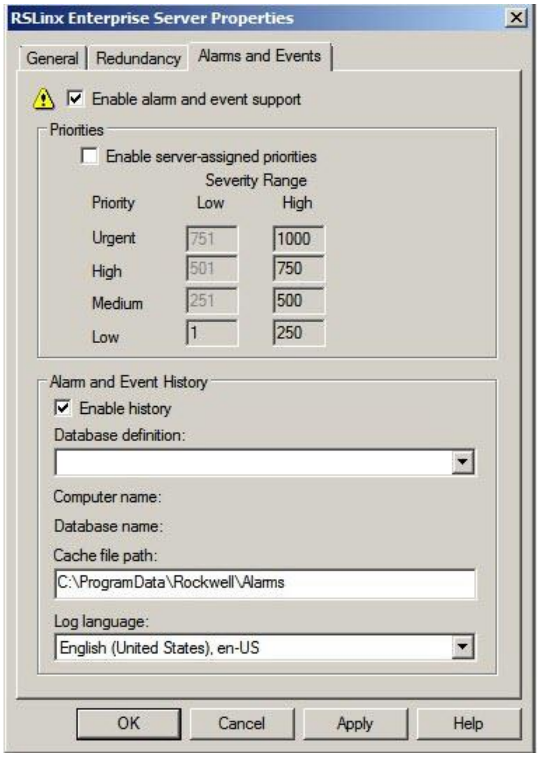
2. Provide a name and an optional description, and then specify the name of the computer hosting RSLinx Enterprise.

Setting up support for FactoryTalk Alarms and Events

After you add an RSLinx Enterprise 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 RSLinx Enterprise 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 check box, Enable alarm and event support.



☑ Enable alarm and event support

Priorities

□ Enable server-assigned priorities

Priority	Severity Range Low	High
Urgent	751	1000
High	501	750
Medium	251	500
Low	1	250

Alarm and Event History

☑ Enable history

Database definition:

Computer name:

Database name:

Cache file path:
C:\ProgramData\Rockwell\Alarms

Log language:
English (United States), en-US

To enable alarm and event history logging on the computer running the device server, select the check box, 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 291.

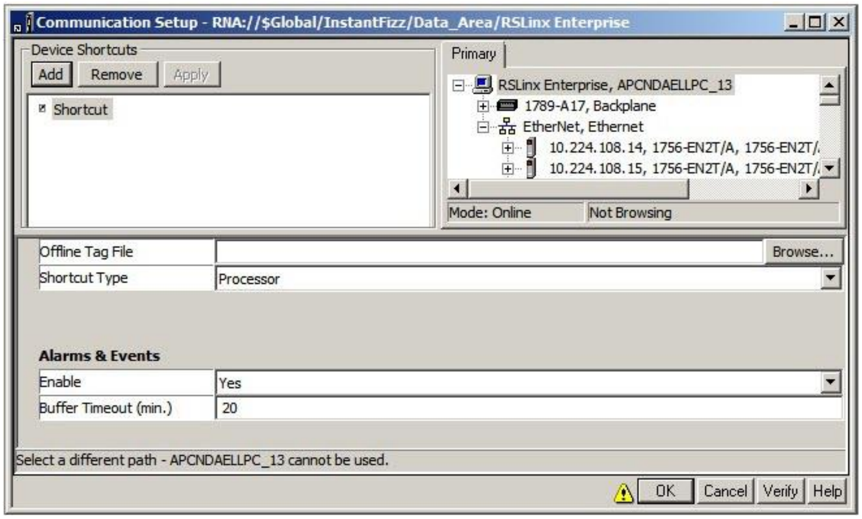
Optionally, you can set up server-assigned priority values for the alarm server. To modify the severity range, change values in the Low boxes only. Priorities defined at an alarm server override the system-wide settings held at the FactoryTalk Directory.

Specifying a device-based alarm source

In RSLinx Enterprise, 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 RSLinx Enterprise server icon, and then double-click **Communication Setup.**


Communication Setup - RNA://$Global/InstantFizz/Data_Area/RSLinx Enterprise
Device Shortcuts
Add
Remove
Apply
Shortcut
Primary
RSLinx Enterprise, APCNDAELLPC_13
1789-A17, Backplane
EtherNet, Ethernet
10.224.108.14, 1756-EN2T/A, 1756-EN2T/.
10.224.108.15, 1756-EN2T/A, 1756-EN2T/.
Browse...
OK
Cancel
Verify
Help

Mode: Online Not Browsing

Offline Tag File	
Shortcut Type	Processor

Alarms & Events

Enable	Yes
Buffer Timeout (min.)	20

Select a different path - APCNDAELLPC_13 cannot be used.

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 282.

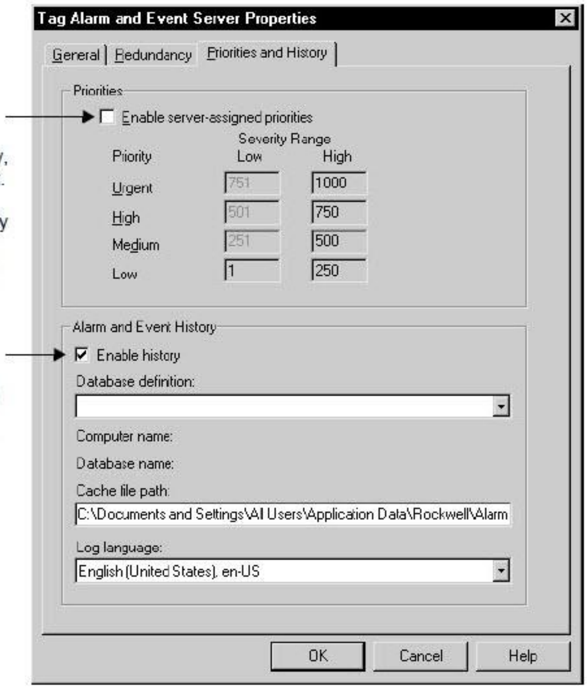
To add a Tag Alarm and Event Server to a network distributed application:

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 check box, Enable history, as shown in the next illustration.



To override priority settings at the FactoryTalk Directory, select this check box.

To modify the severity range for a priority, change the value in the Low boxes.

To enable history logging, select this check box, and then select an existing database definition.

Tag Alarm and Event Server Properties

General | Redundancy | Priorities and History

Priorities

□ Enable server-assigned priorities

Priority	Severity Range Low	Severity Range High
Urgent	751	1000
High	501	750
Medium	251	500
Low	1	250

Alarm and Event History

☑ Enable history

Database definition:

Computer name:

Database name:

Cache file path: C:\Documents and Settings\All Users\Application Data\Rockwell\Alarm

Log language: English (United States), en-US

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 291.

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 Logix5000 controllers that do support device-based alarms.

In the Alarm and Event Setup editor, you can create and modify three types of alarms:

- **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.

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.

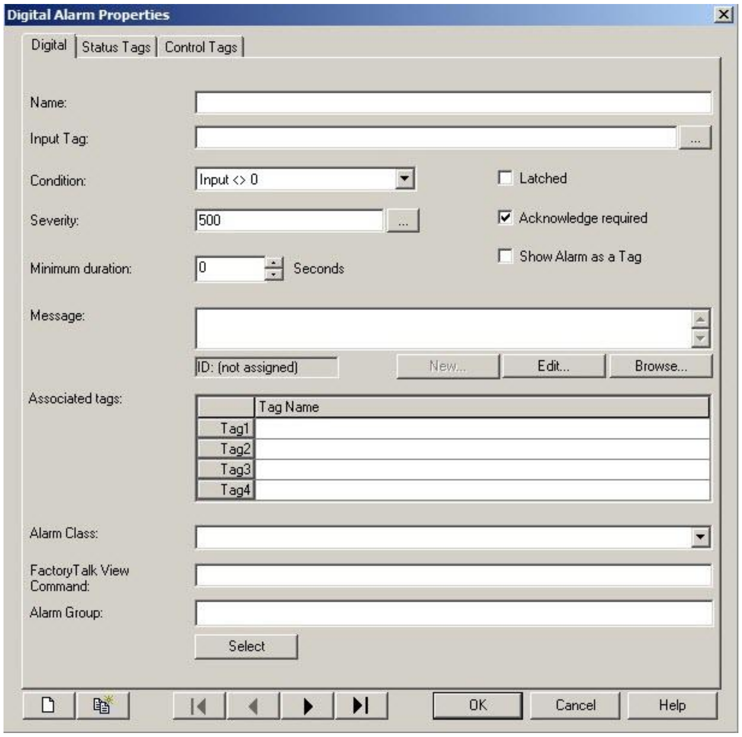
2. Double-click **Alarm and Event Setup**, or right-click the icon, and select **Open**.

Creating tag-based digital alarms

In the Digital tab of the Alarm and Event Setup editor, you can view all the digital alarms set up for this Tag Alarm and Event Server, create a new digital alarm, modify or delete an existing digital alarm, and refresh the list of digital alarms.

To create a new digital alarm:

1. In the Alarm and Events Setup editor, click the Digital tab, and then click **New** on the editor's toolbar.



Digital Alarm Properties

Name:

Input Tag:

Condition: Input <> 0

Severity: 500

Minimum duration: 0 Seconds

☐ Latched

☑ Acknowledge required

☐ Show Alarm as a Tag

Message:

ID: (not assigned)

Associated tags:

	Tag Name
Tag1	
Tag2	
Tag3	
Tag4	

Alarm Class:

FactoryTalk View Command:

Alarm Group:

2. In the Digital tab, set up properties such as the alarm's name, input tag, condition, and severity, whether acknowledgment is required, and whether the alarm is latched.

3. In the Status Tags tab, set up status tags for the alarm.

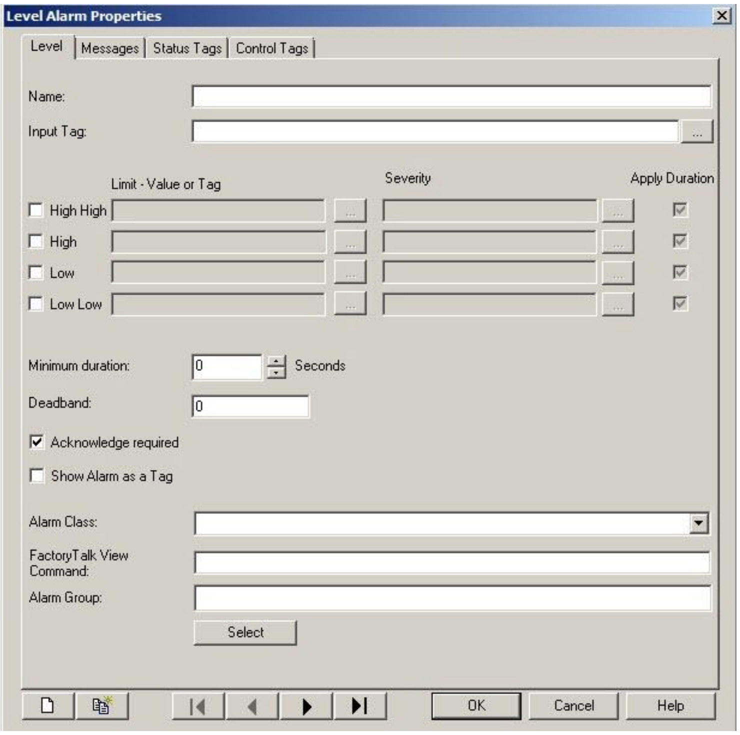
For details about options in the Digital Alarm Properties dialog box, click **Help**.

Creating tag-based level alarms

In the Level tab of the Alarm and Event Setup editor, you can view all the level alarms set up for this Tag Alarm and Event Server, create a new level alarm, modify or delete an existing level alarm, and refresh the list of level alarms.

To create a new level alarm:

1. In the Alarm and Events Setup editor, click the Level tab, and then click **New** on the editor's toolbar.



Minimum duration: 0 Seconds

Deadband: 0

☑ Acknowledge required

☐ Show Alarm as a Tag

2. In the Level tab, set up properties such as the alarm's name, input tag, level, limit type, limit value, and severity, and whether acknowledgment is required.

3. In the Messages and Status Tags tabs, set up messages and status tags for the alarm.

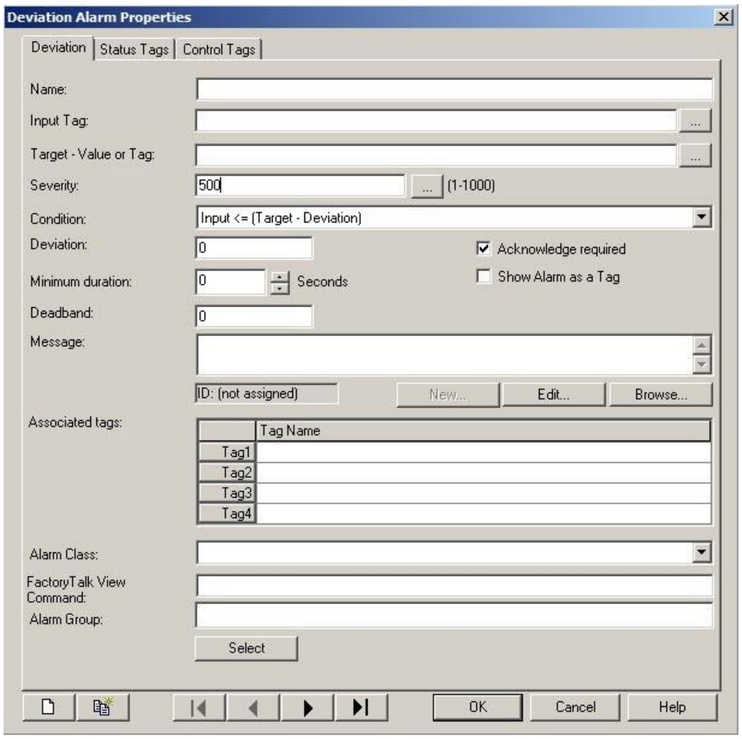
For details about options in the Level Alarm Properties dialog box, click **Help**.

Creating tag-based deviation alarms

In the Deviation tab of the Alarm and Event Setup editor, you can view all the deviation alarms set up for this Tag Alarm and Event Server, create a new deviation alarm, modify or delete an existing deviation alarm, and refresh the list of deviation alarms.

To create a new deviation alarm:

1. In the Alarm and Events Setup editor, click the Deviation tab, and then click **New** on the editor’s toolbar.



Name:

Input Tag:

Target - Value or Tag:

Severity: 500 (1-1000)

Condition: Input <= (Target - Deviation)

Deviation: 0

☑ Acknowledge required

☐ Show Alarm as a Tag

Minimum duration: 0 Seconds

Deadband: 0

Message:

ID: (not assigned)

Associated tags:

	Tag Name
Tag1	
Tag2	
Tag3	
Tag4	

Alarm Class:

FactoryTalk View Command:

Alarm Group:

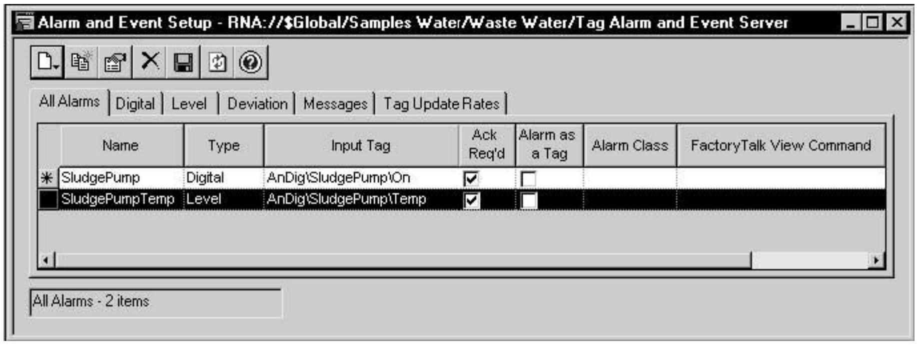
2. In the Deviation tab, set up properties such as the alarm’s name, input tag, target type, target value, deviation value, and severity, and whether acknowledgment required.

3. In the Status Tags tab, set up status tags for the alarm.

For details options in the Deviation Alarm Properties dialog box, click **Help**.

Viewing all tag-based alarms

The All Alarms tab of the Alarm and Event Setup editor shows all the digital, level, and deviation alarms set up for this Tag Alarm and Event Server.



Alarm and Event Setup - RNA://$Global/Samples Water/Waste Water/Tag Alarm and Event Server

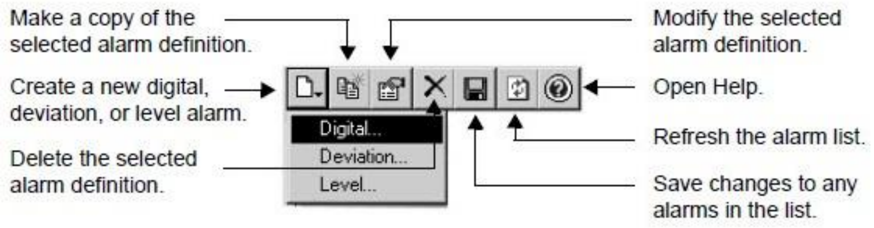
	Name	Type	Input Tag	Ack Req'd	Alarm as a Tag	Alarm Class	FactoryTalk View Command
*	SludgePump	Digital	AnDig\SludgePump\On	☑	☐		
	SludgePumpTemp	Level	AnDig\SludgePump\Temp	☑	☐		

All Alarms - 2 items

The properties shown for each alarm can include the alarm name, input tag, condition, and severity, and whether acknowledgment is required for the alarm. For details, click **Help** in the All Alarms tab.

For information about the properties that digital, level, and deviation alarms have in common, see the FactoryTalk Alarms and Events Help.

Use buttons on the Alarm and Event Setup editor's toolbar, shown in the following illustration, to perform actions quickly in each of the editor's tabs.



Make a copy of the selected alarm definition.
Create a new digital, deviation, or level alarm.
Delete the selected alarm definition.
Digital...
Deviation...
Level...
Modify the selected alarm definition.
Open Help.
Refresh the alarm list.
Save changes to any alarms in the list.

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 Alarm and Event Setup editor, click the Digital, Level, or Deviation tab.

2. Create a new alarm, or select the alarm you want to assign status tags to.

3. 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.

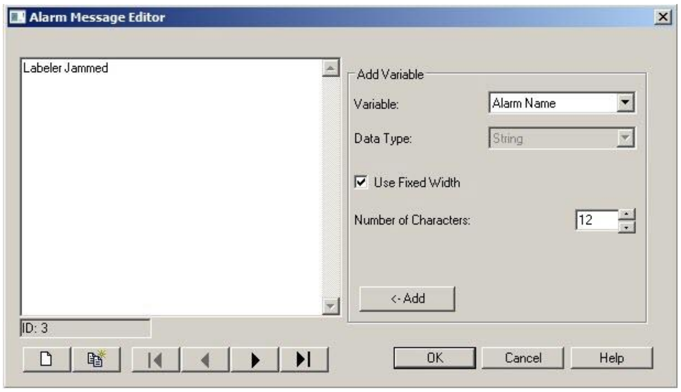
You can create new messages or modify existing ones using the Alarm Message Editor.

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.



Labeler Jammed

ID: 3

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 an 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 run time, the message will show the value of the tag and the alarm severity, at the times 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.

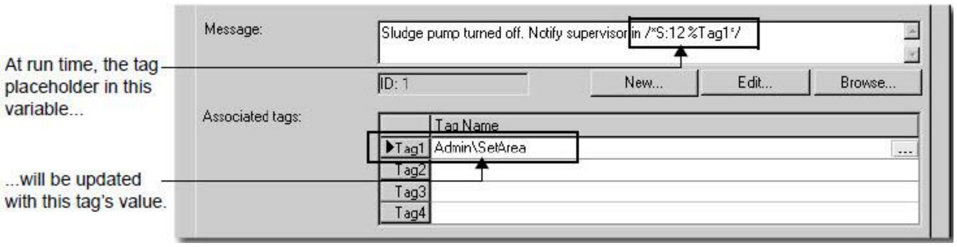
For information about using variables in other parts of a FactoryTalk View SE application, see Creating embedded variables on page 575.

Using tag placeholders in alarm messages

Tag placeholders are another type of variable you can add to an alarm message, to include specific values in the message at run time.

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.



At run time, the tag placeholder in this variable...

...will be updated with this tag's value.

Message: Sludge pump turned off. Notify supervisor in /*S:12 %Tag1*/

ID: 1

Associated tags:

	Tag Name
▶Tag1	Admin\SetArea
Tag2	
Tag3	
Tag4	

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 393.

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 alarm and event history logging

FactoryTalk Alarms and Events generates messages to record the alarm activity that occurs at tag-based and device-based 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.

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, "Defining an alarm and event log database," next.
- 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 292.
- Set up alarm and event log viewers to display logged data at run time. For more information, see Viewing alarm and event history logs on page 310.

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 an alarm server will use, in the server's Properties dialog box. For more information, "Enabling alarm and event history logging," next.

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 (RSLinx Enterprise), 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 282.

Tip: For an RSLinx Enterprise 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 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 run time, 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," next.

- 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 296.
- 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 310.
- 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 306.

For detailed setup instructions, see the FactoryTalk Alarms and Events Help.

About the Alarm and Event graphic library

The Alarm and Event graphic library contains a pre-configured FactoryTalk alarm and event summary, and buttons you can use to acknowledge FactoryTalk alarms.

Use the objects in the library as they are, or drag them into a graphic display, and then modify them to suit the needs of your application.

To open the Alarm and Event library:

In FactoryTalk View Studio, open the Graphics folder, double-click the Libraries icon, and then double-click **Alarm and Event**. You can also drag the Alarm and Event library into the FactoryTalk View Studio workspace.

Setting up an alarm and event summary

When setting up an alarm and event summary, you specify which alarm information will be shown at run time, 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 new 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 click on the Graphic editor's toolbar.

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**).

   Following is an overview of alarm and event summary properties. For details about options in the Alarm and Event Summary Properties dialog box, click **Help**.

Setting up the overall appearance of the summary display

In the Appearance tab, set up which parts of the alarm and event summary to show at run time. For example, you can determine whether the summary will include a vertical scroll bar, the details pane, or a toolbar.

You can also set up fonts, text color, and background color, for various parts of the summary display. For example, you can set up a font, text color, and icon size for the toolbar.

Choosing the columns and toolbar buttons to show

In the Columns tab, set up which columns to show in the alarm and event summary, and in what order. You can also set up the heading text, and the alignment and format of information shown in each column. Use the Sample column to preview your selections.

In the Toolbar tab, you can set up which buttons will show in the alarm and event summary's toolbar. You can also set up the caption, tooltip, and format of each button.

Choosing the status bar contents

In the Status Bar tab, set up which information to display in the status bar, and in what order. You can also set up tooltips for the status bar.

Setting up event subscriptions

In the Event Subscription tab, specify the areas and alarm sources from which the alarm and event banner will obtain data.

You can subscribe to events from any area or alarm source in an application, as long as the source is set up to support FactoryTalk Alarms and Events.You can also filter event subscriptions by priority (Urgent, High, Medium, or Low).

Once you determine where the alarm information will come from, you can decide how to filter and sort the information, to control what appears in the alarm and event summary. By default, all information generated at the alarm source is shown.

Tip: When subscribing to events for an alarm banner, keep in mind that the purpose of the alarm and event banner is to make operators aware of the most serious alarms in a process or area.

Setting up filter and sort criteria

In the Display Filter tab, set up filters for the alarm and event summary, that the operator can apply at run time.

You can build filter criteria using the event fields provided. You can also select an initial display filter for the alarm and event summary.

Tip: Create filters based on what you want to *include* in the alarm and event summary, not what you want to exclude. Anything you don’t specify in the filter will *not* be shown at run time.

In the Sort tab, specify which criteria will be used to sort information in the alarm and event summary, and in what order (ascending or descending).

For information about filtering and sorting information in an alarm an event summary at run time, see Using an alarm summary to monitor and respond to alarms on page 299.

Choosing colors and blink styles for the alarm states

In the States tab, set up text and background colors for the different alarm states (In Alarm Unacknowledged, In Alarm Acknowledged, and Normal Unacknowledged), and the priorities (Urgent, High, Medium, and Low) for each state.

You can also select whether alarm messages blink in the alarm and event summary, and at what rate (slow, medium, or fast). Use the Sample column to preview your selections. To test the blink rate, click **Test Rates.**

Determining run-time behavior and appearance

In the Behavior tab, determine to what extent the operator can interact with the alarm and event summary at run time. For example, determine whether the operator can resize columns, or sort information by clicking on column headings.

In the Common tab, set up the size and position of the alarm and event summary, and give the object a name. You can also determine whether the summary will present a tooltip, or show a highlight when it has focus in the graphic display.

Setting up an alarm and event banner

When setting up an alarm and event banner, you specify which alarm information will be shown at run time, 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 run-time client window on page 299.

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 new 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 click on the Graphic editor's toolbar.

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).

Following is an overview of alarm and event banner properties. For details about options in the Alarm and Event Banner Properties dialog box, click **Help**.

Setting up the appearance of the banner display

In the General tab, set up the appearance of the alarm list and the status bar. For example, you can set up the font, text color, and number of rows in the alarm list.

You can also determine the behavior, when an operator interacts with the banner, such as whether double-clicking a row causes an action.

Choosing the columns and toolbar buttons to show

In the Columns tab, set up which columns to show in the alarm and event banner, and in what order. You can also set up the column width, and the alignment and format of information shown in each column. Use the Sample column to preview your selections.

Choosing the status bar contents

In the Status Bar tab, set up which information to display in the status bar, and in what order. You can also set up tooltips for the status bar.

Setting up event subscriptions

In the Event Subscription tab, specify the areas and alarm sources from which the alarm and event banner will obtain data.

You can subscribe to events from any area or alarm source in an application, as long as the source is set up to support FactoryTalk Alarms and Events.You can also filter event subscriptions by priority (Urgent, High, Medium, or Low).

Once you determine where the alarm information will come from, you can decide how to filter and sort the information, to control what appears in the alarm and event summary. By default, all information generated at the alarm source is shown.

Tip: When subscribing to events for an alarm banner, keep in mind that the purpose of the alarm and event banner is to make operators aware of the most serious alarms in a process or area.

Choosing colors, sound, and blink styles for the alarm states

In the States tab, set up text and background colors for the different alarm states (In Alarm Unacknowledged, In Alarm Acknowledged, and Normal Unacknowledged), and the priorities (Urgent, High, Medium, and Low) for each state.

You can also select whether alarm messages blink in the alarm and event banner, or sound an alarm bell, and at what rate.

Use the Sample column to preview your selections. To test blink and audible notification rates, click **Test Rates.**

Determining run-time behavior and appearance

In the Common tab, set up the size and position of the alarm and event banner, and give the object a name. You can also determine whether the banner will present a tooltip, or show a highlight when it has focus in the graphic display.

About the alarm and event banner at run time

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 run-time client window

At run time, 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 445.

2. In the FactoryTalk View SE Client Wizard, in the Components window, 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.

Using an alarm summary to monitor and respond to alarms

The alarm and event summary is a table that displays active alarm notifications received from the FactoryTalk alarm servers in a FactoryTalk View SE application.

Alarm and event summaries are hosted in graphic displays created in FactoryTalk View Studio, and run in the FactoryTalk View SE Client.

A single alarm and event summary can subscribe to alarm notifications from multiple alarm servers in an application.

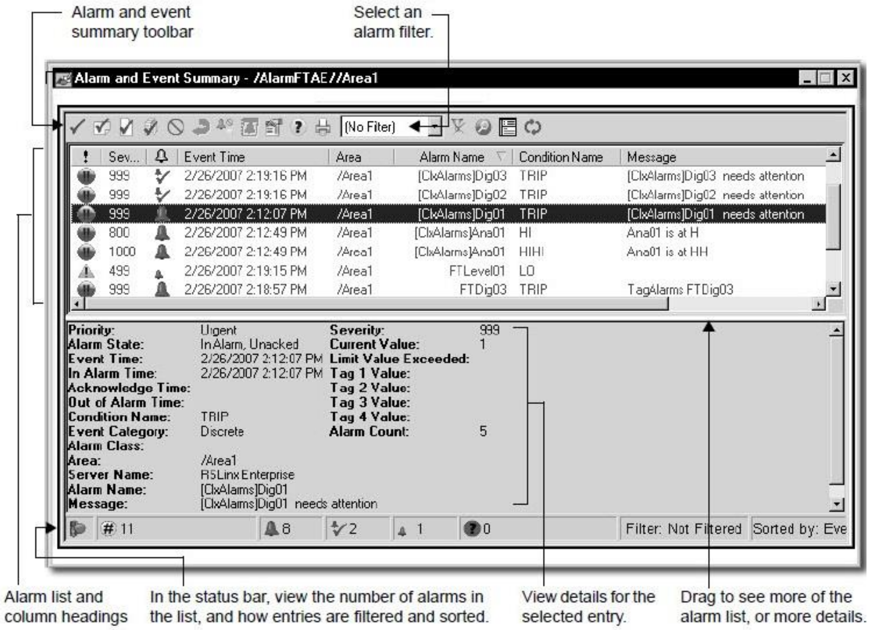
At run time, an operator can use an alarm and event summary to monitor alarm activity, and to acknowledge, suppress, disable, or reset active alarms.

For more information about these actions, see subsequent sections in this chapter.

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 312.

The parts of an alarm and event summary

The following illustration shows an alarm and event summary at run time.



Alarm and event summary toolbar

Select an alarm filter.

Alarm and Event Summary - /AlarmFTAE//Area1

(No Filter)

!	Sev...	🔔	Event Time	Area	Alarm Name	Condition Name	Message
●	999	✓	2/26/2007 2:19:16 PM	/Area1	[ClxAlarms]Dig03	TRIP	[ClxAlarms]Dig03 needs attention
●	999	✓	2/26/2007 2:19:16 PM	/Area1	[ClxAlarms]Dig02	TRIP	[ClxAlarms]Dig02 needs attention
●	999	🔔	2/26/2007 2:12:07 PM	/Area1	[ClxAlarms]Dig01	TRIP	[ClxAlarms]Dig01 needs attention
●	800	🔔	2/26/2007 2:12:49 PM	/Area1	[ClxAlarms]Ana01	HI	Ana01 is at H
●	1000	🔔	2/26/2007 2:12:49 PM	/Area1	[ClxAlarms]Ana01	HIHI	Ana01 is at HH
⚠	499	🔔	2/26/2007 2:19:15 PM	/Area1	FTLevel01	LO	
●	999	🔔	2/26/2007 2:18:57 PM	/Area1	FTDig03	TRIP	TagAlarms FTDig03

Priority: Urgent
Alarm State: In Alarm, Unacked
Event Time: 2/26/2007 2:12:07 PM
In Alarm Time: 2/26/2007 2:12:07 PM
Acknowledge Time:
Out of Alarm Time:
Condition Name: TRIP
Event Category: Discrete
Alarm Class:
Area: /Area1
Server Name: RSLinx Enterprise
Alarm Name: [ClxAlarms]Dig01
Message: [ClxAlarms]Dig01 needs attention

Severity: 999
Current Value: 1
Limit Value Exceeded:
Tag 1 Value:
Tag 2 Value:
Tag 3 Value:
Tag 4 Value:
Alarm Count: 5

11 | 🔔 8 | ✓ 2 | 🔔 1 | ? 0 | Filter: Not Filtered | Sorted by: Eve

Alarm list and column headings

In the status bar, view the number of alarms in the list, and how entries are filtered and sorted.

View details for the selected entry.

Drag to see more of the alarm list, or more details.

An alarm and event summary is composed of:

- An alarm list, that shows alarms that need the operator’s attention.
- An optional status bar, that provides information about the contents of the alarm and event summary, such as the name of the filter applied to the list view.
- An optional details pane, that shows the attributes of the currently selected alarm.
- A shortcut menu, and an optional toolbar, for alarm-related operations.

The following sections describe some of the ways you can use an alarm and event summary at run time. For details, see the FactoryTalk Alarms and Events Help.

Filtering and sorting information at run time

An operator can filter and sort the information shown in the alarm list at run time.

To filter information in the alarm list, the operator can do one of the following:

- Select from a predefined list of filters, set up for the alarm and event summary at design time.
- Clear the current filter. This refreshes the alarm list, to show all alarm information generated by alarm sources the alarm and event summary subscribes to. For information about event subscriptions, see Setting up event subscriptions on page 295.
- Create a custom filter. For details, see the FactoryTalk Alarms and Events Help.

If the Configure Sorting button is available on the toolbar, the operator can select run-time sorting criteria.

Tip: Changes made to the sort order and custom filter at run time are not saved. The next time the alarm and event summary is shown, it will use the sort order defined for it at design time.

Acknowledging alarms

Operators acknowledge alarms to show they are aware that the alarms have occurred.

Acknowledging an alarm does not correct the condition that caused the alarm. For the alarm condition to return to normal, the value of the tag being monitored must be within normal range.

A single tag will generate several alarms if its value crosses more than one alarm limit (also called a threshold). A tag representing temperature, for example, might trigger both Hot and Overheat alarms before it can be acknowledged. In this case, each alarm condition must be acknowledged separately.

A single tag might also generate alarms and then return to normal range several times, before it can be acknowledged. In this case, only the most recent alarm condition can be acknowledged.

In an alarm and event summary, an operator can acknowledge:

- One or more selected alarms (and enter an optional comment).
- All alarms currently visible in the alarm list.
- All alarms, including those that aren’t visible in the current page of the alarm list.

  If an alarm filter is applied, acknowledging all alarms excludes alarms that have been filtered from the list.

For details about acknowledging alarms, see the FactoryTalk Alarms and Events Help.

Tip: Most alarms are configured to require acknowledgment. If you want specific alarms to always be in an acknowledged state, in the Alarm and Event Setup editor for these alarms, clear the check box Acknowledge required.

Acknowledging the selected alarm

To acknowledge selected alarms in an alarm and event summary, use one of these methods:

- Select the alarms, and then click ✓ on the toolbar.
- Select the alarms, right-click, and then select **Acknowledge**.

Acknowledging all alarms

To acknowledge all alarms in an alarm and event summary, use one of these methods:

- To acknowledge all currently visible alarms, click ☑ on the toolbar.
- To acknowledge all alarms, including those that are not visible in the current page of the list, click ⊘ on the toolbar button Acknowledge all alarms contained in the list.

  If an alarm filter is applied, acknowledging all alarms excludes alarms that have been filtered from the list.

  Tip: To provide the button, Acknowledge all alarms contained in the list, you must select it in the Toolbar tab of the alarm and event summary’s Properties dialog box.

Resetting latched digital alarms

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.

An operator can only reset a latched alarm when the alarm condition returns to normal. Otherwise, the attempt to reset the alarm will fail and generate an error.

For details about setting up and resetting digital latched alarms, see the FactoryTalk Alarms and Events Help.

To reset a latched alarm:

1. In the alarm and event summary, select one or more alarms to reset, and then click on the toolbar button**.**
2. In the Reset Alarm dialog box, type an optional comment, and then click **Reset**. The comment is recorded with the reset alarms, in the alarm and event log.

Suppressing and disabling alarms

Suppress alarms that are not needed temporarily, for example, if they are interfering with another alarm that is the root cause of the condition you need to correct.

Suppressed alarms are not shown in alarm and event summaries or banners. However, they are recorded in the alarm and event log.

To completely prevent alarm detection for an alarm, disable the alarm. For example, you might disable an alarm for diagnostic or maintenance purposes.

Tip: Disabling a latched digital alarm while the Tag Alarm and Event Server is running, causes the alarm to become unlatched.

To suppress alarms:

1. In the alarm and event summary, select one or more alarms, and then click ⊘ on the toolbar button.
2. In the Suppress Alarm dialog box, type an optional comment, and then click **Suppress**. The comment is recorded with the suppressed alarms, in the alarm and event log.

To disable alarms:

1. In the alarm and event summary, select one or more alarms, and then click □ on the toolbar button.
2. In the Disable Alarm dialog box, type an optional comment, and then click **Disable**. The comment is recorded with the disabled alarms, in the alarm and event log.

Unsuppressing and enabling alarms

Since suppressed and disabled alarms are not shown in alarm and event summaries or banners, to unsuppress or enable alarms, you must use the Alarm Status Explorer.

You can set up an alarm status explorer in a FactoryTalk View SE graphic display. For more information, see Working with alarm sources in the Alarm Status Explorer on page 306.

You can also open a stand-alone Alarm Status Explorer at run time, by clicking Display the Alarm Status Explorer, on the alarm and event summary toolbar. For more information, see the FactoryTalk Alarms and Events Help.

Shelve or unshelve an alarm

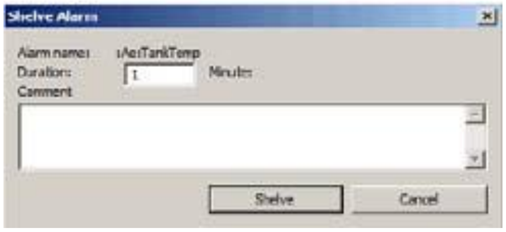
Shelving an alarm temporarily suppresses it. When shelving, the operator specifies a duration for the alarm to remain shelved. The alarm is automatically unshelved when the duration period times out.

A shelved alarm is shown in the Alarm and Event Summary or Alarm Status Explorer. However, a shelved alarm is not shown in the Alarm and Event Banner.

If alarms that were previously shelved need to be monitored again, you can unshelve the specified alarms in the Alarm Status Explorer.

To shelve an alarm:

1. In the Alarm and Event Summary event list, select the alarm you want to shelve and click ≡ on the toolbar. The Shelve Alarm dialog box opens.



Shelve Alarm

Alarm name: ::Ae:TankTemp

Duration: 1 Minutes

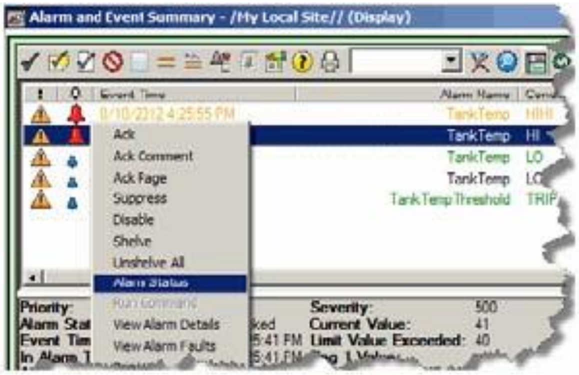
Comment

2. In the Shelve Alarm dialog box, enter the duration (in minutes) to shelve the alarm, and any optional comments.

3. When finished, click **Shelve**.

To unshelve an alarm:

1. In the Alarm and Event Summary window, right-click an alarm, and then from the context menu, select **Alarm Status**.



Alarm and Event Summary - /My Local Site// (Display)

!		Event Time	Alarm Name	Cond
		0/10/2012 4:25:55 PM	TankTemp	HIHI
			TankTemp	HI
			TankTemp	LO
			TankTemp	LO
			TankTemp Threshold	TRIP

Priority: Severity: 500

Alarm Stat... Current Value: 41

Event Tim... 5:41 PM Limit Value Exceeded: 40

In Alarm T... 5:41 PM Tag 1 Value:

2. In the Alarm Status Explorer window, select the alarm or alarms you want to unshelve, and then click ≡ on the toolbar.

In the Unshelve Alarm dialog box, enter a comment to explain why the alarm is being unshelved and then click **Unshelve**.

Tip: You can also use the **Unshelve all alarms** button to unshelve all shelved alarms in the Alarm and Event Summary. For details, see the FactoryTalk Alarms and Events System Configuration Guide.

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 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 312.

To create an alarm status explorer:

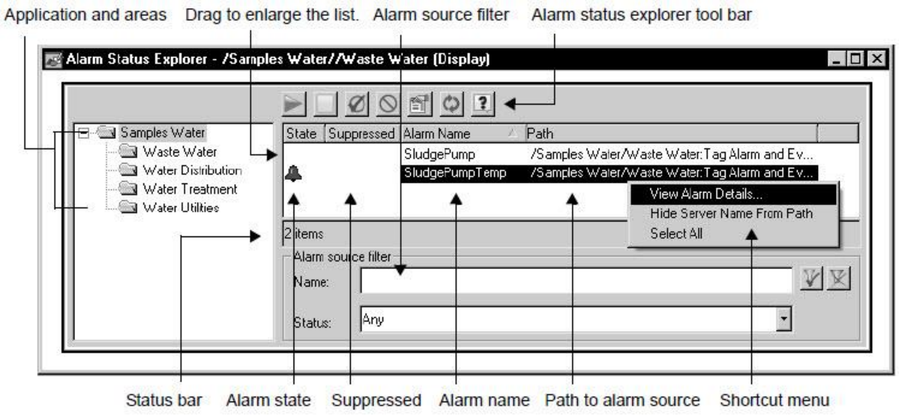
1. In FactoryTalk View Studio, in the Explorer window, expand the application, area, and HMI server icons, where you want to create the alarm status explorer display.
2. To create a new graphic display, expand the Graphics folder, right-click Displays, and then select **New**.
3. In the Graphics editor, select **Objects** > **Alarm and Event** > **Status Explorer**. You can also click on the Graphic editor's toolbar.
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.

The parts of an alarm status explorer

The following illustration shows an alarm status explorer at run time.



State	Suppressed	Alarm Name	Path
		SludgePump	/Samples Water/Waste Water:Tag Alarm and Ev...
		SludgePumpTemp	/Samples Water/Waste Water:Tag Alarm and Ev...

An alarm status explorer is composed of:

- An alarm list, that shows all the alarms contained in the selected areas (and sub-areas).
- An optional tree view of the application, and the areas that contain alarm sources.
- A status bar, that provides information about the contents of the alarm status explorer, such as the number of alarms in the selected alarm source.
- A shortcut menu, and an optional toolbar, for performing common operations.

For details about using an alarm status explorer at run time, see the FactoryTalk Alarms and Events Help.

Tip: At run time, you can also open a stand-alone Alarm Status Explorer, by clicking the **Display the Alarm Status Explorer** button in the alarm and event summary toolbar. For details, see the FactoryTalk Alarms and Events Help.

Browsing alarm sources

In the tree view on the left side of the alarm status explorer, expand folders representing the application, and areas in the application, to find alarm sources.

If the selected area or sub-area contains alarms, the alarms will show in the alarm list on the right side of the explorer.

Refreshing the alarm list

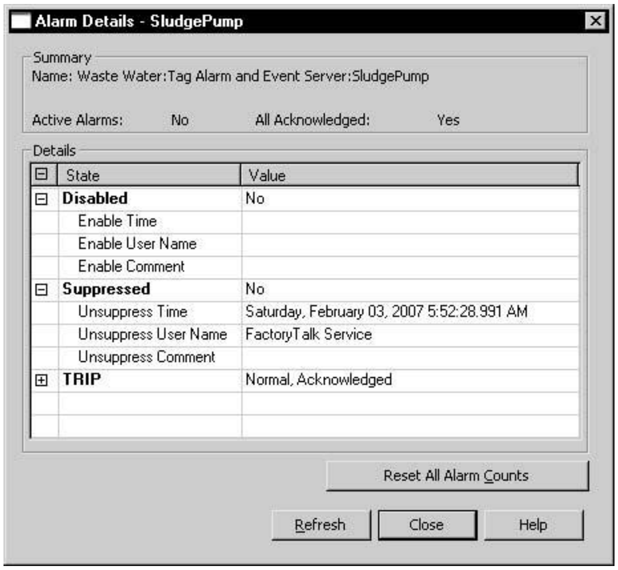
If alarms are added or deleted after the alarm status explorer is opened, the alarm list is not updated dynamically.

To show the most recent list of alarms and their current states, on the alarm status explorer's toolbar, click Refresh alarm list .

Viewing alarm details

To view details for the selected alarm, on the alarm status explorer's toolbar, click (or right-click the alarm, and then select **View Alarm Details)**.

The Alarm Details dialog box (shown in the following illustration) shows the last time an alarm was disabled or enabled, and suppressed or unsuppressed, and by whom. You can also view information about the most recent alarm condition, and the priority and severity associated with the alarm.



Alarm Details - SludgePump

Summary
Name: Waste Water:Tag Alarm and Event Server:SludgePump

Active Alarms: No All Acknowledged: Yes

Details

State	Value
⊟ **Disabled**	No
Enable Time	
Enable User Name	
Enable Comment	
⊟ **Suppressed**	No
Unsuppress Time	Saturday, February 03, 2007 5:52:28.991 AM
Unsuppress User Name	FactoryTalk Service
Unsuppress Comment	
⊞ **TRIP**	Normal, Acknowledged

Alarm condition details vary, depending on the type of alarm. For a level alarm, for example, details are listed for each alarm level (High High, High, Low, and Low Low). For details about options in the Alarm Details dialog box, click **Help**.

Tip: You can also view alarm details from an alarm and event summary.

Unsuppressing and suppressing alarms

Use the alarm status explorer to suppress and unsuppress alarms.

Suppress alarms that are not needed temporarily, for example, if they are interfering with another alarm that is the root cause of the condition you need to correct.

Suppressed alarms are not shown in alarm and event summaries or banners. However, they are recorded in the alarm and event log. To resume monitoring the alarm in alarm and event objects, unsuppress it.

The Suppressed column in an alarm status explorer indicates whether an alarm is currently suppressed.

To check when, why, and by whom the alarm was suppressed, right-click the alarm, and then select **View Alarm Details**. For more information about the Alarm Details dialog box, see Viewing alarm details on page 308.

To unsuppress an alarm:

1. In the alarm status explorer, select one or more suppressed alarms, and then click ⌀ on the toolbar.
2. In the Unsuppress Alarm dialog box, type an optional comment, and then click **Unsuppress**.

To suppress an alarm:

1. In the alarm status explorer, select one or more alarms, and then click ⊘ on the toolbar.
2. In the Suppress Alarm dialog box, type an optional comment, and then click **Suppress**.

Disabling and enabling alarms

Use the alarm status explorer to disable and enable alarms.

Disable alarms to completely prevent alarm detection for the alarm. For example, you might disable an alarm for diagnostic or maintenance purposes. To restore alarm detection for an alarm, enable the alarm.

Tip: Disabling a latched digital alarm while the Tag Alarm and Event Server is running, causes the alarm to become unlatched.

The State column in an alarm status explorer indicates whether an alarm is currently disabled.

To check when, why, and by whom the alarm was disabled, right-click the alarm, and then select **View Alarm Details.**

For information about the Alarm Details dialog box, see page 1-46.

To enable an alarm:

1. In the alarm status explorer, select one or more disabled alarms, and then click ▶ on the toolbar.
2. In the Enable Alarm dialog box, type an optional comment, and then click **Enable**.

To disable an alarm:

1. In the alarm status explorer, select one or more alarms, and then click □ on the toolbar.
2. In the Disable Alarm dialog box, type an optional comment, and then click **Disable**.

Viewing alarm and event history logs

The Alarm and Event Log Viewer shows the contents of 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 new 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 click 📋 on the toolbar.
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**.

The parts of an alarm and event log viewer

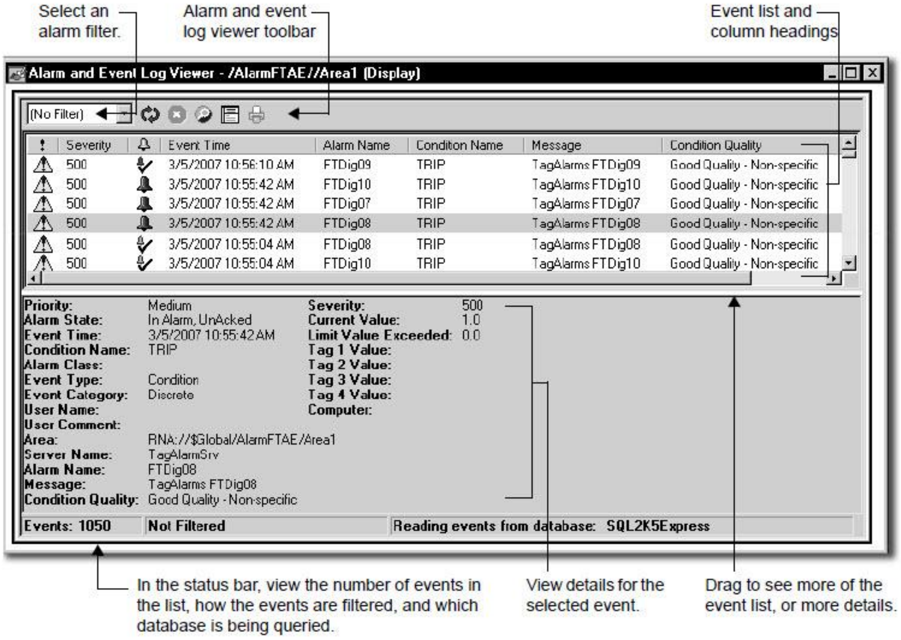
The following illustration shows an alarm and event log viewer at run time.

An alarm and event log viewer is composed of:

- An event list, that shows all the data in the specified alarm and event log.
- An optional status bar, that provides information about the contents of the event list, such as the number of events, and whether a filter is applied to the display.
- A shortcut menu, and an optional toolbar, for performing common operations.
- An optional details pane, that shows the attributes of the currently selected event.

For details about using an alarm and event log viewer at run time, see the FactoryTalk Alarms and Events Help.

Tip: To view alarm-related activity that is sent to diagnostic and audit logs, use the FactoryTalk Diagnostics List or Log Viewer. For details, see the FactoryTalk Alarms and Events Help.



Select an alarm filter.

Alarm and event log viewer toolbar

Event list and column headings

Alarm and Event Log Viewer - /AlarmFTAE//Area1 (Display)

(No Filter)

!	Severity	🔔	Event Time	Alarm Name	Condition Name	Message	Condition Quality
⚠	500	✓	3/5/2007 10:56:10 AM	FTDig09	TRIP	TagAlarms FTDig09	Good Quality - Non-specific
⚠	500	🔔	3/5/2007 10:55:42 AM	FTDig10	TRIP	TagAlarms FTDig10	Good Quality - Non-specific
⚠	500	🔔	3/5/2007 10:55:42 AM	FTDig07	TRIP	TagAlarms FTDig07	Good Quality - Non-specific
⚠	500	🔔	3/5/2007 10:55:42 AM	FTDig08	TRIP	TagAlarms FTDig08	Good Quality - Non-specific
⚠	500	✓	3/5/2007 10:55:04 AM	FTDig08	TRIP	TagAlarms FTDig08	Good Quality - Non-specific
⚠	500	✓	3/5/2007 10:55:04 AM	FTDig10	TRIP	TagAlarms FTDig10	Good Quality - Non-specific

Priority:	Medium	**Severity:**	500
Alarm State:	In Alarm, UnAcked	**Current Value:**	1.0
Event Time:	3/5/2007 10:55:42 AM	**Limit Value Exceeded:**	0.0
Condition Name:	TRIP	**Tag 1 Value:**	
Alarm Class:		**Tag 2 Value:**	
Event Type:	Condition	**Tag 3 Value:**	
Event Category:	Discrete	**Tag 4 Value:**	
User Name:		**Computer:**	
User Comment:			
Area:	RNA://$Global/AlarmFTAE/Area1		
Server Name:	TagAlarmSrv		
Alarm Name:	FTDig08		
Message:	TagAlarms FTDig08		
Condition Quality:	Good Quality - Non-specific		

Events: 1050 | **Not Filtered** | **Reading events from database: SQL2K5Express**

In the status bar, view the number of events in the list, how the events are filtered, and which database is being queried.

View details for the selected event.

Drag to see more of the event list, or more details.

Refreshing the event list

To show the most recent alarm activity logged to the selected database, on the alarm and event log viewer's toolbar, click ⟳.

Filtering information at run time

To filter information in the event list, an operator can do one of the following:

- Select from a predefined list of filters, set up for the alarm and event log viewer at design time.
- Clear the current filter. This refreshes the event list, to show all the alarm data stored in the database being queried.

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. For an example, see page 1-52.

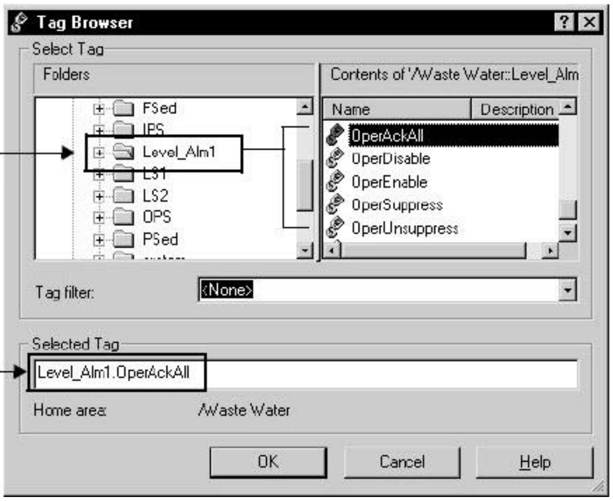
You can also use alarm tags to obtain information about an alarm's state. For a list of alarm tags and their functions, and for information about device-based alarm tags, see the FactoryTalk Alarms and Events Help.

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 314.

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 check box, 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 the alarm in the Tag Browser, its alarm tags become visible in the right-hand window.

You can use the Oper* tags to let an operator respond to instances of the alarm at run time.

Tag filter: <None>

Selected Tag: Level_Alm1.OperAckAll

Home area: /Waste Water

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 check box, 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 on 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 run time, 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 449.

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 555.

Retrieving information about the severity of alarms

FactoryTalk View SE provides alarm functions 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 316.

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 316.

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 you can use in expressions, to retrieve the number of specified alarms that are acknowledged, unacknowledged, or acknowledged and returned to normal.

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 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 316.

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 316.

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 316.

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 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 referencese on page 131.

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 Logix5000 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 cross-hatch character (#) followed by a number from 1 to 500. The tag definition in the following example contains the placeholder **#1**:

```
AE_InAlmAckedCount
("/MixerArea/MixerLine1::[CLX]Line1.Mixer_#1")
```

At run time, 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 428.

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 run time, 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 Logix5000 controller, or if the RSLinx Enterprise alarm server:

- Loses its connection to the Logix5000 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 non-recoverable program fault.
- Cannot subscribe to all of the alarms at the Logix5000 controller.

If the condition causing the disconnection is resolved, the expression can be reevaluated, with the restored alarm function included.

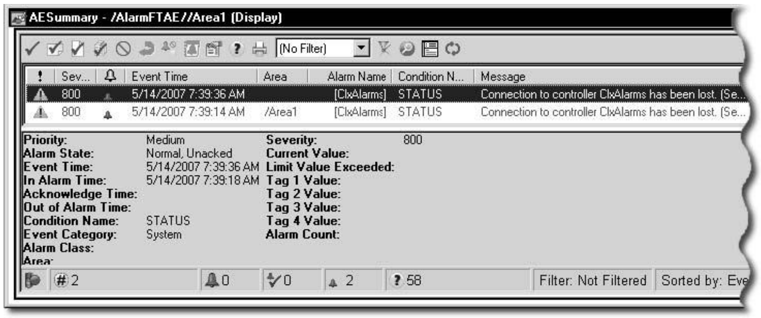
About controller status system alarms

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.

Following is an illustration of a controller status alarm in an alarm and event summary.



!	Sev...	🔔	Event Time	Area	Alarm Name	Condition N...	Message
⚠	800		5/14/2007 7:39:36 AM		[ClxAlarms]	STATUS	Connection to controller ClxAlarms has been lost. (Se...
⚠	800		5/14/2007 7:39:14 AM	/Area1	[ClxAlarms]	STATUS	Connection to controller ClxAlarms has been lost. (Se...

Priority:	Medium	**Severity:**	800
Alarm State:	Normal, Unacked	**Current Value:**	
Event Time:	5/14/2007 7:39:36 AM	**Limit Value Exceeded:**	
In Alarm Time:	5/14/2007 7:39:18 AM	**Tag 1 Value:**	
Acknowledge Time:		**Tag 2 Value:**	
Out of Alarm Time:		**Tag 3 Value:**	
Condition Name:	STATUS	**Tag 4 Value:**	
Event Category:	System	**Alarm Count:**	
Alarm Class:			
Area:			

For details about controller status alarms and the conditions that cause them, see the RSLinx Enterprise 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. Microsoft Excel 2000, XP, 2003, 2007, and 2010 formats are supported.

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 run time.
- 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 run-time language switching.
- Support for multiple languages in the graphic libraries.

About language switching

The FactoryTalk View language switching feature gives operators the ability to switch between up to 40 different languages in one application at run time.

FactoryTalk View SE Clients can run in any of the languages an application supports. In a network distributed application, multiple clients can 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 332.
- Importing translated text, see Importing translated or modified text strings on page 342.

Text strings you can view in different languages at run time

In applications that support multiple languages, the following types of text strings can be viewed in different languages at run time:

- 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: A text object with its SizeToFit property set to True might change in size when shown in different languages.

- Text strings defined for FactoryTalk Alarm and Event summaries, banners, and log viewers. For details, see the FactoryTalk Alarms and Events Help.

Specifying time, date, and numeric formats

During development or at run time, 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 run time, 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 run time.

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 run time.

Text that cannot be exported

The following user-defined text strings are visible at run time, 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
- The text of HMI tag alarm messages
- Headings in an HMI tag alarm summary
- 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 Recipe 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 run time

Some text that you define for an application is never visible to an operator at run time. 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 Image St_Image LinkedObject
	Display list selector	St_Display St_Parameter
FactoryTalk View Site Edition	Recipe	Value
	Numeric input and string input	TagName
	Numeric display and string display	Expression
	Tag label	TagName PropertyName Caption
	Trend	All properties, **except** ChartTitle, Pen Description, and EngineeringUnits. These three properties **are** 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 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 run time.

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 332.
- Importing translated strings, see Importing translated or modified text strings on page 342.
- Setting up language switching at run time, see Setting up run-time language switching on page 347.

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 permissions you might require.

For an overview of FactoryTalk Security services, see Setting up security on page 85. For details, see FactoryTalk Security Help.

Setting up font support for Windows languages

If you are using Windows XP, to ensure that fonts used in a FactoryTalk View application display correctly, regardless of the current language, install font support for all of the types of Windows languages the application will use. Depending on your application, this means that you might need to enable font support for languages that read from right to left, or for East Asian character sets.

The other 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).

Enabling font support for additional Windows languages

The steps for installing font support vary by operating system, and are not required for Windows 8.1, Windows 8, Windows 7, Windows Server 2012, or Windows Server 2008.

To enable font support for additional languages in Windows XP:

1. In Control Panel, double-click the **Regional and Language Options** icon.

2. In the Languages tab, under Supplemental language support, select the appropriate check box, and then click **OK**.

For details, see Windows Help.

Selecting a language for a new FactoryTalk View application

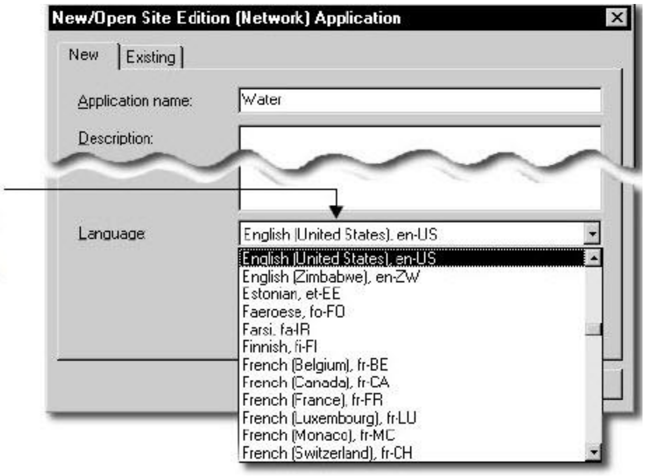
When you create a new FactoryTalk View application, you must select a language for the application. The language you select becomes the default application language.

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 languages you add to the application.

To select a language when creating a new 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 a new application, you select a language from the full list of languages that Windows supports, as shown in the following illustration.



Application name: Water

Description:

Language: English (United States), en-US

English (United States), en-US
English (Zimbabwe), en-ZW
Estonian, et-EE
Faeroese, fo-FO
Farsi, fa-IR
Finnish, fi-FI
French (Belgium), fr-BE
French (Canada), fr-CA
French (France), fr-FR
French (Luxembourg), fr-LU
French (Monaco), fr-MC
French (Switzerland), fr-CH

When creating a new application, you can select any of the languages Windows supports.

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 mean 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 331.

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.

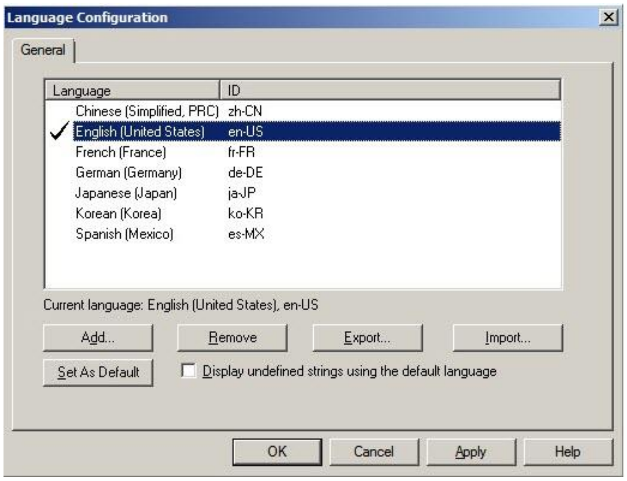
To translate multiple languages at once, add languages to the application first, and then export text strings for all the languages, to a Microsoft Excel spreadsheet. For details, see Exporting application text strings **on** page 332**.**

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 re-open the application.

To add languages to an application:

1. In FactoryTalk View Studio, from the **Tools** menu, select **Languages**.

2. In the Language Configuration dialog box, click **Add**.



	Language	ID
	Chinese (Simplified, PRC)	zh-CN
✓	English (United States)	en-US
	French (France)	fr-FR
	German (Germany)	de-DE
	Japanese (Japan)	ja-JP
	Korean (Korea)	ko-KR
	Spanish (Mexico)	es-MX

Current language: English (United States), en-US

3. In the Add Language dialog box, select the language to add, and then click **OK**.

4. Repeat steps 2 and 3 for each language you want to add to the application.

For details about options in the Language Configuration dialog box, click **Help**.

Setting up a default application language

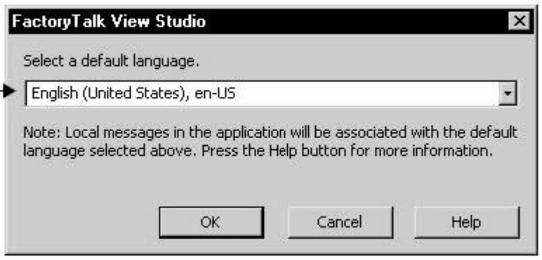
The language you select when creating a new application becomes the default language for the application.

To change the default language, open the Language Configuration dialog box, select an application language from the list, and then click **Set as Default** (shown in the previous illustration).

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.

Select a default language for an application.



FactoryTalk View Studio

Select a default language.

English (United States), en-US

Note: Local messages in the application will be associated with the default language selected above. Press the Help button for more information.

Showing undefined text strings in the default language

By default, any undefined text strings in a FactoryTalk View application, which cannot be viewed in the current application language, are shown as a question mark (?).

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 check box, 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 run time.

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 337.

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. In FactoryTalk View Studio, from the **Tools** menu, select **Languages**.
2. In the Language Configuration dialog box, click **Export**. This opens the String Import Export Wizard.

   Tip: If you are exporting strings for a single language only, before you click **Export**, in the list of application languages, select the language of the strings you want to export.

3. Select to export strings either for the selected language, or for all languages in the application.
4. Click **Next**, and then follow the instructions in the String Import Export Wizard.

Tip: Before exporting text in an application, ensure that all the HMI project components containing text to be exported are not in use. If project components are being modified remotely, and they have not been saved, the exported file might not contain the unsaved changes.

If you cancel the export operation while it is in progress, any text files created prior to canceling might not be complete.

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 following folder on the local computer:

- \Users\Public\Public Documents\RSView Enterprise\Strings
- (Windows XP) \Documents and Settings\All Users\Shared Documents\RSView Enterprise\Strings

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 332.

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 335.

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 information about how to open the Add Language dialog box, see Adding languages to an application on page 330.

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.

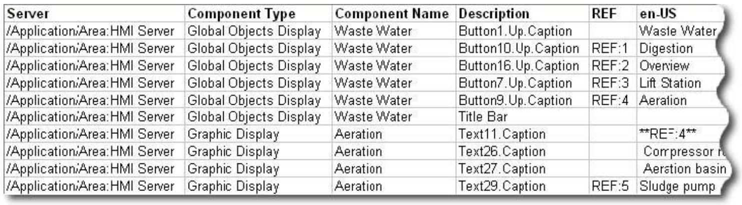
For information about the format and schema of Unicode text files, see Working with strings exported to a Unicode text file on page 338.

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 r
/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 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 check box, **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 330.

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.

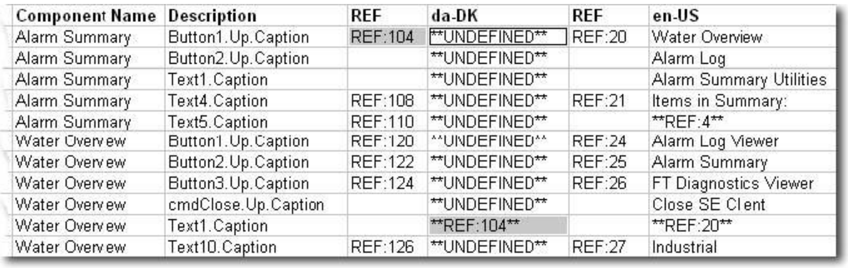
- Do not change the default order of the Server, Component Type, Component Name and Description columns.
- Do not insert any blank rows or columns in the spreadsheet. The import operation will stop at a blank row or column. Anything after that will not be imported.
- You can change the order of the columns 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 Overv ew	Button1.Up.Caption	REF:120	^^UNDEFINED^^	REF:24	Alarm Log Viewer
Water Overv ew	Button2.Up.Caption	REF:122	**UNDEFINED**	REF:25	Alarm Summary
Water Overv ew	Button3.Up.Caption	REF:124	**UNDEFINED**	REF:26	FT Diagnostics Viewer
Water Overv ew	cmdClose.Up.Caption		**UNDEFINED**		Close SE Cl ent
Water Overv ew	Text1.Caption		**REF:104**		**REF:20**
Water Overv ew	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.

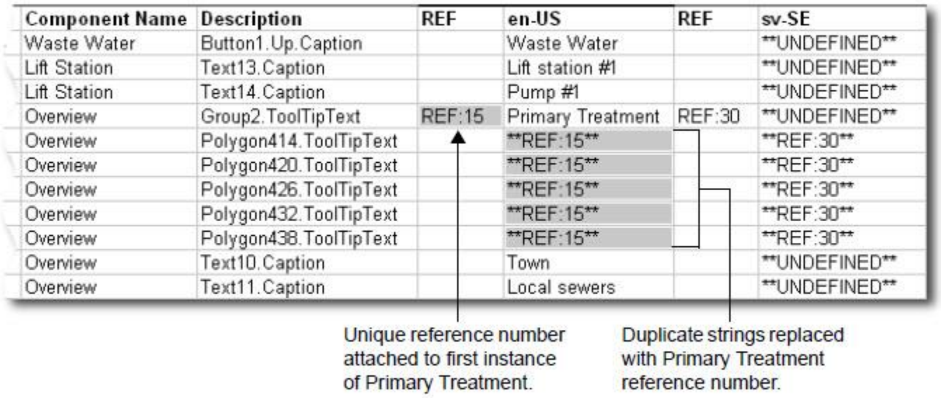
For more information about duplicate text strings, see "Working with duplicate text strings," next.

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 331.

Working with duplicate text strings

If the check box, 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 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.

To specify the file format (Step 1):

1. In Step 1 under Original data type, click **Delimited**.
2. In the Start import at row box, type or select 1.
3. In the File origin list, click **Windows (ANSI).**
4. Click **Next**.

To specify the field delimiter (Step 2):

1. Select the Tab check box. If any other check boxes are selected, clear them.
2. Make sure you clear the check box, **Treat consecutive delimiters as one**.
3. Click **Next**.

To specify the column data format (Step 3):

1. If it is not selected already, under Column data format, click **General**.
2. Click **Finish**.

Saving a Unicode text file in Microsoft Excel

When you click **Save**, Excel warns that the file may contain features that are not compatible with Unicode text.

1. When prompted to keep the workbook in Unicode format, which leaves out incompatible features, click **Yes**.
2. When closing the file, you are prompted to save changes. Click **Yes**.
3. You can save the file using its original name, or you can type a new name.

4. When prompted again to keep the workbook in Unicode format, which leaves out incompatible features, click **Yes**.

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 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 335.
- Working with strings exported to a Unicode text file on page 338.

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. In FactoryTalk View Studio, from the Tools menu, select **Languages**. This opens the Language Configuration dialog box.
2. Export text strings currently in the application, to create a backup.

   If necessary, you can use the backup to restore text already saved in any application language, if an error occurs while importing, or if you cancel the import before it is complete. For details about exporting text, see Exporting text strings in an application on page 332.
3. Click **Import**. This opens the String Import Export Wizard.

   Tip: If you are importing strings from a Unicode text file, before you click **Import**, in the list of application languages, select the language of the strings you are about to import.
4. Choose to import strings from a Unicode text file into the selected application language, or from a Microsoft Excel spreadsheet into all application languages defined in the spreadsheet.
5. Follow the instructions in the String Import Export Wizard.

   For details about options in the String Import Export Wizard, click **Help**.

   Tip: Before importing text into an application, ensure that the HMI project components involved are not in use. If components are being modified remotely, and the changes are saved after you import text, the imported text will be overwritten.

   If you cancel an 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 saved in the application, import the text from the backup file you created in step 2 of this task.

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 following folder on the local computer:

- \Users\Public\Public Documents\RSView Enterprise\Strings

- (Windows XP) \Documents and Settings\All Users\Shared Documents\RSView Enterprise\Strings

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 342.

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 *FileName*.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 "*FileName*", Line *x*. *ComponentType* 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 "*FileName*", Line *x*. *ComponentName* 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 "*FileName*", 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 "*FileName*", Line *x*. The string was not used in the application and was not imported.	This error occurs if: • 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. • the object was deleted from the application after the text was exported. If this is correct, ignore the error.

File "*FileName*", 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""`
File "*FileName*", 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 {*FileName*.txt}. None of the strings from this file were imported.	The text file could not be opened. Make sure 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.

Common 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.	User canceled or interrupted the string import operation.
"*RFC1766 Name*" does not exist in this application. No strings for this language were imported.	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.
"*Component Type*" is not valid. No strings for this component type were imported.	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."
"*Server Name*" does not exist in this application. No strings for this server were imported.	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.
"*Server Name*" is unavailable. No strings for this server were imported.	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.
"*Server Name_Component Type_Component Name*" does not exist in this application. No strings for this component were imported.	Component name for a string in the spreadsheet does not exist in the application. No strings associated with this component name 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 name.

"*Server Name_Component Type_Component Name_Description*" does not exist in this application. No strings for this component were imported.	Description for a string in the spreadsheet does not exist in the application. No strings with this description 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 description.
"*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.	Optimized duplicate string reference in the spreadsheet refers to a string that does not exist. 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.
Unable to save "*Display Name.*" No strings for this component name were imported.	A graphic display or global object display cannot be saved. The import will continue.
Unable to open "*Display Name.*" No strings for this component name were imported.	A graphic display or global object display cannot be opened. The import will continue.
Unable to import strings into "*HMI Server Name*" as the currently logged in user doesn't have write access to the area.	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. Applies to FactoryTalk View SE network distributed applications. No strings associated with this server will be imported. However, the import will continue for any other HMI servers in the application.
Unable to import strings as the currently logged in user doesn't have write access to the application.	The currently logged on user does not have Common Write access to the application whose strings are being imported. Applies to FactoryTalk View SE local station, network station, and FactoryTalk View ME applications. No application strings will be imported.
Unable to import one or more strings, as one or more cells were not formatted as text or numeric values.	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.
Unable to import one or more strings, as one or more cells were not formatted as text or numeric values.	A spreadsheet cell containing a server name, component type, component name, description, or string reference is not formatted for text or numeric values. No strings in that row will be imported. The import will continue. This error will only be reported once for a single import operation.
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 "*Local Message File Name*." No strings for this component name were imported.	Imported local message file cannot be saved. The import will continue.
Unable to open "*Local Message File Name*." No strings for this component name were imported.	Imported local message file cannot be opened. The import will continue.

Setting up run-time language switching

When you create a FactoryTalk View SE Client configuration file, you select the initial run-time 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 run time.

The Language command uses the following syntax:

```
Language <RFC1766 name>
```

where <*RFC1766 name*> is a code name for the language 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 330.

Tip: If you create buttons for switching languages at run time, 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.

You can set up individual libraries to support multiple languages. If you do this, however, the text strings will display only in the current application language. In any other language, the strings will display as undefined.

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 check box, **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, 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 check box, **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 tag and alarm properties at run time.
- 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 353.
- **Disconnected operation**. For example, connected clients can continue to run when the FactoryTalk Directory becomes unavailable. For

information, see Monitoring the status of the Network Directory server on page 356.

- **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 362.
  - 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 RSLinx Enterprise servers, see the product documentation.
  - OPC data servers, including RSLinx 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 run time on page 373.
- **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 366.
- **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 380.

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 149, and Working with local station applications on page 167.

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.

To open the FactoryTalk Help:

Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **FactoryTalk Help**.

Monitoring the status of application servers

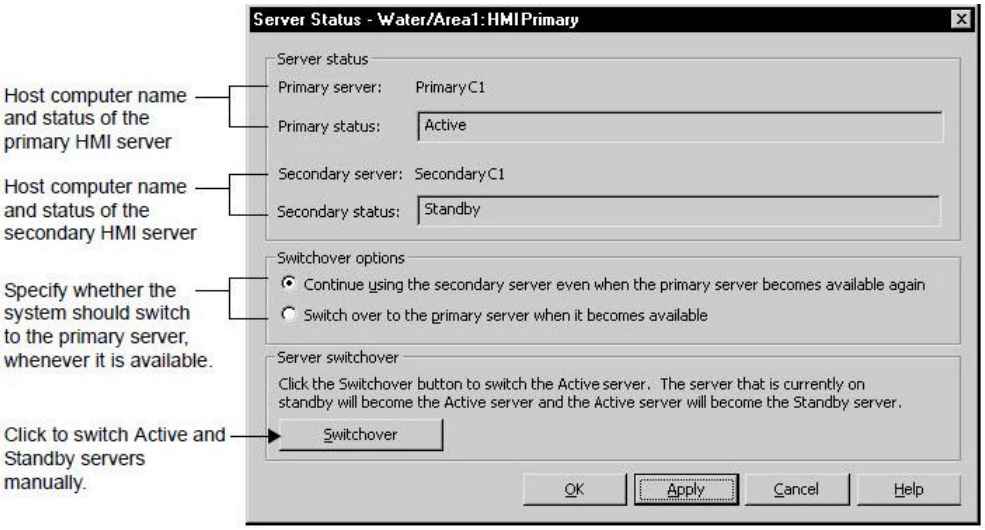
In FactoryTalk View Studio, or in the FactoryTalk View SE Administration Console, you can monitor the operational status of servers in a FactoryTalk View SE application, to determine whether the servers are ready to provide service, and to troubleshoot problems.

In the Server Status dialog box, you can:

- Check the status of non-redundant and redundant FactoryTalk View SE Servers, Rockwell Automation Device Servers (RSLinx Enterprise), 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 370.

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**.


Server Status - Water/Area1:HMIPrimary

Server status

Primary server: PrimaryC1

Primary status: Active

Secondary server: SecondaryC1

Secondary status: Standby

Switchover options

● Continue using the secondary server even when the primary server becomes available again

○ Switch over to the primary server when it becomes available

Server switchover

Click the Switchover button to switch the Active server. The server that is currently on standby will become the Active server and the Active server will become the Standby server.

Host computer name and status of the primary HMI server

Host computer name and status of the secondary HMI server

Specify whether the system should switch to the primary server, whenever it is available.

Click to switch Active and Standby servers manually.

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.

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 service 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. This would occur, for example, if a supporting service is unhealthy. For information about what happens when an HMI server fails, see What happens if a non-redundant HMI server fails on page 355 and What happens when the primary HMI server fails on page 371.
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 service to clients, in the event that the Active server fails. 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 380.
- 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 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

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

The 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 FactoryTalk Directory Server Location Utility, you can view the current status of the Network Directory server:

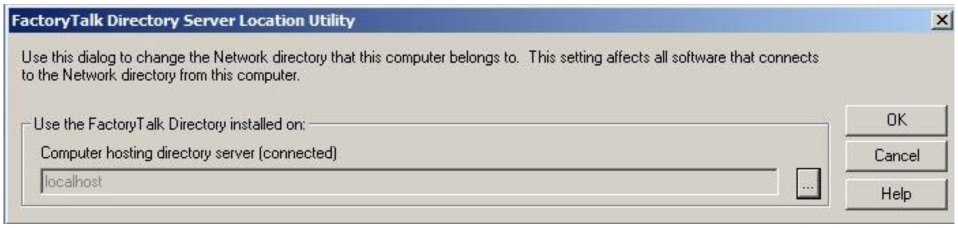
- **(connected)** means that all FactoryTalk products and components participating in a FactoryTalk system, located on the current computer, are connected to and communicating with the Network Directory server computer.
- **(read-only)** means that FactoryTalk system participants on the current computer 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. Select **Start > All Programs > Rockwell Software > FactoryTalk Tools > Specify FactoryTalk Directory Location**.
2. You are prompted to log on to FactoryTalk. Type your name and password, and then click **OK**.

   To use the FactoryTalk Directory Server Location Utility, you must log on 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, look for the status of the Network Directory server beside the label, Computer hosting directory server:



FactoryTalk Directory Server Location Utility

Use this dialog to change the Network directory that this computer belongs to. This setting affects all software that connects to the Network directory from this computer.

Use the FactoryTalk Directory installed on:

Computer hosting directory server (connected)

localhost

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 new security accounts, or change system security policies.

When the Network Directory is available again, all dependent clients in the system 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 Rockwell Automation Logix5000 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 RSLinx Enterprise 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 you plan to keep running.

This could be the same computer used to host a FactoryTalk View SE Server, or a Rockwell Automation Device Server (RSLinx Enterprise).

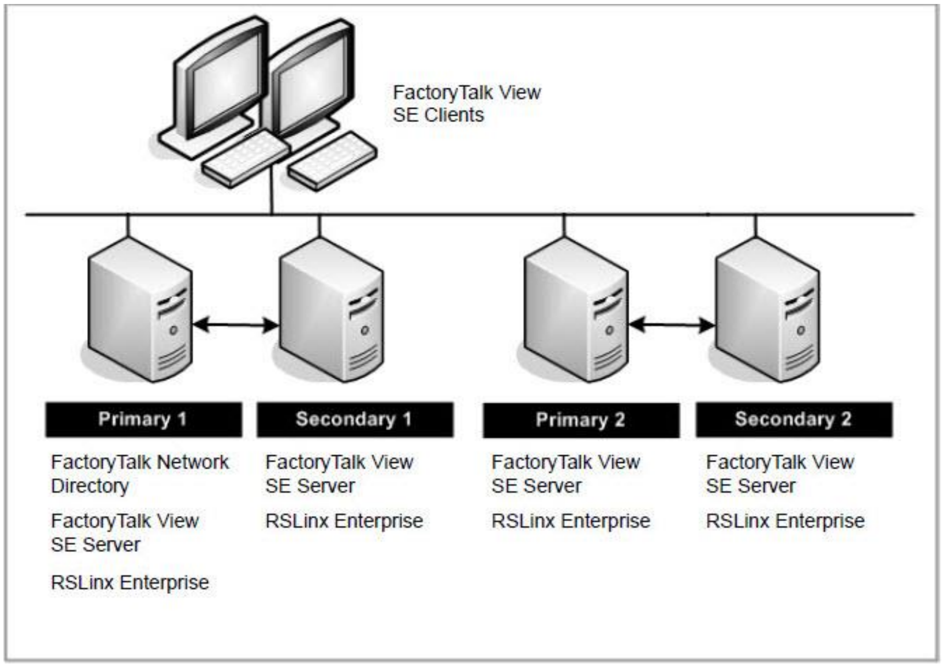
Note: Do not run the FactoryTalk Network Directory on the same computer as a Windows domain controller.

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 RSLinx Enterprise servers.

In this example, two redundant server computer pairs host primary and secondary HMI servers, and RSLinx Enterprise 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 83.


FactoryTalk View SE Clients
Primary 1
FactoryTalk Network Directory
FactoryTalk View SE Server
RSLinx Enterprise
Secondary 1
FactoryTalk View SE Server
RSLinx Enterprise
Primary 2
FactoryTalk View SE Server
RSLinx Enterprise
Secondary 2
FactoryTalk View SE Server
RSLinx Enterprise

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 distributed application, a single application server computer that meets the recommended requirements could host *all* of the following:

- One FactoryTalk View SE Server
- One RSLinx Enterprise data server. (Optionally, you can enable RSLinx Enterprise to distribute device-based FactoryTalk Alarms and Events.)
- One OPC-DA server (RSLinx Classic or some other OPC data server)

- One FactoryTalk Tag Alarm and Event Server
- The FactoryTalk Network Directory

However, if the network distributed application requires more HMI, data, or alarm and event servers, it is recommended that you use multiple host computers to distribute the load.

Tip: If you plan to deploy a network distributed application that uses more than two servers (or two pairs of redundant servers) and 20 clients, it is recommended that you contact your local Rockwell Automation Sales office for architectural assistance. For information about setting up redundant FactoryTalk View SE Servers, see Setting up FactoryTalk system availability on page 351 in the *FactoryTalk View Site Edition User's Guide*.

For information about hardware and operating system requirements that might affect your application, see the *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:

- 50 is the maximum number of FactoryTalk View SE Clients 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 FactoryTalk View SE platform requirements that might affect the design of a redundant system, see the *FactoryTalk View Site Edition Installation Guide.*

For information about limits and requirements that apply to redundant data servers (RSLinx Enterprise, RSLinx Classic, or another OPC-DA server), 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, including RSLinx Enterprise and RSLinx Classic, 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, RSLinx Enterprise server, or OPC data server (including RSLinx Classic) in a network distributed application.

In FactoryTalk View SE, the alternate (or backup) server in a redundant server pair is called the secondary server.

Summary of steps

In FactoryTalk View Studio, after you develop and test the network distributed application, do the following for any type of application server you want to make redundant:

1. Install and activate the necessary software on the secondary computer.

   For information about installing the FactoryTalk View SE software, see the *FactoryTalk View Site Edition Installation Guide.*

2. In Windows Explorer, copy any necessary files from the primary server computer to the secondary server computer.

3. On the secondary server computer, specify the location of the FactoryTalk Network Directory.

4. In the primary server’s Properties dialog box, set up redundancy options, and specify the name of the secondary server computer.

5. Ensure that data on the primary and secondary servers is identical:

   - For HMI servers, you can manually replicate changes from the primary to the secondary server. For details, see Replicate changes to the secondary HMI server on page 366.
   - For data servers, settings for topics and networks, for example, must be identical. For information about setting this up, see the product documentation.

For specific information about setting up redundancy for HMI servers, see "Setting up a redundant HMI server pair" next.

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 77.
   - Copy the primary HMI server’s project files to the secondary server computer. For details, see Copy HMI project files to the secondary computer on page 364.

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 364.
- To ensure that settings for the redundant pair are identical, replicate changes from the primary to the secondary server. For details, see Replicate changes to the secondary HMI server on page 366.

Specify the Network Directory on the secondary computer

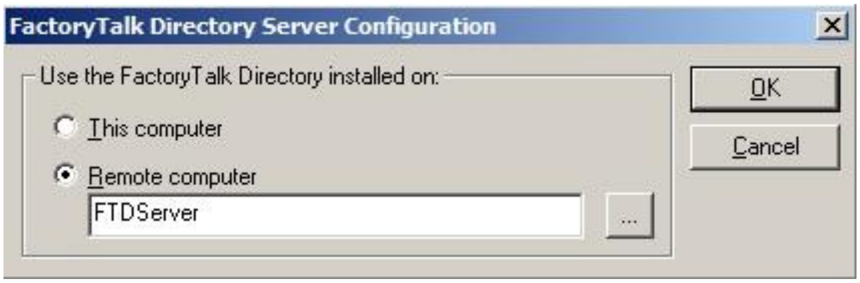
After installing the FactoryTalk View SE Server software on the secondary server computer, specify the location of the FactoryTalk Network Directory for the application.

To specify the Network Directory location:

1. Select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **Specify FactoryTalk Directory Location**.

   Tip: To open the FactoryTalk Directory Server Location Utility, you must log on as a user with administrative privileges at the Network Directory and in Windows, on the computer where the tool is running. For more information, see Setting up security on page 85.

2. In the FactoryTalk Directory Server Location Utility, click the Browse button beside Computer hosting directory server.



FactoryTalk Directory Server Configuration

Use the FactoryTalk Directory installed on:

○ This computer

● Remote computer

FTDServer

3. Select **Remote computer**, type the name of the Network Directory server computer, and then click **OK**.

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

Copy HMI project files to the secondary computer

It is easiest to copy an HMI server when it is not running.

If the HMI server you want to copy is running, stop the server before copying the HMI project files. If you're not sure whether the HMI server is running, follow the steps on, for copying an HMI server while it is running.

To stop a running HMI server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the HMI server's icon, and then select **Properties**.
2. In the Properties dialog box, ensure that the Startup type is set to On Demand.
3. Stop any clients (FactoryTalk View Studio, the FactoryTalk View SE Administration Console, or FactoryTalk View SE Client) that might be connected to the HMI server.

To copy the HMI project files to the secondary computer:

1. In Windows Explorer on the primary server computer, copy the HMI project folder, which has the same name as the HMI server, from the following location:
   - **..\Users\Public\Public Documents\ RSView Enterprise\SE\HMI Projects**
   - (Windows XP) **C:\Documents and Settings\All Users\Shared Documents\RSView Enterprise\SE\HMI Projects**
2. In Windows Explorer on the secondary server computer, paste the HMI project folder into the same location.

Set up HMI server redundancy options

In the primary HMI server's Properties dialog box, in the **Redundancy** tab, 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 in 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.

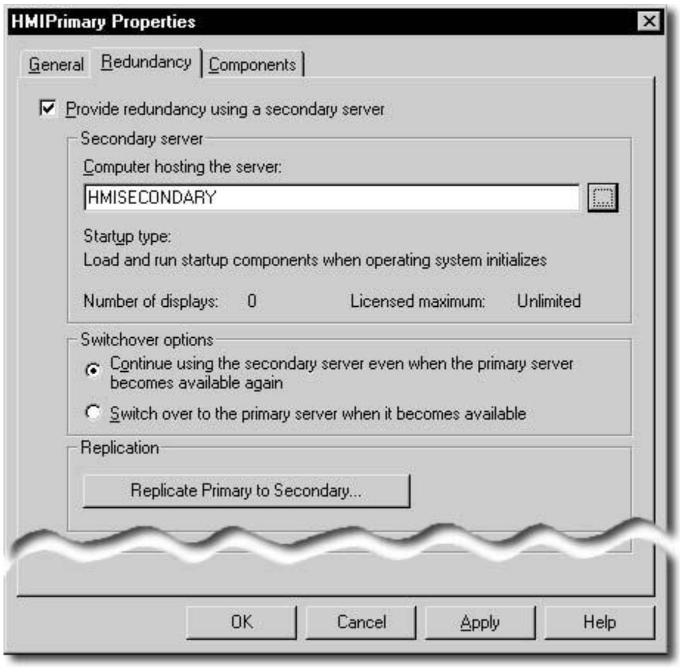
To set up redundancy in the Properties dialog box:

1. In FactoryTalk View Studio, in the Explorer window, right-click the HMI server's icon, and then select **Properties**.

2. Select the option, 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


HMIPrimary Properties

General | Redundancy | Components

☑ Provide redundancy using a secondary server

Secondary server

Computer hosting the server: HMISECONDARY

Startup type: Load and run startup components when operating system initializes

Number of displays: 0 Licensed maximum: Unlimited

Switchover options

● Continue using the secondary server even when the primary server becomes available again

○ Switch over to the primary server when it becomes available

Replication

3. In the **Redundancy** tab, select the check box, **Provide redundancy using a secondary server**.

4. Specify the name of the secondary HMI server's host computer, and the switchover option for this redundant server pair.

For details about options in the Redundancy tab, click **Help**.

Replicate changes to the secondary HMI server

While the primary HMI server is active, you can modify HMI project components, such as graphic displays, configuration files, and log files. However, any changes you make on the primary server are not synchronized automatically on the secondary server.

To synchronize server settings and contents, you must replicate changes manually, from the primary to the secondary server. The replicate operation copies the primary server’s configuration files, including settings in the Components tab, to the secondary server.

For example, if an On Active and a Shutdown macro are selected for the primary server, the settings will be replicated to the secondary. This means that the same macros specified for the primary server will run when the secondary server becomes active or shuts down.

Information that is not included in the replicate operation includes datalog files generated at run time, the current value of HMI memory tags, retentive tags, and the HMI tag alarm suppressed list.

HMI server replication is only available for redundant application server pairs in a FactoryTalk View SE network distributed application.

For replication to succeed, the primary server must remain active and the secondary server must remain on standby, for the duration of the replicate operation.

For information about checking server states, see Monitoring the status of application servers on page 353.

To replicate changes from the primary to the secondary server:

1. In FactoryTalk View Studio, in the Explorer window, right-click the primary HMI server’s icon, and then select **Properties**.

2. Click the Redundancy tab, and then click **Replicate Primary to Secondary**.

   Tip: The **Replicate Primary to Secondary** button is available only if the primary server is active when you select the Redundancy tab.

   Follow the instructions on screen. After replication is complete, the secondary server is restarted.

Determining the Active HMI server in a redundant pair

At run time, 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 364. For information about other server states, see Monitoring the status of application servers on page 353.

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.

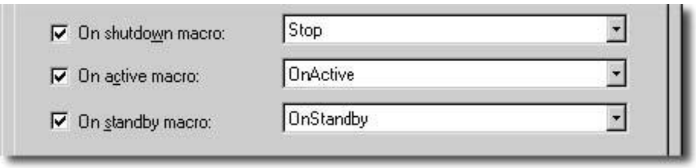
An HMI server is considered to be ready when:

- 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 and HMI tag alarm services are started and completely initialized.
- All alarmed HMI tags are on scan and ready to detect alarms.
- HMI tag alarm state synchronization is finished. For more information, see Synchronize time clocks on redundant computers on page 378.

For information about when other application servers (RSLinx Enterprise 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.



☑ On shutdown macro: Stop
☑ On active macro: OnActive
☑ On standby macro: OnStandby

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 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 failover to the Standby server.

You might have to re-run 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 them, the primary server will become active, and the secondary server will go on standby.

Determining which HMI tag alarm state to use

During the network disconnection, HMI tag alarm states are maintained at both the Active primary and the Active secondary HMI server.

When the network is restored, after the HMI servers determine which server will be active, the following criteria are used to determine which alarm states to retain:

- Most recent user input to alarm information—if the time stamp for most recent user input is the same at both servers, use the number of alarm transactions.
- Number of alarm transactions—if the number of alarm transactions is the same at both servers, use the greatest interval between the first and last transaction.
- Greatest interval between first and last alarm transaction—if the interval between the first and last transaction is the same at both servers, use the most recent transaction.

If all alarm state criteria are the same at the primary and secondary HMI servers, the state of HMI tag alarms on the primary server is used at the new, Active HMI server.

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 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 367.

About 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 check box, beside users you want to give permission to switch servers.

For information about FactoryTalk Security, see Setting up security on page 85.

What happens when the primary HMI server fails

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 380.
- 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.

Within five seconds, Diagnostic messages are logged at connected client computers, notifying operators of the server failure.

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 within 30 seconds of it becoming active.

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 367.

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 368.

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 you select in the **Redundancy** tab of the HMI server's Properties dialog box. For details 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 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 368.

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 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 and alarm properties at run time

You can make changes to HMI tags and their alarm 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 you can modify at run time, and what needs to happen before each change takes effect.

Use 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.
		The Security Code property is read using VBA code.
		The Security Code property is shown in the Tag Browser.
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 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 VBA code.
		The Low EU or High EU property is shown in the Tag Browser.
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 VBA code.
		The Engineering Units property is shown in the Tag Browser.
		An HMI tag alarm summary object is shown for the first time, in a FactoryTalk View SE Client, and the HMI tag alarm system is running. After the initial display, additional changes to the Units property will not show until the client is restarted.
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.
		Or 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 VBA code.
		The Item Canonical Data Type property is shown in the Tag Browser.
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 VBA code.
		The Initial Value property is shown in the Tag Browser.
		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 VBA code.
		The Contact Close or Contact Open property is shown in the Tag Browser.
Length	HMI string tags	Immediately. The Value property is updated to use the new Length value as soon as the property is modified.
		Or when:
		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 VBA code.
		The Length property is shown in the Tag Browser.

Tip: The Name and Tag Type properties of an HMI tag cannot be modified, after the tag is created.

Modifying HMI tag alarm properties

You can modify the properties of HMI tag alarms at run time. For the changes to take effect, you must run the **AlarmAcceptEdits** command.

When AlarmAcceptEdits runs, all alarm property changes made since the last time AlarmOn was run, are reflected in HMI tag alarm summaries on connected clients.

Optionally, you can specify an area name when you run the **AlarmAcceptEdits** command. If you do this, only changes made in the specified area, since AlarmOn was last run, will take effect. For details, click **Help** in the **Command Wizard**.

Tip: If you run AlarmAcceptEdits while a redundant HMI server pair is synchronizing alarm states, the secondary HMI server will not contain the latest changes. To update the secondary server, you must replicate changes made on the primary server. For details, see Replicate changes to the secondary HMI server on page 366.

Example: Modifying HMI tag alarms in a redundant HMI server pair

To ensure that changes to HMI tag alarm properties are copied to the secondary server in a redundant HMI server pair, follow these steps:

1. Modify the property. For example, change the severity for an analog alarm threshold.
2. Run the **AlarmAcceptEdits** command.
3. Replicate the change to the secondary HMI server. For instructions, see Replicate changes from the primary to the secondary server on page 366.

   Note: If you replicate changes from the primary to the secondary HMI server before running the **AlarmAcceptEdits** command, the primary and secondary HMI projects will not be synchronized.

How alarm property changes take effect

The following table lists the HMI tag alarm properties you can modify at run time, and the effect each change will have, the next time AlarmAcceptEdits is run.

Use information in the table to ensure that the changes you make are reflected as expected, in HMI tag alarm summaries on connected clients.

If this property	**Is changed to this value**	**This is what happens when AlarmAcceptEdits is run**
Alarmed (Alarmed or Not Alarmed)	True	The newly alarmed tag is monitored for alarms. Additionally, existing alarm events with the same name as the newly alarmed tag are removed from HMI tag alarm summaries.
	False	Existing alarm transactions for the tag are removed from HMI tag alarm summaries.
Label	Any change	New alarm transactions reflect the new Label value. Existing alarm transactions continue to use the old Label value.
Severity	Any change	If the tag is currently in alarm, a new alarm transaction is generated, based on the new Severity value.
Analog Level	Level is added	Existing alarm transactions for the tag are removed from HMI tag alarm summaries.
	Level is added or removed	If the change results in a new alarm state for the tag, a new alarm transaction is generated based on the new state.
Analog Threshold	Any change	If the change results in a new alarm state for the tag, a new alarm transaction is generated based on the new state.
Analog Direction	Increasing or Decreasing	Existing alarm transactions for the tag are removed from HMI tag alarm summaries. If the change results in a new alarm state for the tag, a new alarm transaction is generated based on the new state.
Digital Alarm Type	On (from Off) *or* Off (from On)	A new alarm transaction is generated. If the tag is currently in alarm, the transaction is OutofAlarm; if the tag is currently out of alarm, the transaction is InAlarm.
	Any Change, Changes to On, or Changes to Off (from Off or On)	If the tag is currently in alarm, an OutofAlarm transaction is generated.
	On (from Any Change, Changes to On, or Changes to Off)	If the tag's value is On, a new alarm transaction is generated. If the tag's value is Off, no new transaction is generated, and existing alarm transactions remain in HMI tag alarm summaries.
	Off (from Any Change, Changes to On, or Changes to Off)	If the tag's value is Off, a new alarm transaction is generated. If the tag's value is On, no new transaction is generated, and existing alarm transactions remain in HMI tag alarm summaries.
In Alarm Messages Out of Alarm Messages Acknowledge Messages Identification Out of Alarm Label Deadband	Any change	The change takes effect, for any new or existing alarm transactions associated with the modified tag. **Tip**: The **AlarmAcceptEdits** command will not apply changes to the contents of User Default messages, for theInAlarm Messages, Out of Alarm Messages, and Acknowledge Messages properties.

Acknowledge (bit)	Any change	The change takes effect, for any new or existing alarm transactions associated with the modified tag. If an Acknowledge Bit tag is added with the Auto Reset property set to True, the Acknowledge Bit tag is set to 0.
Handshake (bit)	Any change	The change takes effect, for any new or existing alarm transactions associated with the modified tag. If a Handshake Bit tag is added, and alarming is started with handshaking turned on (AlarmOn /H), and if the alarm tag is in alarm, the Handshake Bit tag is set to 1. If a Handshake Bit tag is added with the Auto Reset property set to True, and alarming is started with handshaking turned on (AlarmOn /H), and if the alarm tag is *not* in alarm, the Handshake Bit tag is set to 0.

Tip: The **AlarmAcceptEdits** command is for HMI tag alarms only. The command is not required to effect online changes to alarm definitions in a Tag Alarm and Event Server. For information about modifying FactoryTalk tag-based alarm properties, see the FactoryTalk Alarms and Events Help.

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.

Synchronize time clocks on redundant computers

HMI servers manage the synchronization of HMI tag alarm states, between primary and secondary servers.

For example, if there are five unacknowledged alarms at the primary server when it fails, the same five alarms will be show at the secondary server, when the failover is complete. Alarm states are also synchronized when the system switches back to the primary server.

To ensure tight synchronization of alarm states, synchronize clocks on the primary and secondary HMI server computers to a time server.

If the clocks are not synchronized, when a failover occurs, multiple alarms or inconsistent information might be shown in HMI tag alarm summaries, on connected clients.

Tip: You can set up a Microsoft Windows domain to include a time-synchronization service. For details, see Windows Help for setting up the domain.

About alarm monitoring on the secondary server

While the primary HMI server is active, the secondary server runs its alarm monitoring system in backup mode. This means that alarm states are synchronized even if you have not set up the secondary server to start alarm monitoring on demand.

The backup mode that runs on the secondary server only keeps alarm states synchronized; it does not detect alarms.

When the system fails over to the secondary server, alarm monitoring starts on that server automatically, as if it was running on the primary server.

When the system switches back to the primary server, alarm monitoring starts on that server automatically, while the secondary server returns to standby mode.

Tip: If FactoryTalk View SE is monitoring a large number of HMI tags for alarms, it is possible that alarms might be missed for tags that go into and out of alarm quickly. This might happen while the system is failing over to the secondary or switching back to the primary server.

Centralize storage of diagnostic and alarm log 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 and HMI tag alarm 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 and alarm logs to buffer logged data, in the event that communications with the ODBC database are lost.

For information about setting up a central ODBC database:

- For diagnostic logs, see Logging system activity on page 383.
- For HMI tag alarm logs, see Setting up HMI tag alarms on page 217.

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 running 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 368.

For information about creating macros, see Adding logic and control on page 633.

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 from the primary to the secondary HMI server, see Replicate changes to the secondary HMI server on page 366.

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.

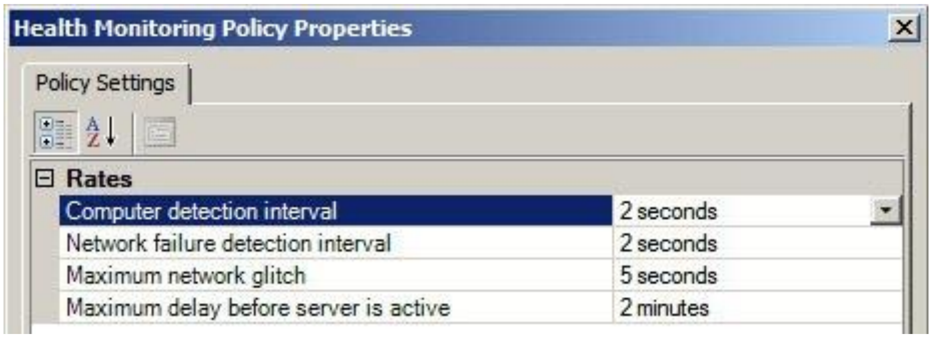
The system does the following connection monitoring:

- 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 373.

You can change the default settings, in the Health Monitoring Policy Settings dialog box.



Computer detection interval	2 seconds
Network failure detection interval	2 seconds
Maximum network glitch	5 seconds
Maximum delay before server is active	2 minutes

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 380.

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 368.

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 387.

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** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **FactoryTalk Help**.

2. In the Contents tab, open the book, Diagnostic messages. Look under:

   - **Log diagnostic messages** for information about FactoryTalk Diagnostics logging.
   - **View diagnostic messages** for information about the Diagnostics Viewer.

   You can also gain access to the FactoryTalk Diagnostics Help by clicking **Help** in dialog boxes used to set up Diagnostics logging and the viewer.

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 run time, if the FactoryTalk Diagnostics List is visible, an operator can alert the plant's control systems engineer 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 operation of the system and make the necessary corrections.

For more information, see Setting up message routing on page 387.

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 run time.
- **Information** messages indicate that a process or action has completed successfully. For example, an information message is logged when a user logs on 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 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.

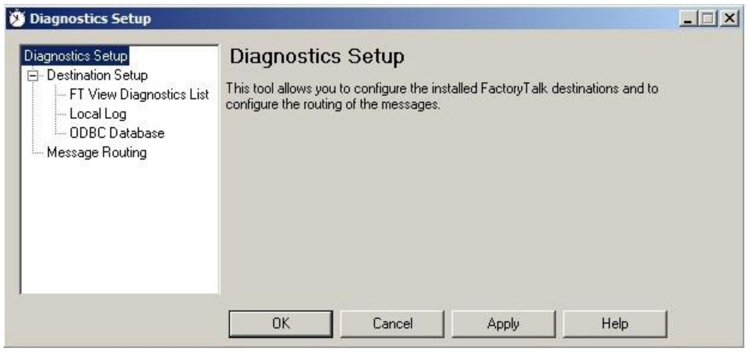
Setting up FactoryTalk Diagnostics in FactoryTalk View

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**.



This tool allows you to configure the installed FactoryTalk destinations and to configure the routing of the messages.

For details about options in the FactoryTalk Diagnostics Setup tool, click **Help**.

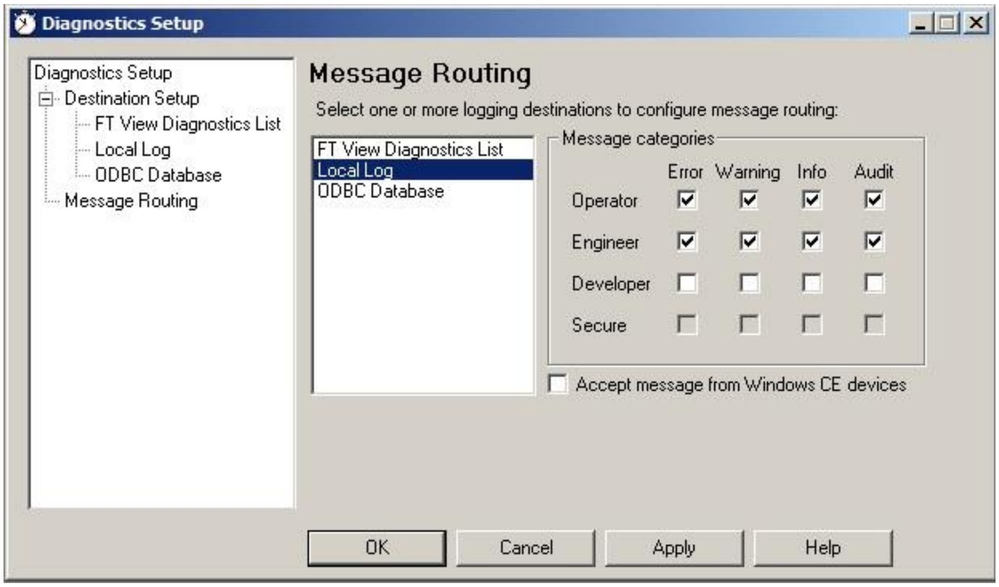
Setting up message routing

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.



Message Routing

Select one or more logging destinations to configure message routing:

FT View Diagnostics List
Local Log
ODBC Database

Message categories

	Error	Warning	Info	Audit
Operator	☑	☑	☑	☑
Engineer	☑	☑	☑	☑
Developer	□	□	□	□
Secure	□	□	□	□

□ Accept message from Windows CE devices

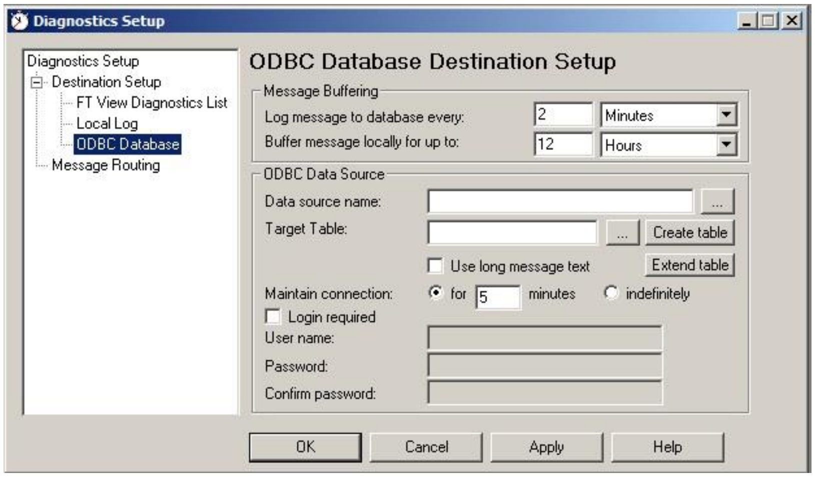
For more information about audience and severity, see Message categories on page 385.

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. FactoryTalk View supports Oracle, and Microsoft SQL Server ODBC-compliant databases.



ODBC Database Destination Setup

Message Buffering

Log message to database every: 2 Minutes

Buffer message locally for up to: 12 Hours

ODBC Data Source

Data source name: ...

Target Table: ... Create table

☐ Use long message text Extend table

Maintain connection: ● for 5 minutes ○ indefinitely

☐ Login required

User name:

Password:

Confirm password:

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 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 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.

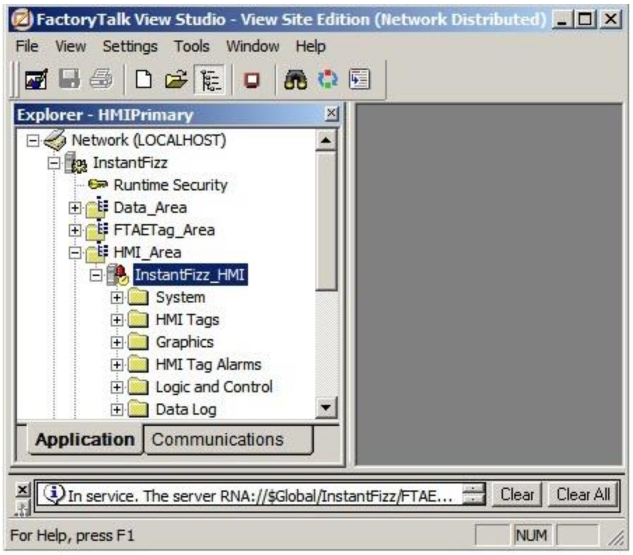
This is useful in the event that the ODBC-compliant database becomes unavailable, for example, due to a network failure. In such cases, messages will remain in the local log until the buffer time expires.

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 keep track of 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 387.



In service. The server RNA://$Global/InstantFizz/FTAE...

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.

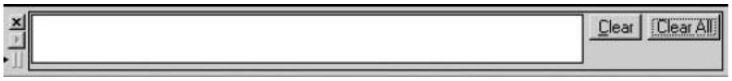
Showing or hiding the Diagnostics List

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 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

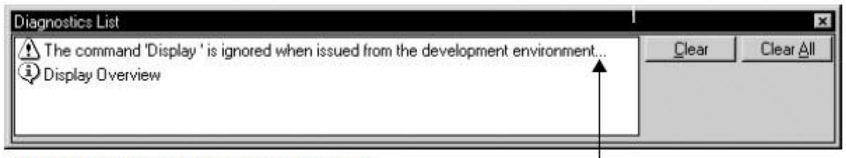
When the Diagnostics List is undocked, you can make it any size you want. For example, to view more than one message at a time, make the list larger.

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 command 'Display ' is ignored when issued from the development environment...
ⓘ Display Overview

An ellipsis indicates a truncated message.
To view the entire message, resize the Diagnostics List.

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 (at the top of the list), or the selected message, click **Clear**.
- To clear all messages in the list, click **Clear All**.

Clearing a message in the Diagnostics List does not delete the message from the Diagnostics log.

Viewing FactoryTalk Diagnostics logs

Use the FactoryTalk Diagnostics Viewer to view the contents of Diagnostics logs.

To open the FactoryTalk Diagnostics Viewer:

- In FactoryTalk View Studio or in the FactoryTalk View SE Administration Console, from the Tools menu, select **Diagnostics Viewer.**

  Alternatively, select **Start** > **All Programs** > **Rockwell Software** > **FactoryTalk Tools** > **Diagnostics Viewer**.

For details about setting up and using the viewer, from the viewer's Help menu, select **FactoryTalk Diagnostics Viewer Help**.

To open the Diagnostics Viewer at run time:

1. In an FactoryTalk View graphic display, create a button that an operator can use to open the Diagnostics Viewer.

2. For the button's press action, type the following command:

   ```
   AppStart "C:\Program Files\Rockwell Software\RSView
   Enterprise\ActivityLogViewer.exe"
   ```

   You must include the quotation marks, because there are spaces in the parameter.

   Tip: At run time, 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. You can also use the programmatic solution documented in Answer ID 9041, in the Rockwell Automation KnowledgeBase.

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 controller instruction faceplates 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 run time.

About graphic displays

A graphic display represents a run-time 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 run time.

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).

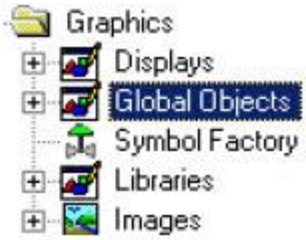
In FactoryTalk View Studio, you can use the Graphics Import Export Wizard to export graphic display information to an XML file, or to import a graphic display XML file into an application.

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 671.

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 you can create in FactoryTalk View can be a global object, except for ActiveX controls, OLE objects, and HMI tag alarm summaries.

Working with global objects

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 their base objects.

At run time, when a standard display that contains reference objects is run, the global object display (or displays) that contains the base objects is loaded in the background.

Changes you make to the base object are reflected in all of the reference objects linked to it, the next time displays containing the reference objects are opened, or refreshed by closing and reopening them.

For information about creating global object displays, see Creating and working with global object displays on page 420.

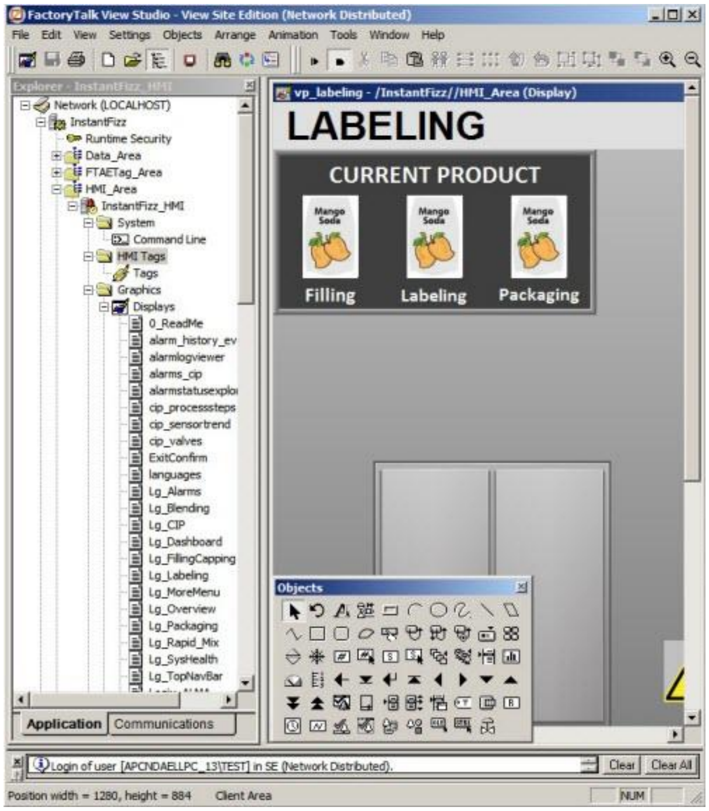
Parts of the Graphics editor

Use the Graphics editor to create and modify standard graphic displays, global object displays, and graphic libraries. Where you open the Graphics editor determines the type of display you create.

To open the Graphics editor:

1. In FactoryTalk View Studio, in the Explorer window, open the Graphics folder.
2. Right-click one of the Displays, Global Objects, or Libraries icons, and then select **New**. Or, double-click an existing standard display, global object display, or graphic library.

When the new or existing display opens, you are working in the Graphics editor.



Login of user [APCNDAELLPC_13\TEST] in SE (Network Distributed).

Position width = 1280, height = 884 Client Area

The Graphics editor consists of these parts:

The toolbars contain buttons for commonly used menu items. The previous illustration shows three toolbars, but the Graphics editor has others, including toolbars for foreground and background colors, pattern styles, and aligning objects.

You can hide or show toolbars using the **View** menu, and move the toolbars around on the screen. For more information about toolbars, see Using the toolbars on page 400.

The display area is the graphic display itself.

To set up the run-time appearance and behavior of a display, right-click the display, and then select **Display Settings**.

In the Display Settings dialog box, you can set up a title bar for the display, specify its size, position, and background color, and determine whether it has a border at run time.

The appearance of a display in the Graphics editor is similar to its run-time appearance (except for the content of the title bar, and the position of the display).

Some changes made at design time, such as resetting the zoom feature, do not affect the appearance of the display at run time.

For more information about display settings, see Setting up the appearance and behavior of a graphic display on page 434.

The Diagnostics List shows information about system activity, such as command and macro usage, tag reads and writes, communication errors, and system warnings for the computer where FactoryTalk View is installed.

The status bar describes the action associated with the selected menu item or button. The status bar also shows the x- and y-coordinates, width, and height of the selected graphic object.

Viewing display contents in the Object Explorer

The Object Explorer shows a tree-list of all the objects in a graphic display. Objects are listed in front-to-back order, with the most recently created objects at the front. Grouped objects are listed as expandable items in the tree, with a + icon.

When you click an object in the Object Explorer, it is selected in the display, and selection handles appear around it automatically. This means you can use the Object Explorer to select an object hidden by others in a display, instead of bringing the object to the front.

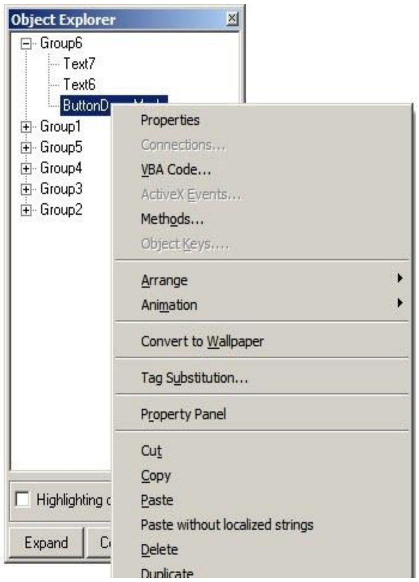
Tip: The Object Explorer does not show wallpaper objects, or the objects within ActiveX composite objects.

You can keep the Object Explorer open while you work on different objects, and in different displays.

You can also use the Object Explorer to highlight specific types of objects, and objects that have animation, or a tag or expression assigned to them.

To open the Object Explorer:

In FactoryTalk View Studio, open the Graphics editor, right-click the graphic display, and then select **Object Explorer**.



To highlight different types of objects in a display:

1. In the Object Explorer, click **Settings**.
2. In the Highlight Settings dialog box, select boxes beside the items that you want highlighted, and then click **OK**.
3. In the Object Explorer, select the check box, **Highlighting on**.

The objects are highlighted in red in the Object Explorer and in the graphic display.

Viewing object properties in the Property Panel

The Property Panel shows the properties of graphic objects and the values assigned to the properties. It also shows the tags or expressions assigned to an object’s connections.

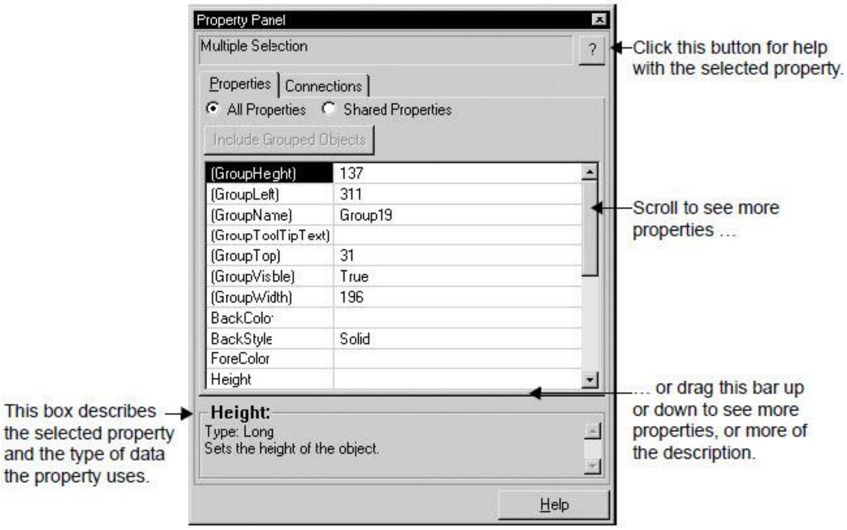
Use the Property Panel to modify the properties of graphic objects, and to assign tags and expressions. You can select multiple objects, and then make changes to the common properties of all of the objects at the same time.

You can keep the Property Panel open as you work in the Graphics editor, and you can drag the panel’s borders to make it larger or smaller.

To open the Property Panel:

Right-click a graphic object in a display, and then select **Property Panel**.

You can also click on the tool bar, to show or hide the panel.



(GroupHeight)	137
(GroupLeft)	311
(GroupName)	Group19
(GroupToolTipText)	
(GroupTop)	31
(GroupVisble)	True
(GroupWidth)	196
BackColor	
BackStyle	Solid
ForeColor	
Height	

Height:
Type: Long
Sets the height of the object.

Click this button for help with the selected property.

Scroll to see more properties ...

... or drag this bar up or down to see more properties, or more of the description.

This box describes the selected property and the type of data the property uses.

For details about options in the Property Panel, click **Help**.

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

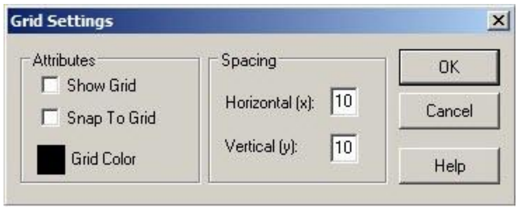
To magnify or reduce the view of a graphic display, use the Zoom In or Zoom Out option on the **View** menu. Zoom In magnifies objects, while Zoom Out reduces magnification.

In the Properties tab of the Display Settings dialog box, if the Size property for a display is set to Specify Size in Pixels, you can use **Zoom to Fit** to resize the display to fit the workspace window.

If the Size property is set to Use Current Size, Zoom to Fit behaves in the same way as Cancel Zoom; it returns a display to its original size.

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 any time during the drawing process.



Attributes

□ Show Grid

□ Snap To Grid

■ Grid Color

Spacing

Horizontal (x): 10

Vertical (y): 10

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.

If you select **Snap to Grid**, the grid is activated, and the next object you draw or position will be pulled to the closest grid point. Turning on the grid does not affect the placement of existing objects.

If you clear **Snap to Grid**, the grid is passive, and does not affect the position of objects. Turn off the grid to draw or position an element between the grid lines.

Using the toolbars

The toolbars are a convenient way to perform an action. You can:

- Hide or show the toolbars using the items on the View menu. If there is a check mark beside the toolbar name, the toolbar is visible.
- Drag the toolbars anywhere on the screen.
- Dock the toolbars to an edge of the FactoryTalk View Studio workspace (except the ActiveX Toolbox).

Selecting a drawing tool

The Objects toolbar contains tools for creating, selecting, and rotating objects. The tools are also available on the **Objects** menu.

To draw an object, click a tool in the toolbox or on the **Objects** menu. When you do this, the pointer changes to show which tool is active.

To stop using a drawing tool, click the Select tool, or click another drawing tool.

Selecting objects

To work with an object in a graphic display, first you must select it.

To select a single object in a display, click on the object using the left mouse button. Handles appear along the edges of a selected object.

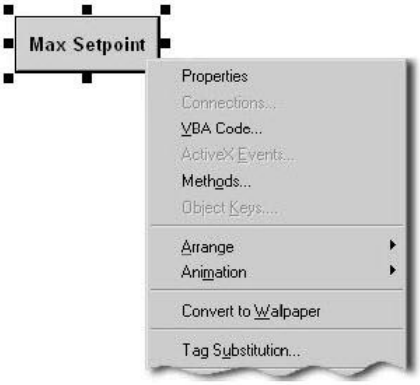
To select several objects, click in the graphic display at one corner of the group, and then drag the mouse diagonally, to draw a bounding box around the objects. When you release the mouse button, all the objects within the bounding box are selected.

You can also select a group of objects by holding down the Ctrl key while you click each individual object.

To cancel selection of an object or a group of objects, click in a part of the graphic display that does not contain any objects.

Using shortcut menus to perform actions quickly

When you right-click in a graphic display, or on a graphic object, a shortcut menu opens, as shown in the following illustration.



When you right-click an object ...

... the shortcut menu opens.

The shortcut menu contains items for working with the selected object.

The items on the shortcut menu depend on where you click: when you right-click an object, the menu contains items relevant to the object; when you right-click a display, the menu contains items relevant to the display.

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 449.

Copying objects

To copy objects, you can:

- Drag and drop objects in the same display.
- Drag and drop objects between displays, or from a graphic library to a display.
- Copy and paste objects.

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 323.

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.

3. 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.

To drag objects between displays:

1. Open both displays (or a graphic library and a display).
2. Position or resize the displays so both are visible.
3. Select one or more objects.
4. Click the selected object and drag it to the new display.

   If you selected several objects, dragging one of the objects copies all the selected objects. The objects maintain their position relative to each other.

Copying and pasting objects

You can cut, copy, or paste objects using the menu items on the **Edit** menu or the buttons on the toolbar.

Once you cut or copy an object, you can paste it anywhere in the drawing area of:

- The same graphic display.
- A graphic display in the same or a different application.
- A graphic library in the same or a different application.

To cut or copy objects:

1. Select one or more objects.
2. From the **Edit** menu, select **Cut** or **Copy**, or click ✂ or ⎘ on the Graphics toolbar.
   - To remove the original object, click **Cut**.
   - To retain the original object, click **Copy.**

To paste objects:

1. Click in the display, or library to paste to.
2. From the **Edit** menu, select **Paste**, or click 📋 on the Graphics toolbar.

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.

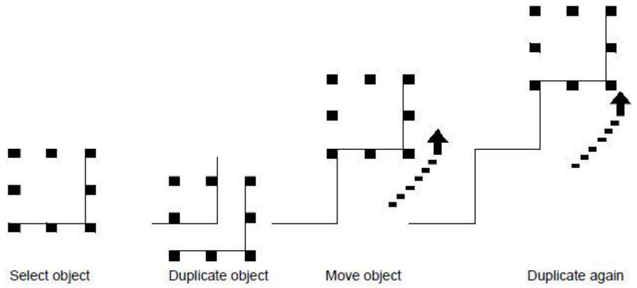
When an object is duplicated, any animation attached to the object is also duplicated.

If an object group is duplicated, the new copy of the group can be ungrouped into individual objects. For more information about grouping objects, see Grouping objects on page 411.

To duplicate an object:

Right-click the object in a display, and then select **Duplicate**, or click on the toolbar.

Duplicate works until the object is no longer selected.

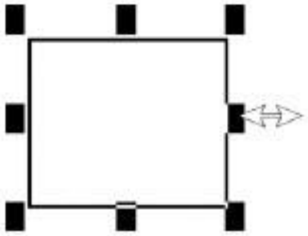


Resizing and reshaping objects

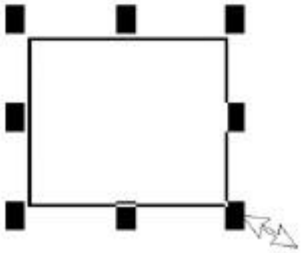
When you select an object, handles appear around it.

When you position the pointer over a handle, the pointer changes to a double-headed arrow. You can then click on 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 on 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 501.

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.

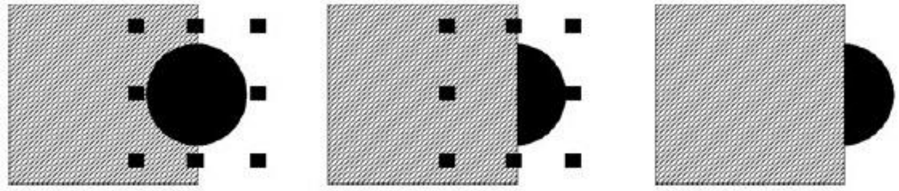
- Combine several objects into a group that behaves as a single object.
- Separate a grouped object into its component objects.

Stacking objects

Objects in a graphic display are stacked in the order they are created, with the most recently created object on top.

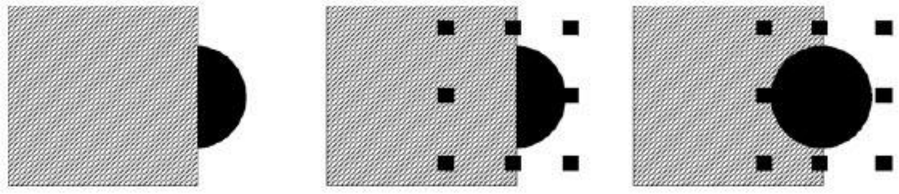
To change the order, move objects to the front or back of the stack.

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.

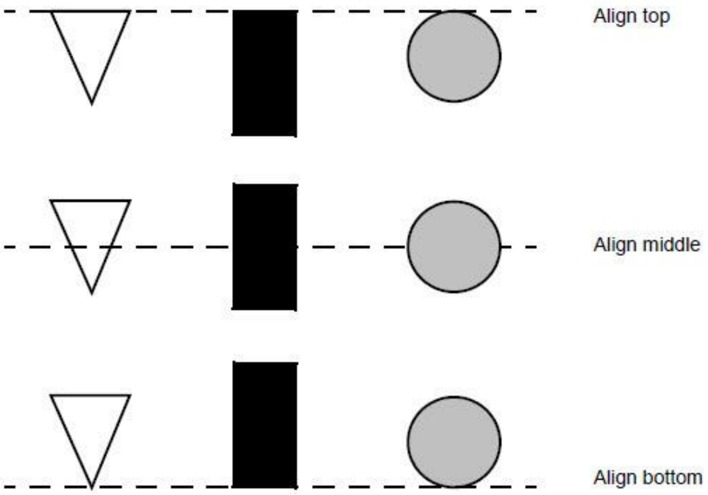
Aligning objects

To line up the tops, bottoms, or sides of objects in a display, align the objects with each other, or with the display grid.

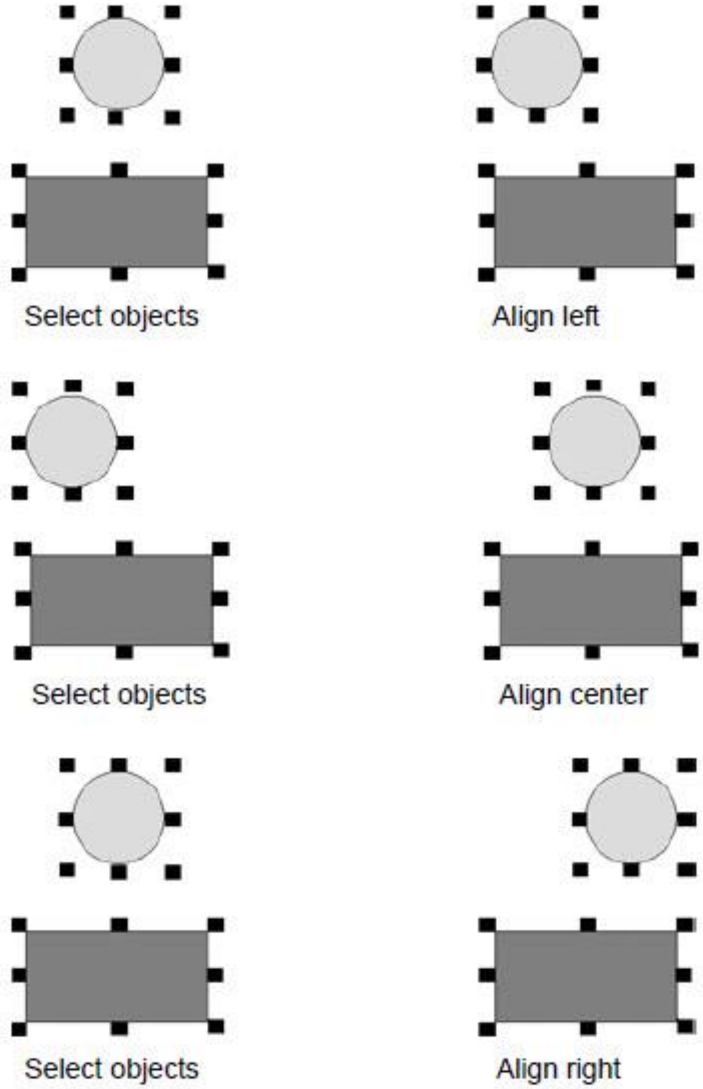
To do this	Click this button or menu item
Align the selected object (or objects) with the left-most selected object.	Align Left
Align the selected object (or objects) with the horizontal center of the largest selected object.	Align Center
Align the selected object (or objects) with the right-most selected object.	Align Right

Align the selected object (or objects) with the top-most selected object.	Align Top
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 bottom-most selected object.	Align Bottom
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

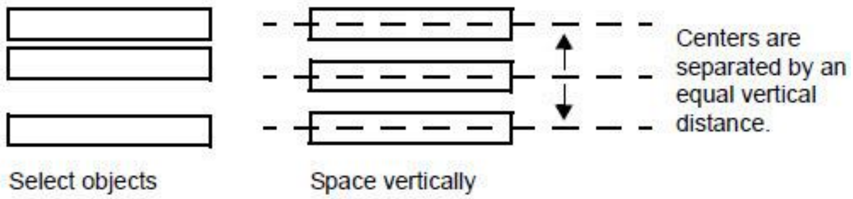


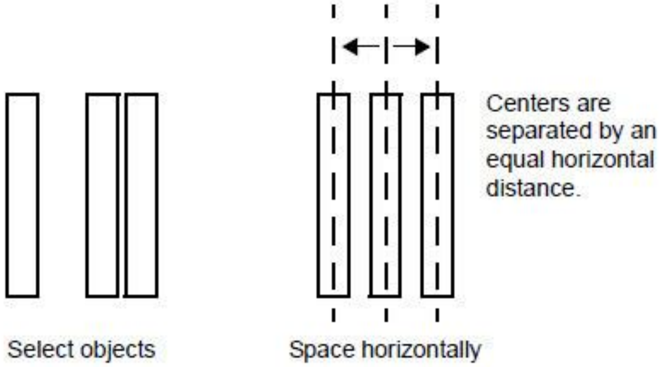
Spacing objects

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 horizontally.	Space Horizontal
Place the centers of the selected objects an equal distance apart vertically.	Space Vertical

Spacing objects vertically and horizontally



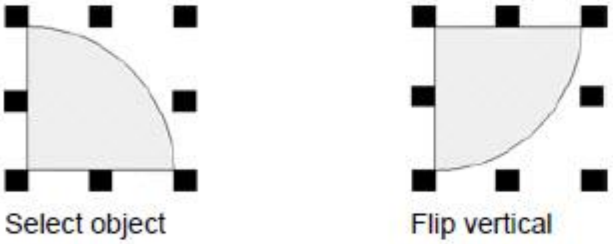


Flipping objects

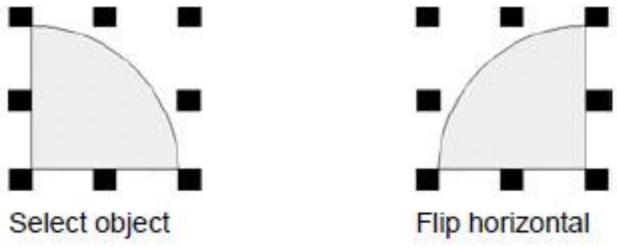
To move an object in a graphic display to a position that is a mirror image of its original position, flip the object vertically or horizontally.

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

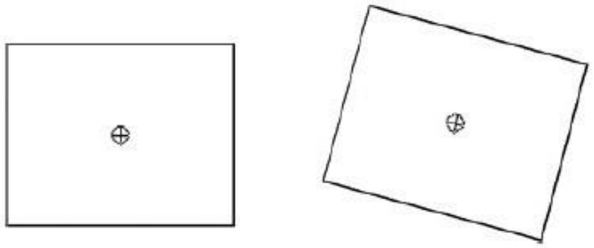


Flipping objects horizontally

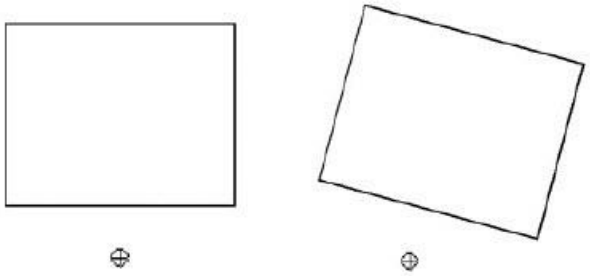


Rotating objects

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.

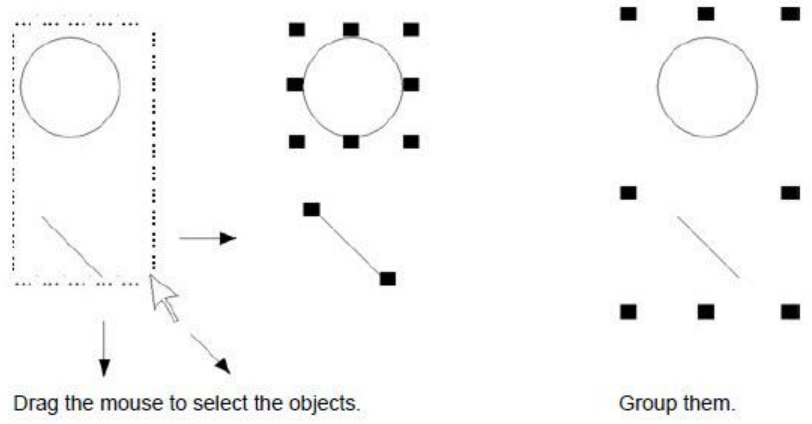
For details, see Rotating objects on page 410.

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.

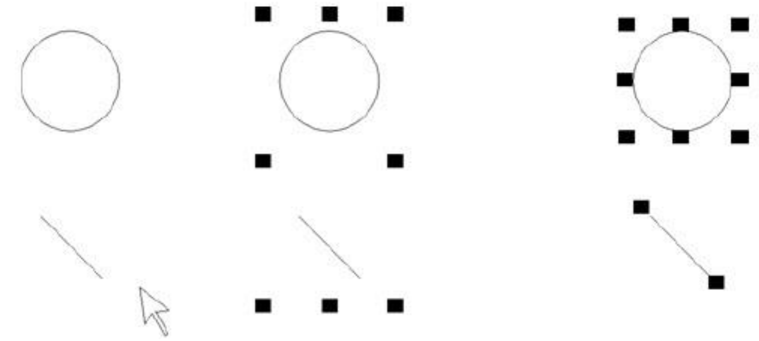
Group animation generally takes precedence over the animation attached to individual objects. For more information, see Applying animation to object groups on page 519.

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 animation attached to the group, because the group no longer exists. However, any animation attached to the individual objects remains active.



Modifying grouped objects

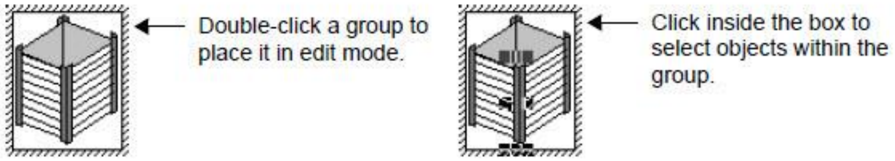
You can modify a group of objects without breaking up the group. This is particularly useful when animation is attached to the group, because ungrouping deletes the animation.

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 2 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 419.

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.

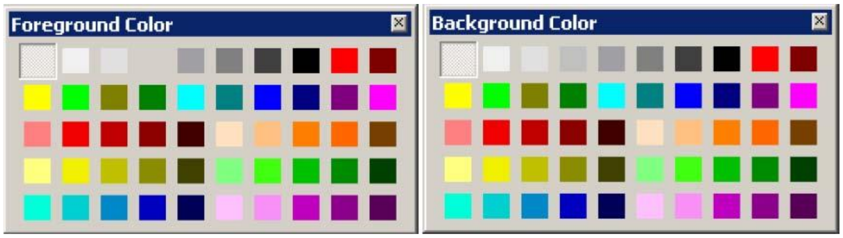
To see which objects are selected, look in the Object Explorer or the FactoryTalk View Studio status bar. For information about using the Object Explorer, see Viewing display contents in the Object Explorer on page 397.

To cancel group edit mode, click outside the group border.

Applying colors

The color palettes contain a range of colors that you can apply to graphic objects. You can select colors before you draw an object, or you can apply them to an existing object.

To show and hide the color palettes, from the **View** menu, select their names.



You can use the:

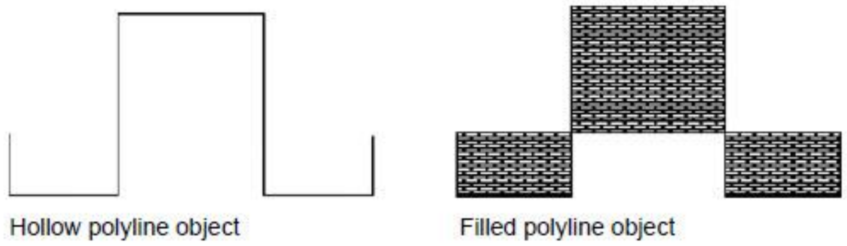
- **Foreground Color** palette to select a color for the outline of an object, for text, or for hollow objects.
- **Background Color** palette to select a color for the inside of an object or for solid objects.

For objects with patterns, you can set the color of the pattern separately.

Applying pattern styles and colors

Use the Pattern Styles toolbar to apply patterns to graphic objects. You can select a pattern before you draw an object, or you can apply a pattern to an existing object.

An object's pattern is visible only if the object's background style is Solid. Objects with a transparent background have no pattern.



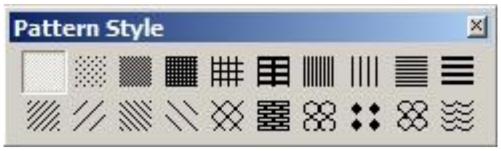
Pattern styles apply to the interior of objects. Closed objects such as rectangles, circles, polygons, text objects, and wedges are completely filled.

Open or partially open objects, such as polylines or freehand objects, can also take a pattern style. The Graphics editor draws an imaginary line from the start and end points, and then fills the object as a closed object.

The pattern color is not determined by the foreground or background color selected for an object. Instead, you must apply pattern colors separately.

To open the Pattern Styles toolbar:

Select **View > Toolbars > Pattern Styles**.

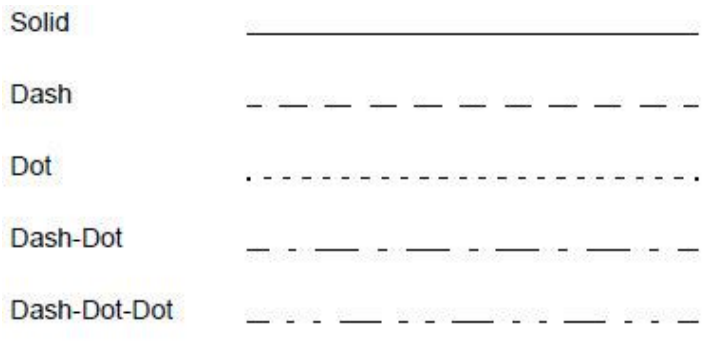


To apply a pattern color:

1. Right-click the graphic object, and then select **Properties**.
2. In the Properties dialog box, select the Pattern color box, and then select a pattern color.

Changing line properties

You can select a line object and change its width and style in the General tab of its Properties dialog box, or in the Property Panel. The line styles are:



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.

For trend objects, you can customize line width and color in the Pens tab of the Trend Properties dialog box. For more information, see Setting up trends on page 603.

Naming graphic objects

When you create a graphic object, it is assigned a name automatically.

In the Common tab of the object’s Properties dialog box, or in the Property Panel, you can give the object a new name.

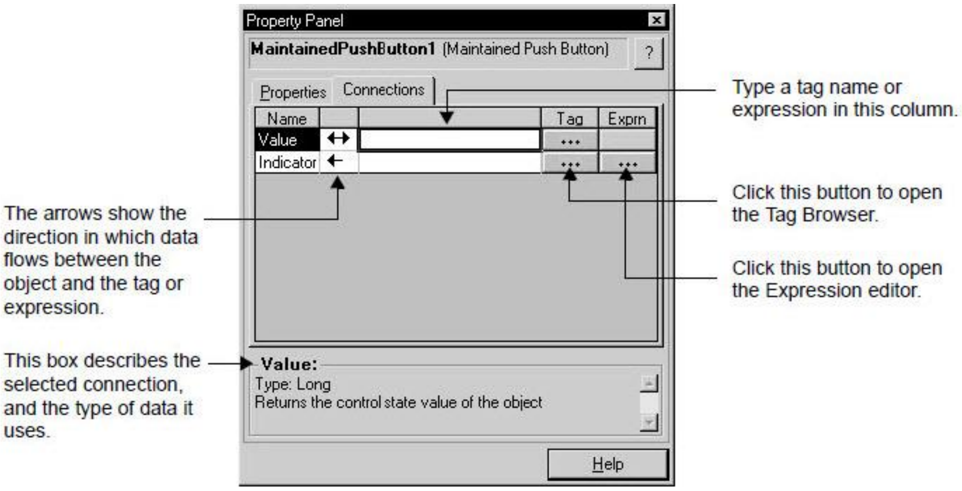
An object’s name is used when logging events for the object. It is also used with commands. For example, when using the **Invoke** command to call a method, you must specify the name of the object in which the method is implemented.

The object’s name is also shown in the Object Explorer. For more information, see Viewing display contents in the Object Explorer on page 397.

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



Property Panel

MaintainedPushButton1 (Maintained Push Button)

Properties | Connections

Name			Tag	Exprn
Value	↔		...	
Indicator	←		...	...

Value:
Type: Long
Returns the control state value of the object

Type a tag name or expression in this column.

Click this button to open the Tag Browser.

Click this button to open the Expression editor.

The arrows show the direction in which data flows between the object and the tag or expression.

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.

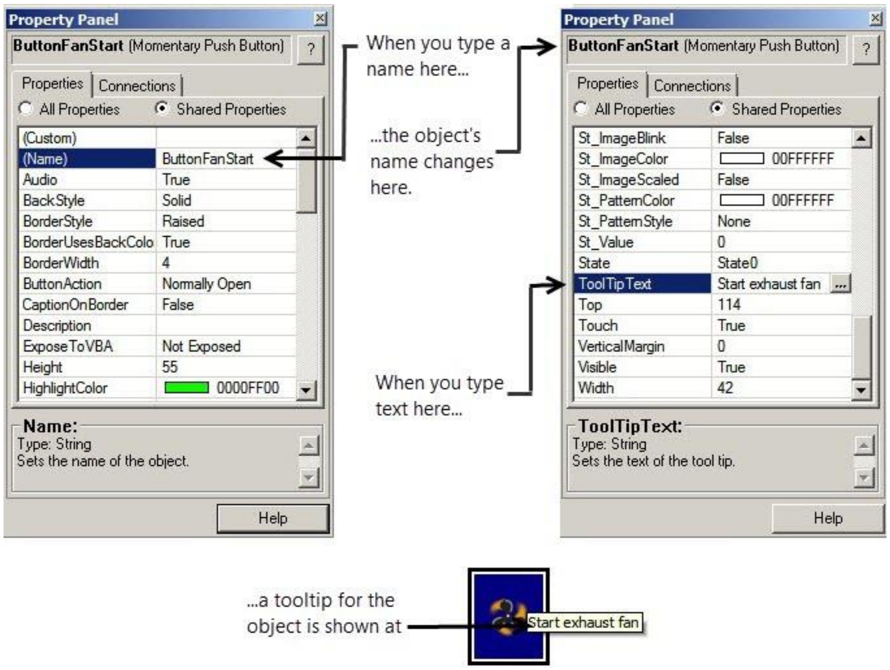
The arrows beside the connection names indicate the direction data flows between the tag or expression and the object:

- 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.

Adding tooltips to graphic objects

To provide information about a graphic object to an operator, add a tooltip to the object.

By default, an object has no tooltip. If you add a tooltip, it shows at run time, when the operator positions the pointer over the object for a few seconds.



Property Panel

ButtonFanStart (Momentary Push Button) ?

Properties | Connections

○ All Properties ● Shared Properties

(Custom)	
(Name)	ButtonFanStart
Audio	True
BackStyle	Solid
BorderStyle	Raised
BorderUsesBackColo	True
BorderWidth	4
ButtonAction	Normally Open
CaptionOnBorder	False
Description	
ExposeToVBA	Not Exposed
Height	55
HighlightColor	🟩 0000FF00

Name:
Type: String
Sets the name of the object.

When you type a name here...

...the object's name changes here.

Property Panel

ButtonFanStart (Momentary Push Button) ?

Properties | Connections

○ All Properties ● Shared Properties

St_ImageBlink	False
St_ImageColor	□ 00FFFFFF
St_ImageScaled	False
St_PatternColor	□ 00FFFFFF
St_PatternStyle	None
St_Value	0
State	State0
ToolTipText	Start exhaust fan ...
Top	114
Touch	True
VerticalMargin	0
Visible	True
Width	42

ToolTipText:
Type: String
Sets the text of the tool tip.

When you type text here...

...a tooltip for the object is shown at

Start exhaust fan

You can create a tooltip for an object in the Property Panel, or in the Common tab of the object's Properties dialog box.

Tooltips can consist of a single line or multiple lines of text.

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.

To search for and replace text strings:

1. Select the graphic object (or objects) that contain text strings you want to replace.

   Tip: To select all the graphic objects in a display, press **Ctrl+A**.

2. From the Edit menu, select **Tag Substitution**.

3. In the Tag Substitution dialog box, type the text you search for and replace, and then click **Replace**.

   For details about options in the Tag Substitution dialog box, click **Help**.

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.

Tip: 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.

To convert objects to wallpaper:

1. Select one or more objects.
2. Select **Edit > Wallpaper > Convert to Wallpaper.**

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.

Testing graphic displays

You can test the objects in a graphic display quickly, by switching to 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 651.

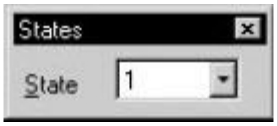
Tip: If objects in a graphic display are connected to tags in devices, to fully test the display you must set up communication with the devices or data servers.

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 Sates toolbar or the Property Panel.



To view an object's states using the States toolbar:

1. Select **View** > **Toolbars** > **States**.
2. Select one or more objects.
3. In the States toolbar, click the state you want to view. If you selected multiple objects, the toolbar shows the states common to all the objects.
4. To view the next state in the list, click it, or press the **Down Arrow** key. To view:

- The previous state, press the **Up Arrow** key.
- The first state, press the **Home** key.
- The last state, press the **End** key.

To view an object's states using the Property Panel:

1. Select one or more objects.
2. In the Property Panel, click the State property, and then click the state to view.
3. To view the next state quickly, double-click the row, or press **Enter** on the keyboard.

Creating and working with global object displays

Use the Graphics editor to create global object displays in the Global Objects folder, the same way you create standard displays in the Displays folder.

All of the objects and groups of objects created in a global object display are global objects. Any graphic object you can create in FactoryTalk View can be a global object, except for ActiveX controls, OLE objects, and HMI tag alarm summaries.

When you copy a global object into a standard graphic display (in the Displays folder), the copy is called a reference object. The original global object (in the Global Objects folder) becomes the copy's 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.

For information about setting up the properties that link base and reference objects, see Setting up the link properties of reference objects on page 501.

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.

Creating global object displays

To create a global object display, create a new 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, graphic objects in the display convert to global objects, except for ActiveX controls, OLE objects, HMI tag alarm summaries, 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, to the following location on the development computer:

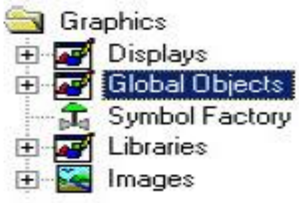
```
\Documents and Settings\All Users\Documents\RSView
Enterprise\SE\HMI Projects\<HMI Project Name>\Global
Objects
```

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*.

To create a new global object display:

- In FactoryTalk View Studio, in the Explorer window, right-click the Global Objects icon, and then select **New**.



The new global object display opens in the Graphics editor. Use tools in the editor to create global objects, just as you would create graphic objects in a standard display.

To create a global object display from an standard display:

1. In FactoryTalk View Studio, in the Explorer window, right-click the Global Objects icon, and then select **Add Component into Application**.

2. Browse to and select the standard display or library to add, and then click **Open**.

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 run time

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.

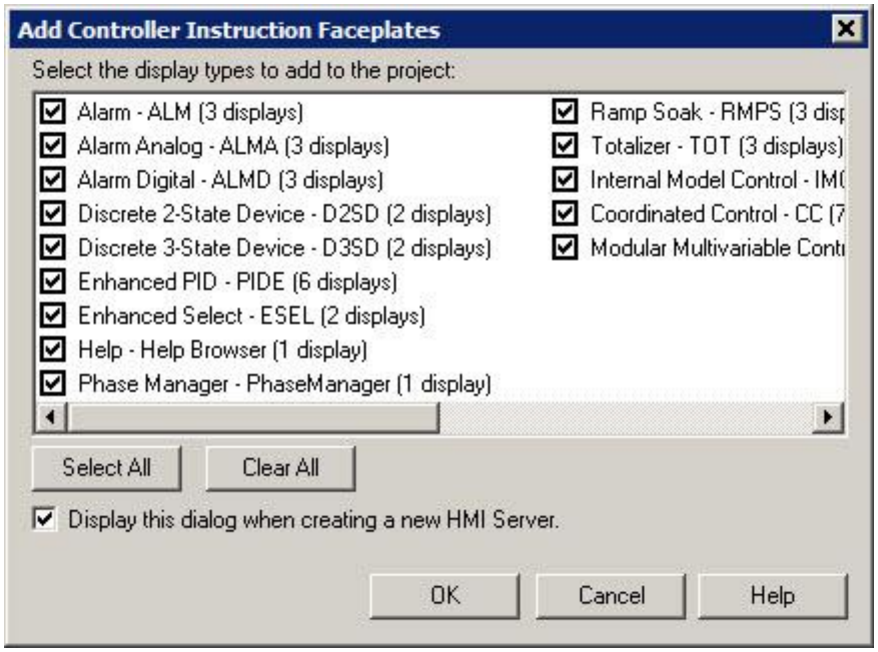
However, you can run a global object display in test display mode, when developing an application in FactoryTalk View Studio.

Tip: To minimize the number of global object displays that must be loaded in the background at run time, store the base objects used in an application on as few global object displays as possible.

Adding controller instruction faceplates to an application

When you add an HMI server to a network distributed or station application, or when you create a new local station application, you have the option to add the controller instruction faceplate displays that are installed with FactoryTalk View SE.

In the Add Controller Instruction Faceplates dialog box, you can select to add some or all of the faceplate displays. For details about options in the dialog box, click **Help**.


Add Controller Instruction Faceplates

Select the display types to add to the project:

☑ Alarm - ALM (3 displays)
☑ Alarm Analog - ALMA (3 displays)
☑ Alarm Digital - ALMD (3 displays)
☑ Discrete 2-State Device - D2SD (2 displays)
☑ Discrete 3-State Device - D3SD (2 displays)
☑ Enhanced PID - PIDE (6 displays)
☑ Enhanced Select - ESEL (2 displays)
☑ Help - Help Browser (1 display)
☑ Phase Manager - PhaseManager (1 display)
☑ Ramp Soak - RMPS (3 disp
☑ Totalizer - TOT (3 displays)
☑ Internal Model Control - IM(
☑ Coordinated Control - CC (7
☑ Modular Multivariable Conti

☑ Display this dialog when creating a new HMI Server.

You can also add the controller instruction faceplates later, after you create the application or HMI server. To do this, right-click the HMI server, and then select **Add Controller Instruction Faceplates**.

If you added a display previously, you can either replace the existing display or remove it from the list of displays.

Tip: Adding faceplate displays to an application affects the license count. Each added faceplate display (.gfx file) counts as one display for activation purposes. The corresponding global object display (.ggfx file) is not included in the count.

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 controller instruction faceplates displays in an application as they are, or you can copy and then paste the faceplates into existing standard and global objects displays.

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

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 and drop objects from the libraries into displays.

Location of library files

Library files are stored in the following folder on the local computer:

- \Users\Public\Public Documents\RSView Enterprise\SE\Libraries
- (Windows XP) \Documents and Settings\All Users\Shared Documents\RSView Enterprise\SE\Libraries

For information about changing the Libraries location, see the FactoryTalk View Site Edition Help.

Importing graphic files from third-party applications

FactoryTalk View can import the following types of files:

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 files, set up the computer's display properties to display more than 256 colors. This will ensure that 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 uses 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 you use, the more memory that is consumed:

- 16-color bitmaps consume 4 bits per pixel (½ 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 an 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.

Using the Image Browser to import images

Use the Image Browser to import images as needed while you set up graphic objects.

In the Image Browser you can:

- Import images into the application.
- Select the image to use on a graphic object.
- Delete images from the application.

To open the Image Browser:

Use one of these methods:

- In an object’s Properties dialog box, click the **Browse** button next to the Image box.

  Depending on the type of object, the Image box could be located on the General tab, Label tab, or States tab.

- With one or more objects selected, in the Property Panel click the Image property, and then click the **Browse** button.

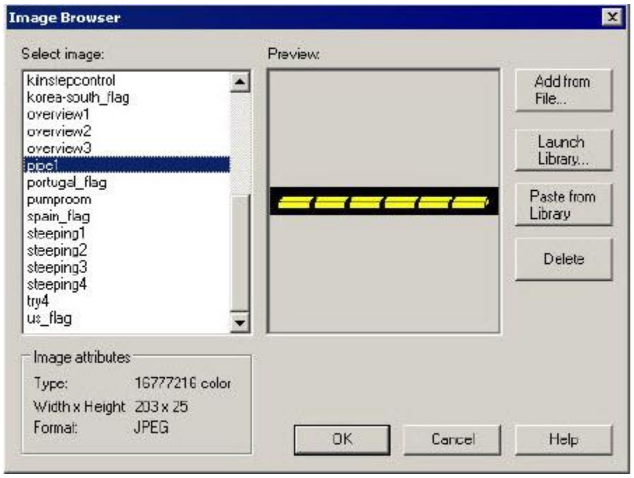


Image attributes
Type: 16777216 color
Width x Height 203 x 25
Format: JPEG

For details about using the Image Browser to import, select, and delete images, see Help.

You can also remove an image by clicking it in the Images folder, and then right-clicking Remove or Delete.

To import Symbol Factory objects directly into image container:

1. From the Image Browser window, click the **Launch Library…** button. This will open a Symbol Factory window.

2. Browse through the Categories and locate the graphic to be used. Select the graphic so it is highlighted. Click the **Copy** button in Symbol Factory. Symbol Factory will be minimized to the system tray.

3. From the Image Browser, click the **Paste from Library** button.

4. A dialog box will open and ask for a unique image name. Type the new name over the default name of the Symbol Factory object shown.

5. Click the **OK** Button in Image name dialog. The dialog box will close returning you to the Image Browser. The new image will be shown and the new image name shown in the image browser explorer.

6. Click the **OK** button in the Image Browser for the graphic to be placed in the current workspace.

Note that any Symbol Factory object, including Bitmap, DIB and metafile can be pasted into Image Browser where it will be stored as a bitmap.

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 run time, 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.

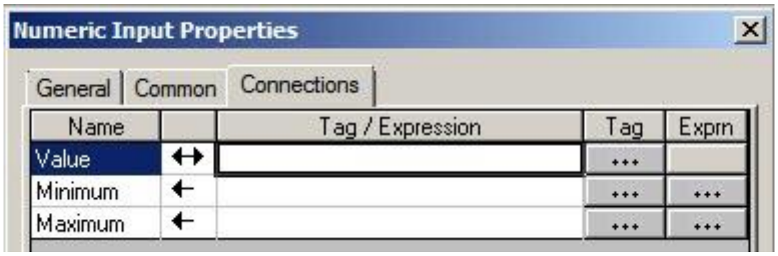
Parameter files and parameter lists are loaded using the **Display** command parameters */P* and */T*, respectively. For details, see the examples on Replacing tag placeholders using parameter lists on page 429.

Creating a tag placeholder

A tag placeholder is the cross-hatch character (#) followed by a number from 1 to 500.

You can use a tag placeholder to specify a value for a graphic object instead of (or as part of) specifying a tag name, expression, command, or embedded variable.

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 run time or at test display mode.



Name		Tag / Expression	Tag	Exprn
Value	↔		...	
Minimum	←		...	...
Maximum	←		...	...

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 run time, 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 run time 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**.

Tip: You can use wildcard characters in a parameter file entry only if the tag placeholder to be replaced is specified for an HMI tag alarm summary. In that case, a parameter file containing the entry **#1 = bean_*** would specify alarms associated with all tags beginning with **bean_** .

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 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 tag placeholder with a list of folder names

The tag database contains these tags:

Corn\Weight
Corn\Level
Corn\Temp

Bean\Weight
Bean\Level
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 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 run-time 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 run time 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 429 and Replacing tag placeholders using parameter lists on page 429.

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.

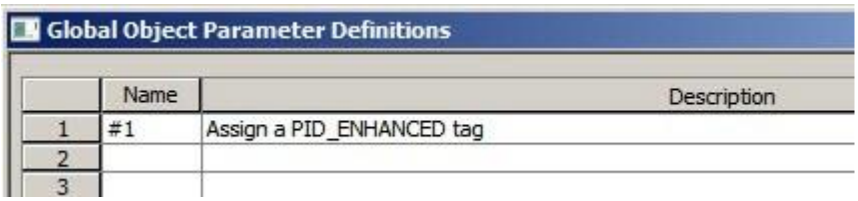
   For information about creating reference objects, see Creating global objects on page 499.

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. For an example, see page 1-39.

To define tag placeholders for the global base object:

1. Right-click the global base object (or group), and then select **Global Object Parameter Definitions**.
2. In the **Global Object Parameter Definitions** dialog box, specify a tag placeholder name (#1, for example) and optional description, as shown in this illustration:

   Global Object Parameter Definitions

	Name	Description
1	#1	Assign a PID_ENHANCED tag
2		
3		

   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.

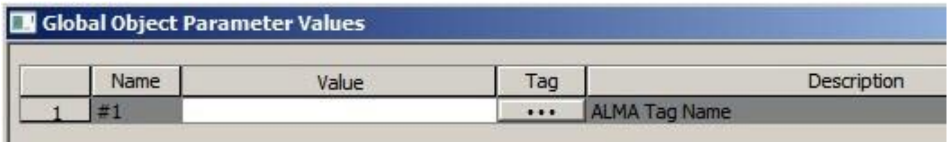
For details about options in the **Global Object Parameter Definitions** dialog box, click **Help**.

To assign a value to a tag placeholder in a reference object:

1. Right-click the global reference object (or group), and then select **Global Object Parameter Values**.

   Tip: If there are no tag placeholders defined for the linked base object, the **Global Object Parameter Values** dialog box is not available.

   The Global Object Parameter Values dialog box shows the Name and Description of all tag placeholders defined for the global base object, as shown in the following illustration.

   Global Object Parameter Values

	Name	Value	Tag	Description
1	#1		•••	ALMA Tag Name

2. In the Value column, specify a value for each tag placeholder, either by typing in the box or by clicking Tag to browse for and select a tag.

For details about options in the **Global Object Parameter Values** dialog box, click **Help**.

Example: Assigning different values to the same placeholder in two global reference objects

To assign different values to two global reference objects that are linked to the same base object, follow these steps:

1. In the global object display, right-click the global base object (or group), and then select **Global Object Parameter Definitions**.

   Tip: If you select any of the individual objects in a group object, the **Global Object Parameter Definitions** dialog box is not available.

2. In the **Global Object Parameter Definitions** dialog box, provide a tag placeholder name. For this example, use #1.

3. Type a description for the tag placeholder, that indicates what type of value to assign to the placeholder.

4. Right-click the global base object, and then select **Global Object Defaults**.

5. In the Global Object Defaults dialog box, ensure that these options are selected:

- For the LinkAnimation default, select **Link with expressions**.
- For the LinkConnections default, select **True**.

Click **OK**.

6. Create a global reference object, by copying the base object and then pasting it into a standard graphic display (in the Displays folder).

7. Duplicate the reference object.

8. Right-click the first reference object, and then select **Global Object Parameter Values**.

   The **Global Object Parameter Values** dialog box shows the tag placeholder name and description you set up for the global base object.

9. In the **Value** column beside tag placeholder #1, type a tag name, or click the **Tag** button (...) to browse for and select a tag. Click **OK**.

10. Repeat steps 8 and 9 for the second reference object. To assign a unique value, select a different tag.

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.

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.

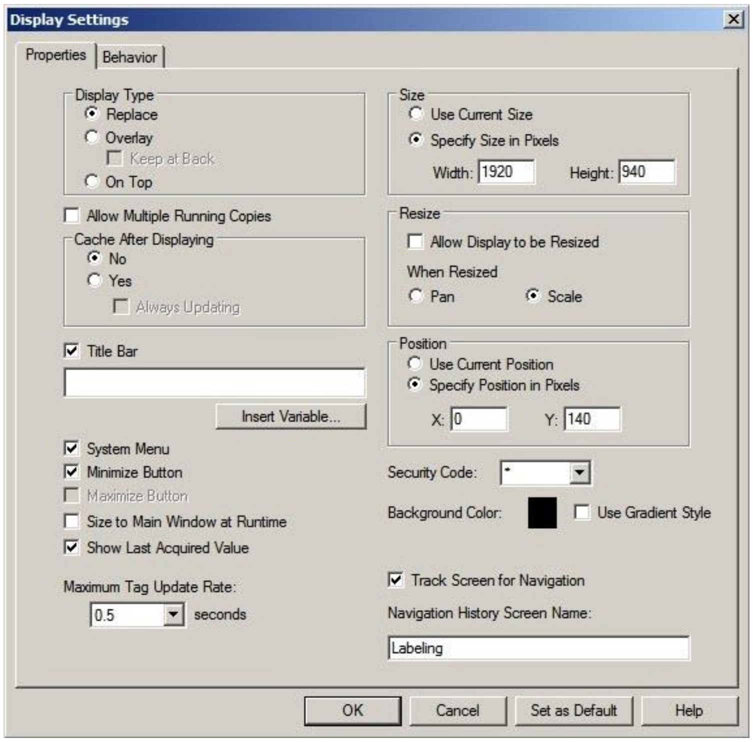
Setting up the properties of a graphic display

In the Properties tab of the Display Settings dialog box, you can specify:

- How the display interacts with other displays at run time.
- Whether multiple copies of the display can run simultaneously on the same client.
- How displays are cached.
- What buttons and text show in the display’s title bar.
- How often the display is updated.
- The display’s size and position, and whether it can be resized at run time.
- Run-time security for the display.
- The display’s background color.
- If gradient style is used.
- If the display is tracked in the navigation history.

To open the Display Settings dialog box:

From the **Edit** menu, select **Display Settings**. You can also right-click an empty area of the display, and then select **Display Settings**.



Display Settings

Properties | Behavior

Display Type
● Replace
○ Overlay
□ Keep at Back
○ On Top

□ Allow Multiple Running Copies

Cache After Displaying
● No
○ Yes
□ Always Updating

☑ Title Bar

Insert Variable...

☑ System Menu
☑ Minimize Button
□ Maximize Button
□ Size to Main Window at Runtime
☑ Show Last Acquired Value

Maximum Tag Update Rate: 0.5 seconds

Size
○ Use Current Size
● Specify Size in Pixels
Width: 1920 Height: 940

Resize
□ Allow Display to be Resized
When Resized
○ Pan ● Scale

Position
○ Use Current Position
● Specify Position in Pixels
X: 0 Y: 140

Security Code: *

Background Color: ■ □ Use Gradient Style

☑ Track Screen for Navigation

Navigation History Screen Name: Labeling

OK | Cancel | Set as Default | Help

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 check box, **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 run time 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 run time. For more information about these commands, see the FactoryTalk View Site Edition Help.

Allowing multiple running copies

Use this option with graphic displays of 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 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 run time, 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 429.

Caching displays

Placing a graphic display in the cache makes the display appear more quickly at run time, 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 441.

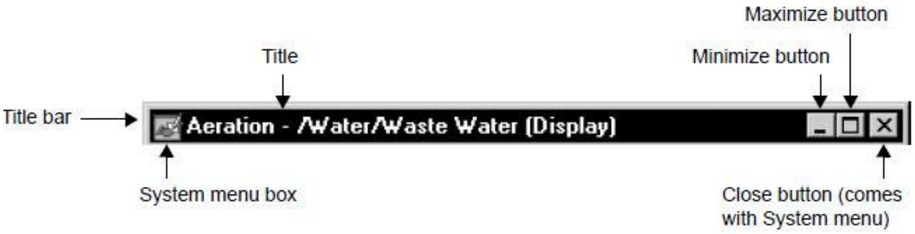
Tip: Cached displays consume memory. Always updating a cached display can add to communications overhead, as data is retrieved for tags whose values might not be needed.

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 run time.

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 title bar text. For more information, see Creating embedded variables on page 575.

The following illustration shows the window style options that are available when you select the **Title Bar** check box. To add or remove an item from the title bar, select or clear the appropriate check box.


Maximize button
Title
Minimize button
Title bar
Aeration - /Water/Waste Water (Display)
System menu box
Close button (comes with System menu)

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 run time.

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 run time, 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 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.

Specifying the size of the graphic display

You can use the current size of the display or specific 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 run time. 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’s 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 run time.

If you do this, you can override the display settings by using size and position parameters with the **Display** command. For details, see the FactoryTalk View Site Edition Help.

Securing the 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 85.

Selecting the 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 check box. 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** check box. When you select this check box, 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.

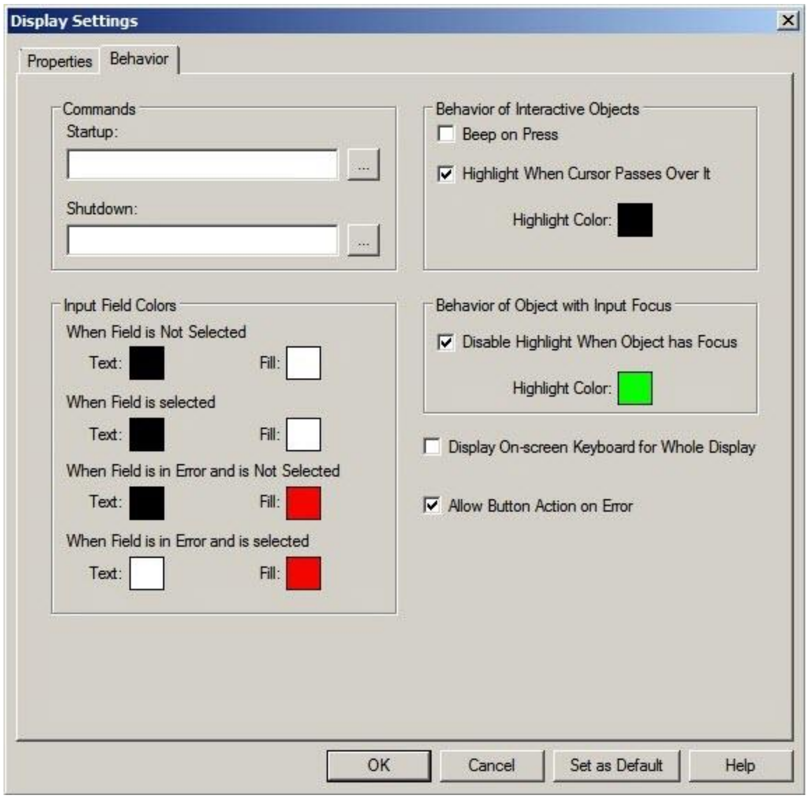
If you do not select this check box, the display will not be added to the navigation history.

Setting up the run-time 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 run time.

- 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.



Commands
Startup:
Shutdown:

Behavior of Interactive Objects
□ Beep on Press
☑ Highlight When Cursor Passes Over It
Highlight Color: ⬛

Input Field Colors
When Field is Not Selected
Text: ⬛ Fill: ⬜
When Field is selected
Text: ⬛ Fill: ⬜
When Field is in Error and is Not Selected
Text: ⬛ Fill: 🟥
When Field is in Error and is selected
Text: ⬜ Fill: 🟥

Behavior of Object with Input Focus
☑ Disable Highlight When Object has Focus
Highlight Color: 🟩

□ Display On-screen Keyboard for Whole Display

☑ Allow Button Action on Error

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 run time, 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 an operator can interact with at run time, 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.

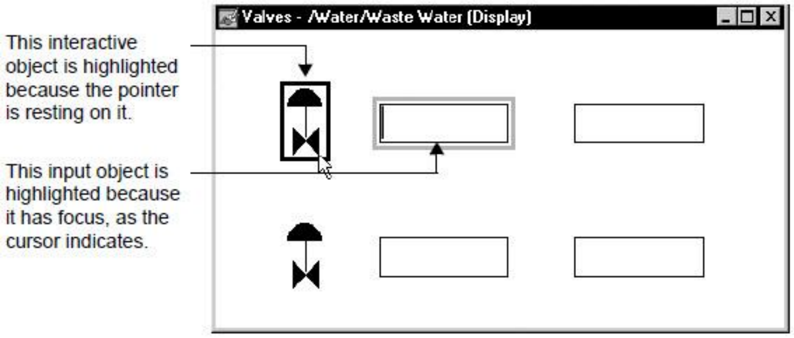
You can specify whether objects with input focus in a graphic display have a highlight box, and what color the highlight will be.

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.


Valves - /Water/Waste Water (Display)
This interactive object is highlighted because the pointer is resting on it.
This input object is highlighted because it has focus, as the cursor indicates.

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 run time, to interact with numeric input, string input, and recipe objects in the graphic display, select the check box, **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 475.

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.

To change the default settings for new graphic displays:

1. Create a new display, and change its display settings to the desired default values.

   Tip: When you make certain changes, the Set as Default button changes to Apply. This lets you apply the changes to the current display without closing the Display Settings dialog box.
   It does not affect the default display settings.

2. Click **OK** to save the changes and to close the Display Settings dialog box.

3. Open the Display Settings dialog box again, and then click **Set as Default** at the bottom of the dialog box.

   The values selected for the previous display become default settings for new displays.

Docking displays to the FactoryTalk View SE Client window

At run time, 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.

Display command parameters for docking displays

To dock graphic displays, run the Display command with any of the following parameters:

To dock a display in this position	Use this parameter
Top edge of the client window	/DT
Bottom edge of the client window	/DB
Left edge of the client window	/DL
Right edge of the client window	/DR

To let an operator dock a graphic display, provide a button object in another display, that uses the Display command with one of the docking parameters as its press action.

In the FactoryTalk View SE Client Wizard, you can also specify a docking parameter for the initial display that runs when the client starts up.

Tip: The Display command parameters /B, /Min., /Max, /X, and /Y are ignored when specified with a docking parameter. For more information, see the FactoryTalk View Site Edition Help.

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 435.

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.

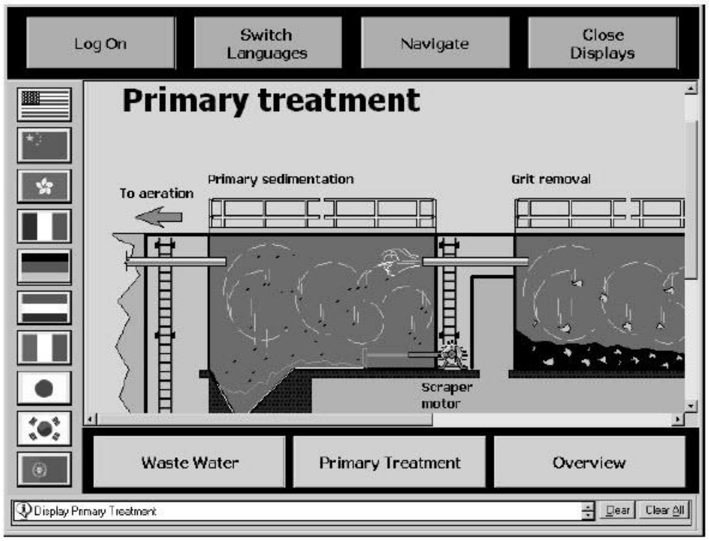
If Line1 is larger than the available client area, scroll bars will be provided for gaining access to the hidden portion of the display. If you opened Line1 first, and then ran the Menu display with a docking parameter, Line1 would be repositioned in the client window, to accommodate the Menu display.

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.

The menu at the top was docked initially, when the FactoryTalk View SE Client started.

The languages (left) and navigation (bottom) menus were docked in that order, by clicking buttons on the initial display.

The Primary Treatment display was opened from the navigation menu.



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 ***.

Printing displays at run time

To let an operator print graphic displays at run time, when you create the displays, provide a way for the operator to run the **PrintDisplay** command.

For example, create a button object, display key, or client key with the **PrintDisplay** command as the press action.

When you use the **PrintDisplay** command, FactoryTalk View prints the entire display, even if parts are covered by other displays. You can use the **ScreenPrint** command to print an image of whatever shows on the monitor.

For more information about the **PrintDisplay** and **ScreenPrint** commands, see the FactoryTalk View Site Edition Help.

Creating graphic objects

This chapter describes how to use, create, and set up 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 393.

Types of graphic objects in FactoryTalk View SE

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 452.
- **Push button** objects are typically used to start or stop processes or actions. For more information, see Creating the different types of push buttons on page 460.
- **Numeric and string** objects are used for entering or displaying numbers or text. For more information, see Creating the different types of data display and input objects on page 471.
- **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 Creating the different types of indicators on page 477.
- **Gauge and graph** objects show a range of values, and relationships between variables. For more information, see Creating the different types of gauges and graphs on page 479.
- **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 482.
- **Advanced** objects, such as trends and HMI tag alarm summaries, provide ways for operators to monitor, analyze, and interact with application data at run time. For more information, see Creating the different types of advanced objects on page 484.

  Some advanced objects are described in detail in other parts of this manual. For information about creating trend objects, see Setting up trends on page 603.

For information about creating HMI tag alarm summaries, see Setting up HMI tag alarms on page 217.

- **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 261.
- **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 information, see Working with OLE objects on page 505.
- **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 506, and Animating ActiveX objects on page 538.
- **Import** objects such as graphics from file.
- **Symbol Factory** a library of graphic objects that can be imported into a Graphic Display.

About global objects

You can create FactoryTalk View global objects to link the appearance and behavior of one graphic object to multiple copies of the object in the same application.

Any graphic object you can create in FactoryTalk View can be a global object, except for ActiveX controls, OLE objects, and HMI tag alarm summaries. For more information, see Creating global objects on page 499.

For information about global object displays, see Creating graphic displays on page 393.

Setting up the properties of graphic objects

After creating a graphic object, you can use the Properties dialog box to set up its appearance and behavior.

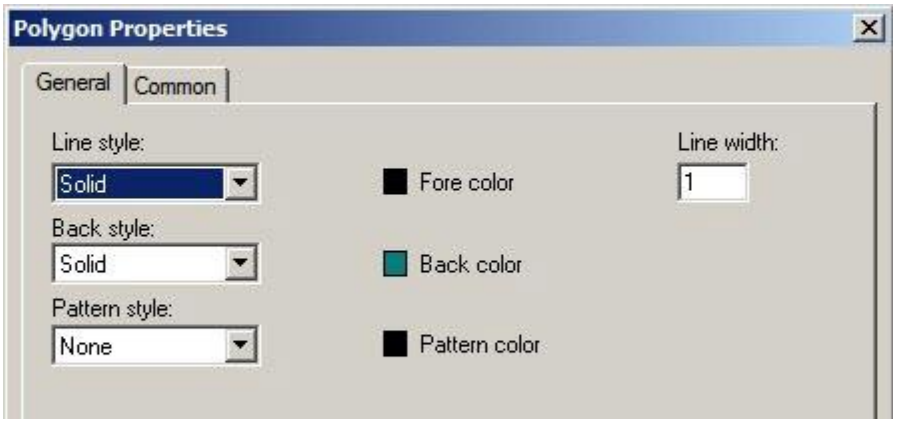
Opening the Properties dialog box:

To open the Properties dialog box for an object, use one of these methods:

- For some objects, the Properties dialog box opens directly, when you create the object.
- For other objects, you open the Properties dialog box by double-clicking the object after you create it.

- You can also right-click the object, and then select **Properties**.

The following illustration shows the Properties dialog box for a polygon object.



Line style: Solid

Back style: Solid

Pattern style: None

Line width: 1

In the Properties dialog box, the features of the object are organized in tabs. The number of tabs and their contents depend on the type of object. For details about options in any Properties dialog box, click **Help**.

Setting up properties common to all objects

There is a Common tab in the Properties dialog box for every FactoryTalk View graphic object. Use the Common tab to set up these properties for the object:

- Height and width
- Top and left position
- Name
- Tooltip
- Visibility

For ActiveX objects, trends, and Alarm and Event objects, you can also set up these properties:

- Focus highlight
- Pointer highlight
- Key navigation
- Tab index

Creating the different types of drawing objects

To draw simple objects such as lines, rectangles, and ellipses, in the Graphics editor, select drawing tools from the **Objects** menu or toolbar.

The drawing tool for an object has the name of the object. To draw a rectangle or square, for example, select the Rectangle tool.

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.

Drawing a rectangle or square

Use □ to draw a rectangle or square.

To draw a rectangle:

- Drag □ diagonally, until the rectangle is the desired size.

To draw a square:

- Press Ctrl while you drag □.

Drawing a rounded rectangle

Use ▢ to draw a rectangle with rounded corners.

To draw a rounded rectangle:

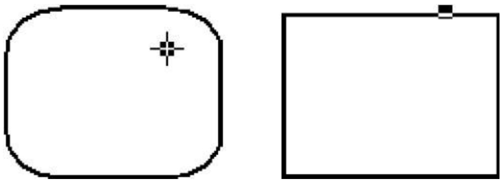
- Drag ▢ diagonally, until the rounded rectangle is the desired size.

To change a rounded rectangle into a right-angle rectangle:

1. Position the pointer on the small box shown inside the rounded rectangle.

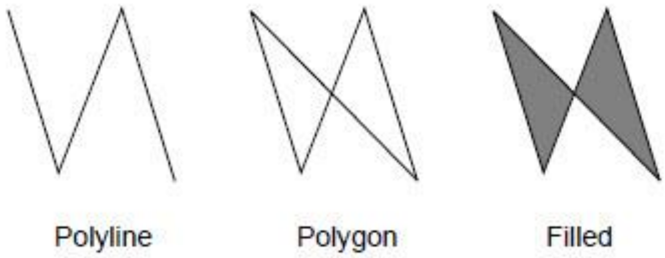
   If the box is not visible, right-click the rounded rectangle, and then click **Edit**.

2. Drag the drawing tool until the rectangle is the desired shape.



Drawing a polyline or polygon

A polyline is a series of connected line segments. A polygon is a closed polyline shape.



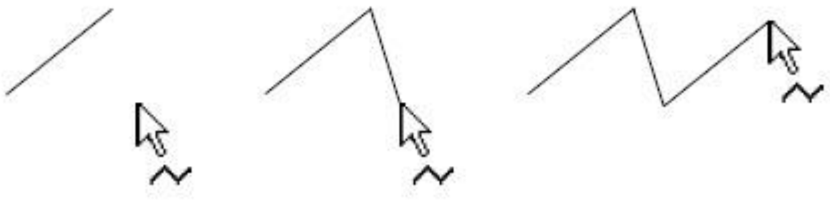
To draw a polyline or polygon:

1. Drag ∿ or ▱ to create the first segment of the object.

   To draw horizontal or vertical lines (not diagonal lines), press **Ctrl** while you drag.

2. Release the mouse button.

3. Move the Polygon or Polyline tool to where the angle of the object is to be, and then click the left mouse button.

   Repeat this step until the object is completed.

4. To finish drawing, click the **Select** tool.

Drawing an ellipse or circle

Use ○ to draw an ellipse or circle.

To draw an ellipse:

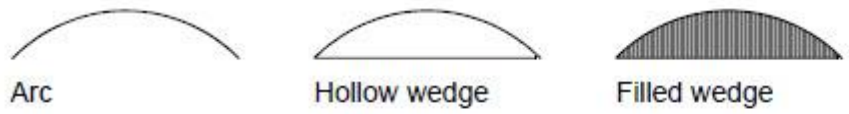
- Drag ○ diagonally, until the ellipse is the desired size.

To draw a circle:

- Press Ctrl while you drag ○.

Drawing an arc or wedge

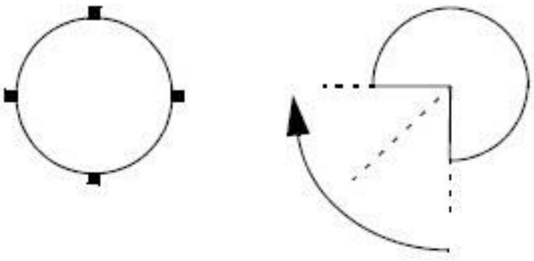
Arcs and wedges are drawn in two steps: first you create an ellipse or circle, and then you reshape the ellipse or circle.



You can also use the Arc and Wedge tools to reshape existing arcs, ellipses, or wedges.

To draw the arc or wedge:

1. Drag ◜ or ⌓ to draw a circle.
2. Release the mouse button.
3. Click a handle, and drag the Arc or Wedge tool to cut out part of the circle.



To draw the arc or wedge in 45-degree increments, press **Ctrl** while you draw.

Changing the properties of drawing objects

You can change the properties of drawing objects:

- In the object's Properties dialog box
- In the Property Panel. For details, see Viewing object properties in the Property Panel on page 398.
- Using the Foreground Color and Background Color palettes, and the Pattern Style toolbar. For details, see Applying colors on page 413.

All drawing objects, except for text, image, and panel objects, have the same properties. Line objects do not take patterns.

For details about setting up a drawing object, click **Help** in its Properties dialog box.

To open the Properties dialog box:

Double-click the object, or right-click the object, and then select **Properties**.

Creating text objects

Use the Text tool to draw a text object. You can then create the text and set up its appearance in the Text Properties dialog box.

Choosing text fonts

You can choose a font before or after you create text, and change the font of any object, including objects that display data or have captions.

When choosing a font, style, size, and color for text, keep the following design principles in mind:

- Choose a font that was designed for on-screen viewing. Examples include Arial, Arial Narrow, Trebuchet MS, and Verdana. You can download these and other fonts free of charge, from the Microsoft Typography web site.
- TrueType and OpenType fonts are recommended because they can be resized easily, without losing text quality.
- Choose only one or two fonts, and use them for all the graphic displays in an application.
- Choose a font size that can be read easily. Test the font size, and adapt it to the screen resolution the operator will see at run time.

- To save screen space, use a condensed font such as Arial Narrow, which fits more characters per line, rather than reducing the font size.
- Sans-serif typefaces are easier to read at small sizes and lower resolutions on the screen than serif typefaces. For example, for normal text use Arial rather than Times New Roman.
- Use high-contrast color combinations, such as yellow on blue, rather than low-contrast combinations like black on green.
- Use colors with recognizable meanings. For example, in some countries the colors red and green mean stop and start.

  Keep color meanings consistent by assigning red only to Stop buttons, and green only to Start buttons.
- Ensure that the fonts used in graphic displays are installed on all computers where the displays will run.

  If a font is not installed, Windows substitutes one that is installed, possibly with unsatisfactory results.

Creating a panel

Use the **Panel** tool to draw rectangles and squares that have borders.

The panel object supports visibility animation. You can also set up panel objects to blink at run time. For details, click **Help** in the Panel Properties dialog box.

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 an image that is not in the Windows bitmap format (.bmp).
- Paste an image copied from another application.

Placing images in graphic displays

Use the Image tool to place a bitmap, jpeg, or .png image in a graphic display.

If the image is not a bitmap, jpeg, or .png file, you must use image editing software to convert it to one of those formats before you can place it 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 run time.

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, .jif, .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.

For details, see Importing graphic files from third-party applications on page 424.

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 run time, 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.

Specifying tag names

At run time, interactive objects obtain data from tags. When creating interactive objects, you specify the name of the tag (or a tag placeholder) that will supply the object with data.

Supplying a tag name:

To supply a tag name for an object that uses data, when creating the object you can:

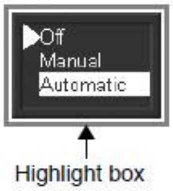
- Type the name of the tag. You do not have to create the tag to use the tag name, but be sure to create the tag later or errors will be reported at run time.
- Click the **Browse** or **Tags** button (whichever is available) to open the Tag Browser and select or create the tag.

For more information about working with tags and the Tag Browser, see Working with tags on page 189.

Determining which objects have input focus

At run time, the object with focus in a graphic display is surrounded by a highlight box, unless the check box, **Disable Highlight When Object has Focus**, is selected in the Display Settings dialog box for the display.

In the Display Settings dialog box, you can also specify the highlight color. For more information, see Specifying the behavior of objects with input focus on page 442.



Focus highlight for ActiveX and trend objects

For ActiveX and trend objects, use the Common tab in the object’s Properties dialog box to specify whether to display a highlight.

If the check box, **Disable Highlight When Object has Focus**, is selected for the graphic display, that setting overrides what is specified for the ActiveX or trend object, in its Properties dialog box.

Using the keyboard to select objects that can take focus

If a mouse or touch screen is not connected to the computer at run time, 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 and HMI tag alarm summaries
- 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 473.

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, input objects, and HMI tag alarm summaries.

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 check box, **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 check box, **Key navigation**.
- In the Property Panel for the object, click the Properties tab, and then set the KeyNavigation property to **False**.

Creating the 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.
- **Momentary** push buttons change a tag to one value when the button is pressed, and another value when the button is released.

  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.

  Momentary push buttons are useful for jogging a motor, and they can be set up to start and stop a machine or process.
- **Maintained** push buttons switch between two values.

  This type of button is useful for changing a setting within a machine or process, but not for starting 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.
- **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.

- **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 interlocked push buttons work as a group, you add them to the display one at a time.
- **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.
- **Navigation** push buttons can be configured to let the operator perform the following actions: show previous display screen, show next display screen, and show a list of previously shown screens.

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.

Creating button push buttons

Use the Button tool to create push buttons that work like standard Windows buttons.

You can assign FactoryTalk View commands to buttons, so they trigger actions when pressed and released.

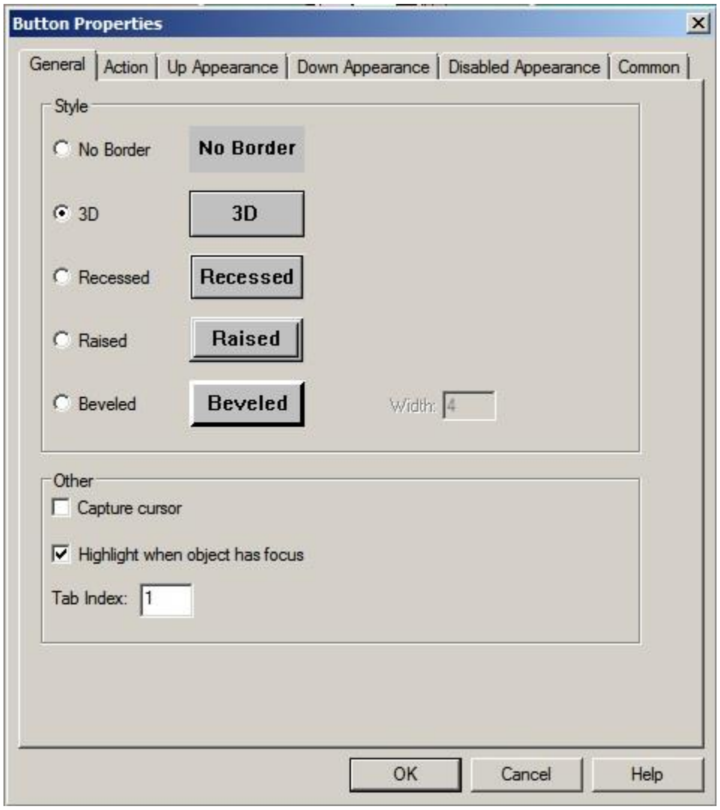
You can also attach different types of animation to a button, except for fill, rotation and touch animation. For more information, see Animating graphic objects on page 515.

Setting up button properties

In the Button Properties dialog box:

- In the **General** tab, specify general characteristics of the button including style, index number, runtime focus highlight, and whether the cursor is captured when the button is pressed.
- In the **Action** tab, set up how the button will behave when the user presses, holds, and releases it at run time, and whether to show a confirmation dialog box before release actions are performed.

- In the **Up Appearance** tab, specify what the button looks like when it is not pressed.
- In the **Down Appearance** tab, specify what the button looks like when it is pressed.
- In the **Disabled Appearance** tab, specify what the button looks like when it is in the disable state.



Style
○ No Border
● 3D
○ Recessed
○ Raised
○ Beveled
Width: 4

Other
☐ Capture cursor
☑ Highlight when object has focus
Tab Index: 1

Creating momentary push buttons

Use the Momentary Push Button tool to create a button that starts a process or action.

A momentary push button changes a tag to one value when pressed, and to another value when released. You can specify values for the press and release actions, or the momentary push button can have these states:

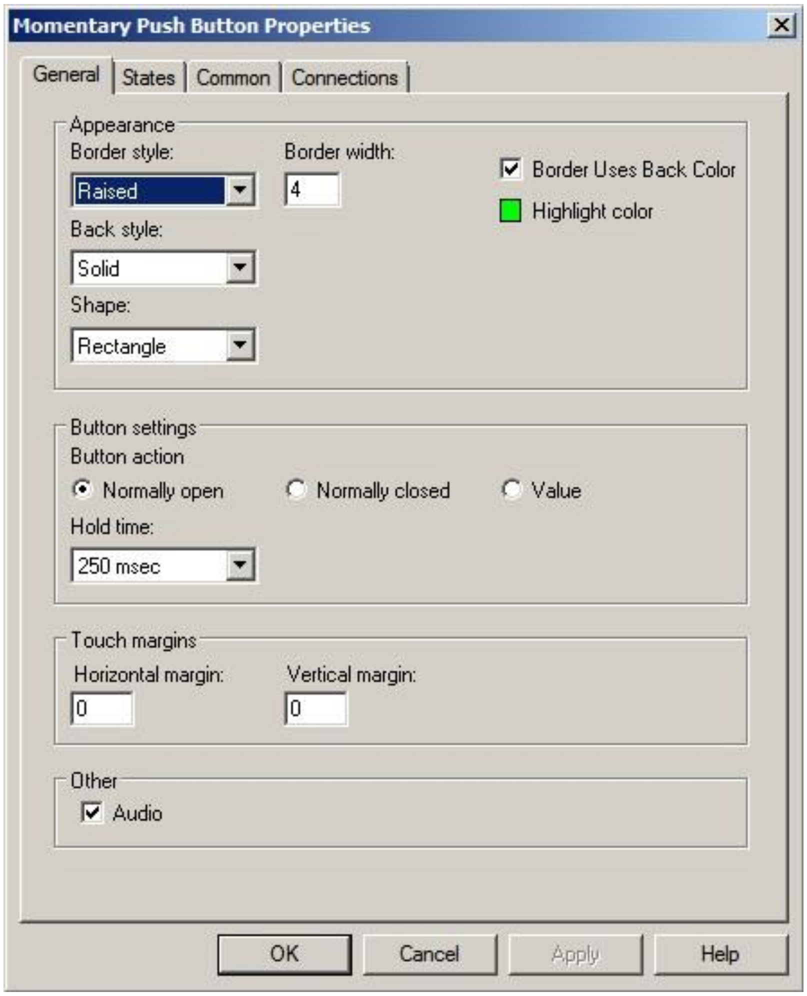
- **Normally open** means the released state of the button is off: when the button is pressed, the tag’s value is set to 1; when the button is released, the tag’s value is set to zero. Pressing the button completes the circuit.
- **Normally closed** means the released state of the button is on: when 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.

Setting up momentary push button properties

In the Momentary Push Button Properties dialog box:

- In the **General** tab, specify the general appearance for all states of the momentary push button at run time, and what type of action the button performs.
- In the **States** tab, specify the appearance and behavior of the button for each state, when it is pressed and released.
- In the **Connections** tab, specify the tags or the expression the button will use for transferring data.



Momentary Push Button Properties

Appearance

Border style: Raised

Border width: 4

☑ Border Uses Back Color

🟩 Highlight color

Back style: Solid

Shape: Rectangle

Button settings

Button action

● Normally open ○ Normally closed ○ Value

Hold time: 250 msec

Touch margins

Horizontal margin: 0

Vertical margin: 0

Other

☑ Audio

Creating maintained push buttons

Use the Maintained Push Button tool to create a button that changes a setting in a machine or process.

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.

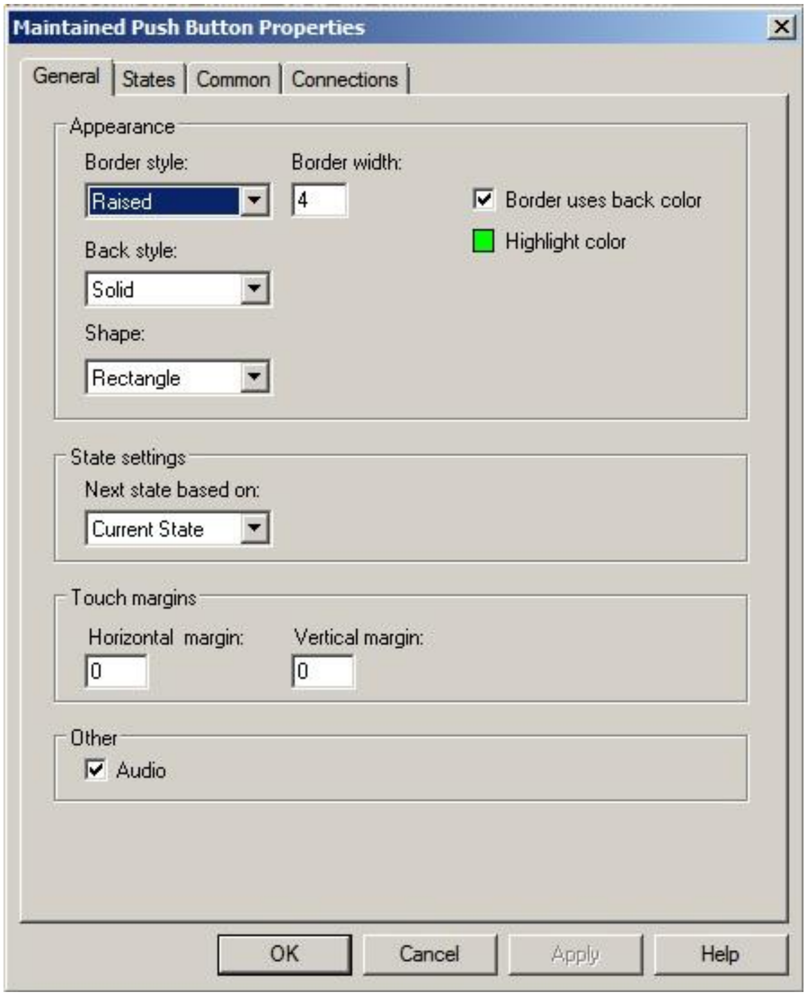
Maintained push buttons are not useful for starting or stopping a machine or process.

Setting up maintained push button properties

In the Maintained Push Button Properties dialog box:

- In the **General** tab, specify the general appearance and touch margins of the button, how it changes states, and whether an audio signal is produced when it is pressed.
- In the **States** tab, specify the appearance and behavior of the button for each state, when it is pressed and released.

- In the **Connections** tab, specify the tags or the expression the button will use for transferring data.



Appearance

Border style: Raised

Border width: 4

☑ Border uses back color

🟩 Highlight color

Back style: Solid

Shape: Rectangle

State settings

Next state based on: Current State

Touch margins

Horizontal margin: 0

Vertical margin: 0

Other

☑ Audio

Creating latched push buttons

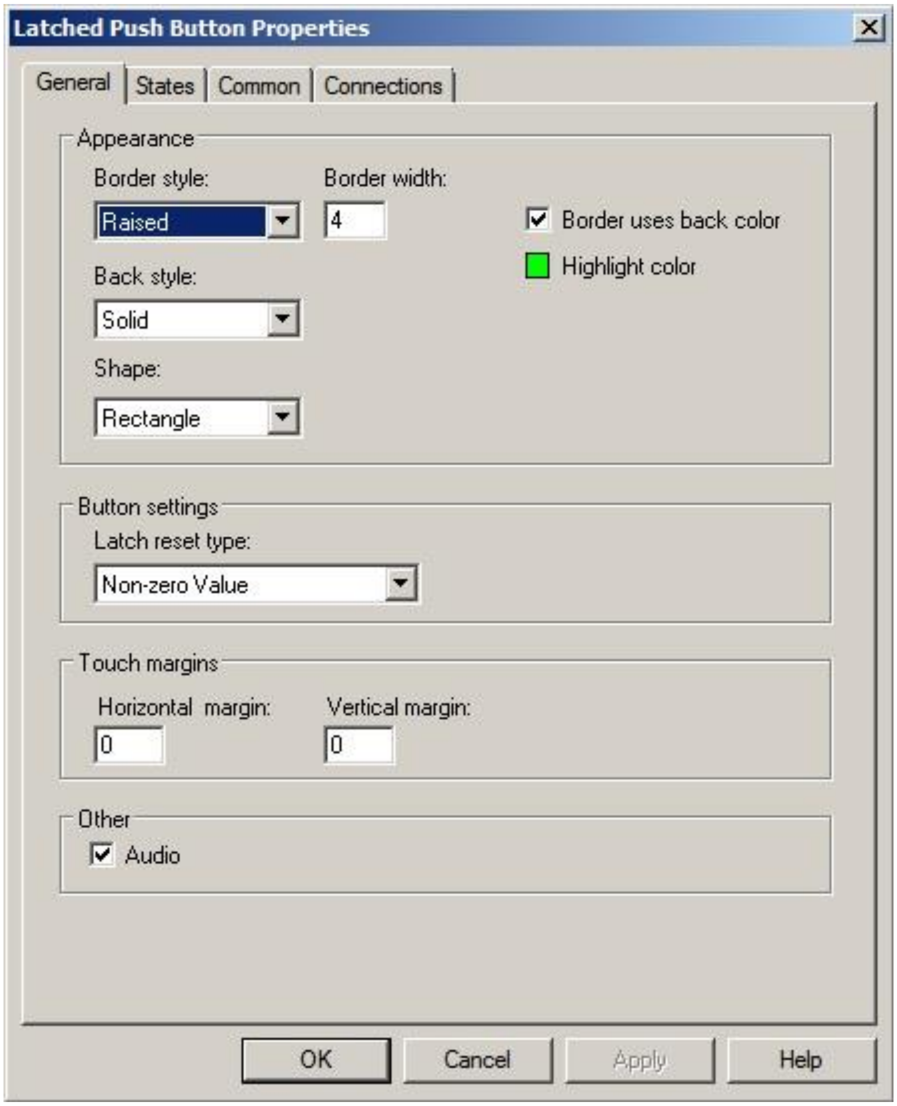
Use the Latched Push Button tool to create a button that latches in the on position, and must be unlatched by another button or process to return to the off position. This type of button is useful for starting a machine or process.

The Handshake tag must be set back to zero before the operator can press the latched push button again.

Setting up latched push button properties

In the Latched Push Button Properties dialog box:

- In the **General** tab, specify the general appearance and touch margins of the button, its latch reset type, and whether an audio signal is produced when it is pressed.
- In the **States** tab, specify what the button does when it is latched and unlatched.
- In the **Connections** tab, specify the tags or the expression the button will use for transferring data.



Appearance

Border style: Raised

Border width: 4

☑ Border uses back color

🟩 Highlight color

Back style: Solid

Shape: Rectangle

Button settings

Latch reset type: Non-zero Value

Touch margins

Horizontal margin: 0

Vertical margin: 0

Other

☑ Audio

Creating multistate push buttons

Use the Multistate Push Button tool to create a button an operator can use to view and cycle through multiple options consecutively.

The multistate push button displays the current state of a process or operation, by showing a different color, caption, or image to reflect different states.

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.

Setting up multistate push button properties

In the Multistate Push Button Properties dialog box:

- In the **General** tab, specify the general appearance and touch margins of the button, how it changes states, and whether an audio signal is produced when it is pressed.
- In the **States** tab, specify what the button does when it is pressed and released.
- In the **Timing** tab, set up whether the button is to repeat automatically when the operator presses and holds it down. You can also set up repeat rate for the button.
- In the **Connections** tab, specify the tags or the expression the button will use for transferring data.

About the run-time 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.

Creating interlocked push buttons

Use the Interlocked Push Button tool to create multiple buttons, one at a time, that work together in a group and share the same tag.

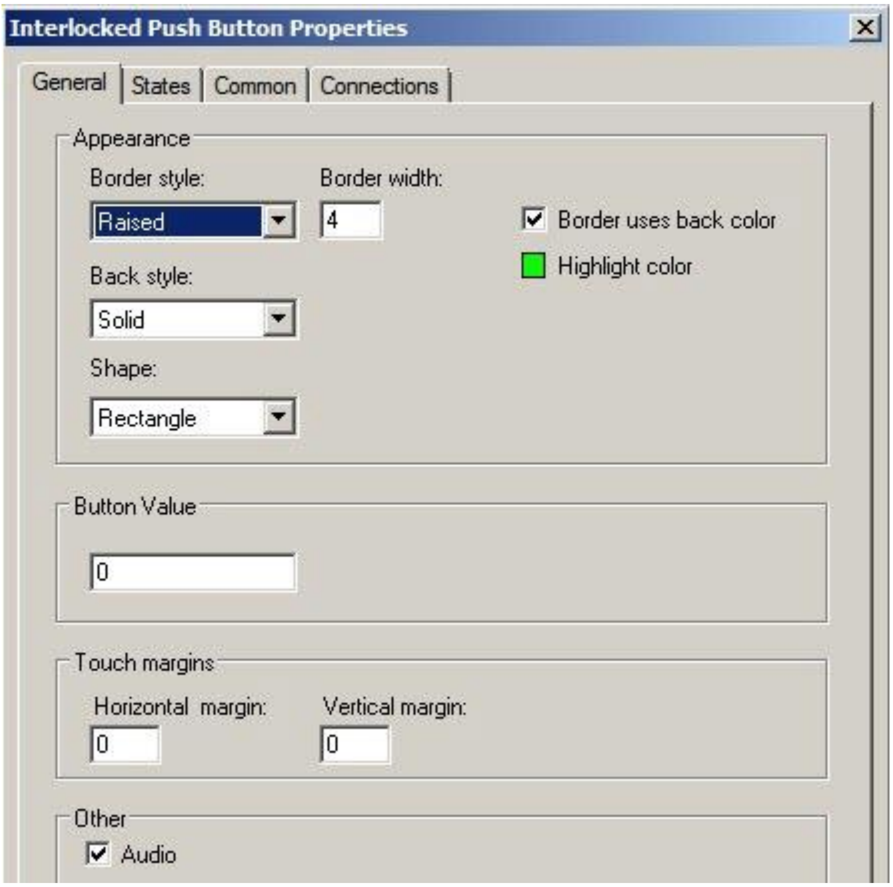
The buttons work together like the preset station selector buttons on a car radio: pressing one button cancels another. When the operator presses one of the interlocked push buttons, the button’s Value tag changes to one value.

When the operator presses a different interlocked button, the buttons’ Value tag changes to another value. All the buttons share the same Value tag.

Setting up interlocked push button properties

In the Interlocked Push Button Properties dialog box:

- In the **General** tab, specify the general appearance and touch margins of the button, its value, and whether an audio signal is produced when it is pressed.
- In the **States** tab, specify what the button does when it is pressed and released.
- In the **Connections** tab, specify the tags or the expression the button will use for transferring data.



Appearance

Border style: Raised

Border width: 4

☑ Border uses back color

🟩 Highlight color

Back style: Solid

Shape: Rectangle

Button Value

0

Touch margins

Horizontal margin: 0

Vertical margin: 0

Other

☑ Audio

Creating ramp push buttons

Use the Ramp Button tool to create a button that an operator can use to increase or decrease the value of a tag.

Ramp push buttons can change a tag by either an integer or floating point value. You can use two ramp buttons together to create an increase/decrease control.

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.

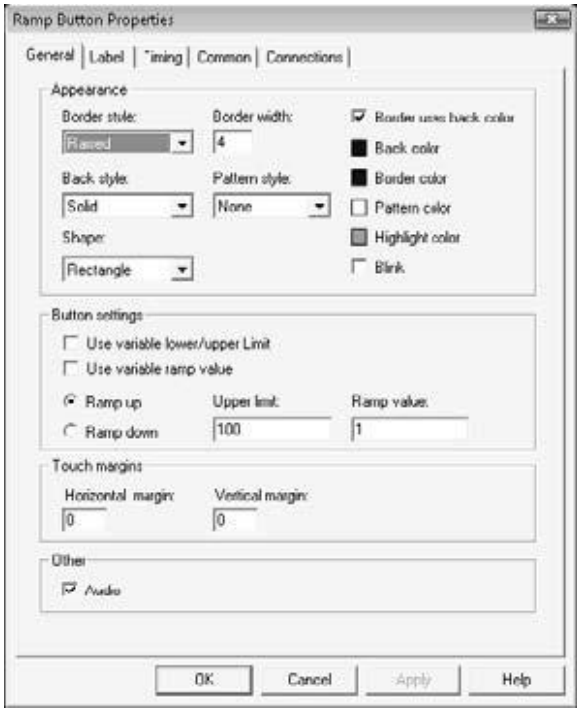
Some data servers do not impose a minimum or maximum limit on the value of a tag. Set the upper and lower limits of the ramp push button to limit the range of values the operator can send to the data server

Setting up ramp push button properties

In the Ramp Push Button Properties dialog box:

- In the **General** tab, specify the general appearance of the button, the iction the button performs, and whether an audio signal is produced when it is pressed.
- In the **Label** tab, create a caption and select an image for the button.
- In the **Timing** tab, set up whether the button action will repeat automatically, when the operator holds the button down. You can also set up the repeat rate.

- In the **Connections** tab, specify the tags or the expression the button will use for transferring data.



Ramp Button Properties

General | Label | Timing | Common | Connections

Appearance

Border style: Raised

Border width: 4

☑ Border uses back color

■ Back color

■ Border color

□ Pattern color

▨ Highlight color

☐ Blink

Back style: Solid

Pattern style: None

Shape: Rectangle

Button settings

☐ Use variable lower/upper Limit

☐ Use variable ramp value

● Ramp up

○ Ramp down

Upper limit: 100

Ramp value: 1

Touch margins

Horizontal margin: 0

Vertical margin: 0

Other

☑ Audio

Creating navigation push buttons

Use the Navigation Button tool to create a button that an operator can use to create buttons that navigate between the opened graphic displays.

A navigation push button can perform three types of actions:

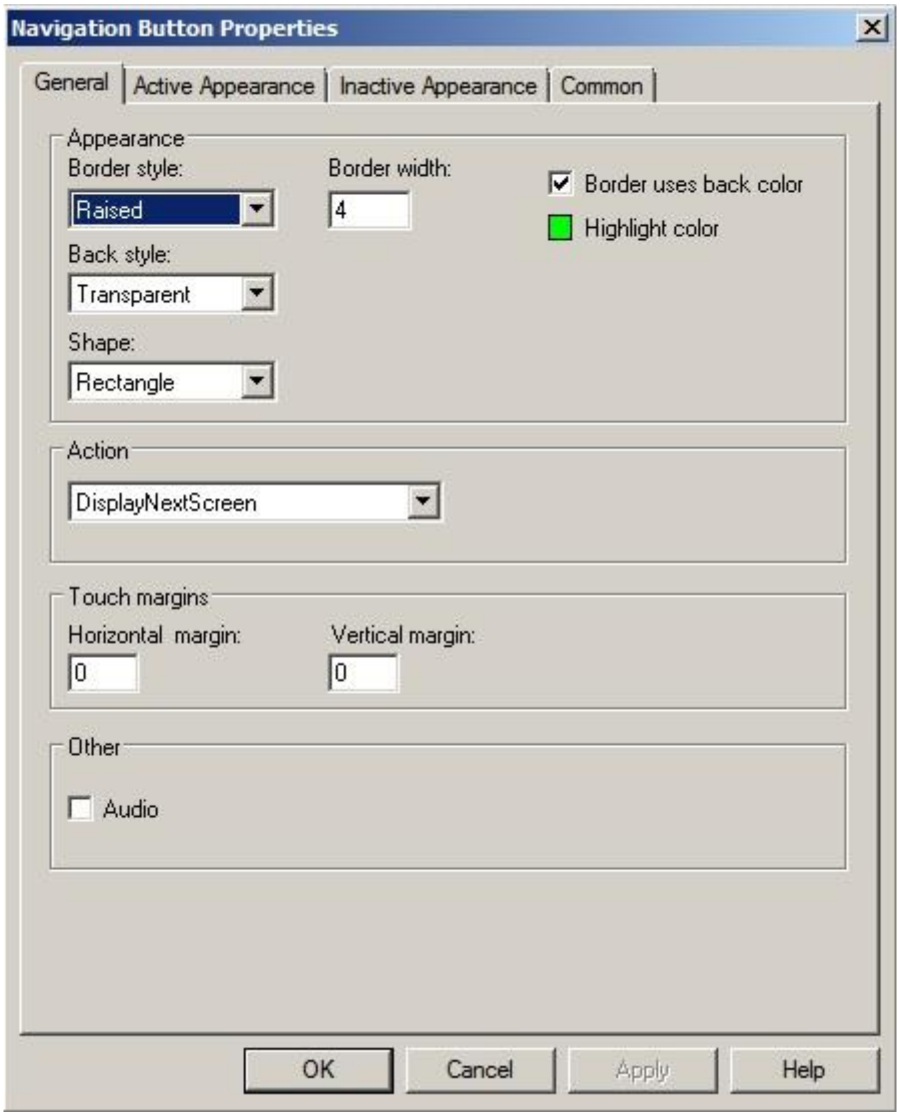
- DisplayPreviousScreen opens the previous graphic display in the navigation history when the button is clicked.
- DisplayNextScreen opens the next graphic display in the navigation history when the button is clicked.
- DisplayNavigationHistory shows a list of previously opened screens when the button is clicked.

Setting up navigation push button properties

In the Navigation Push Button Properties dialog box:

- In the **General** tab, specify the general appearance of the button, the action the button performs, and whether an audio signal is produced when it is pressed.

- In the **Active Appearance** tab, set up the appearance for a navigation button when it is active.
- In the **Inactive Appearance** tab, set up the appearance for a navigation button when it is inactive.



Appearance
Border style: Raised
Border width: 4
☑ Border uses back color
🟩 Highlight color
Back style: Transparent
Shape: Rectangle

Action
DisplayNextScreen

Touch margins
Horizontal margin: 0
Vertical margin: 0

Other
□ Audio

Creating the different types of data display and input objects

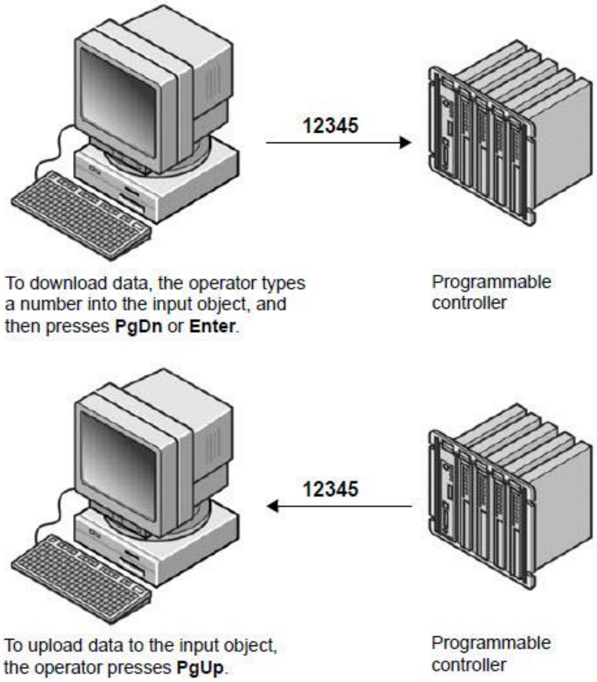
Operators can use data display and input objects to view or change tag data. You can create the following types of data display or input objects:

- **Numeric and string display objects** to let operators view tag or expression values in a graphic display at run time. 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 run time. Use input objects to download or upload tag data. You can set up whether to show a confirmation 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 473.

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 programmable controllers, external OPC servers or DDE devices, or FactoryTalk View memory.


12345
To download data, the operator types a number into the input object, and then presses **PgDn** or **Enter**.
Programmable controller
12345
To upload data to the input object, the operator presses **PgUp**.
Programmable controller

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.

The operator can also retrieve a series of values from a recipe file, change them, write the changed values back to the programmable controller, and then save them to a recipe file. For more information about recipes, see Creating and restoring recipes on page 489.

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.

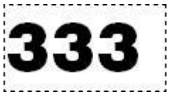
- Each animation of an object (except the Touch animation) is counted as one connection.
- Each connection in the Connections property of an object that is linked to a tag is counted as one connection.
- Each pen configured in a Trend object is counted as one connection.
- Duplicate references of the 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

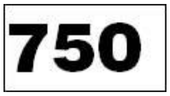
You can set up a numeric or string input object to show a tag's value and update the value continuously. At run time, the appearance of a continuously updating input object changes to reflect which mode it is in.

When the input object is showing a value from the programmable controller or server, it has a dotted border:



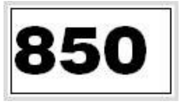
This is called display mode.

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 run time, 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.

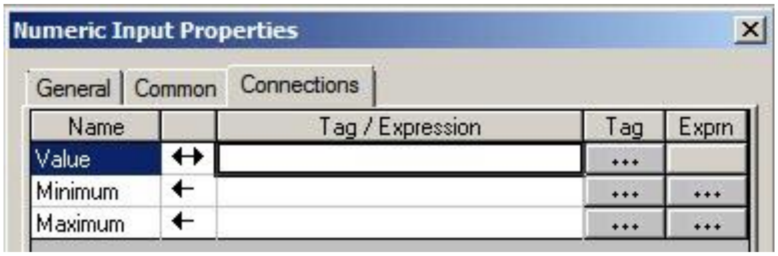
To return to the input object to display mode, press **ESC**.

Validating operator input

To validate operator input, you can specify minimum and maximum values for numeric input objects.

At run time, 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.



Name		Tag / Expression	Tag	Exprn
Value	↔		...	
Minimum	←		...	...
Maximum	←		...	...

In the Displays Settings dialog box, you can select the colors input objects will become when operator input errors occur. For more information, see Setting up the run-time behavior of a graphic display on page 440.

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:

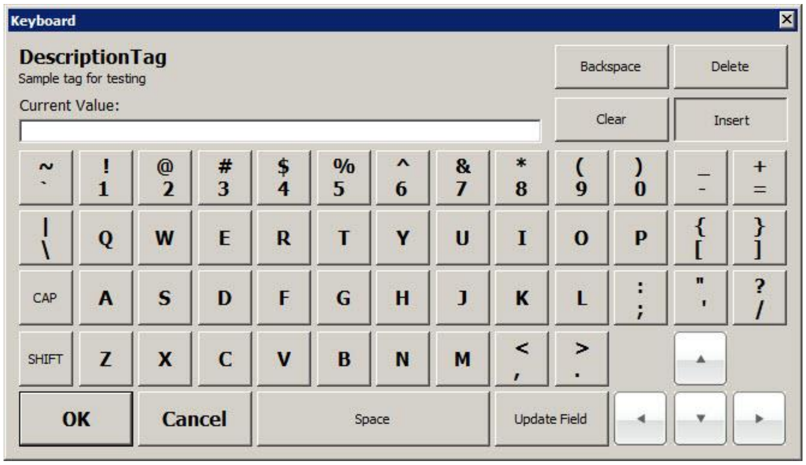
- **Download** downloads the contents of the selected input object to the value table.
- **DownloadAll** downloads the contents of all input objects that are in pending write mode to the value table.
- **Upload** uploads a value from the value table and displays it in the selected input object.
- **UploadAll** 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.


Keyboard

DescriptionTag

Sample tag for testing

Current Value:

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 run-time behavior of a graphic display on page 440.

Creating numeric and string display objects

Use the Numeric Display and String Display tools # $ to create objects an operator can use to view tag or expression data at run time.

In the Numeric Display or String Display Properties dialog box, specify the tag or expression to display, and the appearance of the display object. For details about options in the Properties dialog box, click **Help**.

Creating numeric and string input objects

Use the Numeric Input and String Input tools # $ to create objects an operator can use to enter data for tags that accept numeric values, or text.

In the Numeric Input or String Input Properties dialog box, specify the tag that the operator is to interact with at run time, the appearance of the input object, and whether the object will continuously update the tag’s value.

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

Creating the 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.
- **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.
- **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.

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. In the States tab of the Properties dialog box, you can specify how the indicator will look in each of its different 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.

Set up states for an indicator object in the Connections tab of the Properties dialog box.

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.

Creating multistate indicators

Use the Multistate Indicator tool to create an indicator that shows the current state of a process or operation by showing a different color, caption, or image for each state.

In the Multistate Indicator Properties dialog box, specify state values for the multistate indicator. For details about options in the Properties dialog box, click **Help**.

At run time, the multistate indicator shows the state whose value matches the Indicator tag or expression's value.

Creating symbols

Use the Symbol tool to create an indicator that shows a monochrome image that changes color to match the state of a process or operation. This lets the operator see the status of a process or operation at a glance.

In the Symbol Properties dialog box, specify state values for the symbol indicator, and the image to be shown for each state. For details about options in the Properties dialog box, click Help.

At run time, the symbol indicator shows the state whose value matches the Indicator tag or expression's value.

Creating list indicators

Use the List Indicator tool to create an indicator that shows a list of states for a process or operation, and highlights the current state.

Each state is represented by a caption in the list. This type of indicator is useful to let an 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.

In the List Indicator Properties dialog box, specify state values for the list indicator. For details about options in the Properties dialog box, click **Help**.

At run time, the list indicator highlights the state whose value matches the Indicator tag or expression's value.

Creating the different types of gauges and graphs

Gauges and graphs provide graphical representations of numeric values.

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.

Changing a gauge's fill color at different thresholds

As the needle sweeps higher on a gauge, the area beneath the needle can fill with a color. To help the operator recognize abnormal conditions, you can set up a gauge to change its fill color when the tag value crosses a threshold. For example:

- If the temperature of an oven is lower than required for a recipe, the gauge can show the temperature in blue.
- If the temperature is in the correct range for the recipe, the gauge can show the temperature in green.
- If the temperature is higher than the recipe will accept, the gauge can show the temperature in red.

If you use colored fills on a gauge, make sure enough of the fill is visible when an abnormal condition occurs, so that the operator can recognize the condition.

Tip: Some people are color blind to red and green, so don't rely on color alone to establish meaning.

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.

Changing a bar graph's fill color at different thresholds

To help the operator recognize abnormal conditions, you can set up a graph to change its fill color when the tag value crosses a threshold. For example:

- If the level of the tank of ingredients is lower than the recipe requires, the graph can show the tank's level in red.
- If the level of the tank is in the current range for the recipe, the graph can show the level in yellow.
- If the level is higher than the level the recipe requires, the graph can show the level in green.

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.

Creating gauges

Use the **Gauge** tool to represent a numeric value using a needle on a dial.

In the Gauge Properties dialog box, specify the maximum and minimum values of the gauge, and the tag or expression the gauge is connected to. For details about options in the Properties dialog box, click **Help**.

At run time, the gauge indicates the value of the tag or expression in relation to the gauge's minimum and maximum values.

Creating bar graphs

Use the Bar Graph tool to create a graph that represents a numeric value by filling and emptying as the value rises and falls.

In the Bar Graph Properties dialog box, specify the maximum and minimum values of the bar graph, and the tag or expression the graph is connected to. For details about options in the Properties dialog box, click **Help**.

At run time, the graph shows the value of the tag or expression in relation to the graph's minimum and maximum values.

Creating scales

Use the Scale tool to create a scale for a bar graph.

In the Scale Properties dialog box, specify the appearance of the scale. For details about options in the Properties dialog box, click **Help**.

To place values on the scale as a legend, use text objects. Because the scale doesn't change at run time, you don't need to connect it to a tag.

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 533.
- 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 run time, in the Behavior tab of the Display Settings dialog box, select the check box, **Display on-screen keyboard**. For details, see Setting up the run-time behavior of a graphic display on page 440.
- **Keys** are graphic objects 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 trends.

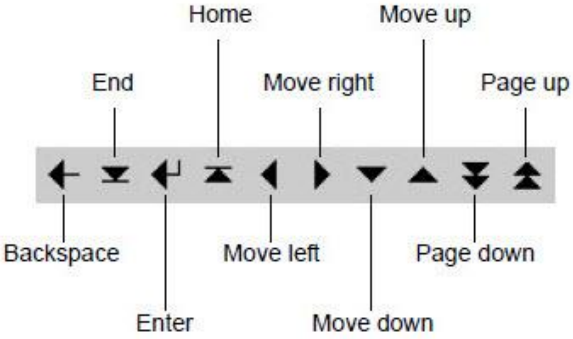
Creating the different types of key objects

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.

Creating the 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:

- **HMI tag alarm summary** objects provide operator with a way to monitor and respond to HMI tag alarm data.
- **Arrow** objects track tag or expression values in a vertical or horizontal path.
- **Control list selectors** let an operator select from a list of states, and write values associated with those states.
- **Piloted control list selectors** 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.
- **Tag labels** show information about a tag's properties, such as the description, engineering units, and minimum and maximum values.
- **Display list selectors** let an operator navigate between graphic displays in a list.
- **Local message displays** provide an operator with information, or prompts about what to do next.
- **Recipe** objects let an operator restore data from a recipe file to input objects in a graphic display.
- **Time and date** shows the current time and date.
- **Trends** provide visual representations, or charts, of real-time or historical data.

For information about creating:

- An HMI tag alarm summary object, see Setting up HMI tag alarms on page 217.
- A trend object, see Setting up trends on page 603.

For details about setting up any of the advanced objects, click **Help** in the object's Properties dialog box.

Creating arrows

Use the **Arrow** tool to create arrows that move based on a tag value or the result of an expression. Arrows can move vertically or horizontally.

Vertical arrows move up or down, and horizontal arrows move left or right, in relation to a tag's Low or High EU (Engineering Units).

For HMI tags, arrows can move in relation to the tag’s minimum and maximum values, or values specified in the General tab of the Arrow Properties dialog box. For data server tags, values must be specified in the General tab.

For vertical arrows, if the value is less than or equal to the minimum value, the arrow is at the bottom of its range. If the value is equal to or greater than the maximum value, the arrow is at the top.

For horizontal arrows, if the value is less than or equal to the minimum value, the arrow is at the left of its range. If the value is equal to or greater than the maximum value, the arrow is at the right.

Creating tag labels

Use the Tag Label tool to show different types of tag information at run time.

In the Tag Label Properties dialog box, specify the tag associated with the label, and the property to show. For details about options in the Properties dialog box, click **Help**.

Some HMI tag properties have different names when shown using a tag label. The following table describes which properties correspond with the types of tag labels.

This tag label	Corresponds with the HMI tag property	For this type of HMI tag	And displays
Low EU	Minimum	Analog	A tag’s minimum value
High EU	Maximum	Analog	A tag’s maximum value
Contact Value	Status	Digital	The current status of a tag. When the tag value is 1, ‘On Label’ is shown. When the tag value is 0, ‘Off Label’ is shown.
Engineering Units (EU)	Units	Analog	A tag’s Units label
Tag Name	Name	All types	A tag’s name
Tag Description	Description	All types	A tag’s description
Contact Open Label	Off Label	Digital	A tag’s Off label
Contact Close Label	On Label	Digital	A tag’s On label

Creating time and date displays

A time and date display shows the current time and date. Use the Time and Date Display tool to create a box that shows the time and date in a graphic display.

In the Time Date Display Properties dialog box, specify the appearance of the object, and the format in which the time and date appear. For details about options in the Properties dialog box, click **Help**.

Creating display list selectors

Use the Display List Selector tool to create list of graphic displays in the application, that operators can select from at run time.

To open a display, the operator can select the corresponding state, and then press **Enter**.

In the Display List Selector Properties dialog box, assign displays to states. For details about options in the Properties dialog box, click **Help**.

Tip: For network distributed applications, display list selectors work with graphic displays located in the home area only. For more information about the home area, see Working with network distributed applications on page 125.

Setting up states for a display list selector

To set up the states for a display list selector, decide how many graphic displays are to be in the list, and then, in the States tab, add that number of states to the display list selector.

For each state, specify a display and its associated parameter file or parameter list. Also specify a caption that identifies the display. This is what the operator will see in the list at run time.

Providing operator instructions in local message displays

Use local message displays to provide an operator with information about a process, or about what to do next, at run time.

For example, you might provide messages that describe the status of a device whose condition cannot be represented graphically with accuracy, or to tell the operator how to deal with a specific situation when it arises.

Setting up local messages

These are the tasks involved in setting up local messages:

1. In the Local Messages editor, create a local message file that contains the messages to display, and the trigger values for these message.
2. In a graphic display, create a local message display that presents a message when its Value tag or expression matches a trigger value in the specified local message file.

You can use multiple local message display objects in a display, and link each object to a different local message file. You can also use the same local message file for multiple local message displays.

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

Trigger values cannot be zero

The trigger value for a local message can be any non-zero integer value (positive or negative). Trigger values do not need to be contiguous, but they must be unique for each message. For example, you could use trigger values of 1, 2, and 3, or of 10, 20, and 30.

Because trigger values cannot be 0, if you use a digital HMI tag, you can only use the value 1 to trigger a message.

If you use an analog tag or an expression, you can use any non-zero integer or floating point value to trigger a message. Floating point values are rounded to the nearest integer.

What is shown at run time

In the local message file, each message is associated with a trigger value. When the local message display’s Value tag or expression matches the trigger value, the message associated with the trigger is shown.

At run time, a local message display shows one message at a time.

Special cases are handled in the following ways:

- If the Value tag or expression is unassigned, the local message display is filled with question marks (?).
- The Value tag or expression is rounded to the nearest integer. If the value does not match any of the trigger values in the message file, the local message display is filled with question marks (?).
- If the message is too long to fit in the list, the last shown character is replaced with an asterisk (*).

- When the Value tag or expression's value is 0, the local message display is cleared.

Creating local message displays

Use the Local Message Display tool to create an object that shows predefined messages at run time.

In the Local Message Display Properties dialog box, specify the message display's Value tag, and the related message file. For details about options in the Properties dialog box, click Help.

Example: Setting up local messages

This example shows how to use local messages to notify the operator about the status of a hoist.

1. Create an analog tag called Hoist_Status.

   This tag points to an address in a programmable controller that is linked to a sensor on the hoist.

When the hoist is	The tag's value is
At bottom	1
Raising	2
Stopped between the top and bottom	3
Lowering	4
At top	5

2. In the Local Messages editor, create the following messages with five trigger values, to match the values of the Hoist_Status tag:

Trigger value	Message
1	The hoist is ready to rise.
2	The hoist is raising the pallet.
3	The hoist has stopped.
4	The hoist is lowering the pallet.
5	The hoist is finished rising.

3. Save the message file with the name Hoist status.

4. In the Graphics editor, create a local message display object.

5. In the object's Properties dialog box, click the General tab, and then select the Hoist status message file.

6. Click the Connections tab, and then type Hoist_Status in the Tag/Expression column.

At run time, when the operator views the graphic display containing the local message display object, the status of the hoist is shown.

Creating and restoring recipes

A recipe file supplies tag values for all the numeric and string input objects in a graphic display. To determine which input objects receive which tag values, the recipe file uses the index numbers assigned to the objects.

FactoryTalk View assigns index numbers to input objects and buttons as you create them. To check the index number for an object, open its Properties dialog box, and then check the number in the Tab index box.

For more information about index numbers, see Using index numbers to navigate to objects in a display on page 530.

You can create a recipe file in the Recipes editor or, at run time, by specifying a file name in the recipe object, and then saving values to that file.

Creating a recipe object

Use the Recipe tool R to create a recipe object in a graphic display, so that an operator can restore or save recipe files at run time.

Rather than entering values one by one into input objects in the display, the operator can use the recipe object to load recipes that provide values for all the objects at once. The operator can also use the recipe object to write the values to network devices.

Tip: Each graphic display can contain only one recipe object.

Restoring and saving recipe values at run time

At run time, an operator can restore values from a recipe file into input objects and send those values to a network device or server.

The operator can also upload values from a network device or server into input objects, and then save those values to a recipe file. If an upload fails because of a communication error, the input object appears in outline form.

To use a recipe object in a graphic display, do one of the following:

- Double-click the recipe object to open the Recipe dialog box, and then save or restore the recipe file.
- Press **Ctrl+R** to move to the object, and then press **Enter** to restore the contents of the recipe file.
- Press **Ctrl+W** to move to the object, and then press **Enter** to save the recipe file.

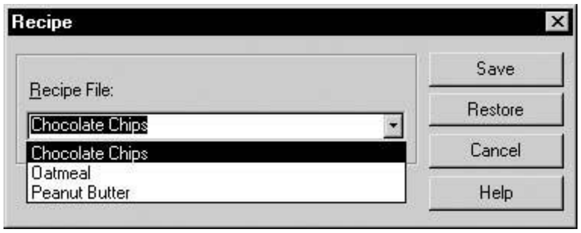
If the display is set up to use the on-screen keyboard, selecting the recipe object and pressing Enter opens the keyboard. To open the Recipe dialog box, press **Download** in the on-screen keyboard.

To restore the values from a recipe file:

1. In the recipe object, type the name of the recipe file you want to restore, and then press **Enter**.

   Tip: If you don’t know the name of the recipe file, just click in the recipe object, and then press **Enter**.

2. In the Recipe dialog box, click **Restore**.



   If you didn’t specify a recipe file name, select a recipe file first, and then click **Restore**.

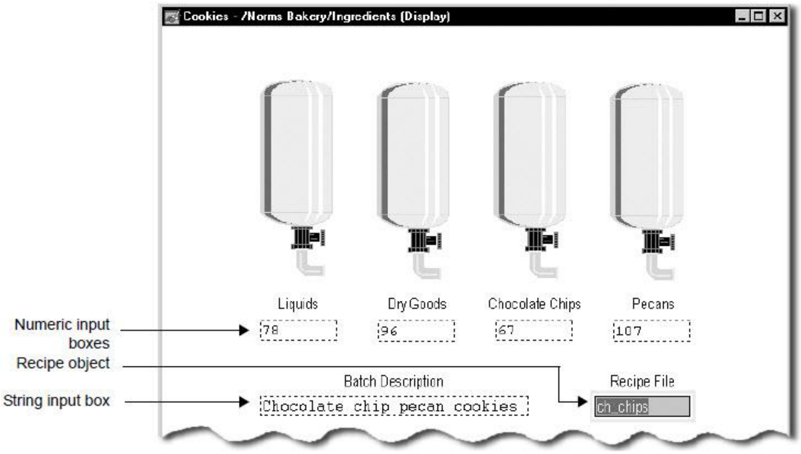
You can also use the **RecipeRestore** command to restore values from a recipe file. For details, see the FactoryTalk View Site Edition Help.

To download recipe values to a network device or server:

After restoring values from a recipe file to input objects in a display, to download the values, press **PgDn**, or use the **DownLoad** or **DownLoadAll** command.

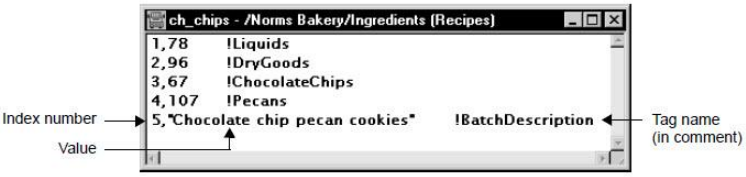
Example: Creating a recipe file at run time

Values typed in input boxes in the following graphic display, are used to create a recipe file for chocolate-chip cookies.



Liquids: 78
Dry Goods: 96
Chocolate Chips: 67
Pecans: 107
Batch Description: Chocolate chip pecan cookies
Recipe File: ch_chips

This is the new recipe file:



```
1,78       !Liquids
2,96       !DryGoods
3,67       !ChocolateChips
4,107      !Pecans
5,"Chocolate chip pecan cookies"      !BatchDescription
```

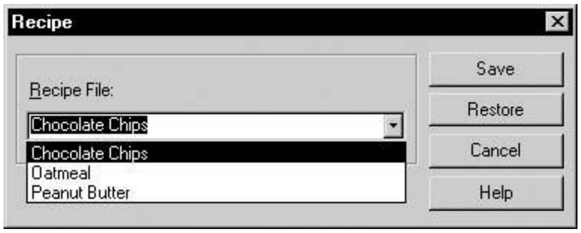
To save recipe values from a network device or server:

1. To upload values you want to save to a recipe file, press **PgUp**, or run the **Upload** or **UploadAll** command.

2. In the recipe object, type the name of the recipe file, and then press **Enter**.

   Tip: If you don't know the name of the recipe file, just click in the recipe object, and then press **Enter**.

3. In the Recipe dialog box, click **Save**.



   If you didn't specify a recipe file name, select a recipe file first, and then click **Save**.

You can also use the **RecipeSave** command to save values to a recipe file. For details, see the FactoryTalk View Site Edition Help.

Creating control list selectors

Control list selectors let an operator scroll through a list of states for a process and select one of the states. A highlight in the list shows the current state.

Selecting states in a control list selector

A control list selector can show several states at the same time, but only one state can be selected at a time. As the operator scrolls through the list, each state is selected automatically.

To let the operator confirm the selection of a particular state before the state's value is written to the network device, include an Enter key with the control list selector.

Using keys to scroll the list

When a piloted control list selector is operator controlled, it works with:

- Key objects. These are graphic objects that duplicate the functions of keyboard keys. Use them with touch-screen terminals.
- The arrow keys and Enter key on a terminal's keypad.
- The arrow keys and Enter key on a keyboard.

The operator presses the keys to scroll up or down the list, or to make selections from the list. The keys can be set up to work with the piloted control list selector that has focus, or with a specific piloted control list selector.

For more information about keys, see Using key objects to simulate keyboard functions on page 482.

Setting the Value tag

At run time, the Value tag changes:

- When the operator selects the next item in the list.
- When the operator presses the Enter key, if the control list selector requires that a selection be confirmed using the Enter key.
- When another process changes the tag's value in the network device.

For example, a control list selector is used to change a tag’s value. If the tag’s value is zero, when the operator selects the next state, the tag’s value changes to the value for the next state, for example 1.

If another process changes the tag’s value to 2, the next time the operator selects the next state, the tag’s value changes to 3.

Using Enter key handshaking to ensure the current value is read

When the operator presses the control list selector’s Enter key, the highlighted state’s value is written to the network device.

Use Enter key handshaking to hold the current value of the tag at the network device for a specified period, to ensure it is read before the control list selector overwrites it with a new value.

Before Enter key handshaking completes for an object, the operator can provide input to other objects in the graphic display. However, if the operator presses the Enter key for an object before its handshake is completed, an error is sent to the Diagnostics log file.

If the graphic display is closed, the Enter tag is reset to 0, and any handshake timing is also reset.

Creating control list selectors

Use the Control List Selector tool to create a list that shows multiple options, which an operator can cycle through consecutively. The control list selector shows the current state of a process or operation by highlighting the state.

Each time the operator presses a key, the control list selector’s highlight changes position, and the Value tag changes to the value for the next state.

When the control list selector is in its last state and the operator presses the key, the control list selector returns to its first state.

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 states off 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.

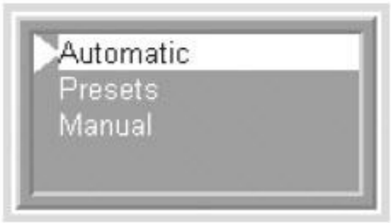
Selecting states in a piloted control list selector

A piloted control list selector can show several states at the same time, but only one state can be selected at a time.

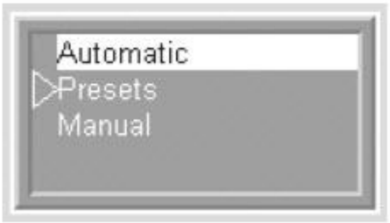
You can set up a piloted control list selector to be operator controlled or remote controlled, by assigning tags or expressions in the Connections tab.

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.

Using keys to scroll the list

When a piloted control list selector is operator controlled, it works with:

- Key objects. These are graphic objects that duplicate the functions of keyboard keys. Use them with touch-screen terminals.
- The arrow keys and Enter key on a terminal's keypad.
- The arrow keys and Enter key on a keyboard.

The operator presses the keys to scroll up or down the list, or to make selections from the list. The keys can be set up to work with the piloted control list selector that has focus, or with a specific piloted control list selector.

For more information about keys, see Using key objects to simulate keyboard functions on page 482.

Controlling the list view

If the piloted control list selector contains more states than the list can display simultaneously, the value of the Top Position tag (if assigned) changes whenever the item at the top of the list changes.

If the Visible States tag is assigned, the values assigned to all visible states are written to the Visible States tag whenever the list scrolls.

If all the visible states are written, the tag must support arrays, and the array must be the same length as the number of visible states in the piloted control list selector.

Selecting items directly or remotely

Each state can be selected directly by an operator, or remotely by a device such as a programmable controller. The operator or the controller scrolls through the list to select a different state or a different group of visible states.

When an operator or remote device selects a state, the value assigned to the selected state is written to the piloted control list selector's Value tag. If the state is turned off, the state's value is not written to the Value tag.

If the operator attempts to select a state that is remote controlled, the state's value is not written to the Value tag.

Setting the Value tag

At run time, the Value tag changes:

- When the operator selects an item in the list, if the check box, **Write on enter**, is cleared in the General tab.
- When the operator selects an item in the list, and then presses the Enter key, if the check box, **Write on enter**, is selected in the General tab.

- When a remote device selects an item in the list.

For more information about the Write on enter option, see the FactoryTalk View Site Edition Help.

Using Enter key handshaking to ensure the current value is read

When the operator presses the piloted control list selector's Enter key, the highlighted state's value is written to the network device.

Use Enter key handshaking to hold the current value of the tag at the network device for a specified period, to ensure it is read before the piloted control list selector overwrites it with a new value.

Tip: Enter key handshaking can be used only if the piloted control list selector is operator controlled and if the check box, Write on enter, is selected.

How the handshaking tags and settings interact

If the Enter tag is not assigned, no handshaking will take place.

If the Enter tag is assigned, when the operator presses the Enter key associated with the piloted control list selector, the following will occur:

1. The timer for the Enter key control delay option begins timing. When the time has expired, the Enter tag is set to 1.

   If the Handshake reset type is set to Non-zero value, the Enter Handshake tag must be 0 when the Enter key control delay expires in order to set the Enter tag to 1.

2. The timers start for the Enter key handshake time, and Enter key hold time.

3. If the Enter Handshake tag is assigned, the Enter tag remains set until the Enter key handshake time expires or until it is reset by the Enter Handshake control, whichever happens first.

   If the Enter Handshake tag is not assigned, the Enter tag remains set until the Enter key hold time expires.

4. The Enter tag is reset to 0.

How the Handshake reset type option works

The Enter Handshake tag resets the Enter tag as follows:

- **Non-zero value** resets the Enter tag when the Enter Handshake tag is a non-zero value.
- **Zero to non-zero transition** resets the Enter tag when the Enter Handshake tag changes from zero to a non-zero value.

Before the Enter tag is reset to 0, the operator can provide input to other objects in the graphic display.

If the operator presses the Enter key for an object whose Enter tag has not yet been reset (using a key, or external keyboard or keypad), an error is logged to the Diagnostics log.

If the graphic display is closed, the Enter tag is reset to 0, and any handshake timing is also reset.

Creating piloted control selectors

Use the Piloted Control List Selector tool to create a list that displays multiple options, and lets an operator cycle through the options consecutively. The piloted control list selector displays the current state of a process or operation by highlighting the state.

A piloted control list selector can be operator controlled, remote controlled, operator and remote controlled, or neither.

Specifying the text and value for each state

Piloted control list selectors change their appearance and the Value tag, based on their states.

In the States tab of the Properties dialog box, for each state, you specify the text that appears in the list and the value for the Value tag.

You can also set up whether each state can be operator controlled, remote controlled, operator and remote controlled, or neither. If **None** is selected, the state is turned off and cannot be selected.

You can set up 255 states for a piloted control list selector. This provides the operator with up to 255 selections in the list.

Setting up connections for a piloted control list selector

To connect with a network device such as a programmable controller, piloted control list selectors use tags or expressions. You set up these tags or expressions in the Connections tab of the Properties dialog box.

Piloted control list selectors have seven tags for sending and receiving data:

- The **Value** tag receives the value of the currently selected state. You can use this value to trigger an action, for example, to set a motor's speed to low, medium, or high.
- The **Indicator** tag selects an item in the list if the piloted control list selector is being controlled remotely, for example, by a network device.
- The **Remote Access** tag or expression determines whether the piloted control list selector can be operated by the operator or remotely, for example, by a network device.

  If the remote access tag has a value of zero, or is not assigned, the operator can select states in the list. If the remote access tag has a non-zero value, the selected state is determined by the value of the Indicator tag or expression.
- The **Top Position** tag receives the value of the state that is the first item in the list whenever the list scrolls.
- The **Visible States** tag receives the values of all the states visible in the list whenever the list scrolls. The tag must support arrays, and the arrays must have as many elements as the number of items visible in the piloted control list selector.

  For example, if the piloted control list selector has 5 visible states, and the state values are written to network device addresses N7:0 through N7:4 using RSLinx Enterprise, the Visible States tag should be an array tag with the following address:

  ```
  ::[RSLinx Device Shortcut ]N7:0,L5
  ```

  where L5 indicates that the tag writes values to five addresses in the array.
- The **Enter** tag lets a network device confirm a selection after the Value tag is written. The Enter tag is set for as long as is specified in the Timing tab.
- The **Enter handshake** tag or expression resets the Enter tag, using the Handshake reset type option specified in the Timing tab.

Creating the Alarm and Event objects

The FactoryTalk Alarm and Event objects provide run-time 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 (RSLinx Enterprise) 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.

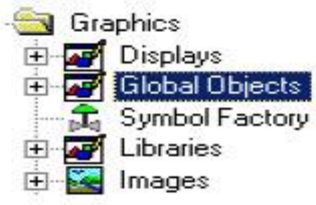
- The **Alarm and Event log viewer** shows information stored in the logs generated by the FactoryTalk Alarms and Events Historian.

  You can use the alarm and event log viewer to view, sort, filter, and export historical alarm information.

For information about creating the FactoryTalk Alarm and Event objects, and using them to monitor and interact with alarms at run time, see Setting up FactoryTalk alarms on page 261.

Creating global objects

FactoryTalk View global objects link the appearance and behavior of one graphic object to multiple copies of the object in the same application.



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. Any graphic object you can create in FactoryTalk View can be a global object, except for ActiveX controls, OLE objects, and HMI tag alarm summaries.

For more information about creating global object displays, see Creating graphic displays on page 393.

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 into a standard graphic display (in the Displays folder), the copy is called a reference object. The original global object (in the Global Objects folder) becomes the copy's 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.

To create a reference object:

Use one of these methods:

- Copy a global object from a global object display into a graphic display in the same application or, in a network distributed application, within the same HMI server.
- Copy an existing reference object from one graphic display to another in the same application or, in a network distributed application, within the same HMI server.
- Add a global object display into the Displays folder in the same application or, in a network distributed application, within the same HMI server. In this case, all objects saved in the display are converted to reference objects.

  Tip: If you add a global object display into the Displays folder in a different HMI server, the global objects are converted to standard graphic objects, rather than reference objects.

To add a global object display into the Displays folder:

1. In FactoryTalk View Studio, in the Explorer window, right-click the Displays icon, and then select **Add Component Into Application**.
2. Browse to and select the global object display to be added, and then click **Open**.

Using placeholders to specify values for global objects

A tag placeholder is the cross-hatch 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 run time.

You can also set up tag placeholders for a global objects, to assign different run-time 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 run-time values to the linked reference objects.

For more information, see Setting up tag placeholders for global objects on page 430.

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

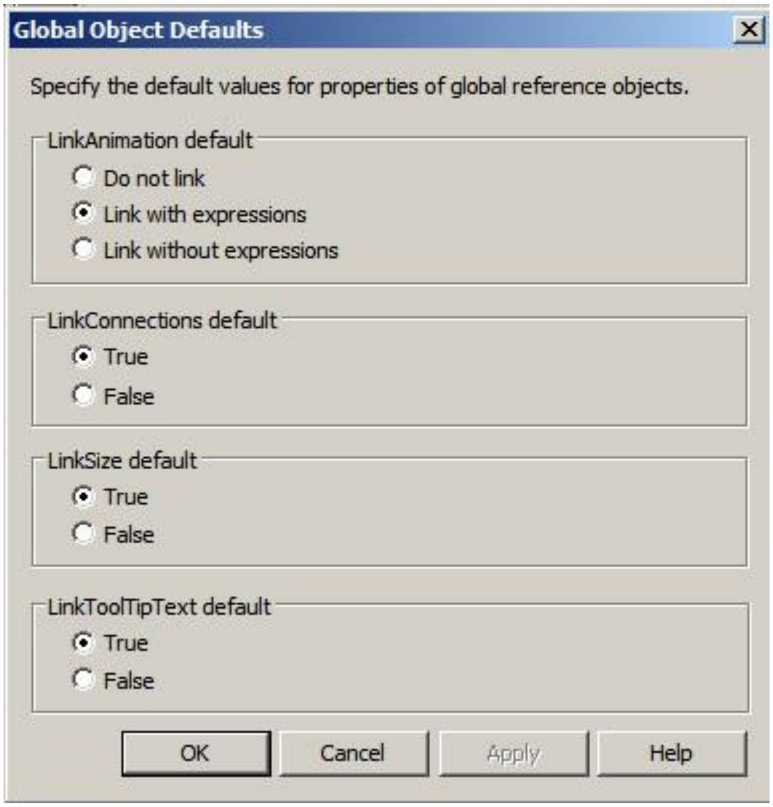
A reference object has properties that link it to its base object. You can set these properties when you copy (or drag and drop) a global object into a standard graphic display, or you can modify them later.

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 you want to modify, and then select **Global Object Defaults**.



Global Object Defaults

Specify the default values for properties of global reference objects.

LinkAnimation default
○ Do not link
● Link with expressions
○ Link without expressions

LinkConnections default
● True
○ False

LinkSize default
● True
○ False

LinkToolTipText default
● True
○ False

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**.

LinkAnimation

The LinkAnimation property determines whether the reference object uses the animation set up for its base object.

Set up the LinkAnimation property of the reference object by selecting:

- **Link with expressions**, to use the animation, including expressions, set up for the base object.
- **Link without expressions**, to use the animation, excluding expressions, set up for the base object. This means you can set up the expressions for the reference object separately.
- **Do not link**, to set up animation and expressions for the reference object separately.

For a Button push button object, the LinkAnimation property determines whether the reference object uses the press, release, or repeat actions set up for the base object. This is true when you select either **Link with expressions** or **Link without expressions**.

LinkConnections

The LinkConnections property determines whether the reference object uses the connections set up for its base object.

Set up the LinkConnections property of the reference object by selecting:

- **True**, to use the connections assigned to the base object.
- **False**, to set up connections for the reference object separately.

Tip: In FactoryTalk View Site Edition, a reference object that is a tag label, trend, or Button push button with an action other than Command, always uses the connections assigned to its base object. You cannot set up connections for these reference objects separately.

LinkSize

The LinkSize property determines whether the reference object uses the height and width set up for its base object.

Set up the LinkSize property of the reference object by selecting:

- **True**, to use the height and width set up for the base object.
- **False**, to set up the height and width of the reference object separately.

Tip: If you attempt to resize a reference object with its LinkSize property set to True, the object will snap back to its original size.

LinkToolTipText

The LinkToolTipText property specifies whether the reference object will use the tooltip text assigned to the base object.

Set up the LinkToolTipText property of the reference object by selecting:

- **True**, to use the base object's tooltip text. The tooltip property on the reference object will be read-only, and cannot be edited. This is the default.
- **False**, to allow the tooltip property in the reference object to be edited. Any change made to the tooltip text on the reference object will not be reflected on the base object. Any change made to this property on the base object will not be reflected on the reference object.

LinkBaseObject

The LinkBaseObject property specifies the name and location of the base object the reference object is linked to.

You cannot modify the LinkBaseObject property. If the specified global object display or base object does not exist, the reference object will appear in red outline, with a red cross on a white background.

To restore the reference object, you must recreate the global object with the same name, on the same global object display.

To find the base object linked to a reference object:

Right click the reference object (or group), and then select **Edit Base Object**.

This opens the global object display containing the base object, and selects the base object on the display.

Link properties of grouped reference objects

Grouped reference objects use the same properties as individual reference objects, to link to their grouped base objects. For grouped reference objects, you can set up:

- LinkBaseObject only for the group, not for individual objects within the group.
- LinkConnections only for individual objects within the group.
- LinkAnimation for the group and for individual objects within the group.

- LinkSize only for the group, not for individual objects within the group.

When the Link properties apply to grouped reference objects, the word Group is added to the property name in the Property Panel.

Breaking links between reference and base objects

To break the link between a reference object (or group) and its base object, use one of these methods:

- Right-click the reference object, and then select **Break Link**.
- Delete, remove, or rename the global object display containing the base object.
- Delete or rename the base object itself.

  Tip: If you delete an object that is part of a grouped based object, the corresponding object in the grouped reference object is also deleted, instead of being treated as a broken link.

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 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.

Creating OLE objects

When you create an OLE Object in a graphic display, you are actually inserting an object made by another application.

To create an OLE object in a graphic display:

1. From the **Objects** menu, select **OLE Object**, and then drag the tool to create a box the size needed to contain the OLE object.
2. In the Insert Object dialog box, select **Create New** to open the application within the graphics display and create an object, or choose **Create from File** to select and insert an existing object.

For details about inserting and modifying OLE objects, see the FactoryTalk View Site Edition Help.

Converting OLE objects

Convert an OLE object to an 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 2003, Visual Basic, and many other third-party applications.

For example, you can embed the Microsoft Forms ActiveX objects in an 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 an 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.

Creating ActiveX objects

When you create an ActiveX object in a graphic display, you are actually inserting an object made by another application.

To create an ActiveX object in a graphic display:

1. From the Objects menu, select ActiveX Control, and then drag the tool to create a box the size needed to contain the object.

2. In the Insert an ActiveX Control dialog box, select an object from the list that is presented.

For details about inserting and registering ActiveX objects, see the FactoryTalk View Site Edition Help.

Setting up ActiveX objects to interact with FactoryTalk View

To set up an ActiveX object to interact with FactoryTalk View you can:

- Connect the object's properties to tags. For details, see Assigning tags and expressions to objects on page 416.
- Connect the object's methods to tags. For details, see Connecting tags to an object's methods on page 540.
- Connect the object's events to tags. For details, see Connecting tags to an ActiveX object's events on page 541.
- Use VBA code. For more information, see Using the SE .Client object model and display code on page 663.

Setting up tools in the ActiveX toolbox

Use the ActiveX toolbox to set up tools for the ActiveX objects you use frequently.

By default, the ActiveX toolbox contains some of the Microsoft Forms 2.0 ActiveX controls that are installed with VBA, and the FactoryTalk View SE Signature Button.

For information about adding objects to the toolbox, see the FactoryTalk View Site Edition Help.

Tip: The ActiveX toolbox cannot be docked to the FactoryTalk View Studio workspace.

Deploying ActiveX components automatically at run time

FactoryTalk View SE clients can automatically install the correct versions of ActiveX controls used in graphic displays.

To deploy 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 located in:

- \Users\Public\Public Documents\RSView Enterprise\ActiveX Control Setup
- (Windows XP) \Documents and Settings\All Users\Shared 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 Symbol Factory

The Symbol Factory is a library of over 5,000 graphics for industrial automation, including pumps, pipes, valves, tanks, mixers, motors, ducts, electrical symbols, flow meters, material handling, sensors, PLCs, transmitters, and ISA symbols.

To open 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 drop down menu item.
- When the **Objects toolbar** is available, click the Symbol Factory icon .
- In the Image Browser, clicking the **Launch LibrarySymbol Factory** button.

To select a graphic:

1. Browse the Categories frame, clicking a category to see the symbols in the right frame.
2. Click the graphic to select it for the Graphic Display.

If the graphic is to be used in this original state, it can be dragged and dropped into the Graphic Display.

Tip: When FactoryTalk View Studio is unlicensed and in Demo mode, the Symbol Factory library will be in demo mode and restricted in the number and type of graphics available for use. Only a licensed FactoryTalk View Studio can enable a fully licensed Symbol Factory library.

Manipulating the graphic:

With the graphic selected, it can be modified using Symbol Factory

1. Select **Options** from the menu.
2. Select **Symbol Options** from the drop down menu.
3. The Symbol Options dialog allows the graphic to be modified in several ways. The orientation can be changed, it can be rotated, the fill color can be changed and applied to the graphic in different modes.
4. The Preview frame will show the changes made to the graphic.
5. When finished modifying, the graphic can be dragged and dropped from the Preview frame to the Graphic Display, or copied and pasted into the Graphic Display.

   Tip: Symbol Factory has its own Help file which explains operations within Symbol Factory and its functions.

For more information, click **Help** in the Symbol Factory.

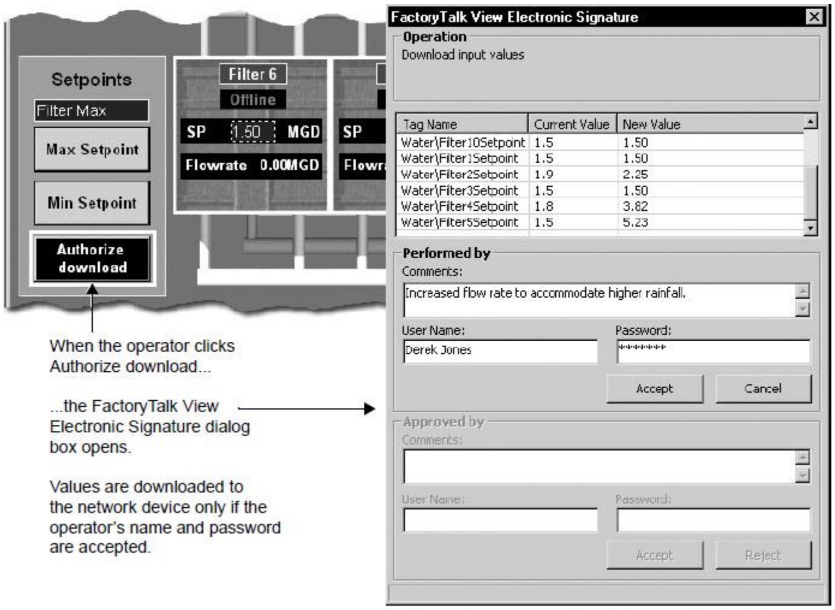
Using electronic signatures to authorize run-time changes

To secure run-time operations in FactoryTalk View SE, use the electronic signature feature to prevent operators from performing actions they aren’t authorized to perform.

The signature button can be used together with other features of FactoryTalk View SE, to meet the security standards required for regulated manufacturing applications, for example, those required for US Government 21 CFR Part 11 compliance.

The signature button verifies the operator’s identity, and allows an action only if the operator enters the appropriate user name and password.

The following illustration shows a signature button for authorizing downloads from a graphic display, and the FactoryTalk View Electronic Signature dialog box that opens when the signature button is pressed.



FactoryTalk View Electronic Signature

Operation
Download input values

Tag Name	Current Value	New Value
Water\Fiter10Setpoint	1.5	1.50
Water\Fiter1Setpoint	1.5	1.50
Water\Fiter2Setpoint	1.9	2.25
Water\Fiter3Setpoint	1.5	1.50
Water\Fiter4Setpoint	1.8	3.82
Water\Fiter5Setpoint	1.5	5.23

Performed by
Comments: Increased flow rate to accommodate higher rainfall.
User Name: Derek Jones
Password: ********

Accept Cancel

Approved by
Comments:
User Name:
Password:

Accept Reject

When the operator clicks Authorize download...

...the FactoryTalk View Electronic Signature dialog box opens.

Values are downloaded to the network device only if the operator's name and password are accepted.

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 85.

Securing tag writes, commands, and downloads

You can use the signature button to secure the following operations:

- Setting a tag value
- Running a FactoryTalk View command
- Downloading all input object values to a network device

At run time, if a user name or password is incorrect, or if other information, such as an operator's comment, is required but not provided, these operations can be prevented and an error message shows.

In addition, the signature button can be set up to require authorization by another person, such as a supervisor, before the operation can be carried out.

For added security, in the Connections tab of the signature button's Properties dialog box, use the security function CurrentUserHasCode in an expression assigned to the button's Enabled property.

This checks whether the current user has the security code assigned to the graphic display.

For more information about securing graphic displays, see Assigning security codes to graphic displays on page 103.

Creating signature buttons

The electronic signature button is an ActiveX object. The tool to create an electronic signature button is in the ActiveX toolbox.

To create an electronic signature button:

1. From the **View** menu, select **ActiveX Toolbox**.
2. In the ActiveX Toolbox, click the **FactoryTalk View SE Signature** button.
3. Drag the tool to create a box on the display that is the size the button is to be.
4. Right-click the signature button, and then select **Properties** to open the FactoryTalk View SE Signature Button Properties dialog box.

Securing objects in graphic displays

The signature button does not change the way graphic objects in a display work.

For example, numeric input objects in a display will download their values to a network device when the operator presses Page Down, even if there is a signature button on the display.

To ensure that values are downloaded only if the electronic signature is verified, you must secure the graphic objects separately. To do this, use one of these methods:

- Turn off the **Enter** key, by using the */E* parameter with the **Display** command. For details about the **Display** command, see the FactoryTalk View Site Edition Help.
- Use display keys to set the Page Up, Ctrl+Page Up, Page Down, and Ctrl+Page Down keys to do nothing. For details about setting up display keys, see Setting up display keys on page 534.

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 name of the person approving the change.

For information about logging system activity, see Logging system activity on page 383.

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.

Following are some of the ways 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 516.

- 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 533.

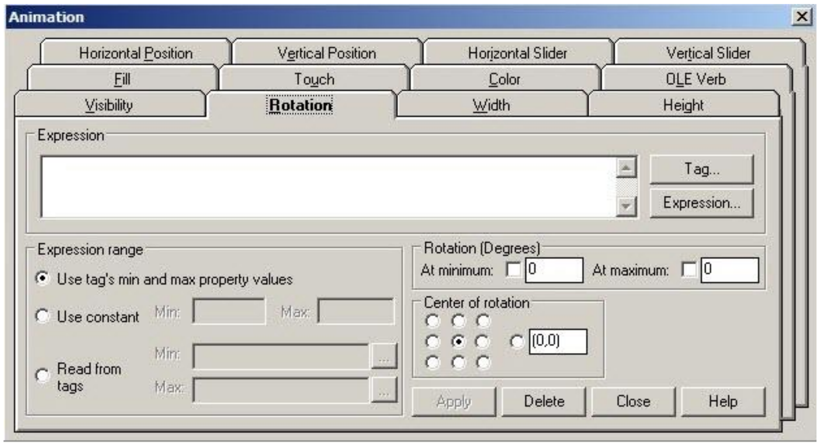
- Animate an ActiveX object:
  - By assigning tags to the ActiveX object's properties, so that the tags' values change when the properties change or, in some cases, the properties change when tag values change.
  - By running commands based on the ActiveX object's events.
  - Mapping tags to the ActiveX 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 538.

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.



● Use tag's min and max property values
○ Use constant
○ Read from tags
At minimum: □ 0
At maximum: □ 0
Center of rotation: (0,0)

For details about options in each Animation tab, click **Help**.

Tip: To set up 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 533.

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 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 518.

To open the Animation dialog box:

1. Select the object or object group you want to apply animation to.
2. Select **View menu** > **Animation**, and then select the type of animation to apply.

   Types of animation an object can't 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 animation to a tag value, you have to specify a tag name or a tag placeholder.

To set up 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 a cross-hatch character (#) followed by a number between 1 and 500.

Tag placeholders used in graphic displays are replaced with tag names when the display is run. To apply 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 428.

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 stand alone as the entire expression.

For more information about expressions, see Creating expressions on page 555.

To set up 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 633.

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 start and end points for the object's range of motion.

When setting up 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 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 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

With Object Smart Path, you can easily set the range of motion for an object, by moving the object on the graphic display. The following example shows how this feature works.

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** check box.
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** check box.
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 run time, 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 412.

Tip: Combining different types of animation can produce unexpected results. Be sure to test group animation, to ensure that the animation achieves the effect you want.

Testing animation

To test the animation applied to objects in a graphic display quickly, switch to test display mode in the Graphics editor.

When you are finished testing, switch back to edit display mode to continue working on the display. To switch between test and edit modes, use the buttons on the toolbar or the items on the View menu.

Testing a 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 651.

Tip: If the animation applied to objects in a graphic display uses device tags, to test the animation, you must set up communications with the devices or data servers. Before deploying an application, test it in a 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. For details, see Viewing display contents in the Object Explorer on page 397.
- To view animation applied to individual objects within a group, use the group edit feature. For details, see Modifying grouped objects on page 412.

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 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 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 501.

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 516.

Showing and hiding objects

Use 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 an object's color

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.

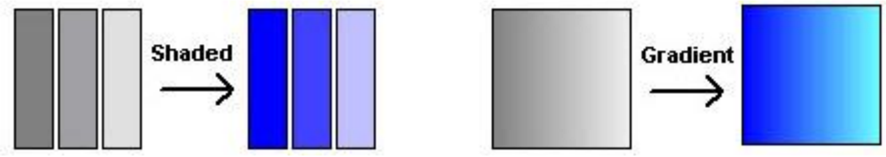
Gradient - Allows you to change the colors, and they will have a gradient look to them. If you select Gradient as the Fill Style, the Fill color box changes to gradient. Click it to bring up the gradient fill dialog box to configure the gradient fill.

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 run time, when the value reaches or crosses the threshold, the color of the object changes.

Tip: Color animation does not affect string input, numeric input, or recipe objects. Color for these objects is defined in the Display Settings dialog box.

Keeping 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**.

Examples

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. For details, Creating text objects on page 455.
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** check box. (Only line color affects text.)
7. Select the Blink color check box next to the **Blink** check box 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 as the fill level changes

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 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 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 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 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 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

Use 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**.

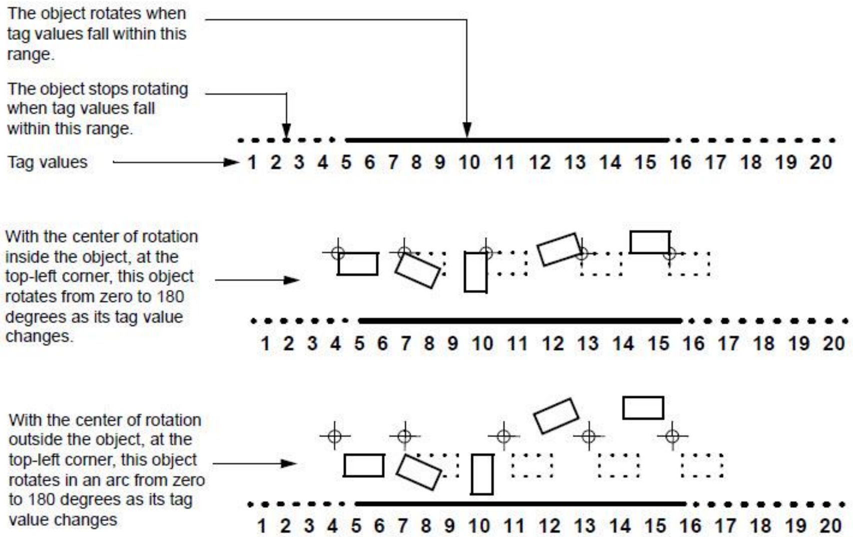
Setting up rotation animation

To set up 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.


The object rotates when tag values fall within this range.
The object stops rotating when tag values fall within this range.
Tag values
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
With the center of rotation inside the object, at the top-left corner, this object rotates from zero to 180 degrees as its tag value changes.
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
With the center of rotation outside the object, at the top-left corner, this object rotates in an arc from zero to 180 degrees as its tag value changes
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

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 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 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 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 run time. 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 460.

For details about Touch options in the Animation dialog box, click **Help**.

Creating a horizontal slider

Use 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 Horizontal Slider options in the Animation dialog box, click **Help**.

Creating a vertical slider

Use 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**.

Animating OLE verbs

Use 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:

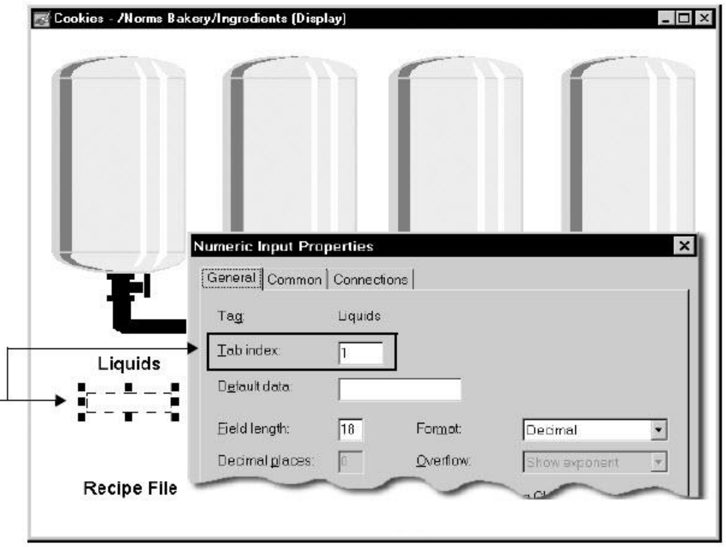
- Numeric and string input objects
- Button objects
- Objects with object key animation
- ActiveX objects

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:


Cookies - /Norms Bakery/Ingredients (Display)

Numeric Input Properties

General | Common | Connections

Tag: Liquids

Tab index: 1

Default data:

Field length: 18

Format: Decimal

Decimal places: 0

Overflow: Show exponent

The Properties dialog box for the selected input object shows the object has tab index number 1.

You can also check the tab index number using one of these methods:

- Right-click the object, select **Property Panel**, and then 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:

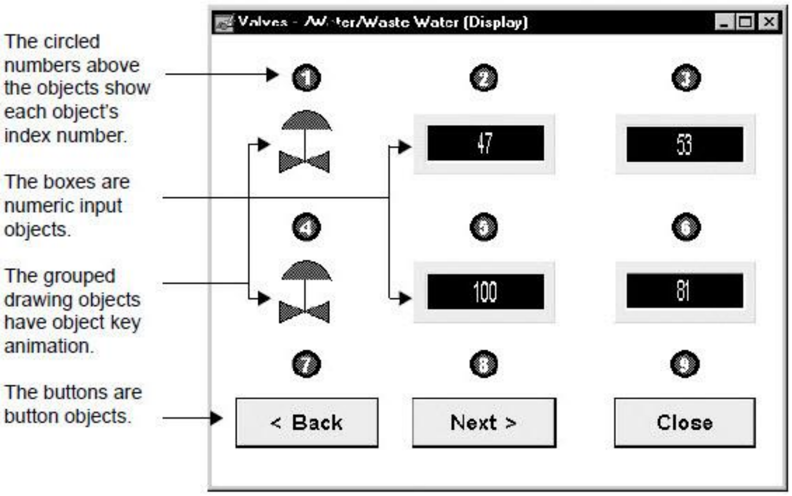
- Determine a tab sequence for objects that an operator can navigate to in a display.

- Move among objects using the **Position**, **NextPosition**, and **PrevPosition** commands at run time. 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. For more information, see Creating and restoring recipes on page 489.

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:


The circled numbers above the objects show each object's index number.
The boxes are numeric input objects.
The grouped drawing objects have object key animation.
The buttons are button objects.
Valves - Water/Waste Water (Display)
1
2
3
47
53
4
5
6
100
81
7
8
9
< Back
Next >
Close

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 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 renumbers 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 run time.

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 550.

About client keys

Client keys are different from object and display keys. Client keys are always active at run time. 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 649. For information about precedence amongst object, display, and client keys, see General rules governing precedence on page 548.

Setting up object keys

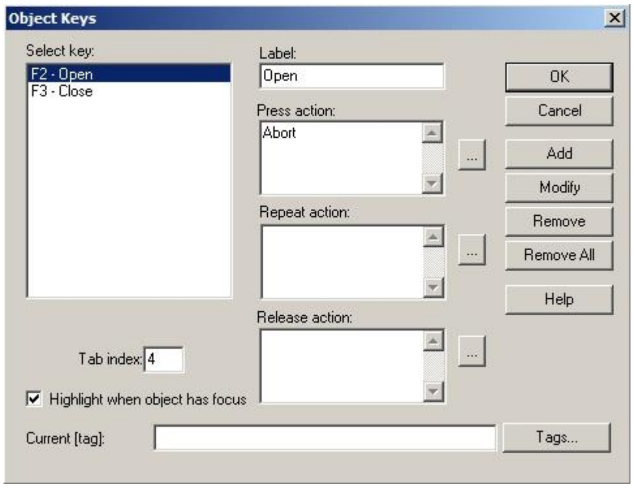
Use object keys to connect a graphic object with one or more keys.

For example, if you connect a rectangle to the F4 key, at run time, 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 run time.

To open the Object Keys dialog box:

Right-click the object you want to set up keys for, and then select **Object Keys**.



Object Keys

Select key:
F2 - Open
F3 - Close

Label: Open

Press action: Abort

Repeat action:

Release action:

Tab index: 4

☑ Highlight when object has focus

Current [tag]:

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 run time, 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 run time, 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.

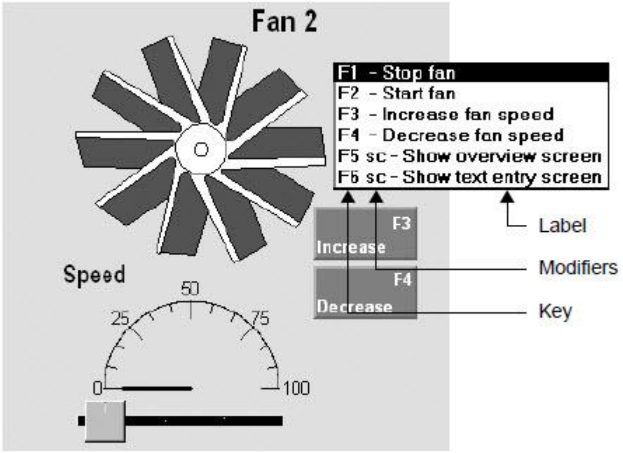
For details about how keys and other interactive objects work at run time, see Specifying the behavior of interactive objects on page 442.

Viewing the key list at run time

An operator can open a key list at run time, 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.

Note: 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 a key is pressed.

To show the key list:

- Use the */O* 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 run time, 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 run time, 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 run time, when the operator selects any equipment object and presses the appropriate Display key, the help file for that piece of equipment is shown.

Example 3: Opening an object-specific HMI tag alarm summary

Imagine you want to have HMI tag alarm information available for a specific machine. If something goes wrong and causes an alarm, you want the operator to be able to show an alarm summary for that machine.

To show the HMI tag alarm summary for the machine, the operator can position the cursor over any object related to that machine in a graphic display, and then press a **Display Key**.

1. Create an HMI tag alarm summary that uses a filter containing only the alarms for the machine. The filter must contain a tag placeholder (for example, #1).

   For information about creating an HMI tag alarm summary, see Setting up HMI tag alarms on page 217.

2. In the Display Keys dialog box, assign a display key. In the **Press Action** box, type *Display AlmSumm /t[tag]*.

3. In the **Object Keys** dialog box, type the name of the machine you want to specify in the alarm summary filter.

   At run time, the value of *[tag]* in the Display Keys, replaces *#1* in the alarm summary filter, and then shows HMI tag alarms for the selected machine only.

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 run time. 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 663.

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 run time.

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. For more information, see Viewing object properties in the Property Panel on page 398.

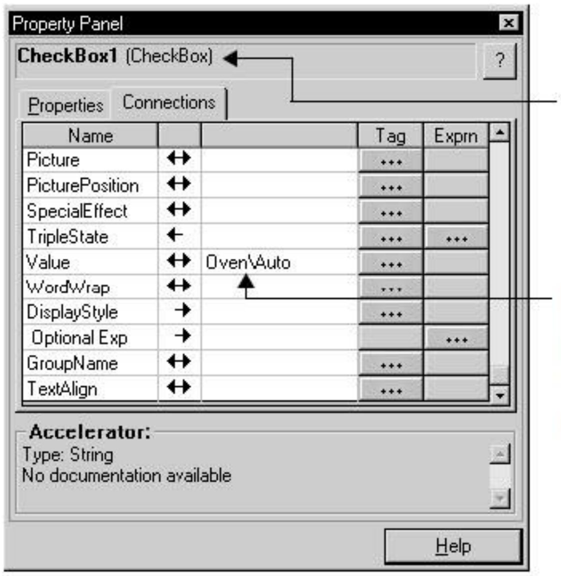
Note: The run-time behavior of ActiveX objects depends on the vendor's implementation. Before making your application available to users, test ActiveX objects thoroughly to ensure they behave as desired.

Connecting tags to an ActiveX object's properties

ActiveX objects have sets of different properties. To view an object's properties, right-click the object, and then select **Property Panel**.

The content of the Property Panel is determined by the application that created the ActiveX object, not by FactoryTalk View.

In the following illustration, the Property Panel is for an ActiveX check box named **CheckBox1**. Clearing this check box changes an oven's temperature control from Automatic mode to Manual mode.



CheckBox1 (CheckBox)

Name			Tag	Exprn
Picture	↔		...	
PicturePosition	↔		...	
SpecialEffect	↔		...	
TripleState	←		...	...
Value	↔	Oven\Auto	...	
WordWrap	↔		...	
DisplayStyle	→		...	
Optional Exp	→			...
GroupName	↔		...	
TextAlign	↔		...	

Accelerator:
Type: String
No documentation available

CheckBox1 is the default name of this object.

The object's Value property is assigned to the Auto tag in the Oven folder.

This connects the check box to an oven's temperature control.

For information about a property, select the property, and then click **Help**.

About ActiveX object names

As shown in 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 541.

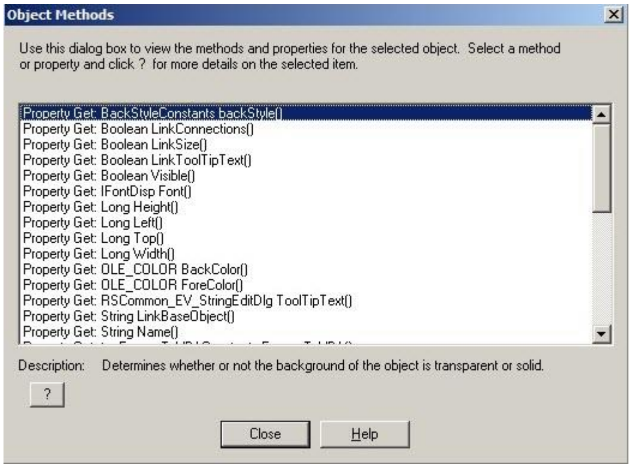
- With commands.

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 540.

You can change the default name for any object. For more information about object naming, see Naming graphic objects on page 415.

Viewing an object's methods

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**.



Object Methods

Use this dialog box to view the methods and properties for the selected object. Select a method or property and click ? for more details on the selected item.

```
Property Get: BackStyleConstants backStyle()
Property Get: Boolean LinkConnections()
Property Get: Boolean LinkSize()
Property Get: Boolean LinkToolTipText()
Property Get: Boolean Visible()
Property Get: IFontDisp Font()
Property Get: Long Height()
Property Get: Long Left()
Property Get: Long Top()
Property Get: Long Width()
Property Get: OLE_COLOR BackColor()
Property Get: OLE_COLOR ForeColor()
Property Get: RSCommon_EV_StringEditDlg ToolTipText()
Property Get: String LinkBaseObject()
Property Get: String Name()
```

Description: Determines whether or not the background of the object is transparent or solid.

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.

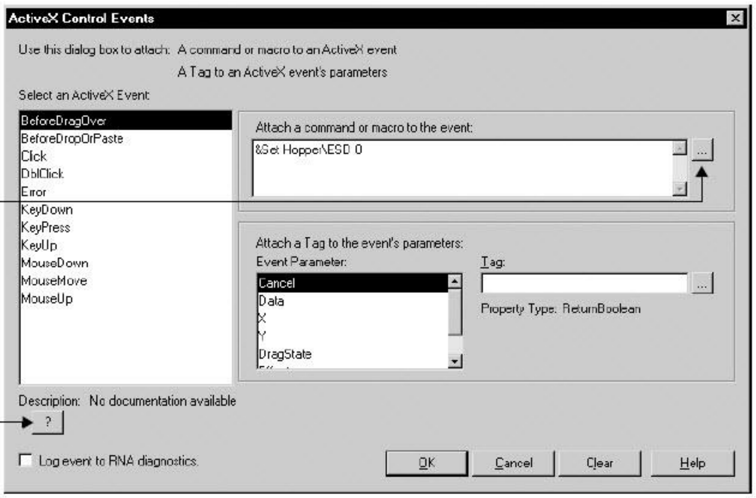
Connecting tags to an ActiveX object’s events

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 run time.

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**.



ActiveX Control Events

Use this dialog box to attach: A command or macro to an ActiveX event
A Tag to an ActiveX event's parameters

Select an ActiveX Event:
BeforeDragOver
BeforeDropOrPaste
Click
DblClick
Error
KeyDown
KeyPress
KeyUp
MouseDown
MouseMove
MouseUp

Attach a command or macro to the event:
&Set Hopper\ESD 0

Attach a Tag to the event's parameters:
Event Parameter:
Cancel
Data
X
Y
DragState

Tag:

Property Type: ReturnBoolean

Description: No documentation available

□ Log event to RNA diagnostics.

Click here to open the Command Wizard.

Click here for information from the vendor, about the object and its events.

For details about options in the ActiveX Control Events dialog box, click **Help**.

Chapter 19

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.

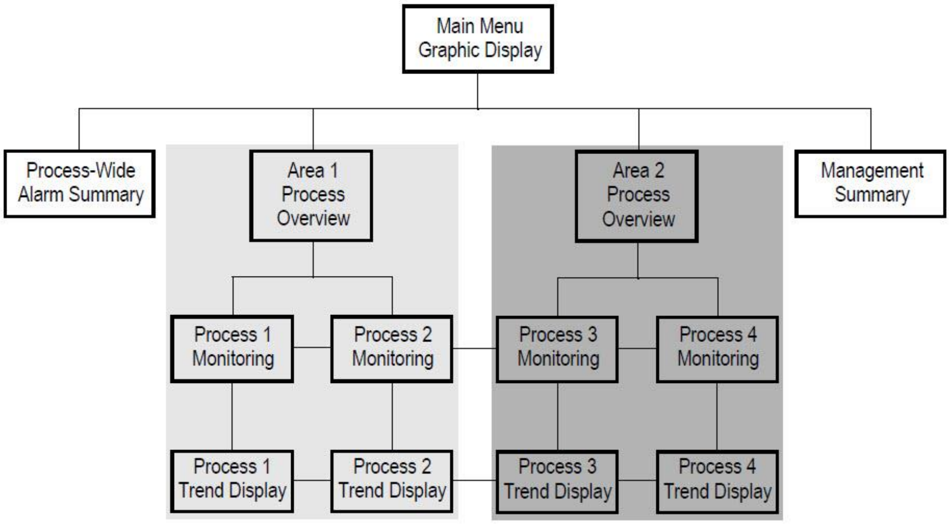
The display hierarchy can represent parts of a plant or process, as well as different types of data displays. For example, the top level might represent an area in the plant, and the bottom level might contain trends and alarm displays specific to each area.

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 located 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 appropriate resolution to run the application.

Setting up ways to move among displays

FactoryTalk View gives you the tools for linking graphic displays and creating an overall application structure that is easy for an operator to use.

You can create an application that is keyboard-based, touch screen-based, or combines both navigation methods.

Although the 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 run time.

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 **Display Settings** dialog box), the Display 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 Display command makes the display active. A display of type: • **Replace** opens on top of other open displays, and closes the ones it overlaps. • **Overlay** opens on top of any open displays, but does not close them. • **On Top** opens on top of any open displays and remains in the foreground.	**Display**
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.	**Abort**
Pull the specified graphic display in front of other open displays. If the specified display is of type Replace or Overlay, the PullForward command gives the display focus, and positions it behind any On Top display that is open.	**PullForward**
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.	**PushBack**

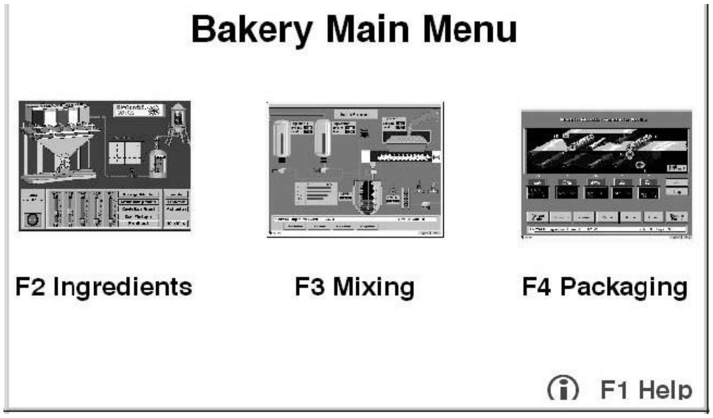
Tip: The **PullForward** and **PushBack** commands provide quick display changes because displays are already open. However, the more displays you have open, and the more complex the displays are, the more memory and CPU are used.

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 651.

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



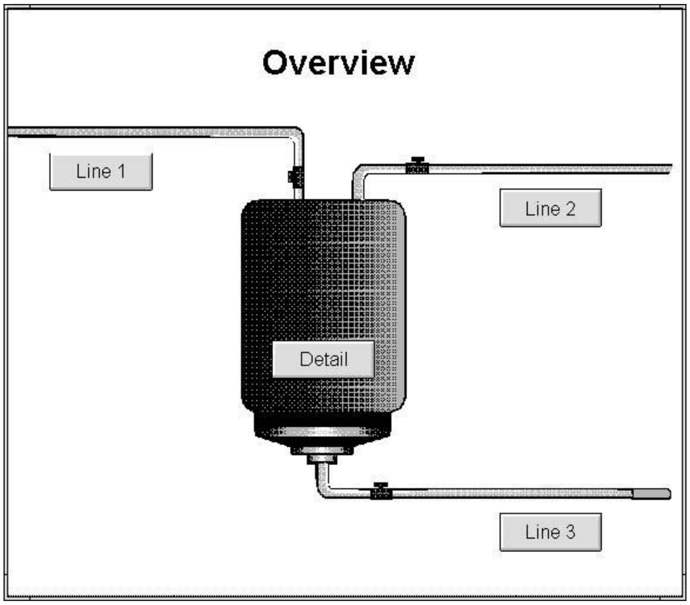
To create this display, the designer assigned various FactoryTalk View commands to keys using the three types of key definitions: object, display, and client. In all cases, keys (not mouse buttons) were defined to run commands.

Object keys and display keys are set up in the Graphics editor. For more information, see Animating graphic objects. on page 515

Client keys are created in the Client Keys editor. For more information, see Creating client keys on page 649.

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 on 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 460.

Choosing display types with navigation in mind

When designing an application, the display types you choose 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 435.

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 445.

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 437.

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 649.

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	Setting up object keys on page 533.
Associate a key with a specific graphic display (display key)	Display key animation in the Graphics editor	Setting up display keys on page 534.
Create a key that works everywhere on an FactoryTalk View client (client key)	A key definition component in the Client Keys editor	Creating client keys on page 649.

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 popup 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 run time, 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.

Say, for example, that 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.

However, if a key that triggers an action in an embedded ActiveX 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.

However, if a key that triggers an action in an embedded 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.	**Tab**
Move focus to the object with the next lowest index number.	**Shift+Tab**
Move focus to the next object, in the direction the arrow key points.	**Ctrl+Up Arrow**, **Ctrl+Left Arrow**, **Ctrl+Down Arrow**, **Ctrl+Right Arrow**
Move focus to the next window.	**Ctrl+F6**
Move focus to the previous window.	**Ctrl+Shift+F6**
Close the active window.	**Ctrl+F4** or **Ctrl+Shift+F4**
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 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 recipe object with focus is set up to display the keyboard.	**Enter**

Upload data into all input objects in the display.	**PgUp**
Upload data into the input object that has focus.	**Ctrl+PgUp**
Download data from all input objects in the display.	**PgDn**
Download data from the input object that has focus.	**Ctrl+PgDn**
Delete the contents of the input object.	**Home+Shift+End+Del**
Move input focus to the recipe object, and prepare for a recipe restore.	**Ctrl+R**
Move input focus to the recipe object, and prepare for a recipe save.	**Ctrl+W**
Open the Recipe dialog box. If Ctrl+W was pressed previously, the recipe is saved. If Ctrl+R was pressed previously, the recipe is restored.	**+** on the numeric keypad
Move the selection bar on the Object Key menu.	**Up Arrow**, **Down Arrow**
Close the Object Key menu, or exit input mode for the updating input object that has focus.	**Esc**
Move the cursor one position left or right.	**Left Arrow**, **Right Arrow**
Delete the character to the left of the cursor.	**Backspace**
Delete the character to the right of the cursor.	**Del**
Delete all characters from the cursor position to the end of the line.	**Shift+End+Del**
Copy the selected items to the clipboard.	**Ctrl+C or Ctrl+Ins**
Cut the selected items and place them in the clipboard.	**Ctrl+X**, or **Shift+Del**
Paste the contents of the clipboard at the current cursor position.	**Ctrl+V**, or **Shift+Ins**
Position the cursor at the beginning of the data entry object.	**Home**

Tip: The arrow keys perform different actions when a trend graphic object has focus. For details, see Using the arrow keys on page 629.

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 649.

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 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 on Help in the FactoryTalk View SE Client wizard. See also Setting up the properties of a graphic display on page 434.

How navigation buttons work

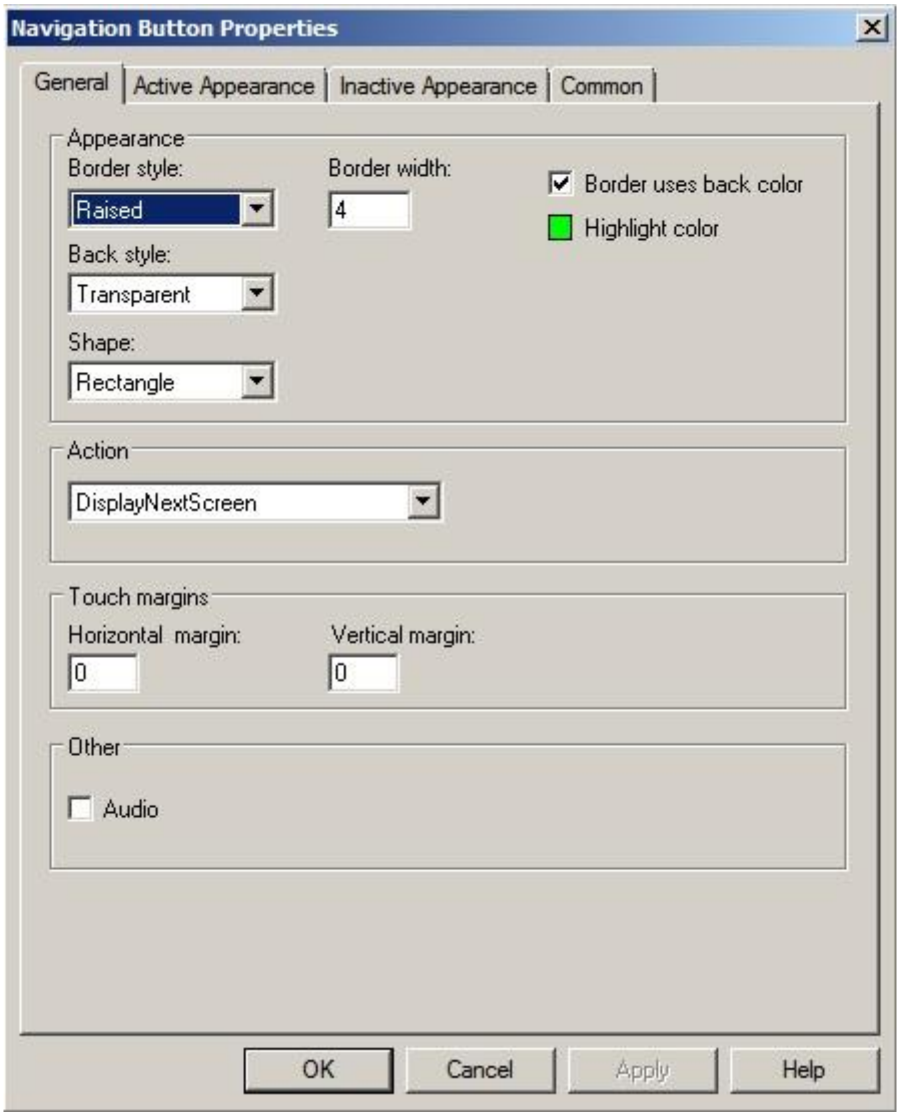
When the button is active, clicking it starts the corresponding action that is set on the **Navigation Button Properties** dialog box. If the button is inactive, clicking it causes no response.

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 a navigation button

1. In the **Graphics** editor, select **Objects** > **Push Button** > **Navigation,** or click the navigation button icon in the **Objects** toolbox.

2. Drag the mouse to position and draw a shape the general size and location you intend the button to be. When you release the mouse button, the **Navigation Button Properties** dialog box opens.



Appearance

Border style: Raised

Border width: 4

☑ Border uses back color

🟩 Highlight color

Back style: Transparent

Shape: Rectangle

Action

DisplayNextScreen

Touch margins

Horizontal margin: 0

Vertical margin: 0

Other

□ Audio

3. Make your changes on the **Navigation Button Properties** dialog box.

4. Close the dialog box.

5. Save your changes.

Setting up the properties for a navigation button

1. In a display, double-click the object that you created for the navigation button (or right-click the object) to open the properties dialog box.

2. In the **Navigation Button Properties** dialog box, click the **General** tab.

3. Specify these properties:

   - General appearance under **Appearance**.
   - Type of action the button initiates under **Action**.
   - Touch-insensitive margins under **Touch margins**.
   - Audio (whether the button beeps at run time) under **Other**.

4. Click the **Active Appearance** and **Inactive Appearance** tabs to specify how the button looks at run time.

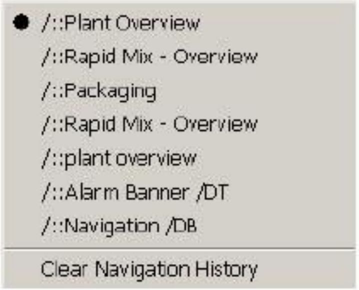
5. Click the **Common** tab to set up how the navigation button will appear at run time.

For additional details on the **Navigation Button Properties** dialog box, click **Help**.

Viewing and clearing the navigation history

The DisplayNavigationHistory button shows a list of all the previously viewed graphic displays, let you view a display in the history list, and let you clear the history list.

- Show the navigation history list by selecting the button you created with the Action set to DisplayNavigationHistory.



The current display is identified with a bullet in front of the display name.

- To clear the navigation history, select **Clear Navigation History**.
- Click a display name in the list to view the display.

  Tip: To change the name of the display for the navigation history list, see Tracking screens for navigation on page 440.

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.

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.

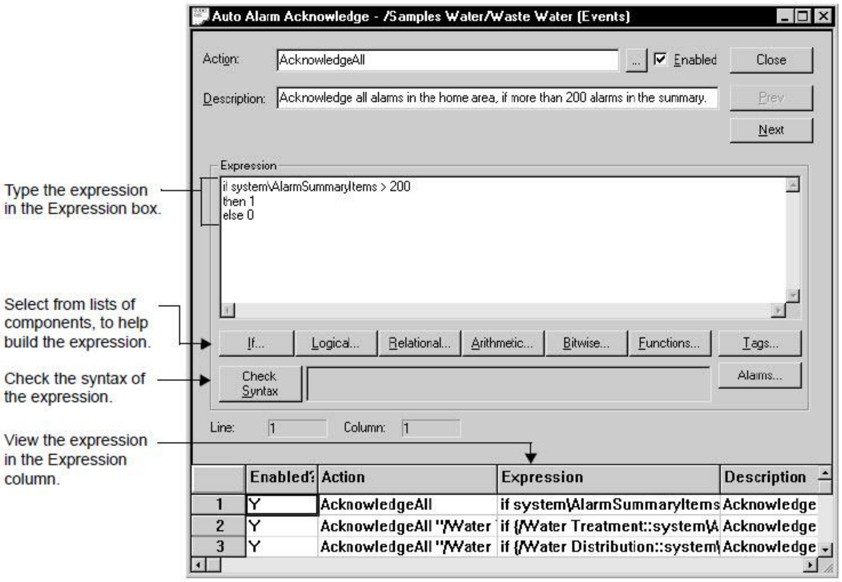
- In the **Derived Tags editor**, specify the name of a tag that will store the result of an expression.
- In the **Events editor**, associate an expression with an action (a command or macro). When the expression changes from *false* to *true* (from *0* to any non-zero value), the action runs.

- 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.



Auto Alarm Acknowledge - /Samples Water/Waste Water (Events)

Action: AcknowledgeAll ☑ Enabled

Description: Acknowledge all alarms in the home area, if more than 200 alarms in the summary.

Expression

```
il system\AlarmSummaryItems > 200
then 1
else 0
```

Line: 1 Column: 1

	Enabled?	Action	Expression	Description
1	Y	AcknowledgeAll	if system\AlarmSummaryItems	Acknowledge
2	Y	AcknowledgeAll "/Water	if {/Water Treatment::system\A	Acknowledge
3	Y	AcknowledgeAll "/Water	if {/Water Distribution::system\	Acknowledge

Type the expression in the Expression box.

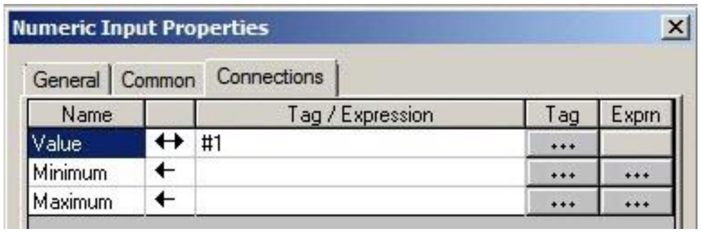
Select from lists of components, to help build the expression.

Check the syntax of the expression.

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.



Numeric Input Properties

General | Common | Connections

Name		Tag / Expression	Tag	Exprn
Value	↔	#1	...	
Minimum	←		...	...
Maximum	←		...	...

Expression components

You can build an expression using any of the following components:

- Tag values (or tag placeholders)
- Constant values

- Mathematical, relational, logical, and bitwise operators
- Built-in functions
- If - then - else logic

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...
FactoryTalk alarms (in the Alarm Source Browser)	Alarms...

Checking the syntax of an expression

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 this 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. You can use wildcards, for example, when using HMI tag alarm functions. For more information about these built-in functions, see Tag functions on page 567.

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 189.

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 195.

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 cross-hatch character (#) followed by a number from 1 to 500.

At run time, 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 428.

Using constants in expressions

To provide a constant value for an expression, use any of the following formats:

- Integer (123)
- Floating point (123.45)
- Scientific notation (1.2345E^{2})
- String constant ("character string")
- Pi string (to represent the symbol π□)

  FactoryTalk View replaces the string with its numeric value.

Using operators to modify values 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.

The following table describes the arithmetic operators, with examples. (In the examples, tag1 = 5 and tag2 = 7.)

Symbol	Operator	Example
+	addition	tag1 + tag2 returns a value of 12
-	subtraction	tag1 – tag2 returns a value of –2
*	multiplication	tag1 * tag2 returns a value of 35

/	division	tag1 / tag2 returns a value of 0.7142857
MOD, %	modulus (remainder)	tag2 MOD tag1 returns a value of 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 **Important:** This operator is for integers only, not floating point numbers.
**	exponent	tag1 ** tag2 returns a value of 78125

Tip: 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 run time.

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 numeric or string values, to provide a true or false result. If the statement is true, the expression returns a value of 1. If the statement is false, the expression returns a value of 0.

The following table describes the relational operators, with examples. (For the numeric examples, tag1 = 5 and tag2 = 7. For the string examples, serial_no = ST009.)

Symbol	Operator	Numeric examples	String examples
EQ, = =	equal	tag1 = = tag2 false	serial_no = = "ST011" false
NE, < >	not equal	tag1 < > tag2 true	serial_no < > "ST011" true
LT, <	less than	tag1 < tag2 true	serial_no < "ST011" true
GT, >	greater than	tag1 > tag2 false	serial_no > "ST011" false
LE, < =	less than or equal to	tag1 < = tag2 true	serial_no < = "ST011" true
GE, > =	greater than or equal to	tag1 > = tag2 false	serial_no > = "ST011" false

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. There are three logical operators: AND, OR, and NOT. If the expression is true, the operators return a non-zero value. If the expression is false, the operators return *0*.

The following table describes the logical operators, with examples. (In the examples, tag1 = 5 and tag2 = 7.)

Symbols	Operator	Action	Example
AND, & &	and	If the statements to the right and to the left of the operator are both true, returns a *1*.	(tag1 < tag2) AND (tag1 = = 5) Both statements are true; returns a *1*.
OR, ||	or	If either or both statements are true, returns a *1*.	(tag1 > tag2) OR (tag1 = = 5) tag1 = = 5 is true; returns a *1*.
NOT	negation	Reverses the logical value of the statement it operates on.	NOT (tag1 < tag2) Although tag1 < tag2 is true, NOT reverses the logical value; returns a *0*.

Tip: The parentheses are essential in the these expressions. For more information, see Evaluation order of operators on page 562

Bitwise operators

Bitwise operators examine and manipulate individual bits within a value.

The following table describes the bitwise operators.

Tip: Bitwise operators are for integers only, not for floating-point numbers.

Symbol	Operator	Action
&	AND	Compares two integers or integer tags on a bit-by-bit basis. If both the corresponding bits in the original numbers are *1*, it bit set to *1*. Otherwise, the resulting bit is *0*.

Tip: Bitwise operators are for integers only, not for floating-point numbers.

|	inclusive OR	Compares two integers or tags on a bit-by-bit basis. If either or both of the corresponding bits in the original numb integer with a bit set to *1*. If both bits are *0*, the resulting bit is
^	exclusive OR (XOR)	Compares two integers or tags on a bit-by-bit basis. If the corresponding bits in the original numbers differ, it eturr to *1*. If both bits are *1* or both are *0*, the resulting bit is *0*.
>>	right shift	Shifts the bits within an integer or tag to the right. Shifts the bits within the left operand by the amount specified bit on the right disappears. Either a *0* or a *1* is shifted in on the left, depending on whethe or a *1*. If the left-most bit is *0*, a *0* is shifted in. If the left-most In other words, the sign of the number is preserved.
<<	left shift	Shifts the bits within an integer or tag to the left. Shifts the bits within the left operand by the amount specified bit on the left disappears and *0* always shifts in on the right.
~	complement	Returns one’s complement; that is, toggles the bits within an i Reverses every bit within the number so every *1* bit becomes

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, evaluation order is determined as follows:

- Operators in parentheses are evaluated first. Use parentheses to change the order.
- Two operators of the same precedence are evaluated from left to right.
- Operators not enclosed in parentheses are evaluated as follows:

Evaluation order	Symbols
1 (highest)	( )
2	NOT ~
3	* / MOD, % ** AND, & & & >> <<
4	+ – OR, || | ^
5	EQ, = NE, < > LT, < GT, > LE, < = GE, > =

Examples: Evaluation order

In the following examples, tag1 = 5, tag2 = 7, and tag3 = 10.

(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).

---

tag1 > tag2 AND tag3

is evaluated in this sequence:

1. tag2 AND tag3 = 1
2. tag1 > 1 = 1

The expression evaluates to 1 (true).

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 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.

This function	Returns this value
SQRT (expression)	The square root of the expression
LOG (expression)	The natural log of the expression
LOG10 (expression)	The base ten log of the expression
SIN (expression)	The sine of the expression in radians
COS (expression)	The cosine of the expression in radians
TAN (expression)	The tangent of the expression in radians
ARCSIN (expression)	The arc sine of the expression in radians
ARCCOS (expression)	The arc cosine of the expression in radians
ARCTAN (expression)	The arc tangent of the expression in radians
SIND (expression)	The sine of the expression in degrees
COSD (expression)	The cosine of the expression in degrees
TAND (expression)	The tangent of the expression in degrees

ARCSIND (expression)	The arc sine of the expression in degrees
ARCCOSD (expression)	The arc cosine of the expression in degrees
ARCTAND (expression)	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 path name, surrounded by quotes. The drive parameter is the drive letter.

This function	Returns this value	Example
FILE_EXISTS("*file*")	1 (true) if the specified file exists	FILE_EXISTS("C:\Documents and Settings\All Users\Shared Documents\ RSView Enterprise\SE\ActivityLog\ Activity.exp") Returns 1 (true) if the file exists or 0 (false) if the file does not exist. Use this function to set a tag when a specified file has been created or deleted.
FREE_BYTES(*drive*)	The number of bytes free on the specified drive	FREE_BYTES(c) Returns the number of bytes available on drive C, up to a maximum of 2.1 GB. Use this function to display a message or trigger an alarm when disk space is getting low.

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.

This function	Returns
TIME("time")	*1* (true) if the time specified is the current time.
BEFORE_TIME("time")	*1* (true) if the expression is evaluated before the specified time.
AFTER_TIME("time")	*1* (true) if the expression is evaluated after the specified time.
INTERVAL("interval")	*1* (true) if the specified time interval has elapsed—the interval timer starts running when an event file starts running.

Tip: The time and interval parameters must be enclosed in quotes.

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 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

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, HMI tag alarm, or HMI alarm event (or multiple tags, or alarm events).

For information about setting up HMI tag alarms, see Setting up HMI tag alarms on page 217.

This function	Returns this value
alm_ack(tag or alarm event) *or* alm_ack(tag*)	*1* (true) if the tag’s alarm or the alarm event has been acknowledged. If examining multiple tags, 1 (true) if one or more alarms have been acknowledged.

alm_allacked(tag or alarm event) *or* alm_allacked(tag *)	*1* (true) if the tag's alarm or the alarm event has been acknowledged. If examining multiple tags, 1 (true) if all tags matching the pattern have been acknowledged. If any of the tags has not been acknowledged, the expression returns 0.
alm_fault(tag or alarm event) *or* alm_fault(tag *)	*1* (true) if there has been an alarm fault for the specified tag or alarm event. If examining multiple tags, 1 (true) if there has been an alarm fault for one or more of the tags.
alm_in_alarm(tag or alarm event) *or* alm_in_alarm(tag *)	*1* (true) if the tag or alarm event is in alarm. If examining multiple tags, 1 (true) if one or more of the tags or alarm events are in alarm.
alm_level(tag or alarm event) *or* alm_level(tag *)	The alarm level or threshold for an analog tag or alarm event. This can be value between 1 and 8, or 0 if the tag is not in alarm. If examining multiple tags, the highest level of the tags in alarm. For example, if the current alarms are levels 2, 6, and 8, this function returns the value 8.
alm_severity(tag or alarm event) *or* alm_severity(tag*)	The severity of the alarm or alarm event. This can be a value between 1 and 8, or 0 if the tag or alarm event is not in alarm. If examining multiple tags, the highest severity of the tags or alarm events that are in alarm. For example, if the current alarms have severities of 1, 3, and 6, this function returns the value 1.
alm_suppress(tag or alarm event) *or* alm_suppress(tag *)	*1* (true) if the tag's alarms are suppressed. For multiple tags, 1 (true) if one or more of the tags' alarms are suppressed.
alm_tags_inalm(tag or alarm event) *or* alm_tags_inalm(tag *)	*1* (true) if the tag or alarm event is in alarm. If examining multiple tags, the number of tags or events in the query that are in alarm, or 0 if none are in alarm.
alm_tags_inalm_unack(tag or alarm event) *or* alm_tags_inalm_unack(tag *)	*1* (true) if the tag or event is in alarm and unacknowledged. If examining multiple tags, the number of tags or events in the query that are both in alarm and unacknowledged, or 0 if none are in alarm and unacknowledged.
alm_tags_unack(tag or alarm event) *or* alm_tags_unack(tag *)	*1* (true) if the tag or alarm event is unacknowledged; 0 if the tag or event is acknowledged. If examining multiple tags, the number of tags or alarm events in the query that are unacknowledged, or 0 if all are acknowledged.
comm_err(tag)	*1* (true) if the last read or write operation for the specified tag indicated a communication failure. You can examine the status of only one tag with this function.

Tip: If a graphic display requires more than 20 HMI tag alarm functions, it is recommended that you use the alarm functions in derived tags. For details, see the FactoryTalk View Site Edition Help.

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 195.

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.

Example: HMI tag alarm functions

ALM_IN_ALARM(/Ingredients::vessel3\TIC3\pv*)

Returns *1* (true) if one or more tags in the specified folder have a name beginning with the letters *pv* and are in alarm. Returns *0* (false) if none of the specified tags are in alarm.

In this example, the tags are in the area called Ingredients.

Security functions

Use the built-in security functions to control access to an application, based on a user's identity or security permissions.

This function	Returns this value	Example
CurrentComputerHasGroup	*1* (True) if the current computer is assigned to the specified group. The group name must be enclosed in double quotes (" "). **Note:** 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.	CurrentComputerHasGroup("Group A") Returns *1* if the current computer is assigned to the group *Group A*. If not, returns *0*.
CurrentUserHasCode	*1* (true) if the user possesses any of the specified security codes. If checking multiple security codes, do not type a space between the security code letters.	CurrentUserHasCode(ABP) Returns *1* if the current user has been assigned one or more of the specified codes. If not, returns *0*.
CurrentUserHasGroup	*1* (True) if the current user is assigned to the specified group. The group name must be enclosed in double quotes (" ").	CurrentUserHasGroup("Group C") Returns *1* if the current user is assigned to the group *Group C*. If not, returns *0*.
CurrentUserName	The name of the current user. The user's domain is not included.	CurrentUserName( ) Returns the name of the current user.

Language function

Use the built-in language function to return the name of the current run-time language. An expression containing this function is evaluated whenever a language switch occurs.

This function	Returns this value	Example
CurrentLanguage	The RFC1766 name of the current run-time language.	CurrentLanguage( ) Returns en-US, if the current run-time language is English (United States).

FactoryTalk alarm functions

Use the built-in FactoryTalk alarm functions to examine the status of one or more FactoryTalk alarms.

This function	Returns this value:
AE_FaultedAlarmsExist(AlarmName)	A boolean representing whether any alarm exists in **Faulted** state with the specified alarms, area path, or group path. Returns 0 if no alarm exists and 1 if one or more alarms exist.
AE_HighPriorityActive(AlarmName)	An integer from 1 to 4 representing the highest priority value associated with the specified alarms, whose states are **In Alarm**. • 1: Low • 2: Medium • 3: High • 4: Urgent If 0 is returned, it means none of the specified alarms were **In Alarm**.
AE_HighSeverityAcked(AlarmName)	An integer from 1 to 1000 representing the highest severity value associated with the specified alarms, whose states are **In Alarm and Acknowledged**. If 0 is returned, it means none of the specified alarms were both **In Alarm and Acknowledged**.
AE_HighSeverityUnacked(AlarmName)	An integer from 1 to 1000 representing the highest severity value associated with the specified alarms, whose states are **In Alarm and Unacknowledged**. If 0 is returned, it means none of the specified alarms were both **In Alarm and Unacknowledged**.
AE_InAlmAckedCount(AlarmName)	A long integer from 0 to the number of FactoryTalk alarms that are **In Alarm and Acknowledged**.
AE_InAlmUnackedCount(AlarmName)	A long integer from 0 to the number of FactoryTalk alarms that are **In Alarm and Unacknowledged**.
AE_NormalUnackedCount(AlarmName)	A long integer from 0 to the number of FactoryTalk alarms that are **Normal and Unacknowledged**.
AE_ShelvedAlarmsExist(AlarmName)	A boolean representing whether any alarm exists in **Shelved** state with the specified alarms, area path, or group path. Returns 0 if no alarm exists and 1 if one or more alarms exist.
AE_UnackedAlarmsExists(AlarmName)	A boolean representing whether any alarm exists in **In Alarm and Unacknowledged** 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.

This wildcard character	Table Heading
?	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 261.

About using FactoryTalk alarm names in expressions

To supply a FactoryTalk alarm name for an expression, type it in the **Expressions** box, or click the **Alarms** button to browse for and select one or more tags in the **Alarm Source** browser.

You must specify an alarm source when you use the built-in FactoryTalk alarm functions in an expression. An alarm source might be a single alarm, or an area containing several alarms.

When referring to an alarm source, you can use an absolute or relative reference.

Enclose alarm names that contain dashes or start with a number in brackets {} when you use them in an expression. Also use brackets when using wildcard characters (* or ?) to represent multiple alarms.

You can also use tag placeholders in alarm names, to substitute values for the placeholders at run time. For more information about tag placeholders, see Using placeholders to specify tag values on page 428.

Using if -then-else logic in expressions

Use 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.

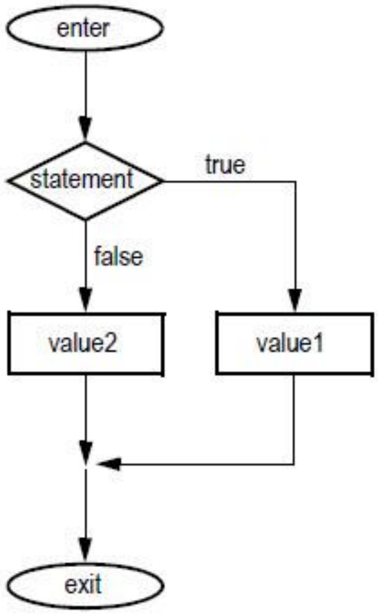
To build conditional expressions, use the relational operators and the logical operators for the statement, and values.

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 non-zero value, and false means zero.

Following is an illustration of the if – then – else structure:



Nested if - then - else

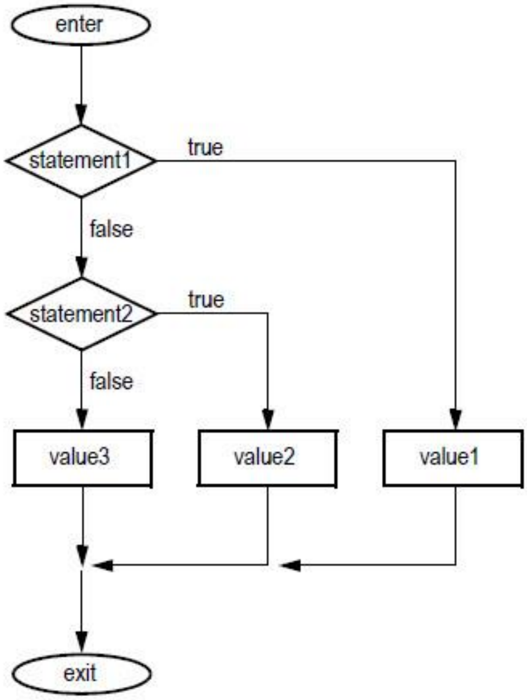
It is common to nest an if- then - else structure inside the then or else part of an if - then - else structure.

Example 1: Nested if - then- else

This expression:

```
If statement1 Then value1
Else if statement2 Then value2
Else value3
```

is interpreted in this way:

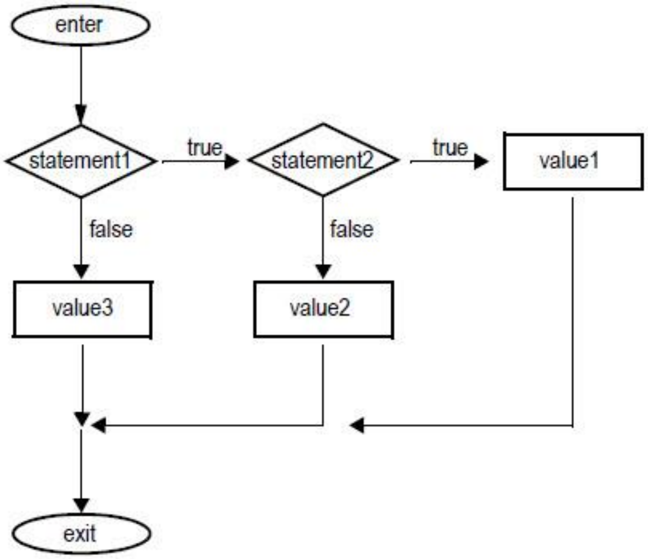


Example 2: Nested if – then – else

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 run time.
- How embedded variables are updated at run time.

About embedded variables

Use embedded variables to provide information in a graphic display that updates dynamically at run time.

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 text captions on graphic objects, tooltip text, local message text, and the title bar text of graphic displays.

For example, if you embed a tag value and a time variable in a local message, when the local message is shown at run time, 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 run time, see How embedded variables are updated at run time on page 584.

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 261.

Inserting embedded variables

You can insert one or more embedded variables in the:

- 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 449.

- Local messages you create in the Local Messages editor. For information about creating local messages, see Setting up local messages on page 487.

- Tooltips you create for graphic objects. For information about creating tooltips, see Adding tooltips to graphic objects on page 416, or see the FactoryTalk View Site Edition Help.

- Title bar text you create for graphic displays. For information about setting up display titles, see Setting up the properties of a graphic display on page 434.

  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.

To insert an embedded variable in the caption of a graphic object:

1. Open the graphic object's **Properties** dialog box.
2. Click the tab containing the **Caption** box (the General, Up Appearance, Label, or States tab, depending on the type of object).
3. Click the **Insert Variable** button beneath the **Caption** box, or right-click the **Caption** box, and then select **Insert Variable**.
4. Click the type of variable you want to insert (**Numeric**, **String**, or **Time/Date**).
5. Fill in the options in the dialog box that opens. For details, click **Help**.

To insert an embedded variable in a local message:

1. Open the **Local Messages** editor
2. Right-click the **Message** column, and then select **Edit String**.
3. Click the **Insert Variable** button beneath the **String Edit** box, or right-click the **String Edit** box, and then select **Insert Variable**.

4. Click the type of variable you want to insert (**Numeric**, **String**, or **Time/Date**).

5. Fill in the options in the dialog box that opens. For details, click **Help**.

To insert an embedded variable in the tooltip of a graphic object:

1. Open the graphic object’s **Properties** dialog box.

2. In the **Common** tab, click the **Insert Variable** button beneath the **ToolTip** text box, or right-click the **ToolTip** text box, and then select **Insert Variable**.

3. Click the type of variable you want to insert (**Numeric**, **String**, or **Time/Date**).

4. Fill in the options in the dialog box that opens. For details, click **Help**.

Tip: Tooltips can consist of a single line of text, or multiple lines.

To insert an embedded variable in the title bar of a display:

1. Open the **Display Settings** dialog box.

2. In the **Properties** tab, click the **Insert Variable** button beneath the **Title Bar** box, or right-click the **Title Bar** box, and then select **Insert Variable**.

3. Click the type of variable you want to insert (**Numeric**, **String**, or **Time/Date**).

4. Fill in the options in the dialog box that opens. For details, click **Help**.

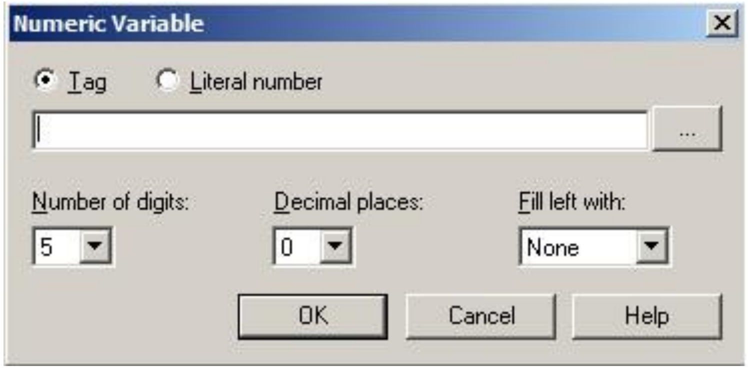
To modify an embedded variable:

1. Right-click the embedded variable string in the **Caption**, **String Edit**, **ToolTip** text, or **Title Bar** box, and then select **Edit Variable**.

2. Make the necessary changes in the **Numeric**, **String** or **Time and Date Variable** dialog box, and then click **OK**.

Creating numeric embedded variables

Insert numeric embedded variables to show analog or digital tag values or literal numbers in local message text, in a graphic object's caption or tooltip, or in a graphic display's title bar.



● Tag ○ Literal number

Number of digits: 5

Decimal places: 0

Fill left with: None

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 run time.

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 run time 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 run time, see How embedded variables are updated at run time on page 584.

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 run time, the numeric embedded variable

```
/*N:3 Oven_temp NOFILL DP:0 */
```

would show the current value of an analog tag called Oven_temp, with 3 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 run time, the numeric embedded variable

```
/*LN:3 9999 NOFILL DP:0 */
```

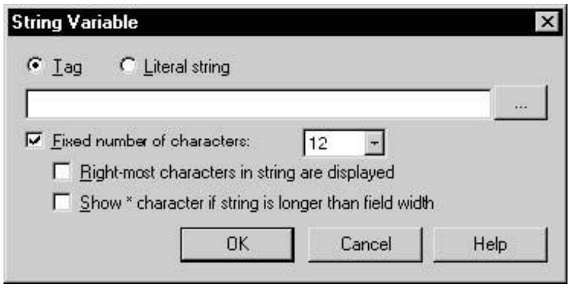
would show the number

```
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 local message text, in a graphic object's caption or tooltip, or in a graphic display's title bar.



String Variable

● Tag ○ Literal string

☑ Fixed number of characters: 12

□ Right-most characters in string are displayed

□ Show * character if string is longer than field width

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 run time.

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 you type (or specify in a parameter file) cannot contain spaces.

Formatting options for 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 run time 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 run time, see How embedded variables are updated at run time on page 584.

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 run time, 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 run time, the string embedded variable

```
/*LS:0 September28*/
```

would show the text string

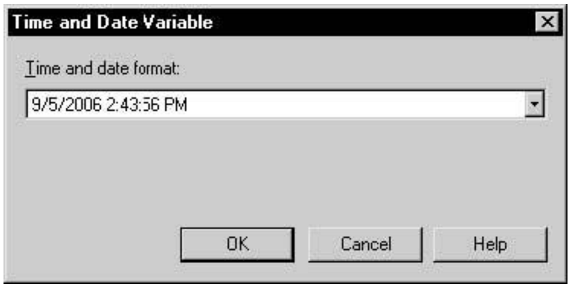
```
September28
```

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 577.

Creating time and date embedded variables

Insert time and date embedded variables to show the current time and date in local message text, in a graphic object’s caption or tooltip, or in a graphic display’s title bar.



Time and date format: 9/5/2006 2:43:56 PM

In the **Time and Date Variable** dialog box, select the format, in which you want the current time and date shown at run time.

For details about options in the **Time and Date Variable** dialog box, click **Help**.

For information about how embedded variables are updated at run time, see How embedded variables are updated at run time on page 584.

Syntax for time and date embedded variables

Time and date embedded variables use this syntax:

```
/*Time_date_ format */
```

where

Time_date_ format uses one of these character sequences:

These characters	Specify this format
SD	Short date
LD	Long date
SDT	Short date and time
LDT	Long date and time
T	Time
TSD	Time and short date
TLD	Time and long date

Example: Time and date embedded variable

At run time, the time and date embedded variable

```
/*SDT*/
```

would show the short date followed the time, for example:

```
9/28/2006 1:12:25 PM
```

A space is inserted between the time and date at run time.

How embedded variables are shown at run time

If there is no valid data available for an embedded variable, the caption, local message, tooltip, or title bar that contains 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 or 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 you set up on the run-time computer. For example, if the run-time 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 check box, **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 run time.

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 run time.

Time and date formats

Time and date embedded variables use the time and date formats you set up on the run-time computer. For example, if the time and date variable shows the short date, at run time the variable will use the short date format set up for the run-time computer.

How embedded variables are updated at run time

How an embedded variable is updated at run time 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 443.

- 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.

- 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.
- Data log storage formats.
- Creating a data log model.
- Working with ODBC data sources.
- Creating new files at run time.
- Switching log paths at run time.
- Logging on demand.
- Modifying existing data log models.
- Starting and stopping data logging.

About data logging

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, and can be:

- Shown in trends. To show historical data in a trend, assign a tag from a data log model to a pen. For more information, see Setting up trends on page 603.
- Archived for future use.
- Analyzed using any ODBC-compatible reporting software, such as Microsoft Excel, or Business Objects Crystal Reports.

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 run time, 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

Logged data is stored either in an internal file set or in an ODBC-compliant database.

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 cannot view the contents of the file sets, except in trends.

If you log tag values to an ODBC database, you can view the data in trends, or 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 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 the sequence files were created in. You can haveThe 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 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 Oracle, Microsoft SQL Server, or Microsoft Access.

Tip: When creating the tables for Float and String data logging to MS SQL Server 2008, ensure that Microsoft SQL Server 2008 Service Pack 1 has been installed first, or table creation will not be possible.

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- 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 586.

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*	The index number for the tag. The tag name that corresponds to this number is listed in the tag name table.	SQL_SMALLINT, *or* SQL_INTEGER	2 *or* 4
TagName	If the tag name table is not used, the tag name appears in this column	SQL_VARCHAR, *or* SQL_CHAR	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. *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.	SQL_VARCHAR, *or* SQL_CHAR	1
Marker	Internal tag marker.	SQL_VARCHAR, *or* SQL_CHAR	

Format for 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

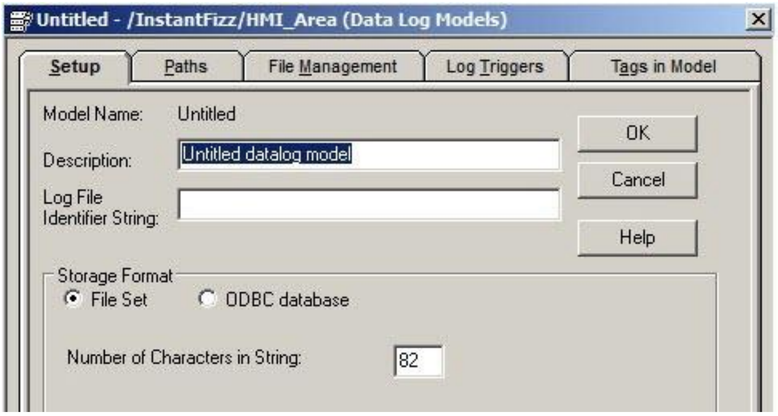
In the **Data Log Models** editor, you can create data log models that specify:

- 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**.

   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.



Untitled - /InstantFizz/HMI_Area (Data Log Models)

Model Name: Untitled

Description: Untitled datalog model

Log File Identifier String:

Storage Format: ● File Set ○ ODBC database

Number of Characters in String: 82

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 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 new 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 extend the length of the message to 1999 for a Microsoft SQL Server or Oracle database, click **Extend table.** The default length of the message is 254. You can only extend the table when the length is between 254 and 1999.

To log a longer diagnostic messages to the database, select the **Use long message text** check box. The message can be up to 1999 characters. If the check box is not selected, the message can be no more than 254 characters.

Tip: Make sure the table is successfully extended before you choose to log long messages. Otherwise the message longer than 254 and all subsequent messages cannot be logged until the length is extended.

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 592.

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 sets, 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 run time on page 594.

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.

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 new file sets

If the data log model uses file sets, you can set up data logging to create new 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 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 check boxes under **Delete Oldest Files**.

Data log file sets are deleted only when a new file set is created. If the application creates a new 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 598.
- You can also combine types of logging. For more information, see Combining logging on page 598.

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 a new 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 new 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.

   Tip: When creating the tables for Float and String data logging to MS SQL Server 2008, ensure that Microsoft SQL Server 2008 Service Pack 1 has been installed first, or table creation will not be possible.

To create new 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 new 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 a new ODBC data source

To create a new 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 a new 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 592.

Setting up security to log data to a remote computer

If you are logging data to an ODBC database that is not located 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 new files at run time

To create new file sets or ODBC backup files at run time, 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 new 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 new 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 new file set for the specified data log model.
- **DataLogNewFile *** creates a new 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 new files at run time, 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 run time

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 new 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 new 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 597.

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 run time, 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, whether or not the model is running.
- DataLogMergeToPrimary * moves data for all data log models whether or not they are running.

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 run time, 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 (an example of how to do this follows).

To provide operators with a way to log on demand at run time, 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 run time, using either FactoryTalk View Studio or the FactoryTalk View SE Administration Console.

If you change a model at run time, 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 600.

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 run-time changes without modifying the model

At run time, 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 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** check box, 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** check box, 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 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 160.

Setting up trends

This chapter describes:

- What trends are.
- How to create trends.
- The parts of a trend.
- Trend chart styles.
- Choosing colors, fonts, lines, and legends for a trend.
- Using shading to compare pens.
- Using overlays to compare real-time and historical data.
- Using trend templates.
- Working with trends at run time.

About trends

A trend is a visual representation, or chart, of current or historical tag values. A trend provides an operator with a way to track plant activity as it is happening.

In a trend, you can:

- Plot data for as many as 100 tags or expressions.
- Plot data over time, in a standard trend chart.
- Plot one tag against another, in an XY Plot chart.
- Use shading to emphasize a comparison between two pens.
- Display isolated or non-isolated graphs.

  In an isolated graph, each pen is placed in a separate band of the chart. In a non-isolated graph, pen values can overlap.

Charting current versus historical data

A trend can show:

- Real-time data from a data server.
- Historical data from a data log model’s set of files.

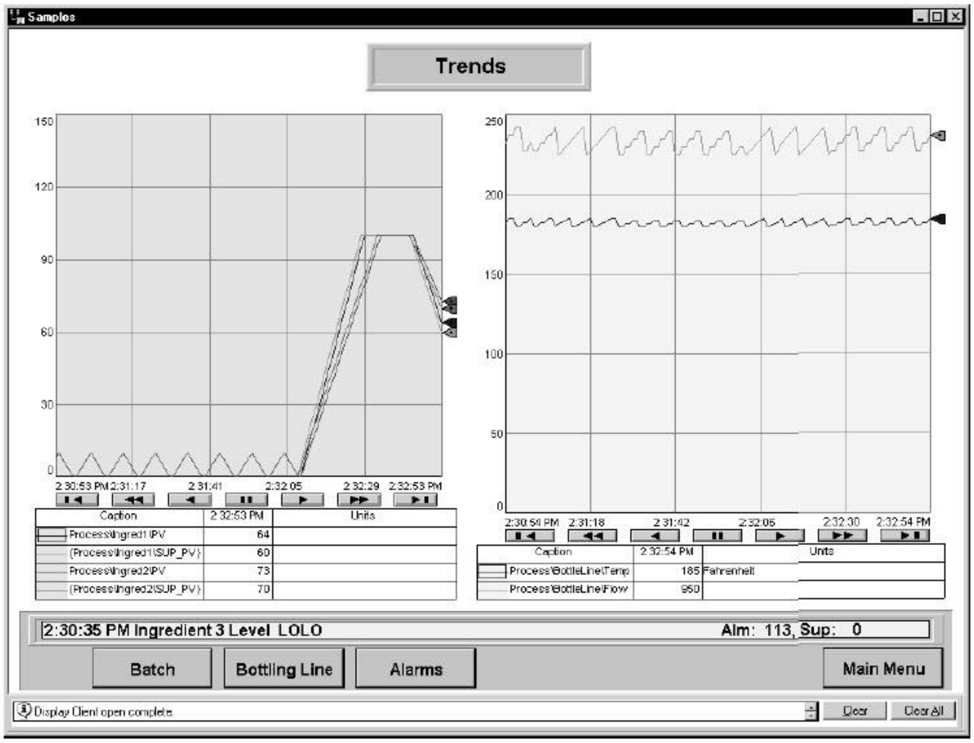
- Historical data from a FactoryTalk Historian Site Edition server and a FactoryTalk Historian Machine Edition module (only for network distributed or station applications).

  In a network distributed application, you can plot historical data from data log models in local or remote areas of the application. For more information about data log models, see Setting up data logging on page 585.

Creating trend objects

You can create a trend using the Trend drawing tool, or you can use the ready-made trend chart in the Trend graphic library. For more information, see About the Trend graphic library on page 623.

The following illustration is of a FactoryTalk View graphic display that contains two different trend displays.



Caption	2:32:53 PM	Units
Process\Ingred1\PV	64	
{Process\Ingred1\SUP_PV}	60	
Process\Ingred2\PV	73	
{Process\Ingred2\SUP_PV}	70	

Caption	2:32:54 PM	Units
Process\BottleLine\Temp	185	Fahrenheit
Process\BottleLine\Flow	950	

2:30:35 PM Ingredient 3 Level LOLO Alm: 113, Sup: 0

Display Client open complete

To create a trend object:

1. In the Graphics editor, create a new graphic display, or open an existing one.
2. Select **Objects** > **Advanced** > **Trend**. You can also click on the Objects toolbar.
3. On the graphic display, drag the mouse to create a box approximately the size you want the trend to be.

Providing a name for the trend

When you create a trend object, it is given a name by default. To change the name, right-click the trend object, and then select **Property Panel**.

In the **Properties** tab, in the (Name) row, type the trend's new name.

The name can contain letters, numbers, and the underscore character (_); however, the first character must be a letter, and the name cannot contain spaces.

Setting up trend properties

The **Trend Properties** dialog box contains several tabs, in which you can set up the trend's data, appearance, and behavior:

- In the **General** tab, set up the trend's data server (real-time or historical polling), chart title, chart style, and update mode.
- In the **Display** tab, set up the appearance of the trend display.
- In the **Pens** tab, set up pen tags and expressions, and the appearance of pens in the trend.
- In the **X-Axis** tab, set up the trend's horizontal axis.
- In the **Y-Axis** tab, set up the trend's vertical axis.
- In the **Overlays** tab, add or remove snapshots of historical data.
- In the **Template** tab, set up properties for trend templates.
- In the **Runtime** tab, set up ways to interact with the trend at run time.
- In the **Common** tab, set up the trend's size and position in the graphic display.

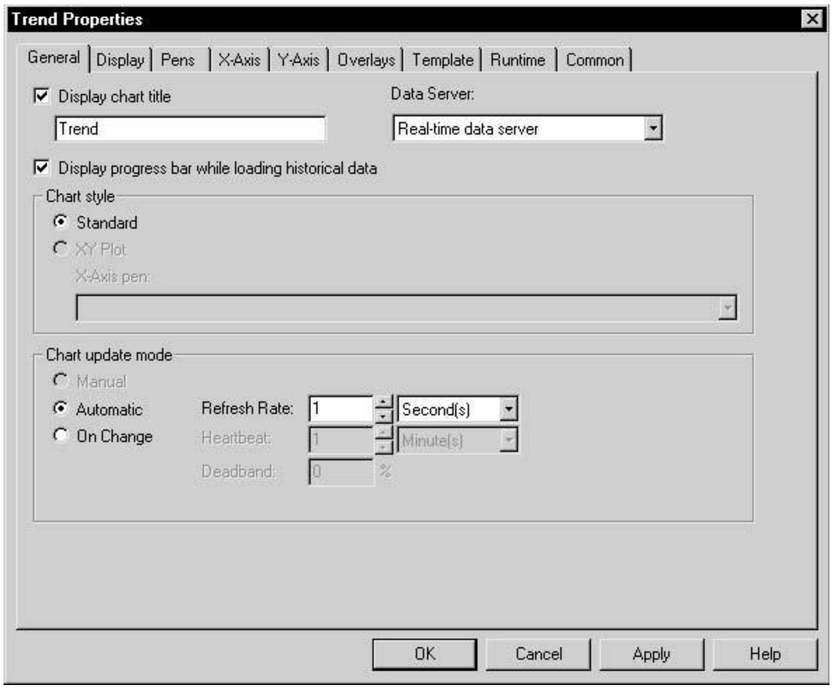
Once you have set up the trend, you can modify it as you would any other graphic object. You can move it, resize it, attach animation to it, and so on.

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 393, and Creating graphic objects on page 449.

To open the Trend Properties dialog box:

Double-click the trend object. You can also right-click the object, and then select **Properties**.



Trend Properties

☑ Display chart title

Trend

Data Server: Real-time data server

☑ Display progress bar while loading historical data

Chart style

● Standard

○ XY Plot

X-Axis pen:

Chart update mode

○ Manual

● Automatic

○ On Change

Refresh Rate: 1 Second(s)

Heartbeat: 1 Minute(s)

Deadband: 0 %

For details about options in the **Trend Properties** dialog box, click **Help**.

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 pens will plot values.

When you are finished testing, switch back to **Edit Display** mode to continue setting up trend properties.

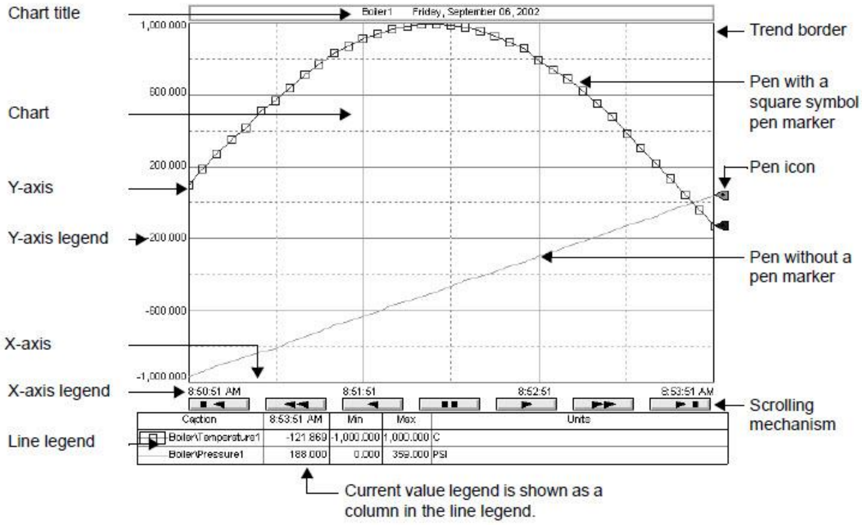
To switch between Test and Edit Display modes:

In the **Graphics** editor, from the **View** menu, select **Test Display** or **Edit Display**. You can also click the Test Display or Edit Display buttons on the toolbar.

The parts of a trend

There are two trend chart styles—the standard chart, and the XY Plot chart.

The following illustration shows the parts of a standard trend chart with two pens, and a three-minute time span.



Caption	8:53:51 AM	Min	Max	Units
Boiler\Temperature1	-121.869	-1,000.000	1,000.000	C
Boiler\Pressure1	188.000	0.000	359.000	PSI

Chart

The chart contains the plotted trend data. The chart is bounded by the y-axis on the left, and the x-axis at the bottom.

Plotted data is shown using pen lines, pen markers, and grid lines.

Chart title

The chart title is the name you give to the trend chart. Naming the trend is optional.

X-axis

The x-axis is the bottom edge of the chart. It is also known as the horizontal axis, or the time axis.

X-axis legend

For standard charts, the x-axis legend indicates the time span covered by the trend.

For XY Plot charts, the x-axis legend shows the scale, or range, of values associated with the selected pen.

The number of timestamps or values shown depends on the size of the trend object and the number of vertical grid lines.

At run time, you can double-click the x-axis legend, to open the **X-Axis** tab of the **Trend Properties** dialog box. You can set up a trend to exclude the x-axis legend.

Y-axis

The y-axis is the left edge of the chart. It is also known as the vertical axis.

Y-axis legend

The y-axis legend shows the scale, or range, of values for the pens. The minimum and maximum values for the scale can be:

- Determined automatically (using the best fit for the current data).
- Derived from a pen's minimum and maximum values.
- Constant values.
- Controlled by tags.

All pens in the trend can use the same scale, or each pen can have its own scale on the y-axis. When the operator clicks a pen in the line legend, the vertical axis shows the selected pen's scale.

For example, if Pen 1 has a minimum value of 10 and a maximum value of 100, the scale on the vertical axis is 10 to 100 when the pen is selected.

If Pen 2 has a minimum of -10 and a maximum of 50, the scale on the vertical axis changes to -10 to 50 when the operator selects the next pen in the line legend.

At run time, you can double-click the y-axis legend, to open the **Y-Axis** tab of the **Trend Properties** dialog box. The legend can also be used to pan data. For more information, see Panning the trend chart on page 629.

You can set up a trend to exclude the y-axis legend.

Pens

Pens are the lines or symbols used to represent values on a trend chart. The values can be tags you are monitoring, expressions that manipulate tag values, or constant values.

To make it easier for an operator to distinguish one pen from another, use different colors, widths, and symbols (also called markers) for each pen.

For an example of how pen markers are used, see Pen markers on page 610.

Legends

There are four kinds of trend legends:

- X-axis legend (described on X-axis legend on page 607)
- Y-axis legend (described on Y-axis legend on page 608)
- Line legend
- Current value legend

The line legend shows the color, name, description, minimum value and maximum values for each pen line shown on the trend chart.

The current value legend shows the pen icon, current pen value, and current time for each pen line shown on the trend chart.

The position of the line legend determines where the current value legend is shown on the chart. When the position of the line legend is set to:

- Left, the line legend is on the left side of the chart, and the current value legend is on the right side.
- Bottom, the current value legend is a column in the line legend.

For more information about the current value legend and the line legend, see Showing a current value legend on page 616 and Showing a line legend on page 617.

Pen icons

Pen icons, if specified, are shown at the right edge of the trend chart.

At run time, if a pen’s current value is within the scale shown on the vertical axis, the pen’s icon is shown.

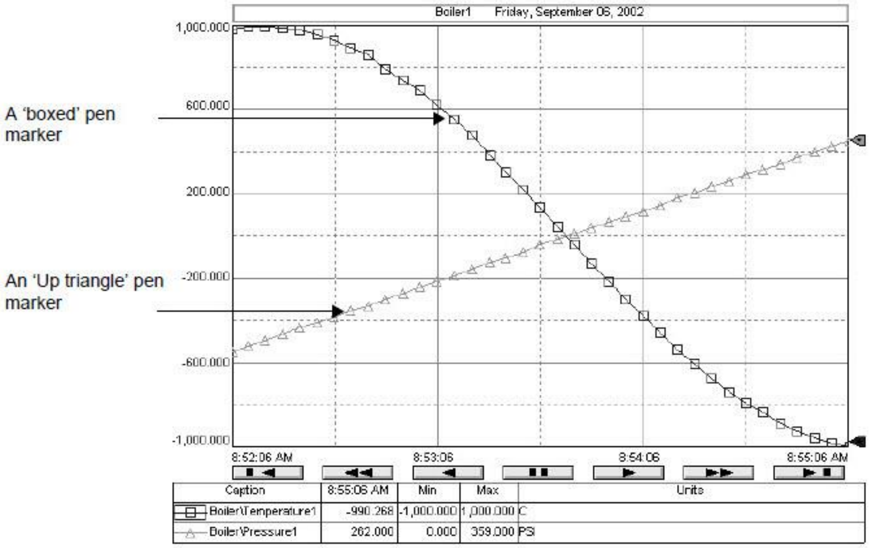
The position of the icon indicates the pen’s most recently recorded value, even if the trend is paused, or if the most recent value has not yet been plotted.

Pen markers

As shown in the next illustration, pen markers are symbols that indicate data points. A data point is the exact position of a value plotted on a trend chart.

You can use pen markers to show when data is sampled. For example, when the trend is updated as the value changes, the pen marker indicates when the trend is updated.

You can also use pen markers to read pen values at run time.


A 'boxed' pen marker

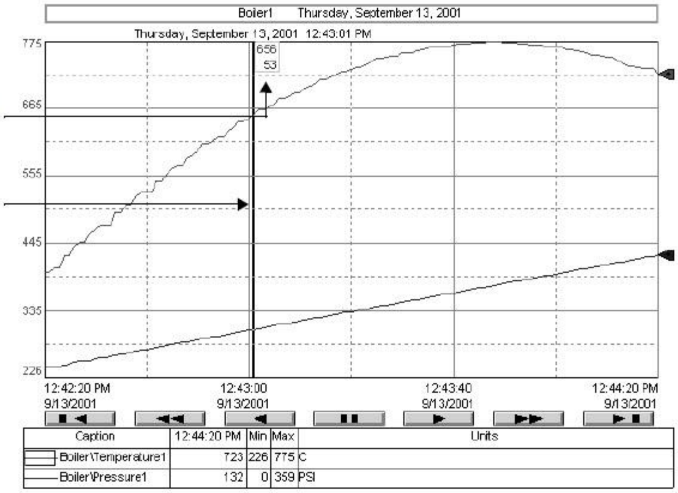
An 'Up triangle' pen marker

Caption	8:55:06 AM	Min	Max	Units
Boiler\Temperature1	-990.268	-1,000.000	1,000.000	C
Boiler\Pressure1	262.000	0.000	359.000	PSI

Tip: If data is plotted frequently, the markers might not show as distinct, separate symbols on the trend chart.

Value Bar

The value bar marks the value of each pen in a trend, at the selected position on the trend chart, as shown in the following illustration.



The value of each pen plotted on the trend chart at this position

The value bar

Caption	12:44:20 PM	Min	Max	Units
Boiler\Temperature1	723	226	775	C
Boiler\Pressure1	132	0	359	PSI

You can position the value bar by clicking anywhere on the trend chart at run time.

If you position the value bar on a pen marker, it shows the value of the data point represented by the pen marker.

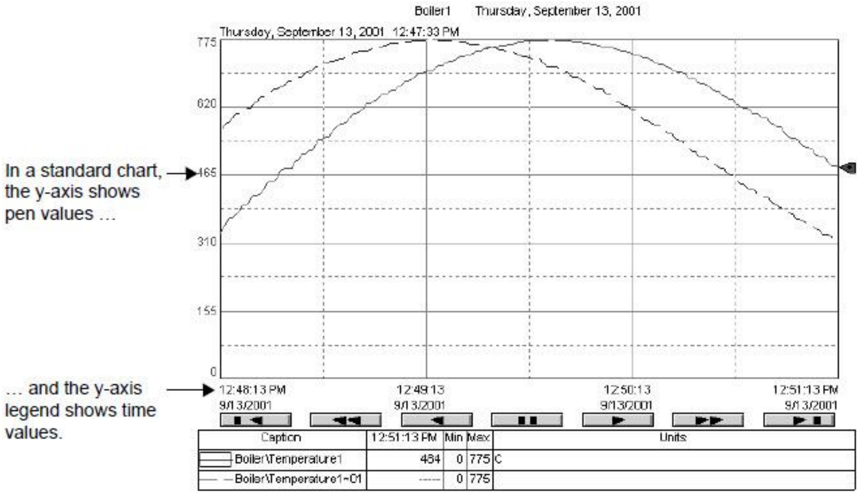
Trend chart styles

There are two trend chart styles—the standard chart, and the XY Plot chart.

You can set up the chart styles for a trend object in the **General** tab of the **Trend Properties** dialog box. For details about options in the General tab, click **Help**.

The Standard chart style

A standard chart plots tag values against time, as shown in the following illustration.



In a standard chart, the y-axis shows pen values …

… and the y-axis legend shows time values.

Caption	12:51:13 PM	Min	Max	Units
Boiler\Temperature1	484	0	775	C
Boiler\Temperature1~01	-----	0	775	

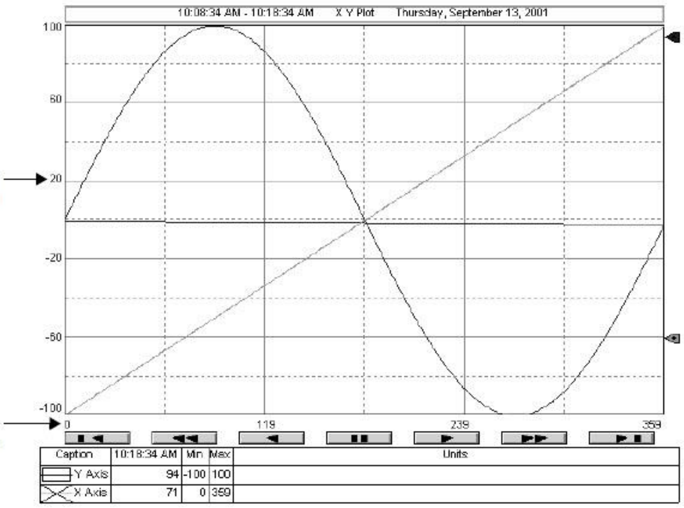
The XY Plot chart style

An XY Plot chart plots the values of one or more tags against another tag.

Use the XY Plot chart to plot one (or more) tag’s values against another tag’s values instead of plotting one tag’s values against time.

For example, you could plot the temperature of a tank against the pressure of the tank.

In the following illustration, the y-axis pen's data is plotted as a diagonal line on the trend chart. The x-axis legend shows the scale for the x-axis pen. The time period covered by the chart is at the upper left.


10:08:34 AM - 10:18:34 AM X Y Plot Thursday, September 13, 2001
100
60
20
-20
-60
-100
0
119
239
359

In an XY Plot, the y-axis shows pen values …

… and the x-axis legend shows pen values.

Caption	10:18:34 AM	Min	Max	Units
Y Axis	94	-100	100	
X Axis	71	0	359	

Isolated graphing

For charts with multiple pens, the pen values can overlap, or you can isolate each pen in its own horizontal band on the chart.

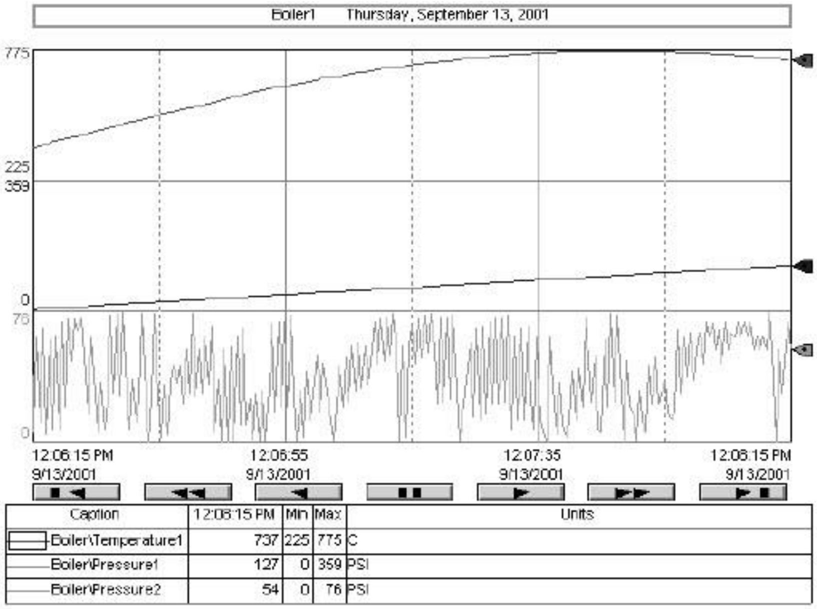
Use isolated graphing to view each pen individually on the trend chart.

The band is the area of the trend chart used by an isolated pen. Use the isolated graphing percentage to specify the space between each pen band.

With isolated graphing, a grid line is placed automatically above each pen's band.

The following illustration shows isolated graphing, with 0% isolation between each pen's band.

Tip: In the illustration, each pen has its own scale. Another option is to use the same scale for all pens in the trend.



Caption	12:08:15 PM	Min	Max	Units
Boiler\Temperature1	737	225	775	C
Boiler\Pressure1	127	0	359	PSI
Boiler\Pressure2	54	0	76	PSI

Plotting a value across the full width of the chart

Use horizontal lines on a trend chart to provide an orientation for your tag data.

For example, you can define upper and lower limits for a tag’s value, and then show the limits as horizontal lines in the trend. When the value crosses one of the limits, the tag’s condition is obvious on the trend.

The vertical position the pen in the trend depends on value of the tag, expression, or constant used to define the pen. If the value changes, the position of the horizontal line also changes.

To plot a constant value across the chart’s full width:

1. In the **Trend Properties** dialog box, click the **Pens** tab.
2. Add a pen with a constant value.

When values for the pen are plotted across the full width of the chart, the pen shows as a solid horizontal line.

To plot a tag or expression across the chart’s full width:

1. In the **Trend Properties** dialog box, click the **Pens** tab.

2. Add a pen with a tag or expression that will determine the position of the pen line.

3. In the **Pen Attributes** spreadsheet, in the **Type** column, select **Full Width**.

When the trend runs, the pen shows as a solid horizontal line across the full width of the chart.

Choosing trend colors, fonts, lines, and legends

In the Trend Properties dialog box, you can set up colors, fonts, lines, and markers for a trend chart and pens.

To set up this attribute	In this tab	Click this box, button, or list
Chart background color	Display	Background color
Text color for the x-axis legend	Display	Text color
Text font, style, and size	Display	Font
Pen line, pen marker, pen icon, and text color for the y-axis legend	Pens	Color
Pen line width	Pens	Width
Pen line style	Pens	Style
Pen marker	Pens	Marker
Color of vertical lines in the grid	X-Axis	Grid color
Color of horizontal lines in the grid	Y-Axis	Grid color

You can also set up these attributes in the **Property Panel**. To open the **Property Panel**, right-click the trend object, and then select **Property Panel**.

Changing the trend highlight color

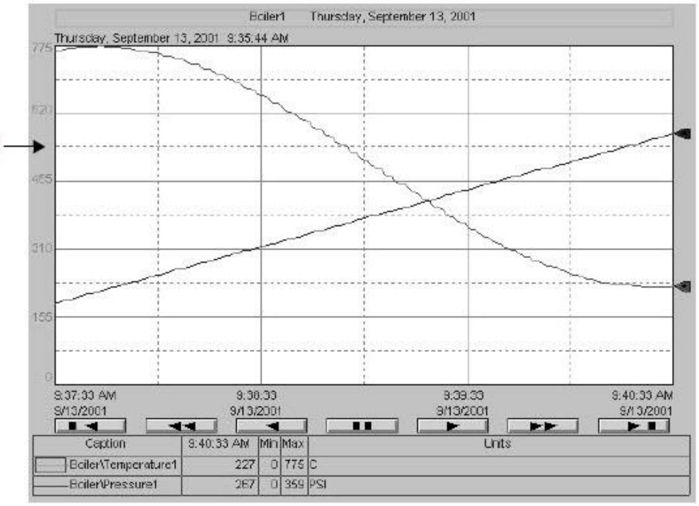
The chart title, y-axis legend, x-axis legend, and line legend can have a highlight box at run time.

Whether these parts of the trend are highlighted, and in what color, is determined by settings for the graphic display that contains the trend. For more information about display settings, see Setting up the appearance and behavior of a graphic display on page 434.

Changing the trend object background

The space behind and around the trend chart and legends is the background of the trend object (shown in the following illustration).

The shaded area is the background of the trend object.



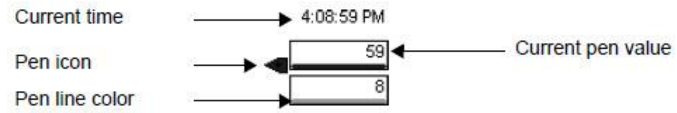
Caption	9:40:33 AM	Min	Max	Units
Boiler\Temperature1	227	0	775	C
Boiler\Pressure1	267	0	359	PSI

You can change the color and style settings for the background using the **WindowStyle** and **WindowColor** options in the **Property Panel**. For details about using the Property Panel, see Viewing object properties in the Property Panel on page 398.

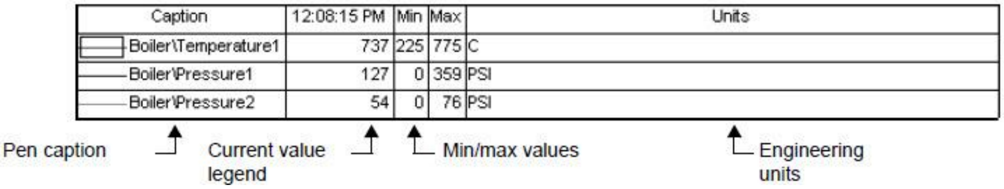
Showing a current value legend

The current value legend can be shown by itself or as part of the line legend, depending on the position of the line legend.

When the line legend is shown to the left of the trend chart, the current value legend is shown to the right of the trend, as shown in the following illustration.



When the line legend is shown at the bottom of the trend chart, the current value legend is shown as a column in the line legend, as shown in the following illustration.



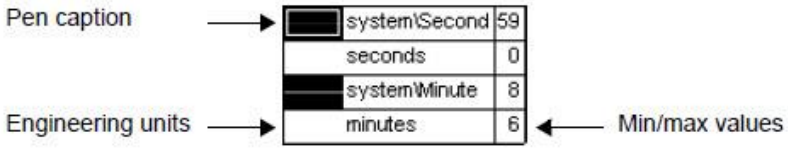
Caption	12:08:15 PM	Min	Max	Units
Boiler\Temperature1	737	225	775	C
Boiler\Pressure1	127	0	359	PSI
Boiler\Pressure2	54	0	76	PSI

Showing a line legend

The line legend shows the details of a pen line, including the line color, pen name, the minimum and maximum values, and the engineering units.

The line legend’s appearance differs depending on where it is located relative to the trend chart. The previous illustration shows the line legend as it appears at the bottom of the trend.

The following illustration shows the line legend as it appears to the left of the trend:



■	system\Second	59
	seconds	0
■	system\Minute	8
	minutes	6

Using shading to compare pens

To compare the values of two or more pens in a trend, use shading in the trend.

For example, you can set up Pen 1 to show a tag’s value, and Pen 2 to show a constant value that is an alarm threshold.

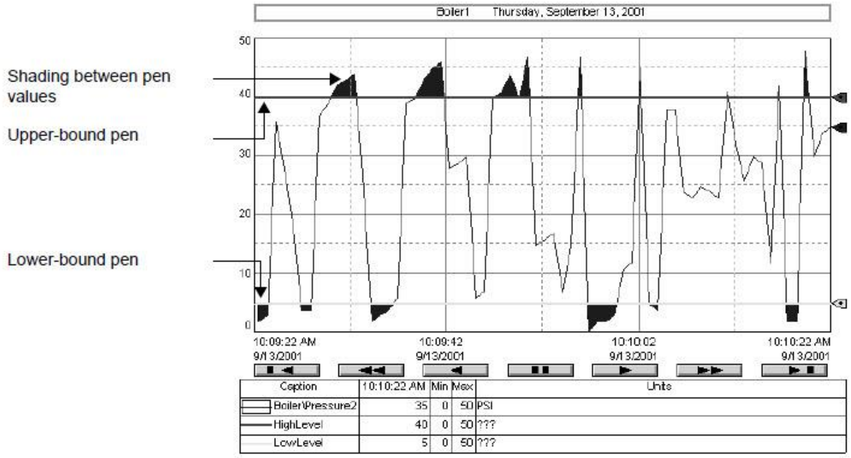
If Pen 2 is the upper boundary for Pen 1, the shading will indicate when the tag’s value exceeds the threshold, by filling in the area between the two pens with the line color for Pen 1.

At least two pens must be set up for a trend, before you can use shading.

Set up shading in the Pens tab in the Trend Properties dialog box. For details about options in the Pens tab, click **Help**.

Tip: Shading is determined by the position of lines on the trend chart—*not* by the actual tag values.

The trend in the following illustration uses shading to indicate tag values in alarm. The difference between the tag's value and the tag's alarm level is shaded.

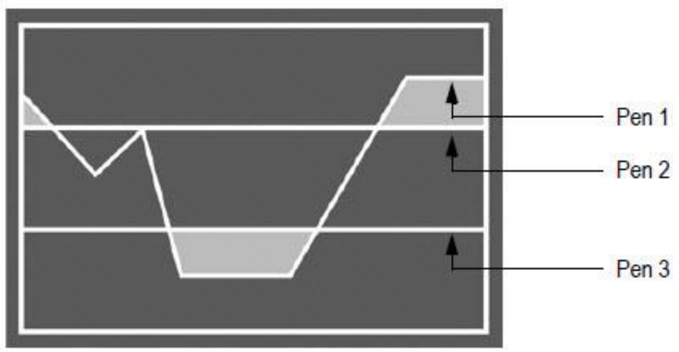


Caption	10:10:22 AM	Min	Max	Units
Boiler\Pressure2	35	0	50	PSI
HighLevel	40	0	50	???
LowLevel	5	0	50	???

Example 1: Shading with three pens

The upper pen, Pen 2, has a constant value of 75 and the lower pen, Pen 3, has a constant value of 25.

Pen 1 represents the tag called VIN_LEVEL. The trend is shaded whenever Pen 1 goes above Pen 2 or below Pen 3. A plot for this trend looks like this:

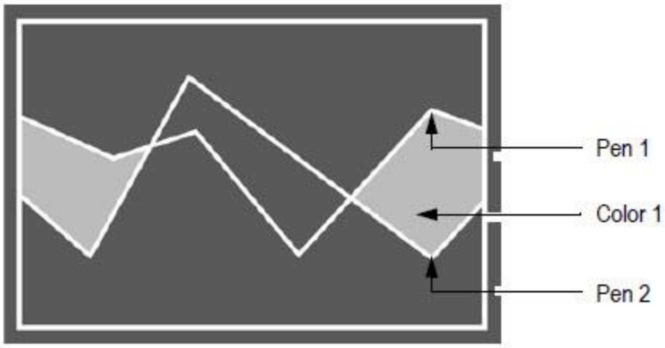


To produce the shading shown in the illustration, use the following settings:

Pen	Tag name or constant value	Shading: Upper Bound	Shading: Lower Bound
Pen 1	VIN_LEVEL	Pen 2	Pen 3
Pen 2	75		
Pen 3	25		

Example 2: Shading with two pens

Whenever Pen 1 goes above Pen 2, the trend is shaded like this:

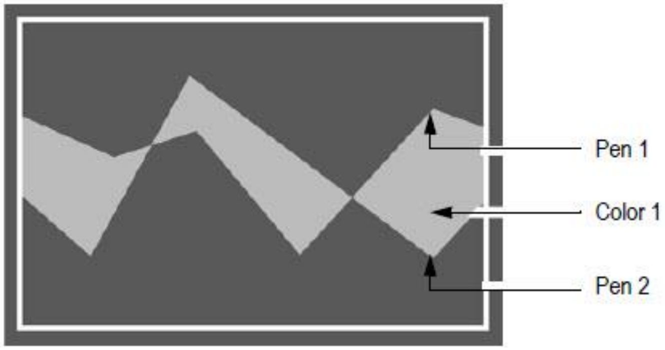


To produce the shading shown in the illustration, use the following settings:

Pen	Tag name or constant value	Shading: Upper Bound	Shading: Lower Bound
Pen 1	VIN_LEVEL	Pen 2	
Pen 2	OIL_LEVEL		

Example 3: Shading between two pens

Whenever Pen 1 goes above or below Pen 2, the trend is shaded like this:



To produce the shading shown in the illustration, use the following settings:

Pen	Tag name or constant value	Shading: Upper Bound	Shading: Lower Bound
Pen 1	VIN_LEVEL	Pen 2	Pen 2
Pen 2	OIL_LEVEL		

Using overlays to compare real-time and historical data

You can layer a trend with a snapshot of previously charted data. This is called an overlay.

Use overlays to compare current charted data to a historical snapshot. For example, you might set up overlays to compare data for a particular shift or batch process.

Setting up snapshots and overlays

To create an overlay, create the data snapshot first, and then add the snapshot as an overlay, to the trend charting current data.

When you add a snapshot to a trend as an overlay, the pens from the snapshot are added automatically to the Pens tab in the **Trend Properties** dialog box. You can change the attributes of pens used in overlays, just as you can change the attributes of regular pens.

An operator can show or hide overlays on the trend chart at run time, using the trend's shortcut menu. For more information, see Using overlays at run time on page 630.

For information about setting up the trend's run-time options, click **Help** in the **Runtime** tab of the **Trend Properties** dialog box.

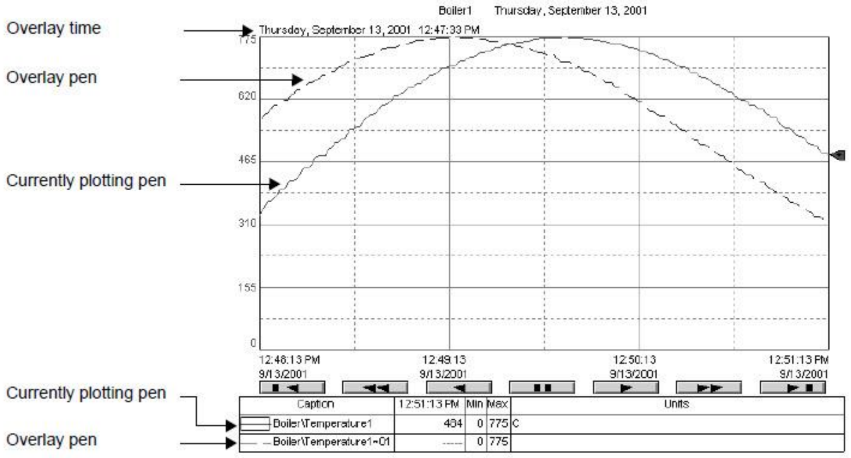
To create a snapshot:

1. Right-click the trend you want to get a data snapshot from, and then select **Create Snapshot.**
2. In the **Component Name** dialog box, type a name for the snapshot, and then click **OK**.

To add a snapshot to the trend as an overlay:

1. In FactoryTalk View Studio, open the graphic display containing the trend that will chart current data.
2. Double-click the trend. The **Trend Properties** dialog box opens.
3. In the **Overlays** tab, click **Add**.
4. In the **Component** browser, find and select the snapshot you want to add, and then click **OK**.

The following illustration is of a trend overlay.



Caption	12:51:13 PM	Min	Max	Units
Boiler\Temperature1	484	0	775	C
Boiler\Temperature1~01	-----	0	775	

Using trend templates

You can use a trend template to:

- Apply a consistent appearance to trend charts.
- Create a set of different views for the same data.
- Return to a standard display after experimenting with display options.
- Save pen attribute data.

Use the **Trend Properties** dialog box to create a trend template. For details about options in the dialog box, click **Help**.

Applying a consistent appearance to trend charts

Create a trend template to standardize a trend chart for all your applications.

Use a standard template when you have several different applications, and you need the trend chart to work the same way in each application.

When you have created and saved the template, load the template into each application's trend chart.

Creating a set of different views for the same data

To assess the same data in different ways, create trend templates that provide different views of the data.

Load the trend templates at run time, to change the appearance of the trend

Returning to a standard display

Create a trend template to use as a standard trend display.

Load the standard display template after an operator changes the trend chart settings, and you want to return to the standard trend chart.

Saving pen attribute data

Create a trend template to save pen attribute data.

You can add a number of new pens at run time, create a template, and then save pen attribute data with the template. The next time you run the trend, if you want the same pen attribute data, you can load the template.

Loading trend templates

Load a template to return to the original trend settings, or to apply a consistent trend appearance across applications.

To load a template:

1. In the **Trend Properties** dialog box, click the **Template** tab, and then click **Load template**.
2. In the **Component** browser, find and select the template you want to load, and then click **OK**.

   Note: Saving, loading, or deleting a template takes effect immediately, and cannot be undone. If you load a template at run time, any existing pens in the trend are deleted and replaced with the pens in the template.

Example: Using the Invoke command to load trend templates at run time

A trend template named *Isolated* uses isolated graphing, and a template named *Non-isolated* uses non-isolated graphing.

To provide an operator with a way to switch between the two templates at run time, do the following in the trend's graphic display:

1. For the isolated graphing template, create a button that runs the following command as its press action:

   ```
   Invoke Me.Trend1.LoadTemplate ("Isolated")
   ```

2. For the non-isolated graphing template, create a button that runs the following command as its press action:

```
Invoke Me.Trend1.LoadTemplate ("Non-Isolated")
```

To load the templates at run time, the operator can press either button.

Tip: In the **Invoke** commands in this example, **Trend1** is the default name of the trend object. Be sure to use the correct name of the trend in the **Invoke** command.

About the Trend graphic library

The Trend graphic library contains a real-time trend, and objects for controlling the trend. You can use the trend and objects as they are, or you can modify them to suit your needs. To use the objects, drag and drop them into a graphic display.

To open the Trend graphic library:

1. In FactoryTalk View Studio, in the Explorer window, open the **Graphics** folder.

2. In the **Libraries** folder, right-click **Trend**, and then select **Open**.

Working with trends at run time

How you can interact with a trend at run time depends on the options you set up in the **Runtime** tab of the **Trend Properties** dialog box.

You might be able to do some or all of the following:

- View and modify trend properties.
- Modify legend properties.
- Pan or zoom on the trend chart.
- Use the trend chart’s shortcut menu.

For details about options in the Runtime tab, click **Help**.

Collecting data in the background at run time

Real-time trends collect data only while the trend is shown. This means that a real-time trend will not contain any data, when the graphic display opens.

This can be a problem if the trend chart is not open for the entire time the application is running, or if other graphic displays are opened before the trend display.

If you want a trend to contain data when the graphic display opens, you can use a data log model, or you can run startup macro, to load and update the display in the background.

This loads data into the data buffer for the trend.

Tip: You can run a startup macro when the HMI server runs. For information about setting up startup components for an HMI server in various types of networks, see Working with network distributed applications on page 125, Working with network station applications on page 149, and Working with local station applications on page 167.

Example: Using a startup macro to load and update a trend in the background

1. Create a startup macro that includes this command:

   ```
   Display Trend1 /ZA
   ```

   where *Trend1* is the name of the graphic display containing the real-time trend, and */ZA* is the parameter for loading the trend data into the data buffer.

   Trend data will remain in the cache, and continue to update, until you run the **FlushCache** command, or until you stop running the application.

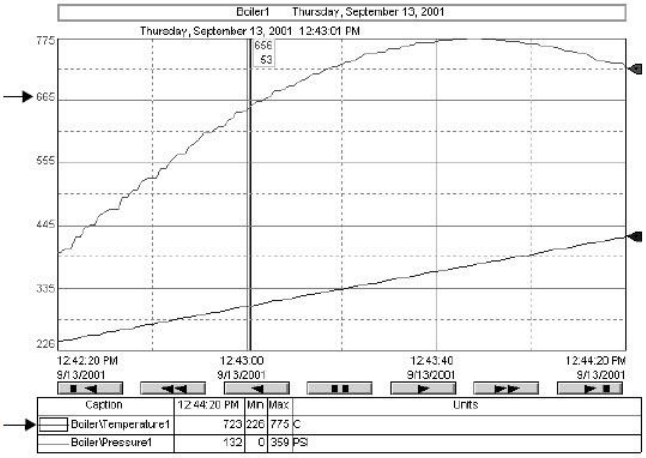
2. When you want the real-time trend to become visible, run the following command:

   ```
   Display Trend1
   ```

   When the graphic display opens, the trend will contain data.

Selecting pens at run time

To show details about a pen, select the pen by clicking it in the line legend. When a pen is selected, the vertical scale changes to reflect the scale of values for the pen.



Caption	12:44:20 PM	Min	Max	Units
Boiler\Temperature1	723	226	775	C
Boiler\Pressure1	132	0	359	PSI

The y-axis legend shows the minimum and maximum scale for a pen ...

... when you click the pen line in the line legend.

Modifying trend properties at run time

Changes made to trend properties at run time are not saved.

The changes affect the trend that is running, but as soon as the graphic display is changed or closed, the changes to the trend are lost.

To make permanent changes to a trend, open and modify it in FactoryTalk View Studio.

To open the Trend Properties dialog box at run time:

Right-click the trend chart, and then select **Chart properties**.

Tip: At run time, the **Trend Properties** dialog box shows only the tabs specified in the Runtime tab, when you set up the trend. You can make changes only in the available tabs. The Runtime tab itself, is not available at run time.

To change the trend chart title:

1. Double-click the trend chart title.
2. In the **Graph** title dialog box, type a name for the trend chart, and then click **OK**.

To open the Y-Axis tab:

1. Double-click the y-axis legend.
2. In the **Y-Axis** tab, make the desired changes, and then click **OK**.

To open the X-Axis tab:

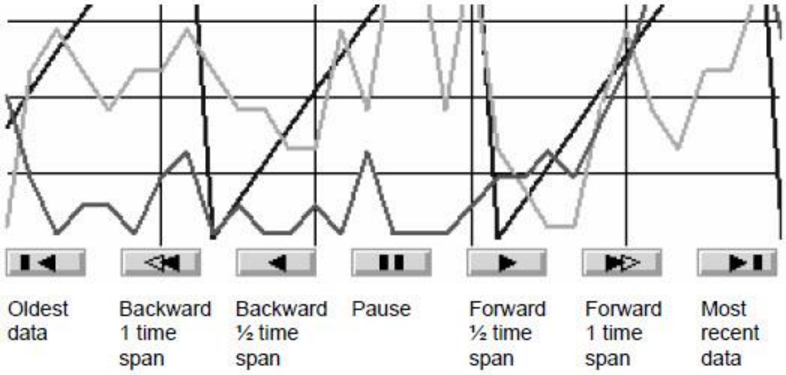
1. Double-click the x-axis legend.
2. In the **X-Axis** tab, make the desired changes, and then click **OK**.

To open the Pens tab:

1. Double-click the line legend.
2. In the **Pens** tab, make the desired changes, and then click **OK**.

Scrolling the trend chart at run time

To view different areas of charted data, an operator can use the set of VCR-style buttons included with a trend. The following illustration describes the function of each button.

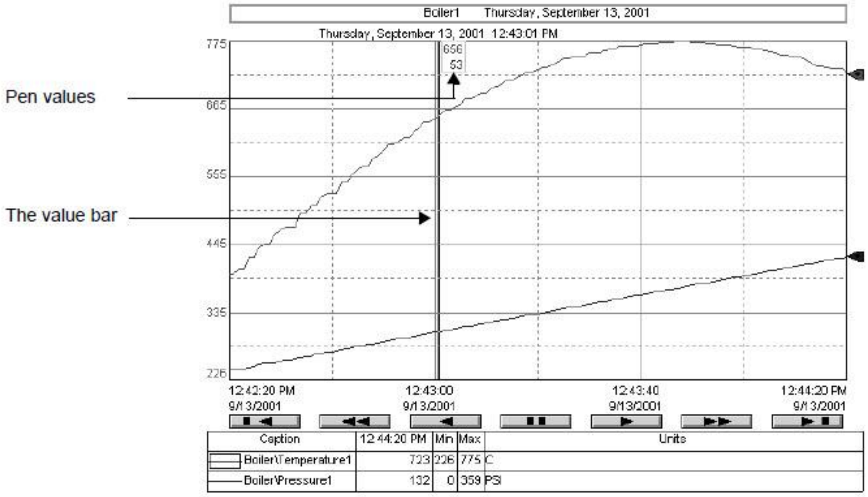


To make the VCR-style buttons available at run time, in the **Display** tab of the **Trend Properties** dialog box, select the check box, **Display scrolling mechanism**.

You can also use the buttons in the Trend graphic library. For information about the library, see About the Trend graphic library on page 623.

Showing the value of pens at various positions

The value bar shows the value of each pen in the trend, at the position you select on the trend chart.



Caption	12:44:20 PM	Min	Max	Units
Boiler\Temperature1	723	226	775	C
Boiler\Pressure1	132	0	359	PSI

The color of the value bar changes automatically, so it is visible for all background colors. For example, if the trend background color is black, the value bar color will be white. If the trend background color is white, the value bar will be black.

A tilde (~) indicates an approximate value. An approximate value is shown when there is no value for the pen at the exact time stamp of the value bar. The approximate value is based on the nearest available reading.

To show the value bar:

- Click the trend where you want the value bar to appear.

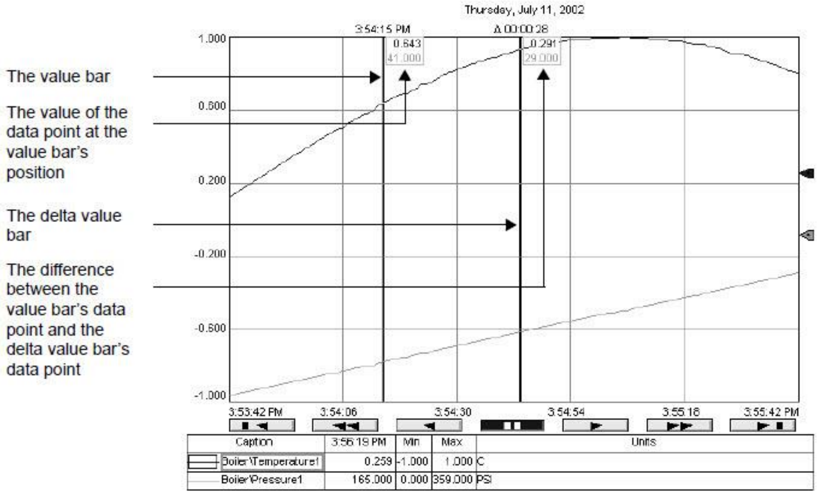
To hide the value bar:

- Click the trend’s vertical axis.

  You can also right-click the trend, and then select **Show value bar**. A check mark indicates the value bar is visible. No check mark indicates the value bar is hidden.

Displaying the difference in pen values for two points

The delta value bar works with the value bar to show the difference in value between two x-axis points for standard and XY Plot chart types.



Caption	3:56:19 PM	Min	Max	Units
Boiler\Temperature1	0.259	-1.000	1.000	C
Boiler\Pressure1	165.000	0.000	359.000	PSI

To show the delta value bar:

1. Right-click the trend chart, select **Active value bar**, and then click **Value**.
2. Click the trend where you want the value bar to appear.
3. From the shortcut menu, select **Active value bar**, and then select **Delta**.
4. Click the trend where you want to see the difference between the value at the value bar's data point and the value at the delta value bar's data point.

To move the delta value bar:

1. Right-click the trend, select **Active value bar**, and then click **Delta**.
2. Click the trend chart at the position where you want to know the difference between the value indicated by the value bar, and the value indicated by the delta value bar.

To remove the delta value bar:

Right-click the trend, and then select **Show value bar.**

This removes both the value bar and the delta value bar from the trend. However, if you select **Show value bar** again, only the value bar will reappear.

Zooming the trend chart

Use zoom to view a selected area of the trend chart in more detail.

To zoom, use one of these methods:

- Right-click and drag the y-axis to zoom vertically.
- Right-click and drag the x-axis to zoom horizontally.
- Left-click and drag the chart to zoom into the selected area.

To restore the original trend chart view:

Right-click the trend chart, and then select Undo Zoom/Pan.

Panning the trend chart

Use pan to view areas of the trend chart that are outside the area of the chart you are viewing. You must pause the trend before you can pan to an area.

To pause the trend:

Click the **VCR-style pause** button at the bottom of the trend. You can also right-click the trend chart, and then select **Scroll**.

To pan vertically:

Click and then drag the y-axis legend.

To pan horizontally:

Click and then drag the x-axis legend.

To restore the original trend chart view:

Right-click the trend chart, and then select **Undo Zoom/Pan**.

Using the arrow keys

You can use the arrow keys on the keyboard to perform some of the same functions as the mouse and VCR-style buttons.

To do this	Press and hold this key	And then press this arrow key
Move back half a time span.	(none)	**Left**
Move forward half a time span.	(none)	**Right**
Pan up 10%.	(none)	**Up**
Pan down 10%.	(none)	**Down**
Move back a full time span.	**Shift**	**Left**
Move forward a full time span.	**Shift**	**Right**
Switch to the next pen’s y-axis.	**Shift**	**Up** or **Down**
Move the value bar to the next data point.	**Ctrl**	**Left** or **Right**
Pan up or down 10%.	**Ctrl**	**Up** or **Down**
Pause.	**Shift** and **Ctrl**	**Left** or **Right**
Switch to the next pen’s y-axis.	**Shift** and **Ctrl**	**Up** or **Down**

Printing the trend chart

To print the trend chart, right-click the trend, and then select **Print Trend**.

Using overlays at run time

Before you can use overlays at run time, a snapshot must be created and the overlay must be set up in the **Overlays** tab. For more information about snapshots and overlays, see Setting up snapshots and overlays on page 620.

To align an overlay to the left of the trend chart:

Right-click the trend chart, select **Overlays** > **Align Left** > the name of the overlay to align to the left of the trend chart.

To relocate an overlay:

Right-click the trend chart, select **Overlays** > **Relocate** > the name of the overlay to relocate.

To show or hide an overlay:

Right-click the trend chart, select **Overlays** > **Show/Hide** > the name of the overlay you want to show or hide.

A check mark beside the overlay name indicates it is visible on the trend chart. No check mark beside the overlay name indicates it is hidden.

To open the Overlays tab:

Right-click the trend chart, select **Overlays > Properties**.

Fixing run-time errors

If data for the trend is not available at run time due to communication errors, messages are sent to the Diagnostics List. For information about setting up FactoryTalk Diagnostics, see the FactoryTalk Help.

For information about troubleshooting common trend problems, see Help.

To view information in the Diagnostics List:

In FactoryTalk View Studio, From the **Tools** menu, select **Diagnostics Viewer**.

You can also select **Start > All Programs > Rockwell Software > FactoryTalk Tools > Diagnostics Viewer**.

Chapter 24

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 run time.

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 run time.

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

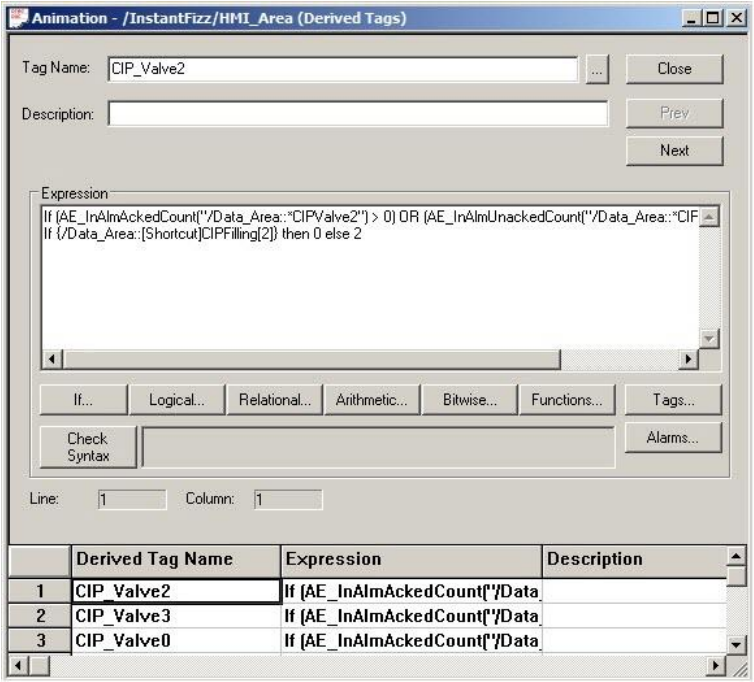
Use the **Derived Tags** editor to create derived tag components for an application.

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.



Animation - /InstantFizz/HMI_Area (Derived Tags)

Tag Name: CIP_Valve2

Description:

Expression

```
If (AE_InAlmAckedCount("/Data_Area::*CIPValve2") > 0) OR (AE_InAlmUnackedCount("/Data_Area::*CIF
If {/Data_Area::[Shortcut]CIPFilling[2]} then 0 else 2
```

Line: 1 Column: 1

	Derived Tag Name	Expression	Description
1	CIP_Valve2	If (AE_InAlmAckedCount("/Data_	
2	CIP_Valve3	If (AE_InAlmAckedCount("/Data_	
3	CIP_Valve0	If (AE_InAlmAckedCount("/Data_	

Accept and Discard buttons

When you enter information in the **Derived Tags** editor, the **Prev** and **Next** buttons change to **Accept** and **Discard** buttons.

Click **Accept** to save derived tags information. Click **Discard** to cancel changes to the derived tags file.

Form and spreadsheet

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 555.

Checking the syntax of an expression

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 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 re-evaluated 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 555.
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 events 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 an 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 an derived tags component at run time, 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 637.

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 start and stop derived tags processing, use FactoryTalk View commands and macros in conjunction with graphic objects. For details about 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 **HMI Server Properties** dialog box, click the **Components** tab, select the check box, **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** check box, 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 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 check box, **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 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 555.
- 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 651.

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

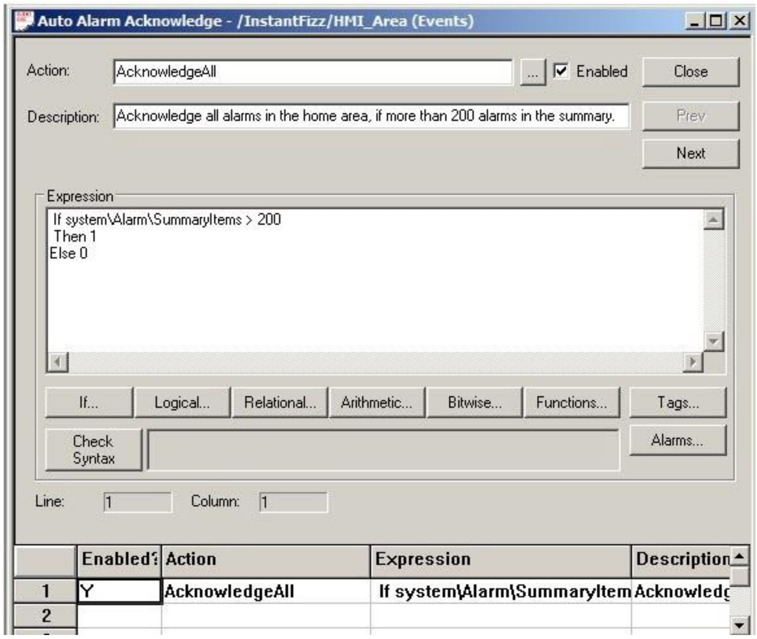
Use the **Events** editor to set up events processing for an application.

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.



Auto Alarm Acknowledge - /InstantFizz/HMI_Area (Events)

Action: AcknowledgeAll ☑ Enabled

Description: Acknowledge all alarms in the home area, if more than 200 alarms in the summary.

Expression

```
If system\Alarm\SummaryItems > 200
Then 1
Else 0
```

Line: 1 Column: 1

	Enabled?	Action	Expression	Description
1	Y	AcknowledgeAll	If system\Alarm\SummaryItem	Acknowledg
2				

Accept and Discard buttons

When you enter information in the **Events** editor, the **Prev** and **Next** buttons change to **Accept** and **Discard** buttons. Click **Accept** to save information. Click **Discard** to cancel changes to the event component.

Form and spreadsheet

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 555.

Checking the syntax of an expression

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 555.

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 events 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 run time, 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 start and stop events processing, use FactoryTalk View commands and macros in conjunction with graphic objects. For details about 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** check box, 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** check box, 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 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 check box, **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 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 160.

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 new 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 $*tag*$ run, 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 646.

Note: Do not use dollar signs or percent signs in comments inserted in macros. This causes errors at run time.

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 all are located 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 3: 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 4: 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 5: 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 run time, 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 142.
- In a local station application, see Selecting startup and shutdown components on page 174.

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 logs on to the FactoryTalk View SE Client.

If the client is already running when the user logs on, 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 located in the **Bottling** area, when the user logs on to the 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 run time, and about assigning login and logout macros to users, see Setting up security on page 85.

Creating client keys

A client key is associated with commands or macros that run when the key is pressed.

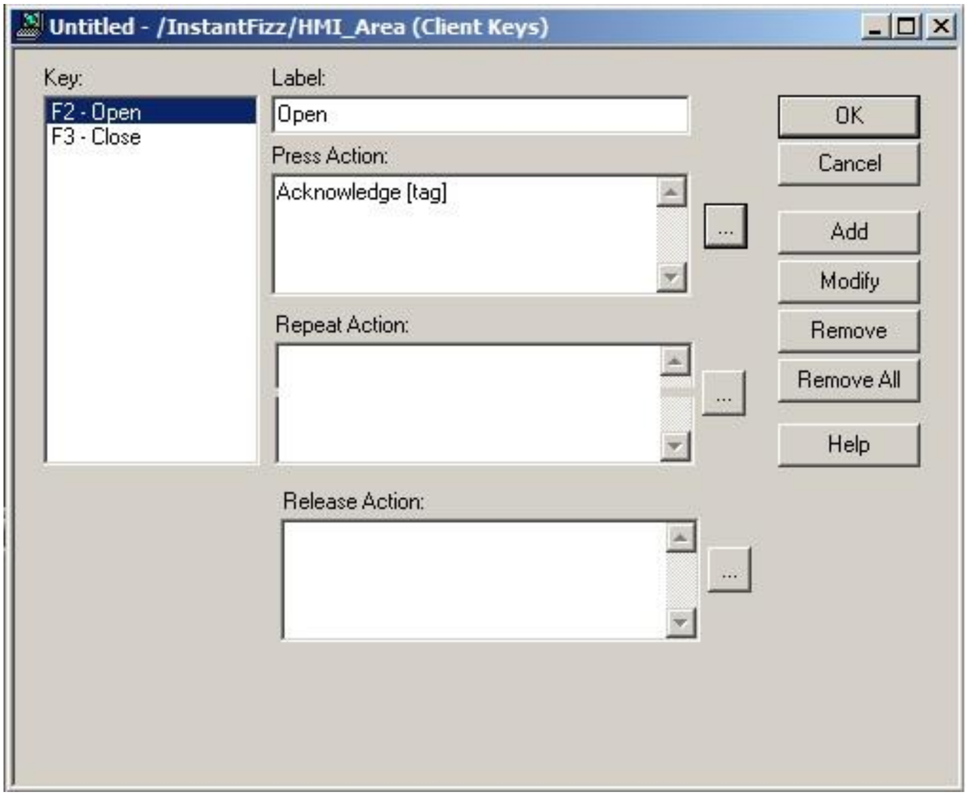
Client keys are similar to object keys and display keys, except client keys are not dependent on a particular graphic object or display. Client keys are associated with the FactoryTalk View SE Client, and are active at all times.

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**.



Untitled - /InstantFizz/HMI_Area (Client Keys)

Key:
F2 - Open
F3 - Close

Label: Open

Press Action: Acknowledge [tag]

Repeat Action:

Release Action:

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 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.

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 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 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 ambiguous area name is one that is the same as another parameter for a command. For example, **AlarmOn "/H"** would turn alarm monitoring on in area *H*; whereas **AlarmOn /H** would turn alarm monitoring on with handshaking, in the current area.

- 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:

- Floating point tags can be a maximum of 17 digits long. If the value is longer than 17 digits, it is represented in scientific format.
- The maximum command length is 5000 characters. If tag values substituted for placeholders in a command cause the command to exceed this length, then the command will be truncated.

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

Acknowledge	AlarmPrintOn	DerivedOn
AcknowledgeAll	DataLogChangeRate	EventOff

AlarmAcceptEdits	DataLogMergeToPrimary	EventOn
AlarmEvent	DataLogNewFile	HandShakeOff
AlarmLogNewFile	DataLogOff	HandShakeOn
AlarmLogOff	DataLogOn	Silence
AlarmLogOn	DataLogRenameFile	SilenceAll
AlarmLogSendToODBC	DataLogSnapShot	SuppressOff
AlarmOff	DataLogSwitchBack	SuppressOffAll
AlarmOn	DerivedOff	SuppressOn
AlarmPrintOff		

Commands that run at the FactoryTalk View SE Client

Abort	Key	PrevWindow
Define	Identify	PrintDisplay
Display	Invoke	PullForward
DisplayClientClose	Language	PushBack
DisplayClientOpen	Login	RecipeRestore
DisplayNavigationHistory	Logout	RecipeSave
DisplayNextScreen	NextPosition	ScreenPrint
DisplayPreviousScreen	NextWindow	SendKeys
Download	OpenRSLogix5000	SetFocus
DownloadAll	Password	Undefine
FlushCache	Position	Upload
Help	PrevPosition	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	Key	PrintDisplay
Display	Login	PullForward
DisplayClientClose	Logout	PushBack
DisplayClientOpen	NextWindow	SetFocus
FlushCache	PrevWindow	

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)	Beep	Ramp
AlarmLogRemark	DDEExecute	Remark
AppAbort	If	Set

AppActivate	Pause	Toggle
AppStart	PlayWave	

If you are using Windows 7 Professional or Windows Server 2008, the **AppStart** command should not be used with interactive actions that use **Events**.

Using absolute and relative references

You can specify several command parameters using either absolute or relative references. Relative references work like file names. Absolute references work like file paths.

A relative reference is a reference to an application component, excluding its path. The component's path is determined from the context in which the component's name is specified.

An absolute reference is a reference to an application component, including its path.

For example:

- **/MixingArea::IngredientDisplay** is an absolute reference to a graphic display called *IngredientDisplay* in the area called *MixingArea*.
- **IngredientDisplay** is a relative reference to a graphic display called *IngredientDisplay* in the current area.

In local applications you only need to use relative references.

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.

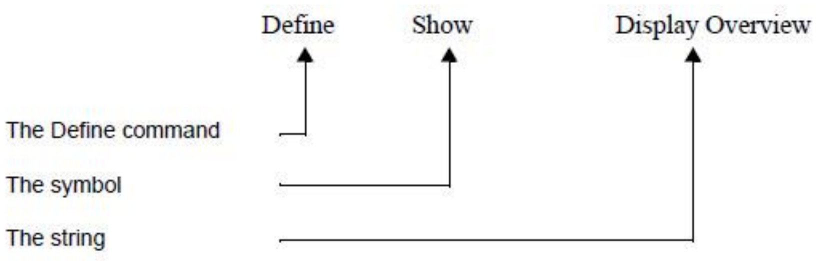
To define a symbol:

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
`Undefine <symbol >` <symbol > is the name of the symbol you want to delete.	Clears the symbol.
Undefine *	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 85.

- If a symbol and a macro have the same name, the symbol runs instead of the macro.

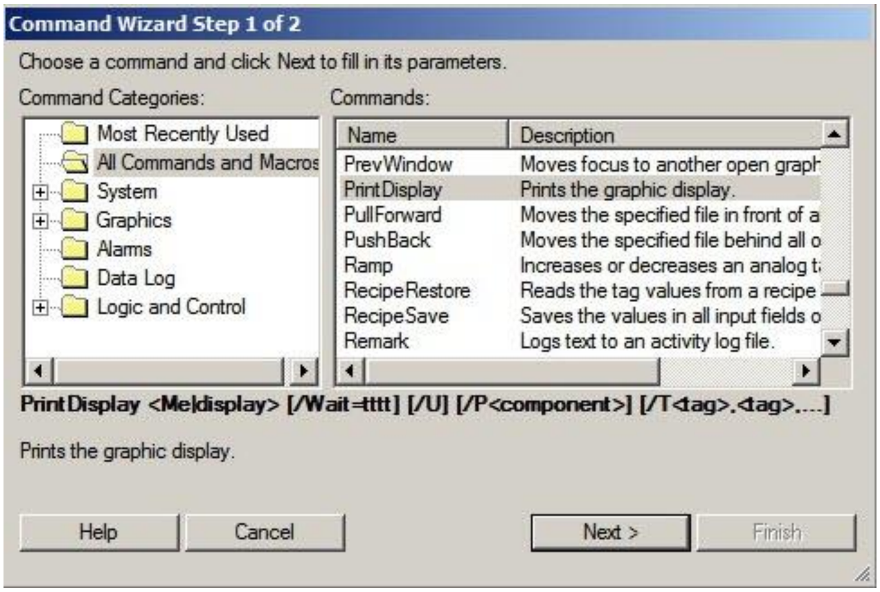
  The order of precedence is: symbol, command, macro.

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.



Choose a command and click Next to fill in its parameters.

Name	Description
PrevWindow	Moves focus to another open graph
PrintDisplay	Prints the graphic display.
PullForward	Moves the specified file in front of a
PushBack	Moves the specified file behind all o
Ramp	Increases or decreases an analog t
RecipeRestore	Reads the tag values from a recipe
RecipeSave	Saves the values in all input fields o
Remark	Logs text to an activity log file.

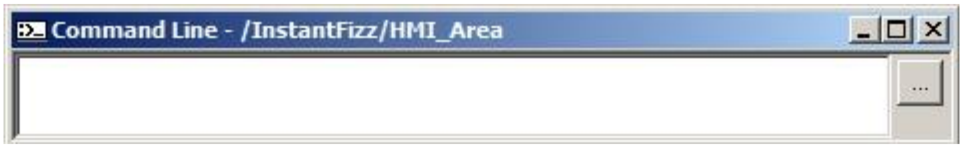
PrintDisplay <Me|display> [/Wait=tttt] [/U] [/P<component>] [/T<tag>,<tag>,...]

Prints the graphic display.

For details about using the **Command Wizard**, and about individual FactoryTalk View commands, see Help.

To open the command line:

In FactoryTalk View Studio, in the **Explorer** window, double-click the **Command Line** icon.



To open the Command Wizard:

Use one of these methods:

- Double-click in an action box, or anywhere in the **Macro** editor.
- Click the **Browse** button beside an action box or any box requiring a FactoryTalk View command.

In the **Command Line, Runtime Secured Commands** editor, **Events** editor, and **Macro** editor, from the **Edit** menu, select **Commands.**

Setting up DDE communications for HMI tags

This appendix describes:

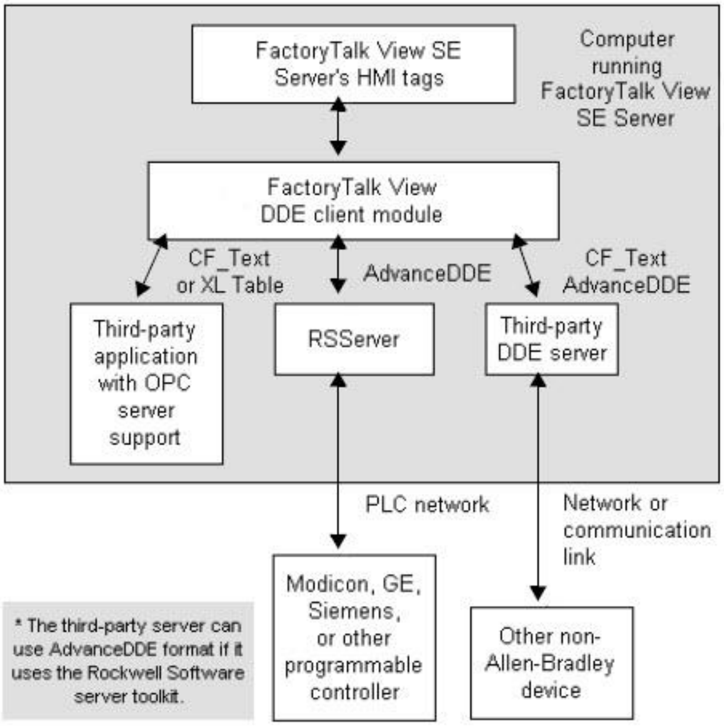
- What DDE communications are.
- How to set up DDE communications with HMI tags.
- Scanning for new tag values.

Tip: This section does not apply to Microsoft Windows 7 Professional or Windows Server 2008.

Overview of DDE communications

To support legacy applications FactoryTalk View supports DDE communications.

The following illustration shows how FactoryTalk View SE might use DDE.


FactoryTalk View SE Server's HMI tags
Computer running FactoryTalk View SE Server
FactoryTalk View DDE client module
CF_Text or XL Table
AdvanceDDE
CF_Text AdvanceDDE
Third-party application with OPC server support
RSServer
Third-party DDE server
PLC network
Network or communication link
Modicon, GE, Siemens, or other programmable controller
Other non-Allen-Bradley device
* The third-party server can use AdvanceDDE format if it uses the Rockwell Software server toolkit.

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 661.

   Tip: If the DDE server is running as an application instead of as a service, for more information, see Answer ID 26798 in the Rockwell Automation KnowledgeBase.

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 189.

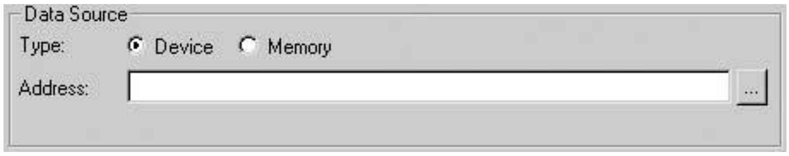
Specifying Device as the data source

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**.



Data Source
Type: ● Device ○ Memory
Address:

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

Use the FactoryTalk View SE Client object model with Microsoft VBA (Visual Basic for Applications) code in graphic displays to customize and extend the capabilities of the FactoryTalk View SE Client.

For example, you can use the object model and display code to do the following:

- **Use data with other applications.** If you regularly require FactoryTalk View data for use in programs such as Microsoft Excel or SQL Server, consider using the FactoryTalk View SE Client object model and VBA display code to integrate these applications with FactoryTalk View.
- **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 run time.

  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.
- **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 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 393.

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 665. For information about VBA Help, see Finding information about VBA on page 670.

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 you create are called in response to events that occur in a graphic display at run time.

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.

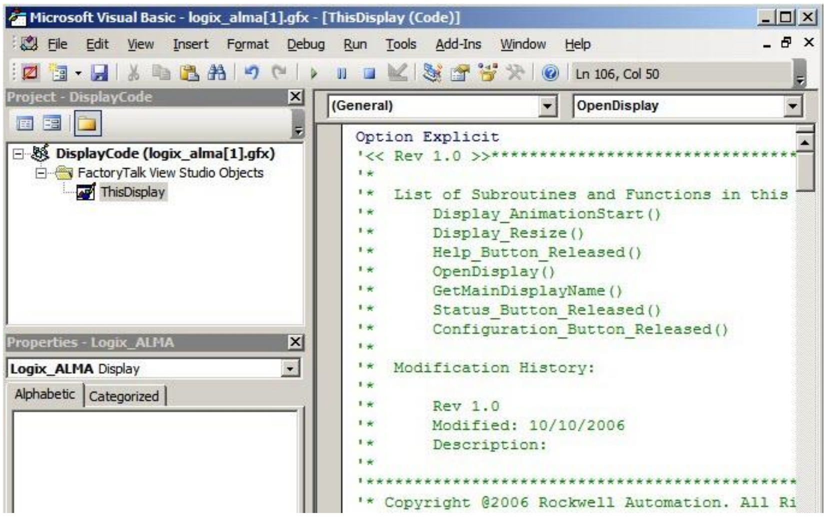
Parts of the VBA integrated development environment

The integrated development environment for VBA IDE, 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 run time.

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**.



```
Option Explicit
'<< Rev 1.0 >>********************************
'*
'*  List of Subroutines and Functions in this
'*      Display_AnimationStart()
'*      Display_Resize()
'*      Help_Button_Released()
'*      OpenDisplay()
'*      GetMainDisplayName()
'*      Status_Button_Released()
'*      Configuration_Button_Released()
'*
'*  Modification History:
'*
'*      Rev 1.0
'*      Modified: 10/10/2006
'*      Description:
'*
'*********************************************
'* Copyright @2006 Rockwell Automation. All Ri
```

For information about opening VBA Help, see Finding information about FactoryTalk View SE Client objects.

Project Explorer window

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 drop-down list at the upper left of the window. When an object is selected in the list, the object's events are listed in the drop-down 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.

VBA can interact with the following objects in the FactoryTalk View SE Client Object Model:

This object in the model	Represents this feature, collection, or graphic object
ActiveXControl	Properties and methods implemented by FactoryTalk View graphics, with those of an ActiveX object.
AlarmSummary	The HMI tag alarm summary.
Application	The FactoryTalk View SE Client container program. Use this object to gain access to the FactoryTalk View graphics object model.

Arc	The arc object.
Arrow	The arrow object.
BackspaceButton	The backspace button object. This object works like the Backspace key on a keyboard.
BarGraph	The bar graph object.
Button	The button object.
ControlListSelector	The control list selector object.
Display	A graphic display.
DisplayListSelector	The display list selector object.
Displays	A collection of Display objects.
Element	Any graphic object in a graphic display. The Element object contains the base properties and methods for all FactoryTalk View objects. These base properties and methods are merged with object-specific properties, methods, and events.
Elements	A collection of Element objects.
Ellipse	The ellipse object.
EmbeddedOLE	Third-party, embedded objects such as Excel charts.
EndButton	The end button object. This object works like the End key on a keyboard.
EnterButton	The enter button object. This object works like the Enter key on a keyboard.
Freehand	The freehand object.
Gauge	The gauge object.
Group	Objects that are grouped together on a display. To write VBA code that responds to a group’s events, you must create the group in a graphic display before you open the VBA IDE. If you don’t create the group first, the Group object will not generate events, and code written for the group will not respond to the group’s events at run time.
HomeButton	The home button object. This object works like the Home key on a keyboard.
Image	The image object.
InterlockedPushButton	The interlocked push button object.
LatchedPushButton	The latched push button object.
Line	The line object.
ListIndicator	The list indicator object.
LocalMessageDisplay	The local message display object.
MaintainedPushButton	The maintained push button object.
MomentaryPushButton	The momentary push button object.
MoveDownButton	The move down button object. This object works like the Down arrow key on a keyboard.
MoveLeftButton	The move left button object. This object works like the Left arrow key on a keyboard.
MoveRightButton	The move right button object. This object works like the Right arrow key on a keyboard.
MoveUpButton	The move up button object. This object works like the Up arrow key on a keyboard.
MultistateIndicator	The multistate indicator object.
MultistatePushButton	The multistate push button object.
NavigationButton	The navigation button object. You can use this object to navigate between previously viewed graphic displays.
NumericDisplay	The numeric display object.
NumericInput	The numeric input object.

PageDownButton	The page down button object. This object works like the Page Down key on a keyboard.
PageUpButton	The page up button object. This object works like the Page Up key on a keyboard.
Panel	The panel object.
Picture	The bitmap object.
PilotCtrlListSelector	The piloted control list selector object.
Polygon	The polygon object.
PolyLine	The polyline object.
PolyPolygon	The polypolygon object.
RampButton	The ramp button object.
Recipe	The recipe object.
RoundedRectangle	The rounded rectangle object. Use the polygon object to represent FactoryTalk View rectangle objects.
Scale	The scale object.
StringDisplay	The string display object.
StringInput	The string input object.
StringList	A collection of unique strings.
Symbol	The symbol object.
Tag	The tag object. Use the Tag object to provide run-time information for a tag, or to set the value of a tag.
TagGroup	A collection of Tag objects.
TagLabel	The tag label object.
Text	The text object.
TimeDateDisplay	The time and date display object.
Wedge	The wedge object.

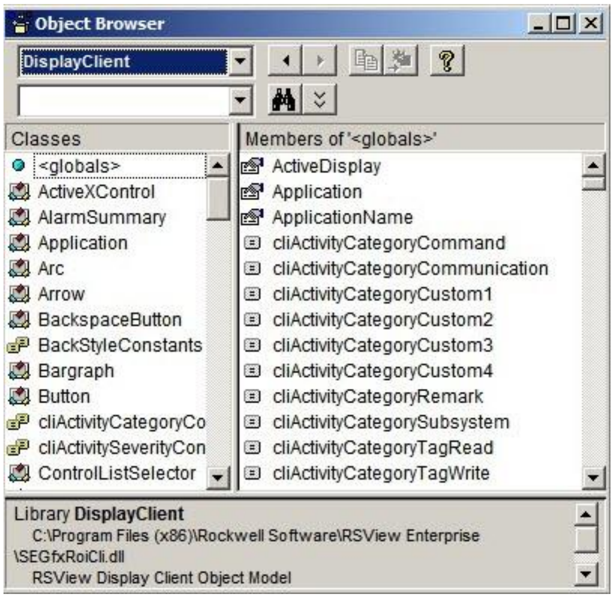
Viewing the objects

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.



Classes

- <globals>
- ActiveXControl
- AlarmSummary
- Application
- Arc
- Arrow
- BackspaceButton
- BackStyleConstants
- Bargraph
- Button
- cliActivityCategoryCo
- cliActivitySeverityCon
- ControlListSelector

Members of '<globals>'

- ActiveDisplay
- Application
- ApplicationName
- cliActivityCategoryCommand
- cliActivityCategoryCommunication
- cliActivityCategoryCustom1
- cliActivityCategoryCustom2
- cliActivityCategoryCustom3
- cliActivityCategoryCustom4
- cliActivityCategoryRemark
- cliActivityCategorySubsystem
- cliActivityCategoryTagRead
- cliActivityCategoryTagWrite

Library **DisplayClient**
C:\Program Files (x86)\Rockwell Software\RSView Enterprise
\SEGfxRoiCli.dll
RSView Display Client Object Model

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 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 web page about XML at http://www.w3.org/XML http://www.w3.org/xml\n.

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 displays names in the BatchImport_*Application name.*xml file.

Modifying exported XML files

When you modify an XML graphic display file, you can make changes to 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

Save XML files created or modified in Notepad, using either the UTF-8 or UTF-16 file format. Notepad's Unicode file type corresponds to 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.

The first line of every XML file contains XML version and encoding attributes. Make sure the encoding attribute matches the format that you are going to use when you save the file.

For example, if the original file was saved in UTF-8 format and you plan to save it in UTF-16 format, make sure the first line specifies

```
encoding="UTF-16"
```

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 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

The graphic display XML file is an 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 this way:

```
<element name>
```

The end is marked this way:

```
</element name>
```

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.

Here is a sample structure for a graphic display XML document containing two graphic objects. *Animations, Connections,* and *States* are all group objects.

For more information about graphic object elements and their attributes, see Help.

Element	Description
<gfx>	Root element.

<displaySettings>	Contains attributes from the **Display Settings** dialog box in the **Graphic Displays** editor.
<*object1*>	Contains attributes from the **General** and **Common** tabs in the object's **Properties** dialog box, as well as elements for the object's caption, image, animation, and connections.
<caption>	Contains attributes for the object's caption.
<imageSettings>	Contains attributes for the object's image.
<animations>	Is a group object that contains an animation element for each type of animation set up for the object.
<animateVisibility>	Contains attributes for Visibility animation.
<animateColor>	Contains attributes for Color animation.
</animations>	Indicates the end of the animations object.
<connections>	Is a group object that contains a connection element for each connection assigned to the object.
<connection name= "Value">	Contains attributes for the Value connection.
<connection name= "Indicator">	Contains attributes for the Indicator connection.
</connections>	Indicates the end of the connections object.
<confirm>	Contains attributes and elements for the confirmation dialog box settings of the object.
</confirm>	Indicates the end of the confirm element.
</*object1*>	Indicates the end of the *object1* element.
<*object2*>	Contains attributes from the **General** and **Common** tabs in the object's **Properties** dialog box, as well as elements for the object's states and connections.
<states>	Is a group object that contains state elements for each of the object's states.
<state stateid="0">	Contains attributes for the object's first state, as well as elements for the state's caption and image.
<caption>	Contains attributes for the state's caption.
<imageSettings>	Contains attributes for the state's image.
</state>	Indicates the end of the state element.
<state stateid="1">	Contains attributes for the object's second state, as well as elements for the state's caption and image.
<caption>	Contains attributes for the state's caption.
<imageSettings>	Contains attributes for the state's image.
</state>	Indicates the end of the state element.
</states>	Indicates the end of the states object.
<connections>	Contains a connection element for each connection assigned to the object.
<connection name= "Value">	Contains attributes for the Value connection.
<connection name= "Indicator">	Contains attributes for the Indicator connection.
</connections>	Indicates the end of the connections element.
</*object2*>	Indicates the end of the *object2* element.
<vbaProject>	Contains elements for the VBA project.
<vbaItem>	Contains elements for a VBA code item.
<vbaCode>	Contains the VBA source code.
</vbaCode>	Indicates the end of the vbaCode element.

<encryptedData>	Contains the encrypted code information if the VBA code item is a user form, module, class module, or procedure.
</encryptedData>	Indicates the end of the encryptedData element.
<vbaItem>	Indicates the end of the vbaItem element.
<vbaProject>	Indicates the end of the vbaProject element.
</gfx>	Indicates the end of the gfx element.

Legal Notices

Copyright notice

End User License Agreement (EULA)

You can view the Rockwell Automation End-User License Agreement ("EULA") by opening the License.rtf file located in your product's install folder on your hard drive.

Trademark Notices

Allen-Bradley, ControlLogix, Data Highway Plus, DH+, Data Highway II, FactoryTalk, Logix5000, PanelBuilder, PanelView, PLC-2, PLC-3, PLC-5, Rockwell, Rockwell Automation, Rockwell Software, RSLinx, RSLogix, RSNetWorx, RSView, SLC, Studio 5000, VersaView, and XM are trademarks of Rockwell Automation, Inc.

Any Rockwell Automation software or hardware not mentioned here is also a trademark, registered or otherwise, of Rockwell Automation, Inc.

Other Trademarks

ActiveX, Microsoft, Microsoft Access, SQL Server, Visual Basic, Visual C++, Visual SourceSafe, Windows, Windows ME, Windows NT, Windows 2000, Windows Server, Windows 8.1, Windows 8, Windows 7, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

Adobe, Acrobat, and Reader are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and other countries.

Oracle is a registered trademarks of Oracle Corporation.

ControlNet is a registered trademark of ControlNet International.

DeviceNet is a trademark of the Open DeviceNet Vendor Association, Inc. (ODVA).

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

OLE for Process Control (OPC) is a registered trademark of the OPC Foundation.

All other trademarks are the property of their respective holders and are hereby acknowledged.

Warranty

This product is warranted in accordance with the product license. The product's performance may be affected by system configuration, the application being performed, operator control, maintenance, and other related factors. Rockwell Automation is not responsible for these intervening factors. The instructions in this document do not cover all the details or variations in the equipment, procedure, or process described, nor do they provide directions for meeting every possible contingency during installation, operation, or maintenance. This product's implementation may vary among users.

This document is current as of the time of release of the product; however, the accompanying software may have changed since the release. Rockwell Automation, Inc. reserves the right to change any information contained in this document or the software at any time without prior notice. It is your responsibility to obtain the most current information available from Rockwell when installing or using this product.

Environmental compliance

Rockwell Automation maintains current product environmental information on its website at http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page http://www.rockwellautomation.com/rockwellautomation/about-us/sustainability-ethics/product-environmental-compliance.page

Contact Rockwell

Customer Support Telephone — 1.440.646.3434
Online Support — http://www.rockwellautomation.com/support/ http://www.rockwellautomation.com/support/

Rockwell Automation support

Rockwell Automation provides technical information on the web to assist you in using its products. At http://www.rockwellautomation.com/support http://www.rockwellautomation.com/support you can find technical and application notes, sample code, and links to software service packs. You can also visit our Support Center at https://rockwellautomation.custhelp.com https://rockwellautomation.custhelp.com for software updates, support chats and forums, technical information, FAQs, and to sign up for product notification updates.

In addition, we offer multiple support programs for installation, configuration, and troubleshooting. For more information, contact your local distributor or Rockwell Automation representative, or visit http://www.rockwellautomation.com/services/online-phone http://www.rockwellautomation.com/services/online-phone.

Installation assistance

If you experience a problem within the first 24 hours of installation, review the information that is contained in this manual. You can contact Customer Support for initial help in getting your product up and running.

United States or Canada	1.440.646.3434
Outside United States or Canada	Use the Worldwide Locator available at http://www.rockwellautomation.com/locations http://www.rockwellautomation.com/locations, or contact your local Rockwell Automation representative.

New product satisfaction return

Rockwell Automation tests all of its products to ensure that they are fully operational when shipped from the manufacturing facility. However, if your product is not functioning and needs to be returned, follow these procedures.

United States	Contact your distributor. You must provide a Customer Support case number (call the phone number above to obtain one) to your distributor to complete the return process.
Outside United States	Please contact your local Rockwell Automation representative for the return procedure.

Documentation feedback

Your comments will help us serve your documentation needs better. If you have any suggestions on how to improve this document, complete the feedback form, publication RA-DU002 http://literature.rockwellautomation.com/idc/groups/literature/documents/du/ra-du002_-en-e.pdf.

Rockwell Otomasyon Ticaret A.Ş., Kar Plaza İş Merkezi E Blok Kat:6 34752 İçerenköy, İstanbul, Tel: +90 (216) 5698400

www.rockwellautomation.com

Power, Control and Information Solutions Headquarters

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

Rockwell Automation Publication VIEWSE-UM006J-EN-E

Supersedes Publication VIEWSE-UM006I-EN-E